Overview

This document provides information and instructions for setting up Lightweight Directory Access Protocol (LDAP) authentication in BusinessObjects Enterprise XI Release 2 using Secure Socket Layer (SSL). This document has complete instructions for both Windows and UNIX. It discusses the key SSL and LDAP concepts and provides detailed information on how these concepts fit into the BusinessObjects Enterprise security model. An appendix is included on troubleshooting LDAP and SSL.

Contents

INTRODUCTION .............................................................................................................. 3
CONCEPTS ...................................................................................................................... 3
LDAP ............................................................................................................................... 3
SSL ................................................................................................................................. 3
Trust ............................................................................................................................... 4
Certificate Authority ................................................................................................. 4
Keystores ..................................................................................................................... 4
Server Authentication ............................................................................................... 4
Mutual Authentication ............................................................................................... 5
Enterprise LDAP and SSL process flows ............................................................... 5
Connecting to LDAP for logon or admin in the CMC ........................................... 5
Authentication in InfoView ....................................................................................... 5
BEFORE GETTING STARTED ..................................................................................... 6
Tools Needed ................................................................................................................ 6
Certutil .......................................................................................................................... 6
Keytool ........................................................................................................................ 6
Services Needed ......................................................................................................... 7
Certificates .................................................................................................................. 7
CA certificate ............................................................................................................... 7
Client certificate ......................................................................................................... 7
GETTING THE CA CERTIFICATE ............................................................................. 7
Windows Certificate Services example ............................................................... 8
## CREATING THE KEYSTORES
- Cert7.db and Key3.db files
- Cacerts file

## GETTING YOUR CLIENT CERTIFICATE
- Creating the certificate request
- Submitting the request to the CA
- Adding the client certificate to the keystores

## CONFIGURING LDAP IN THE CMC
- Connecting to LDAP
- Mapping users and groups

## LOGGING ON IN THE CMC AND INFOVIEW
- Logging on to the CMC using LDAP
- Logging on to InfoView using LDAP

## APPENDIX A: GLOSSARY

## APPENDIX B: TROUBLESHOOTING LDAP AND SSL
- Common Issues

## APPENDIX C: FINDING MORE INFORMATION
Introduction

This document provides instructions for setting up LDAP authentication in BusinessObjects Enterprise XI Release 2 using SSL. It also covers in detail SSL concepts including trust relationships; certificates and keys; protocols; and authentication methods. LDAP concepts such as distinguished names and common names are also discussed.

LDAP and SSL is like a puzzle, with various pieces that need to fit together properly in order for everything to work. If you understand the core concepts and the overall picture of how everything needs to work, then putting the pieces together is much more intuitive. A good analogy is the picture on the box of a jigsaw puzzle.

This guide follows the typical chronological order of setting up an SSL LDAP configuration. However, the conceptual framework for everything is explained first. The additional steps required for mutual (client) authentication are placed in chronological order, but a summary is also provided.

What is not discussed?

This document does not discuss the installation and configuration of your Enterprise, LDAP, web application, or certificate authority servers. It assumes that you either have access to a certificate authority or know how to install the Windows 2003 Certificate Services component. It assumes that you have an existing LDAP server that is configured for SSL. A technical brief for configuring Sun One Directory Server for SSL is in progress.

Concepts

This section explains LDAP and SSL on a conceptual level. For more information, see the Appendix “Finding More Information.”

LDAP

A really useful introduction to LDAP, written by Michael Donnelly, can be found at http://www.ldapman.org/articles/intro_to_ldap.html. Configuring the LDAP server for SSL is outside the scope of this document.

SSL

Secure Socket Layer communication is a way for two peers to talk to each other using encryption so that nobody else is able to understand what is being said. An example of two peers is a web browser and a bank’s web server; or Enterprise XI Release 2 and an LDAP server. These two examples are conceptually identical.
NOTE

Transport Layer Security (TLS) and SSL are often used interchangeably. TLS is the successor to SSL, and includes enhancements and recommendations. The current accepted versions are SSL version 3 and TLS 1.1. This guide uses SSL primarily.

Two peers communicate using SSL with keys that are shared with each other. One peer will use the other peer’s public key to encrypt data, and only the other peer will be able to decrypt the data using a non-shared private key.

Trust

SSL uses encryption, but relies heavily on trust. When Enterprise XI Release 2 connects to the LDAP server, it first verifies that the server is the real, trusted LDAP server, using X.509 certificates. The process is explained below.

Certificate Authority

A certificate authority (CA) is an entity that is responsible for issuing signed certificates. A CA is typically a service running on a server in a company’s network. For example, Windows 2003 has a Certificate Services tool that can be installed. You can configure one such computer on a domain as a CA. In the public, large companies such as Verisign are considered CAs. For example, if you are running a web server with SSL, you would want your certificate signed by a company such as Verisign, because almost all web browsers automatically trust any certificates that are issued by this CA.

The role of the CA in configuring LDAP with SSL is simple. You need to import the CA’s certificate into the keystores that you will create later and declare that you trust any certificates signed by that CA. Later on, this will be used when the LDAP server sends Enterprise its server certificate (which will be signed by that CA).

Keystores

Keystores are files that contain certificates and keys for use with SSL. Typically, a server will have a keystore, and a client will have a keystore. All web browsers, for example, contain keystores for use with SSL. In a typical SSL configuration, the client and server keystores will both contain a CA certificate, which means that both the server and the client will trust any certificate that is signed by that CA.

Server Authentication

This is used in setting up SSL communication to verify that a given server’s certificate is valid and genuine. Typically, the server’s certificate is signed by a CA, and the client knows that the CA is trustworthy (because the CA certificate is stored in the client’s keystore as a trusted CA). Therefore, the server certificate must be trustworthy. In the case of Enterprise LDAP and SSL, the “server” is the LDAP server, because it is providing a service (verifying users) to the Enterprise server.
**Mutual Authentication**

Mutual authentication, also called “client authentication”, actually includes server authentication, but additionally allows the (LDAP) server to verify that the client (BusinessObjects Enterprise) is genuine. In mutual authentication, Enterprise must have its own client certificate that is signed by a CA that the server trusts. When Enterprise contacts the LDAP server, it will send its client certificate, and the server will see that it is signed by a valid CA and automatically know that the Enterprise server is genuine. In this way, both parties are fully aware who is on the other end of the SSL connection.

| TIP | Why is mutual authentication better? After all, Enterprise merely asks whether a given username and password is valid. All that the LDAP server returns is a “yes” or “no”. With server authentication, if an intruder is able to gain access to an LDAP server with a username and password, it may be able to anonymously and securely modify LDAP records in the LDAP server. With mutual authentication, the server knows the exact identity of the person accessing the server. |

**Enterprise LDAP and SSL process flows**

Enterprise uses LDAP and SSL in two different ways, because the Central Management Console (CMC) and the InfoView (desktop.war) components have different requirements in order for SSL to work for LDAP.

**Connecting to LDAP for logon or admin in the CMC**

Enterprise connects to the LDAP server on its secure LDAP port. The LDAP server responds with its server certificate (with an embedded public key). When Enterprise gets this information, it has to decide whether the LDAP server is what it claims it is instead of an imposter that is hijacking packets. Enterprise can tell that the certificate is genuine if it meets two conditions. First, it must be signed by a CA. Second, Enterprise must know of and trust the CA that has signed it. It does this by looking up the CA certificate in its keystore.

Once Enterprise verifies and trusts the server based on the certificate it provided, it can send requests securely.

If mutual authentication is being used, there is an additional step where Enterprise will send its own certificate to the LDAP server. This certificate will have to be pre-signed by a CA. It will most likely be signed by the same CA that the server used. The server will verify that Enterprise’s certificate has a valid CA signature. Then, both parties will be properly authenticated and communication will begin.

**Authentication in InfoView**

The process for InfoView is similar to the CMC. The difference is that InfoView relies on a different keystore than does the CMC. In fact, when
using mutual authentication, InfoView requires two keystores (explained later).

Before Getting Started

Tools Needed

Certutil

(For more information see
http://www.mozilla.org/projects/security/pki/nss/tools/)

Version: NSS Tools up to v3.6
This tool allows you to create the keystore needed by the CMC to authenticate to and map LDAP users through SSL. Newer versions of Certutil generate cert8.db files instead of cert7.db files. Enterprise XI Release 2 requires cert7.db and key3.db. This is the latest release of the NSS tools that uses a cert7.db-based Certutil. The NSS Tools package that includes Certutil is based on the Netscape Portable Runtime (NSPR), so you may need to download the NSPR package. To download either, you can access the Mozilla FTP mirror from the following links: (as of December 2006):

NSS:

NSPR:

Windows:
Download the WIN95.4.0_OPT.OBJ build. You will need to move all of the lib files to the bin folder, or add the lib folder to your system path.

UNIX:
Download the OPT.OBJ builds. For example, for Solaris 8-10, download SunOS5.8_OPT.OBJ.

You will most likely need NSS and the Netscape Portable Runtime (NSPR) that it uses. Download both. Add the NSS and NSPR lib folders to your library path environment variable.

Keytool

Version: included with Java (see above)
This tool is used to create the keystores needed by InfoView. It is included with the Java JRE/JDK (you may need to download one), so you can use this tool as long as the JAVA_HOME environment variable
is pointing to a valid runtime environment. For example:

**WINDOWS:**
JAVA_HOME=C:\Program Files\Java\jre1.5.0_06

**UNIX:**
JAVA_HOME=/opt/software/java/java1.4.2_09/

## Services Needed

Aside from your Enterprise install, you will need an LDAP server (for example, Sun One Directory Server). You will also need a certificate authority server if you do not have the certificate for the CA that signed the LDAP server’s certificate or if you are using mutual authentication and you need a client certificate. You will also need a Java web application server such as Tomcat (which is included with Enterprise on all platforms).

## Certificates

### CA certificate

You must know where to find the certificate for the CA that issued your LDAP server’s certificate so that you can add it to your keystores as a trusted CA. With Windows 2003 certificate services server, you would usually connect to a web interface and request the CA certificate using a URL similar to the following:

http://caserver.company.com/certsrv/

### Client certificate

If you are setting up mutual (client) authentication, you will additionally need a client certificate. You can use Certutil to generate a certificate request, and then submit it to your CA for approval. Once you have retrieved your signed client certificate from the CA, import it into your keystore(s).

<table>
<thead>
<tr>
<th>NOTE</th>
<th>Mutual authentication in the CMC requires that the client certificate be included in the cert7.db file.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mutual (client) authentication in InfoView requires two separate keystore files: Cacerts, which contains your CA certificate, and .keytool, which must contain your client certificate.</td>
</tr>
</tbody>
</table>

## Getting the CA certificate

The CA certificate is required so that Enterprise can verify that the LDAP server certificate comes from a trusted source. Because the server certificate is signed by the CA, if you trust the CA, then you also trust the server certificate, and SSL will work.
Windows Certificate Services example

(Part of Windows 2003 Server, can be added with Add/Remove)

This service provides a web interface and a Microsoft Management Console (MMC) snap-in (used to approve/deny/manage existing requests).

You can retrieve the CA certificate using the web interface shown below (see Figures 1 and 2):

Figure 1

![Microsoft Certificate Services](image)
Creating the keystores

All of the commands are almost identical between Windows and UNIX platforms.

Cert7.db and Key3.db files

The cert7.db and key3.db files are created and managed using Certutil.

From the command line, you create a new cert7/key3 set using the following command:

```
certutil -N -d .
```

The command will prompt you for a password. It is not important, so you can type “password”. Remember to use the same password for all of your keystores. The following example is using UNIX, but they are the same:

```
[bmcrae@vanpgsol64 ~]$ certutil -N -d .

In order to finish creating your database, you must enter a password which will be used to encrypt this key and any future keys.

The password must be at least 8 characters long, and must contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
```

[bmcrae@vanpgsol64 ~]$
This command creates a cert7.db, key3.db, and secmod.db file. You will only need the first two. Once the keystore is created, you can add your CA certificate to it:

```
Command 2
certutil -A -n "MyCA" -t "CT" -d . -i cacert.cer
```

The “MyCA” parameter does not matter, and does not be customized. The C and T attributes are explained in the output below.

Windows:

```
Certificate Name  Trust Attributes
MyCA                             CT,,
p    Valid peer
P    Trusted peer (implies p)
c    Valid CA
T    Trusted CA to issue client certs (implies c)
C    Trusted CA to certs(only server certs for ssl)(implies c)
u    User cert
w    Send warning
```

UNIX:

```
[bmcrae@vanpgsol64 ~]$ certutil -A -n "MyCA" -t "CT" -d . -i cacert.cer
[bmcrae@vanpgsol64 ~]$ certutil -L -d .
MyCA                             CT,,
```

Notice that in UNIX, the trust attributes legend does not appear. It is also worth noting that while Windows allows you to apply other attributes such as “P” and “u”, UNIX will ignore these, as a CA certificate cannot be a user certificate.

Once the CA certificate is added as a trusted CA, this keystore is finished, unless you intend to enable mutual authentication. In that case, refer to the section “Getting Your Client Certificate.”

**Cacerts file**

Creating the cacerts keystore using Keytool, which is required by our InfoView application, is described below. The creation of the keystore (called “cacerts”) and importing of the CA certificate is performed with one command:
Command 3

```
keytool -import -v -alias MyCA -file cacert.cer -trustcacerts -keystore cacerts
```

```
C:\>keytool -import -v -alias MyCA -file cacert.cer -trustcacerts -keystore cacerts
Enter keystore password:  password
Owner: CN=BMAUTH1, OU=CS, O=BOBJ, L=Van, ST=BC, C=CA
Issuer: CN=BMAUTH1, OU=CS, O=BOBJ, L=Van, ST=BC, C=CA
Serial number: 938606c60e59b3442d9512637721e8
Certificate fingerprints:
MD5:
SHA1:
Trust this certificate? [no]:  yes
Certificate was added to keystore

[Saving cacerts]
C:\>
```

You can verify that the import was successful using the following command:

Command 4

```
keytool -list -v -keystore cacerts
```

```
Windows:
C:\>keytool -list -v -keystore cacerts
Enter keystore password:  password
Keystore type: jks
Keystore provider: SUN
Your keystore contains 1 entry:
Alias name: myca
Creation date: Thu Nov 23 17:03:10 PST 2006
Entry type: trustedCertEntry
Owner: CN=BMAUTH1, OU=CS, O=BOBJ, L=Van, ST=BC, C=CA
Issuer: CN=BMAUTH1, OU=CS, O=BOBJ, L=Van, ST=BC, C=CA
Serial number: 938606c60e59b3442d9512637721e8
Certificate fingerprints:
MD5:
SHA1:
```

*******************************************
*******************************************
C:\>
```

NOTE

The .keystore file (mutual authentication only) is created after you have obtained your client certificate (see below).
Getting your client certificate

The client certificate is required only for mutual authentication. This is addressed when you retrieve the CA certificate.

Creating the certificate request

In order to get a signed client certificate, generate a certificate request which then gets submitted to and approved by your CA. To generate the request, you need an existing cert7.db/key3.db pair, and to specify a few attributes:

```
Command 5
certutil -R -s "CN=serverName, OU=CS, O=BOBJ, L=Van, ST=BC, C=CA" -a -o serverName.req -d .
```

Windows:

```
C C:\>certutil -R -s "CN=serverName, OU=CS, O=BOBJ, L=Van, ST=BC, C=CA" -a -o serverName.req -d .
```

A random seed must be generated that will be used in the creation of your key. One of the easiest ways to create a random seed is to use the timing of keystrokes on a keyboard.

To begin, type keys on the keyboard until this progress meter is full. DO NOT USE THE AUTOREPEAT FUNCTION ON YOUR KEYBOARD!

```
|**********************************************************
**|
```

Finished. Press enter to continue:

```
Enter Password or Pin for "NSS Certificate DB":
```

Generating key. This may take a few moments...

```
C:
```

A Base64-encoded certificate request will be placed in a file called “servername.req” in this example.

Submitting the request to the CA

In order to get a signed client certificate, generate a certificate request which then gets submitted to and approved by your CA:

1. To submit the certificate, click advanced certificate request (see Figure 3).
2. Next, paste the text for the Base64-encoded certificate request file that you created with Command 4 into the text box (see Figure 4). Depending on the version of certificate services, you may have the option to choose a template from the dropdown (Web Server fits best and was used in this example).

3. The submitted certificate may be approved automatically, or it could go into a pending state (see Figure 5).
4. Once you have submitted your request, you may need to approve it in Microsoft Certificate Services if using your own CA (see Figure 8).
5. Once the certificate is approved (it may be approved automatically), you can retrieve it from the same interface (see Figure 9).

6. Save the .cer file to the same directory as your cert7.db, key3.db, and cacerts files.

**Adding the client certificate to the keystores**

Once you have retrieved your certificate from the CA, add it into two keystores. Do not put the client certificate in the cacerts keystore. For mutual authentication, create a third keystore called ".keystore":

**Cert7.db and Key3.db**

Much like Command 2, but for the client certificate, thus with different attributes:

```
Command 6

certutil -A -n "Client" -t "Pu" -d . -i client.cer
```
Verify that the certificate is imported, and that the private key is recognized in the key3.db file using the commands “certutil -L -d .” and “certutil -K -d .”

Windows:

C:\>certutil -A -n "Client" -t "Pu" -d . -i client.cer
C:\>certutil -L -d .

<table>
<thead>
<tr>
<th>Certificate Name</th>
<th>Trust Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>Pu,</td>
</tr>
<tr>
<td>MyCA</td>
<td>CT,</td>
</tr>
<tr>
<td>Valid peer</td>
<td></td>
</tr>
<tr>
<td>Trusted peer (implies p)</td>
<td></td>
</tr>
<tr>
<td>Valid CA</td>
<td></td>
</tr>
<tr>
<td>Trusted CA to issue client certs (implies c)</td>
<td></td>
</tr>
<tr>
<td>Trusted CA to certs(only server certs for ssl)(implies c)</td>
<td></td>
</tr>
<tr>
<td>User cert</td>
<td></td>
</tr>
<tr>
<td>Send warning</td>
<td></td>
</tr>
</tbody>
</table>

C:\>certutil -K -d .
Enter Password or Pin for "NSS Certificate DB":
<0> Client
C:\>

.keystore

This is similar to command 3:

Command 7: keytool -import -v -alias Client -file client.cer -trustcacerts -keystore .keystore

C:\>keytool -import -v -alias Client -file client.cer -trustcacerts -keystore .keystore
Enter keystore password: password
Owner: CN=serverName, OU=CS, O=BOBJ, L=Van, ST=BC, C=CA
Issuer: CN=van-w-bmauth1, DC=brodie, DC=businessobjects, DC=com
Serial number: 15ac14c90000000000003
Certificate fingerprints:
Trust this certificate? [no]: yes
Certificate was added to keystore
[Saving .keystore]
C:\>
Finally, place this file along with the other keystores in a directory on your system, for example C:\SSL\, or /r2/bobje/SSL.

Configuring LDAP in the CMC

The client certificate is required only for mutual authentication. However, you would address this when you retrieve the CA certificate.

Connecting to LDAP

1. Type the host name and SSL port (usually 636) of your LDAP server.
2. Type the appropriate LDAP server type, or specify custom attribute mappings.
3. Type the base DN for your LDAP server (for example, 'cn=companyname, cn=com')
4. Type the RDN (for example, 'cn=admin') and password for the Admin account.
5. Type referral information if necessary (normally left alone).
6. Select Server or Mutual Authentication from the Type of SSL authentication dropdown list.
7. Select the appropriate option for server-side SSL strength.

| NOTE | Java applications, such as desktop.war (Java InfoView), automatically use the middle option Accept server certificate if it comes from a trusted certificate authority. It is recommended that you select this option (see Figure 10). |

Figure 10

8. Clear the use default value checkboxes for Path to the certificate and key database files and Password for the key database.
9. Type the path to the cert7.db, key3.db, and cacerts files in the first field and the keystore password in the second. For Mutual authentication only, if you are using a client certificate, you will need
the alias name that you specified in Commands 6 and 7 of this guide. Additional hosts and locations may be added using the SSL host field (see Figure 11). This may be necessary if, for example, you are running web application servers in a distributed farm, or if you have clustered CMS servers and each machine has a different local path to its local SSL certificates. If you leave it blank, all hosts will use the default settings (which would not work if your web server is on UNIX and the CMS is on Windows). Each computer will query the CMS database for information on how to connect using LDAP, and depending on the SSL host values, the correct properties will be returned.

10. Select whether or not to use Siteminder SSO. If selected, configure the Siteminder settings. This configuration is outside the scope of this document.

11. You can specify **New aliases will be added and new users will be created** on the LDAP users and aliases configuration page.

12. Click **Finish** to apply the plugin configuration:
Figure 12

Mapping users and groups

1. In the LDAP tab of the auth page, add the mappings in the Add LDAP group (by cn or dn) field (see Figure 13).
2. Click Update at the bottom of the page. Enterprise will map in the LDAP objects that you have specified.

Logging on in the CMC and InfoView

The client certificate is required only for mutual authentication. This is addressed when you retrieve the CA certificate.

Logging on to the CMC using LDAP

When logging on, choose LDAP as the Authentication Type (see Figure 14).

Figure 14

Logging on to InfoView using LDAP

When logging on, choose LDAP as the Authentication Type (see Figure 15).
Figure 15

CAUTION Mutual authentication requires that the client certificate be stored in the .keystore file. See the section “Adding the client certificate to the keystores”

Appendix A: Glossary

This section is a quick reference for the acronyms and terms used in this guide.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>Certificate Authority. A digital “Notary Public”</td>
</tr>
<tr>
<td>CN</td>
<td>Common Name (LDAP)</td>
</tr>
<tr>
<td>DN</td>
<td>Distinguished Name (LDAP)</td>
</tr>
<tr>
<td>JRE</td>
<td>Java Runtime Environment</td>
</tr>
<tr>
<td>JVM</td>
<td>Java Virtual Machine</td>
</tr>
<tr>
<td>LDAP</td>
<td>Lightweight directory access protocol</td>
</tr>
<tr>
<td>NSPR</td>
<td>Netscape Portable Runtime - used by NSS</td>
</tr>
<tr>
<td>NSS</td>
<td>Netscape Security Services – contains the Certutil tool</td>
</tr>
<tr>
<td>RDN</td>
<td>Relative Distinguished Name (LDAP)</td>
</tr>
<tr>
<td>Siteminder</td>
<td>Tool used to manage user credentials for SSO</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Socket Layer (predecessor to TLS)</td>
</tr>
</tbody>
</table>
SSO | Single Sign On
---|---
TLS | Transport Layer Security
Trust | The basis for SSL authentication, where Person A trusts Person B’s certificate based on a signature by a CA that Person A trusts.

### Appendix B: Troubleshooting LDAP and SSL

This section includes tips for troubleshooting common problems with setting up SSL for LDAP authentication.

Several different error messages can be diagnosed by running the Filemon utility (Windows) on the server computer and then looking at the files being read and any errors that may be surfacing.

### Common Issues

The following table describes a few error messages that can be encountered:

<table>
<thead>
<tr>
<th>Problem:</th>
<th>Cause, resolution:</th>
</tr>
</thead>
</table>
| **InfoView error message:**  
“Account Information Not Recognized: The secLdap plugin failed to connect to the specified hosts” (using mutual authentication) |  
**Possible Cause:** The web application server cannot find the .keystore file.  
**Resolution:** You must have your client certificate in a keystore called “.keystore” in the same path as your cert7.db file.  
**Possible Cause:** The web application server cannot find cacerts file.  
**Resolution:** If .keystore is found, it then looks for “cacerts”. You must have your CA certificate in a keystore called “cacerts” in the same path as your cert7.db file. |
| **CMC error message while updating LDAP properties:**  
“Switching between server and mutual SSL authentication, changing the serv...” |  
**Cause:** If you have established, or attempted to establish, a connection with the server or mutual authentication, you cannot switch.  
**Resolution:** Disable LDAP authentication if it is active, then restart the CMS. You usually do not need to log back on. |
CMC error message while updating LDAP properties:
“The secLdap plugin failed to retrieve the client certificate from the certificate database. Either the key database password is incorrect, or the certificate specified does not exist” (using mutual authentication).

<table>
<thead>
<tr>
<th>Cause:</th>
<th>Either the client certificate is not stored in the cert7.db file or the private key is not stored in the key3.db file.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution:</td>
<td>Ensure that you have both the CA certificate, with CT trust attributes, as well as the Client certificate, with “Pu” trust attributes, active. Use the command <code>certutil -K -d</code> to verify that you have the client private key stored in your key3.db file.</td>
</tr>
</tbody>
</table>
Appendix C: Finding more information

For more information and resources, refer to the product documentation and visit the support area of the web site at:
http://www.businessobjects.com/

For more information on the various trust attributes, visit:

For more information on LDAP, visit:
http://www.ldapman.org/articles/intro_to_ldap.html

For more information on cert7.db and its database format, visit:
http://www.mozilla.org/projects/security/pki/nss/db_formats.html