

Oracle® Endeca Information Discovery Integrator

Security Guide for Integrator

Version 3.2.0 • January 2016

Copyright and disclaimer

Copyright © 2003, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. UNIX is a registered trademark of The Open Group.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Table of Contents

Copyright and disclaimer	2
Preface	5
About this guide	5
Who should use this guide	5
Conventions used in this guide	5
Contacting Oracle Customer Support	6
Chapter 1: About Security in Integrator	7
Overview of security in Integrator	7
Deployment configurations	8
Operating system security	9
Container security	10
Finding additional information	10
Chapter 2: Securing Integrator ETL Installations	11
Securing the file system	11
Configuring SSL	12
Configuring SSL in Integrator ETL Designer	12
Configuring SSL in Integrator ETL Server	13
Configuring SSL between Integrator ETL Designer and IAS	13
Database security	14
Updating the default user	14
Chapter 3: Implementing Integrator ETL Security	15
Managing users and groups	15
Implementing LDAP	15
Sandbox permissions	15
WebDAV authentication	16
Securing graph parameters	16
Chapter 4: Configuring SSL in the Integrator Acquisition System	17
About configuring SSL in the Integrator Acquisition System	17
About configuring SSL for IAS in WebLogic Server installations	18
Enabling SSL for IAS in WebLogic Server	19
Enabling HTTPS redirects for IAS in WebLogic Server	21
Enabling the Endeca Web Crawler to write to an SSL enabled Record Store instance	22
Enabling SSL for the IAS Command-line Utilities	24
About enabling SSL in Jetty Web server installations	26
Enabling SSL for the Endeca IAS Service	26
Enabling SSL for the IAS Command-line Utilities	29
Enabling the Endeca Web Crawler to write to an SSL enabled Record Store instance	32

Chapter 5: Configuring SSL in the IKM SQL to Endeca Server	34
About configuring SSL in the IKM SQL to Endeca Server	34
About securing the IKM SQL to Endeca Server within Oracle Data Integrator	34
Chapter 6: Security Considerations for Integrator ETL Developers	35
Securing individual components	35
Configuring SSL for the Web Service Client component	35
Encrypting database access passwords	35
SOAP authentication	36

Preface

Oracle® Endeca Information Discovery Integrator is a powerful visual data integration environment that includes:

The Integrator Acquisition System (IAS) for gathering content from delimited files, file systems, JDBC databases, and Web sites.

Integrator ETL, an out-of-the-box ETL purpose-built for incorporating data from a wide array of sources, including Oracle BI Server.

In addition, Oracle Endeca Web Acquisition Toolkit is a Web-based graphical ETL tool, sold as an add-on module. Text Enrichment and Text Enrichment with Sentiment Analysis are also sold as add-on modules. Connectivity to data is also available through Oracle Data Integrator (ODI).

About this guide

This guide explains how to install, configure, and use Oracle Endeca Information Discovery Integrator securely.

Who should use this guide

This manual is intended for data architects and system administrators who are responsible for installing, maintaining, and using Integrator.

Conventions used in this guide

The following conventions are used in this document.

Typographic conventions

The following table describes the typographic conventions used in this document.

Typographic conventions

Typeface	Meaning
User Interface Elements	This formatting is used for graphical user interface elements such as pages, dialog boxes, buttons, and fields.
Code Sample	This formatting is used for sample code phrases within a paragraph.
<Variable Name>	This formatting is used for variable values, such as <install path>.
File Path	This formatting is used for file names and paths.

Symbol conventions

The following table describes symbol conventions used in this document.

Symbol conventions

Symbol	Description	Example	Meaning
>	The right angle bracket, or greater-than sign, indicates menu item selections in a graphic user interface.	File > New > Project	From the File menu, choose New, then from the New submenu, choose Project.

Contacting Oracle Customer Support

Oracle Customer Support provides registered users with important information regarding Oracle software, implementation questions, product and solution help, as well as overall news and updates from Oracle.

You can contact Oracle Customer Support through Oracle's Support portal, My Oracle Support at <https://support.oracle.com>.



Chapter 1

About Security in Integrator

Oracle Endeca Information Discovery Integrator is a high-performance platform you can use to extract source data from a variety of source types, process and transform these records, and load them into an Endeca data domain. This guide describes how you can secure Integrator ETL Designer, Integrator ETL Server, Integrator Acquisition System, and their communication with Endeca Server.

[Overview of security in Integrator](#)

[Deployment configurations](#)

[Operating system security](#)

[Container security](#)

[Finding additional information](#)

Overview of security in Integrator

Configuring security in Integrator includes securing the Integrator ETL Server and configuring secure communication between Integrator Acquisition System, Integrator ETL Designer, and Integrator ETL Server, as well as between Integrator and Endeca Server.

Integrator consists of the following parts:

- Integrator ETL Designer (often referred to as simply Integrator ETL) is an integrated development environment (IDE) you can use to develop graphs (applications) to extract data from an existing repository, process it, and load it into Endeca Server.
- Integrator ETL Server provides a runtime environment for the graphs, allowing you to integrate graphs into enterprise applications.
- Integrator Acquisition System, or IAS, is a set of components that crawl source data stored in a variety of formats including: file system, Web servers, and custom data sources. IAS transforms the data, if necessary, and outputs the data to an XML file or a Record Store that can be accessed by Integrator ETL for use in the Endeca Server.

Integrator ETL Server is not required to load data to Endeca Server data domains. Integrator ETL Designer can connect to Endeca Server to load data. This option, however, is not scalable, as each user would need a unique installation of Integrator ETL Designer, and would need to run graphs manually. In an enterprise environment, it is more effective to use Integrator ETL Server, which allows multiple users and groups of users to access and run graphs. Moreover, Integrator ETL Server provides the ability to automatically schedule graphs to run at designated times. Integrator ETL Server also provides tools for monitoring the progress and success of graph execution.

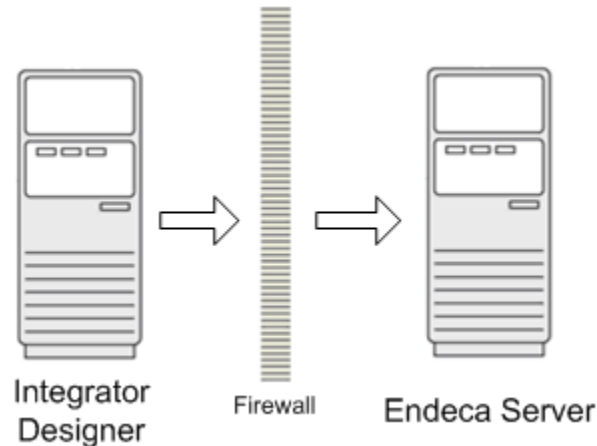
Integrator Acquisition System, Integrator ETL Designer, and Integrator ETL Server, and Endeca Server use HTTP/HTTPS to communicate.

Deployment configurations

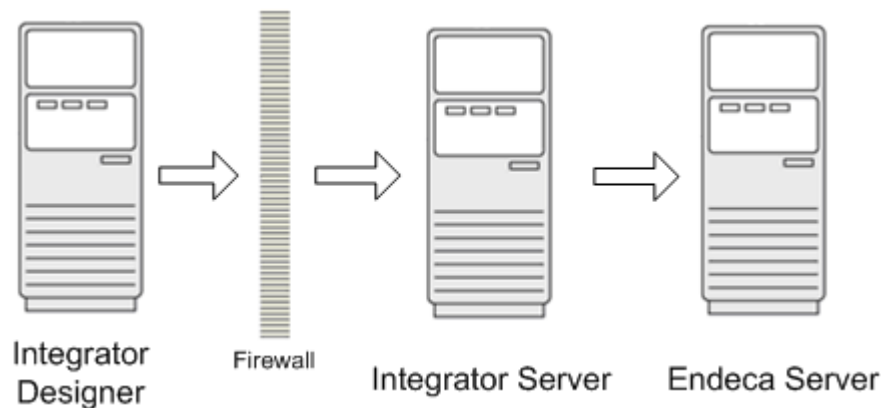
This topic illustrates typical deployment scenarios for Integrator ETL.

The simplest deployment configuration is Integrator ETL Designer and Endeca Server on the same machine. While this configuration can be used for evaluation and for training, it is not a recommended configuration in a development or production environment.

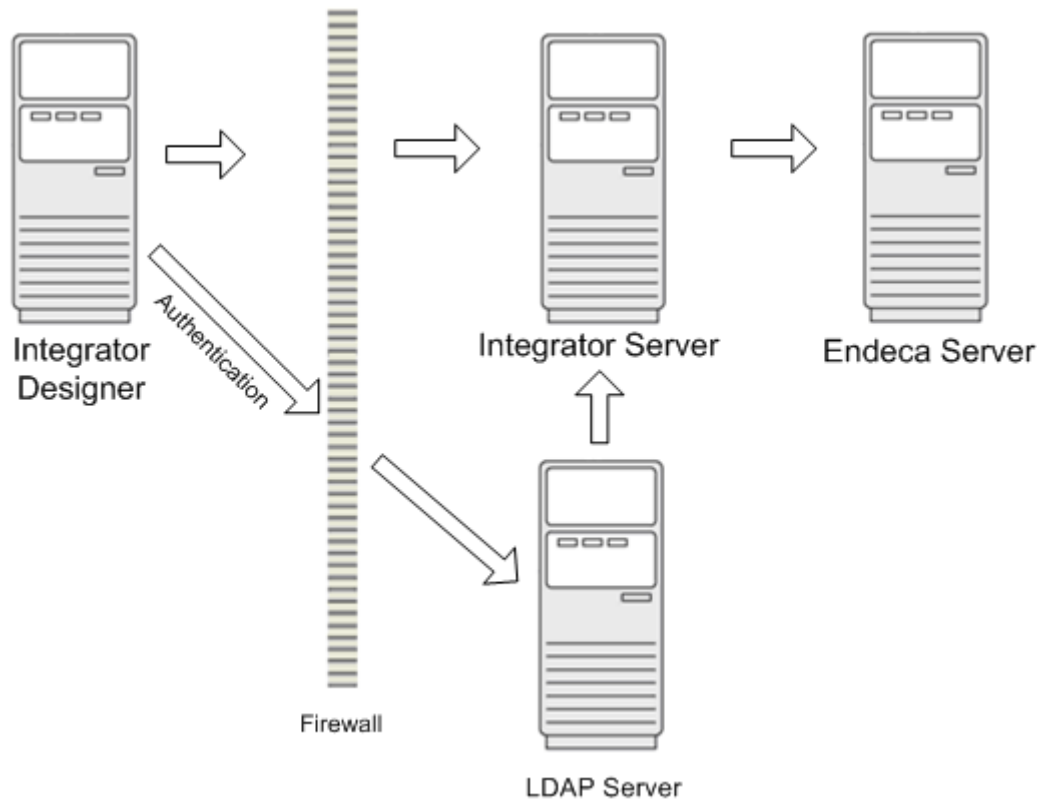
A more complex deployment consists of Integrator ETL Designer deployed on one machine and Endeca Server deployed on another machine. In this scenario, Endeca Server may be deployed behind a firewall.



An enterprise deployment scenario consists of Integrator ETL Designer connected to Integrator ETL Server for development and maintenance of graphs, with Integrator ETL Server connected to Endeca Server to upload data.



An alternative enterprise deployment scenario includes an LDAP server for authentication.



Operating system security

This topic describes where to find information about securing the operating system for Integrator ETL.

The supported operating system for Integrator ETL Designer and Integrator ETL Server is Oracle Linux 6. See the following resources for recommendations about securing and hardening the operating system:

- <http://www.oracle.com/technetwork/articles/servers-storage-admin/secure-linux-env-1841089.html>
- <http://www.oracle.com/technetwork/articles/servers-storage-admin/tips-harden-oracle-linux-1695888.html>

Note the following exceptions to the recommendations above:

- If you are installing Integrator ETL Designer, do not disable XWindows.
- If you want to use SSH to connect to the machine, do not disable SSH. Also, configure access to port 22 (the default SSH port).
- Configure access to the following ports:
 - HTTP
The well-known port is port 80. The default port for the WebLogic Server host for Integrator ETL Server is 7001.
 - HTTPS

The well-known port is port 443. The default port for the WebLogic Server host for Integrator ETL Server is 7002.

- Endeca Server dgraph ports

Endeca Server uses a range of ports for Dgraph processes. The range of ports is configurable. For details, see "Endeca Server configuration" in the *Oracle Endeca Server Administrator's Guide*.

Container security

This topic describes where to find information about securing the container for Integrator ETL Server.

The supported container for Integrator ETL Server is WebLogic Server 12.1.3. For details about securing and hardening WebLogic Server, see <http://docs.oracle.com/middleware/1213/wls/SECMG/toc.htm>.

Integrator ETL Server can also be installed on Apache Tomcat 6.0.

Create a user specifically to install and run the container. This user should have the minimum necessary permissions for the file system.

Finding additional information

In addition to this guide, the following documents will also help you secure your Information Discovery implementation.

The *Oracle Endeca Server Security Guide* describes how to secure Oracle Endeca Server. This is the first step in securing Oracle Endeca Information Discovery.

The *Oracle Endeca Information Discovery Studio Security Guide* describes how to secure Studio and the Provisioning Service.

The *Oracle Endeca Information Discovery Integrator ETL User's Guide*, *Integrator ETL Designer Guide*, and *Integrator ETL Server Guide* describe how to use Integrator ETL.



This section provides recommendations for securing Integrator ETL Designer and Integrator ETL Server installations.

[Securing the file system](#)

[Configuring SSL](#)

[Database security](#)

[Updating the default user](#)

Securing the file system

This topic provides recommendations for securing the installed file system.

Securing the Integrator ETL Designer file system

Integrator ETL Designer typically runs on a workstation. Follow the security guidelines of your organization to determine the correct security implementation for your installation. A very restrictive environment might allow full access to the owner, with no access for group or other. A less restrictive environment might allow full access to the owner, read and execute access to group, and no access to other.

Securing the Integrator ETL Server file system

When installing Integrator ETL Server into a WebLogic Server container, set the following permissions on the *WL_Home* and *Domain_Home*:

- On these directories and all of their subdirectories, set permissions as follows:
 - Owner: full access (read/write/execute)
 - Group: read and execute access
 - Other: No access

In other words, set permissions to **750**.

- On the files within these directories, set permissions as follows:
 - Owner: read and write access
 - Group: read only access
 - Other: no access

In other words, set permissions to **640**

Configuring SSL

You can configure SSL to secure communication among the components of Oracle Endeca. This includes the Integrator Acquisition System, Integrator ETL Designer, Integrator ETL Server, Studio, and Endeca Server. For details about securing Endeca Server, see the *Endeca Server Security Guide*.

Very broadly speaking, you configure SSL in an Endeca Information Discovery application by generating certificates using Java keytool. You then copy the keystore and truststore files to Endeca Server, Integrator ETL Designer, and the Integrator ETL Server, and Integrator Acquisition System.



Note: You may not want to generate the certificates with the `generate_ssl_keys` utility provided with Endeca Server, because the `generate_ssl_keys` utility does not include fully qualified host names, which are required by the Integrator Acquisition System.

[Configuring SSL in Integrator ETL Designer](#)

[Configuring SSL in Integrator ETL Server](#)

[Configuring SSL between Integrator ETL Designer and IAS](#)

Configuring SSL in Integrator ETL Designer

To configure SSL in Integrator ETL Designer:

1. Copy the keystore and truststore files from your Endeca Server installation to a location in your Integrator ETL Designer installation.
2. Start Integrator ETL Designer.
3. Select **Preferences** from the **Window** menu.
4. From the **Preferences** menu, select **Java>Installed JREs**.
5. In the **Installed JREs** menu, click on the checked JRE and then click **Edit**.
The **Edit JRE** menu is displayed.
6. In the **Default VM Arguments** field, enter the following on a single line. Replace the filenames and passwords with the ones you created, and specify the paths to the location where you placed the files:

```
-Djavax.net.ssl.keyStore=path\to\yourcertkeystorefile.jks  
-Djavax.net.ssl.keyStorePassword=keystorepass  
-Djavax.net.ssl.trustStore=path\to\yourtruststorefile.jks  
-Djavax.net.ssl.trustStorePassword=truststorepass
```

7. Click **Finish** to apply your change and close the **Edit JRE** menu.
8. Click **OK** to close the **Preferences** menu.
9. In a simple text editor, open the `IntegratorETL.ini` file in the root of your Integrator ETL installation. Under the "`-vmargs`", add the code from Step 6. For example:

```
-vmargs  
-Dosgi.requiredJavaVersion=1.5  
-XX:MaxPermSize=256m  
-Xms40m  
-Xmx512m  
-Djavax.net.ssl.trustStore=/endeca_server/ssl/endecaServerTrustStore.ks  
-Djavax.net.ssl.trustStorePassword=endeca  
-Djavax.net.ssl.keyStore=/endeca_server/ssl/endecaServerClientCert.ks
```

```
-Djavax.net.ssl.keyStorePassword=endeca
```

You must enable SSL on graph components that communicate with the Endeca Server. For details, see [Securing individual components on page 35](#).

Configuring SSL in Integrator ETL Server

To implement SSL on Integrator ETL Server, copy the keystore and truststore files from the Endeca Server installation to the Integrator ETL Server container, then add Java options with the path and password of the keystore and truststore files.

WebLogic Server container on Linux

Add the keystore and truststore files to your WebLogic installation.

In the file `$DOMAIN_HOME/startWebLogic.sh`, after setting the `USER_MEM_ARGS` variable, set `JAVA_OPTIONS` with the properties for the keystore and the truststore file locations and passwords.

For example:

```
export JAVA_OPTIONS="
-Djavax.net.ssl.trustStore=/endeca_server/ssl/endecaServerTrustStore.ks
-Djavax.net.ssl.trustStorePassword=endeca
-Djavax.net.ssl.keyStore=/endeca_server/ssl/endecaServerClientCert.ks
-Djavax.net.ssl.keyStorePassword=endeca"
```



Note: The formatting above is for clarity on the printed page. In your file, the variables should be entered on one line.

WebLogic Server container on Windows

Add the keystore and truststore files to your WebLogic installation.

In the file `$DOMAIN_HOME/startWebLogic.cmd`, after setting the `USER_MEM_ARGS` variable, set `JAVA_OPTIONS` with the properties for the keystore and the truststore file locations and passwords.

For example:

```
set JAVA_OPTIONS=
-Djavax.net.ssl.trustStore=C:\endeca_server\ssl\endecaServerTrustStore.ks
-Djavax.net.ssl.trustStorePassword=endeca
-Djavax.net.ssl.keyStore=C:\endeca_server\ssl\endecaServerClientCert.ks
-Djavax.net.ssl.keyStorePassword=endeca
```



Note: The formatting above is for clarity on the printed page. In your file, the variables should be entered on one line.

Configuring SSL between Integrator ETL Designer and IAS

You can configure Integrator ETL to securely communicate with IAS using server-only authentication ("one-way") or mutual authentication ("two-way").

To enable server-only authentication, import the IAS certificate into Integrator ETL's truststore. For example, if you are using Java keytool to manage your SSL environment, you run the command:

```
keytool -importcert -v -keystore ietl-truststore.ks -alias <aliasname> -storepass <keystorepass>
-file ias-ca.crt -noprompt
```

where `ietl-truststore.ks` is the name of the Integrator ETL truststore, `ias-ca.crt` is the name of the IAS certificate, and `keystorepass` is the keystore password. This results in Integrator ETL authenticating IAS but not vice versa.

To enable mutual authentication, import the IAS certificate into Integrator ETL's truststore and also import the Integrator ETL certificate into the IAS truststore. For example, if you are using Java keytool to manage your SSL environment, you run two commands:

```
keytool -importcert -v -keystore ietl-truststore.ks -alias <aliasname> -storepass <keystorepass>
-file ias-ca.crt -noprompt
keytool -importcert -v -keystore ias-truststore.ks -alias <aliasname> -storepass <keystorepass>
-file ietl-ca.crt -noprompt
```

where `ietl-truststore.ks` is the example name of the Integrator ETL truststore, `ias-ca.crt` is the name of the IAS certificate, and `keystorepass` is the keystore password. This results in Integrator ETL authenticating IAS and IAS authenticating Integrator ETL.

Database security

Integrator ETL is installed with an embedded Apache Derby database, which is adequate for evaluation. Recommended practice in a production environment is to use an Oracle database.

For details about securing an Oracle database, see the *Oracle Database Security Guide* and *Oracle Database Advanced Security Administrator's Guide*.

Updating the default user

After installation, the default user should be updated.

Integrator ETL Server is installed with a default user named "clover" that has full admin permissions. The default password for this user is "clover".

At a minimum, the default password should be changed. Ideally, you should create a new admin user with full permissions and a unique password, then disable the "clover" user. For details about creating a new user, see "Edit user record" in the *Oracle Endeca Information Discovery Integrator ETL Server Guide*. For details about disabling a user, see "Disabling/enabling users" in the *Oracle Endeca Information Discovery Integrator ETL Server Guide*.



Chapter 3

Implementing Integrator ETL Security

This section describes security features of Integrator ETL.

[Managing users and groups](#)

[Implementing LDAP](#)

[Sandbox permissions](#)

[WebDAV authentication](#)

[Securing graph parameters](#)

Managing users and groups

Integrator ETL Server includes a security module that manages users and groups.

For details about managing users and groups, see "Users and Groups" in the *Oracle Endeca Information Discovery Integrator ETL Server Guide*.

Implementing LDAP

You can use LDAP to authenticate Integrator Server users.

For details about implementing LDAP, see "LDAP authentication" in the *Oracle Endeca Information Discovery Integrator ETL Server Guide*.

Sandbox permissions

Each sandbox is owned by the user that created it. That user can define permissions for other users.

A sandbox is a container for a project on the Integrator ETL Server. A sandbox may exist independently on the server, or may be to a project in Integrator ETL Designer.

The user that creates the sandbox has full control over the sandbox. Users with administrative privileges also have full control over sandboxes. All other users have only the privileges granted by the owner of the sandbox or an administrator.

For details about sandbox privileges, see "Sandbox Content Security and Permissions" in the *Oracle Endeca Information Discovery Integrator ETL Server Guide*.

WebDAV authentication

The WebDAV API of Integrator ETL Server uses HTTP Basic Authentication by default.

You can configure Integrator ETL Server to use HTTP Digest Authentication for WebDAV. Using HTTP Digest Authentication allows you to use WebDAV clients that do not support HTTP Basic Authentication.

To configure Integrator ETL Server to use HTTP Digest Authentication, add the property `security.digest_authentication.features_list` to the configuration properties file. The value of this property should include `/webdav`. Also see "List of Properties" in the *Oracle Endeca Information Discovery Integrator ETL Server Guide*.

Securing graph parameters

You can secure graph parameters in Integrator by setting up a Master Password.

After setting up the Master Password for the Secure Graph Parameters feature, you can make a given graph parameter as secure. For example, your graph can read data from an Oracle database using an encrypted database connection password. The sensitive information in secure parameters is persisted in encrypted form on file system. Decryption of secure parameters is automatically performed in graph runtime.

For details about setting up the Master Password, see the *Oracle Endeca Information Discovery Integrator ETL User's Guide*. To use the Secure Graph Parameters feature in Integrator ETL Server, see "Secure parameters" in the *Oracle Endeca Information Discovery Integrator ETL Server Guide*.



Chapter 4

Configuring SSL in the Integrator Acquisition System

This section describes how to configure the Integrator Acquisition System to use SSL in WebLogic Server installations and in Jetty Web server installations.

[About configuring SSL in the Integrator Acquisition System](#)

[About configuring SSL for IAS in WebLogic Server installations](#)

[About enabling SSL in Jetty Web server installations](#)

About configuring SSL in the Integrator Acquisition System

Configuring SSL in the Integrator Acquisition System enables SSL communication among all the IAS components.

Here is the high-level process to configure SSL for IAS in WebLogic Server installations:

1. Enable SSL for IAS in WebLogic Server. Optionally, enable mutual authentication.
2. Optionally, enable HTTPS redirects.
3. Enable SSL for the IAS Command-line Utilities.
4. Enable the Endeca Web Crawler to write to an SSL-enabled Record Store instance.

The high-level process for Jetty Web server installations is very similar, except you secure the Endeca IAS Service rather than the IAS application in WebLogic:

1. Enable SSL for the Endeca IAS Service. Optionally, enable mutual authentication. On Jetty, HTTPS redirects are enabled by default.
2. Enable SSL for the IAS Command-line Utilities.
3. Enable the Endeca Web Crawler to write to an SSL-enabled Record Store instance.

HTTPS redirects

Although enabling HTTPS redirects is optional, it is highly recommended to simplify IAS configuration. You can use the default IAS ports during installation and then perform minimal configuration to redirect requests from the default port (HTTP) to a secure port (HTTPS).

In Jetty Web server installations, the IAS configuration files have HTTPS redirects enabled by default. In WebLogic Server installations, see [Enabling HTTPS redirects for IAS in WebLogic Server on page 21](#).

Mutual authentication and server-only authentication

The Integrator Acquisition System supports both mutual authentication (also called "two-way" authentication between both client and server) and server-only authentication (also called "one-way" authentication from the server to the client). Oracle recommends configuring your environment for mutual authentication.

Mutual authentication requires a keystore and truststore for clients of IAS. Server-only authentication requires only truststore configuration.

SSL version 3.0

The Integrator Acquisition System supports Version 3.0 of the Secure Sockets Layer (SSL) protocol.

About enecerts, Java keytool, and fully qualified host names

The SSL certificates used for IAS must be issued to the fully qualified host name for the server running IAS. The fully qualified host name must match either the first common name (CN) or any of the subject-alts in the server certificate. A wildcard may occur in the CN and in any of the subject-alts. Also, certificates may be issued to all hosts in a domain by specifying a wildcard such as `*.endeca.com`.

If you generated keystores and truststores by running the `generate_ssl_keys` utility (included as part of the Endeca Server domain running on WebLogic Server), the keystores and truststores do not include the fully qualified host name. You must generate your own keystore and truststore using another utility, for example, Java `keytool`. This is available as part of the Java instance installed with IAS in `<install path>\Oracle\Endeca\IAS\<version>\java\bin\keytool.exe` for Windows and `usr/local/oracle/endeca/IAS/<version>/java/bin/keytool.sh` on UNIX.

When running Java `keytool`, you specify the fully qualified host name or wildcard in response to the prompt "What is your first and last name?" For example:

```
Enter keystore password: endeca
What is your first and last name?
[Unknown]: machine.endeca.com
```

In general, Oracle recommends that you create one truststore for your entire environment (it can contain multiple entries) and a keystore per machine. You can place the truststore in a common directory, for example, `C:\Oracle\Endeca\truststore\truststore.ks` or `/usr/local/oracle/endeca/truststore/truststore.ks`, and then point to that location for IAS configuration.

About configuring SSL for IAS in WebLogic Server installations

The following topics apply only to IAS installations in WebLogic Server.

[Enabling SSL for IAS in WebLogic Server](#)

[Enabling HTTPS redirects for IAS in WebLogic Server](#)

[Enabling the Endeca Web Crawler to write to an SSL enabled Record Store instance](#)

[Enabling SSL for the IAS Command-line Utilities](#)

Enabling SSL for IAS in WebLogic Server

This procedure describes how to secure the IAS application running in WebLogic Server. This allows secure connections from any IAS client to IAS. Secure connections can be either mutually authenticated or authenticated by the server only.

This task requires that you have already created keystore and truststore files.



Note: For simplicity, the paths below use Windows syntax. The UNIX paths are equivalent.

To enable SSL for IAS in WebLogic Server:

1. Stop the IAS application in WebLogic Server.
For example, from a command prompt, run `stopWebLogic` in `C:\Oracle\Middleware\user_projects\domains\ias\bin`.
2. On the machine running WebLogic Server, create a directory to store the keystore and truststore files. Oracle recommends you create an `ssl` directory under `<IAS domain>\config`.
For example, in a default Windows installation, this is `C:\Oracle\Middleware\user_projects\domains\ias\config\ssl`.
3. Copy the keystore file you previously created to `<IAS domain>\config\ssl`.
4. If you want to enable mutual authentication, also copy the truststore file you previously created to `<IAS domain>\config\ssl`.
5. Start the WebLogic Server Administration Console and configure the location of the keystore and truststore files for the IAS application. (This step specifies paths, passphrases, and so on, in the **Configuration>Keystores** tab of the Administration Console.)
This WebLogic task is described in the WebLogic documentation at <http://docs.oracle.com/middleware/1213/wls/WLACH/taskhelp/security/ConfigureKeystoresAndSSL.html>.
6. In the WebLogic Server Administration Console, configure private key alias and password values for the IAS application. (These values are set on the **Configuration>SSL** tab of the Administration Console.)
7. In the WebLogic Server Administration Console, enable the SSL Listen Port and specify a port number of 8402. (These values are set on the **Configuration>General** tab of the Administration Console)
8. If you want to enable mutual authentication, also configure the **Two Way Client Cert Behavior** settings in Administration Console. (These values are set on the **Configuration>SSL>Advanced** section of the Administration Console.)
This WebLogic task is described in the WebLogic documentation at <http://docs.oracle.com/middleware/1213/wls/WLACH/taskhelp/security/ConfigureTwowaySSL.html>.
9. Modify the IAS domain environment script to enable SSL settings:
 - (a) Navigate to `<IAS domain>\bin` on Windows or `<IAS domain>/bin` on UNIX.
 - (b) In a text editor, open either `setIasEnv.cmd` (for Windows) or `setIasEnv.sh` (for UNIX).

- (c) Uncomment the Java options for the truststore location (-Djavax.net.ssl.trustStore), type (-Djavax.net.ssl.trustStoreType), and password (-Djavax.net.ssl.trustStorePassword).
- (d) Replace the tokens listed below for the truststore location path, type, and password values as appropriate for your environment.

Property	Token to replace
javax.net.ssl.trustStore	@TRUSTSTORE_FILE@
javax.net.ssl.trustStoreType	JKS
javax.net.ssl.trustStorePassword	@TRUSTSTORE_PASSWORD@

For example, on Windows, uncomment and modify options similar to the following:

```
REM Set the JVM default trust store and key store locations.
SET JVM_ARGS=-Djavax.net.ssl.trustStore
="C:\Oracle\Endeca\IAS\workspace\conf\truststore.ks" %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.trustStoreType=JKS %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.trustStorePassword=endeca %JVM_ARGS%
```

For example, on UNIX, uncomment and modify options similar to the following:

```
# Setup the Trust Store
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.trustStore=$IAS_WORKSPACE/conf/truststore.ks"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.trustStoreType=JKS"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.trustStorePassword=endeca"
```

- (e) If you enabled mutual authentication, uncomment the Java options for the keystore location (-Djavax.net.ssl.keyStore), type (-Djavax.net.ssl.keyStoreType), and password (-Djavax.net.ssl.keyStorePassword).
- (f) If you uncommented the keystore options, replace the tokens listed below for the keystore location path, type and password values as appropriate for your environment.

Property	Token to replace
javax.net.ssl.keyStore	@KEYSTORE_FILE@
javax.net.ssl.keyStoreType	JKS
javax.net.ssl.keyStorePassword	@KEYSTORE_PASSWORD@

For example, on Windows you uncomment and modify options similar to the following:

```
SET JVM_ARGS=-Djavax.net.ssl.keyStore
="C:\Oracle\Endeca\IAS\workspace\conf\keystore.ks" %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.keyStoreType=JKS %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.keyStorePassword=endeca %JVM_ARGS%
```

For example, on UNIX you uncomment and modify options similar to the following:

```
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.keyStore=$IAS_WORKSPACE/conf/keystore.ks"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.keyStoreType=JKS"
```

```
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.keyStorePassword=endeca"
```

(g) Save and close `setIasEnv.cmd` (for Windows) or `setIasEnv.sh` (for UNIX).

10. Re-start IAS in WebLogic Server.

For example, from a command prompt, run `startWebLogic` in `C:\Oracle\Middleware\user_projects\domains\ias\bin`.

You can confirm that SSL is enabled for the IAS application by starting a Web browser and loading the IAS Server WSDL on the SSL port, or if you have redirects enabled, loading it on the non-SSL port. For example, specify `https://<hostname>:8402/ias-server/ias/?wsdl`, or if you have redirects enabled specify `http://<hostname>:8401/ias-server/ias/?wsdl`. The following WSDL displays:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions name="IasCrawlerService" targetNamespace="http://endeca.com/eidi/ias
/2011-12" xmlns:ns1="http://schemas.xmlsoap.org/soap/http" xmlns:soap="http://schemas.xmlsoap.org
/wsd/soap/" xmlns:tns="http://endeca.com/eidi/ias/2011-12" xmlns:wsdl="http://schemas.xmlsoap.org
/wsd/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <wsdl:types>
...

```

Enabling HTTPS redirects for IAS in WebLogic Server

If both the IAS port and IAS port for SSL are enabled, you can configure HTTPS redirects so that any requests using HTTP (non-secure) are automatically redirected to the same URL using HTTPS.

To enable HTTPS redirects for IAS in WebLogic Server:

1. Ensure that the IAS port for SSL is enabled in WebLogic Server.
Remember that in WebLogic Server, the IAS port for SSL is set on the **Configuration>General** tab of the Administration Console. See the **SSL Listen Port** setting.
2. On the file system, navigate to `<install path>\IAS\3.2.0\webapps\weblogic` and open `Plan.xml` in a text editor.
3. Ensure that `transport-guarantee` is set to `CONFIDENTIAL`.

For example:

```
<variable>
  <name>transport-guarantee</name>
  <value>CONFIDENTIAL</value>
</variable>
```

4. Save and close `Plan.xml`.
5. Apply the updated `Plan.xml` to IAS using the WebLogic Server Administration Console:
 - (a) Open the Administration Console at `http://<hostname>:8401/console` and log in.
 - (b) Click **Lock & Edit**.
 - (c) Under Domain Structure, click Deployments.
 - (d) In the Deployments list, check the box for the IAS Server deployment and click **Update**.
 - (e) In **Deployment plan path**, click the **Change Path**.
 - (f) In the **Path** field, specify the full path to `Plan.xml` including the file name.
 - (g) Click the radio button next to the file name.

(h) Click **Next** and then click **Finish**.

(i) Click **Activate Changes**.

You can confirm that redirects are enabled for the IAS application by starting a Web browser and loading the IAS Server WSDL on the non-SSL port. If redirects are working correctly, the browser redirects to the SSL listen port and loads the WSDL. For example, specify `https://<hostname>:8401/ias-server/ias/?wsdl`. The browser redirects to `http://<hostname>:8402/ias-server/ias/?wsdl` and the following WSDL displays:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions name="IasCrawlerService" targetNamespace="http://endeca.com/eidi/ias
/2011-12" xmlns:ns1="http://schemas.xmlsoap.org/soap/http" xmlns:soap="http://schemas.xmlsoap.org
/wsd/soap/" xmlns:tns="http://endeca.com/eidi/ias/2011-12" xmlns:wsdl="http://schemas.xmlsoap.org
/wsd/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <wsdl:types>
...

```

Optionally, if you need to disable HTTPS redirects, update the `transport-guarantee` to `NONE` and re-apply the updated `Plan.xml` to IAS as described above.

Enabling the Endeca Web Crawler to write to an SSL enabled Record Store instance

This procedure establishes a secure connection between the Endeca Web Crawler and the Record Store Web service running in IAS.

This task requires that you have already configured the Endeca Web Crawler to write to a Record Store instance. If you have not done so, see "Configuring Web crawls to write output to a Record Store instance" in the *IAS Web Crawler Guide*.

To enable SSL in the Endeca Web Crawler:

1. Navigate to `<install path>\IAS\<version>\bin` on Windows or `<install path>/IAS/<version>/bin` on UNIX.
2. Open either `web-crawler.bat` (for Windows) or `web-crawler.sh` (for UNIX) in a text editor.
3. Uncomment the Java options for the truststore location (`-Djavax.net.ssl.trustStore`), type (`-Djavax.net.ssl.trustStoreType`), and password (`-Djavax.net.ssl.trustStorePassword`).
4. Replace the tokens listed below for the truststore location path, type, and password values as appropriate for your environment.

Property	Token to replace
<code>javax.net.ssl.trustStore</code>	@TRUSTSTORE_FILE@
<code>javax.net.ssl.trustStorePassword</code>	@TRUSTSTORE_PASSWORD@
<code>javax.net.ssl.trustStoreType</code>	JKS

For example, on Windows, uncomment and modify options similar to the following:

```
SET JVM_ARGS=-Djavax.net.ssl.trustStore
="C:\Oracle\Endeca\IAS\workspace\conf\truststore.ks" %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.trustStoreType=JKS %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.trustStorePassword=endeca %JVM_ARGS%
```

For example, on UNIX, uncomment and modify options similar to the following:

```
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.trustStore=$IAS_WORKSPACE/conf/truststore.ks"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.trustStoreType=JKS"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.trustStorePassword=endeca"
```

- If you enabled mutual authentication in `jetty.xml`, uncomment Java options for the keystore location (`-Djavax.net.ssl.keyStore`), type (`-Djavax.net.ssl.keyStoreType`), and password (`-Djavax.net.ssl.keyStorePassword`).
- If you uncommented the keystore options, replace the tokens listed below for the keystore location path, type and password values as appropriate for your environment.

Property	Token to replace
<code>javax.net.ssl.keyStore</code>	@KEYSTORE_FILE@
<code>javax.net.ssl.keyStorePassword</code>	@KEYSTORE_PASSWORD@
<code>javax.net.ssl.keyStoreType</code>	JKS

For example, on Windows you uncomment and modify options similar to the following:

```
SET JVM_ARGS=-Djavax.net.ssl.keyStore
="C:\Oracle\Endeca\IAS\workspace\conf\keystore.ks" %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.keyStoreType=JKS %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.keyStorePassword=endeca %JVM_ARGS%
```

For example, on UNIX you uncomment and modify options similar to the following:

```
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.keyStore=$IAS_WORKSPACE/conf/keystore.ks"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.keyStoreType=JKS"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.keyStorePassword=endeca"
```

- Save and close either `web-crawler.bat` (for Windows) or `web-crawler.sh` (for UNIX).
- If you disabled HTTPS redirects, open either `default.xml` (for global configuration of the Endeca Web Crawler) or `site.xml` (for per-crawl property overrides) in a text editor. (You do not need to change this file or the `isPortSsl` property if you are using HTTPS redirects.)
- Add the `isPortSsl` configuration property to specify how to interpret the IAS port setting. Setting the property to `true` instructs the Web Crawler to use the IAS port as an HTTPS port. Setting the property to `false` means the IAS port is treated as an HTTP port. If you enabled redirects for the Endeca IAS Service, you typically set the property to `false` and let the redirect forward the request to the SSL port.

For example:

```
<property>
  <name>output.recordStore.isPortSsl</name>
  <value>>false</value>
  <description>
    Whether or not to use the IAS port as an HTTPS port to
    communicate with the record store service.
    Default: false
```

```
</description>
</property>
```

Enabling SSL for the IAS Command-line Utilities

This procedure establishes a secure connection between all of the IAS Command-line Utilities and the Web services running in the Endeca IAS Service. Recall that the IAS Command-line Utilities include the IAS Server Command-line Utility, the Component Instance Manager Command-line Utility, and the Record Store Command-line Utility.

To enable SSL in the IAS command-line utilities:

1. Navigate to <install path>\IAS\<version>\bin on Windows or <install path>/IAS/<version>/bin on UNIX.
2. To enable SSL in the IAS Server Command-line Utility, open either `ias-cmd.bat` (for Windows) or `ias-cmd.sh` (for UNIX) in a text editor.
3. Uncomment the Java options for the truststore location (`-Djavax.net.ssl.trustStore`), type (`-Djavax.net.ssl.trustStoreType`), and password (`-Djavax.net.ssl.trustStorePassword`).
4. Replace the tokens listed below for the truststore location path, type, and password values as appropriate for your environment.

Property	Token to replace
<code>javax.net.ssl.trustStore</code>	@TRUSTSTORE_FILE@
<code>javax.net.ssl.trustStorePassword</code>	@TRUSTSTORE_PASSWORD@
<code>javax.net.ssl.trustStoreType</code>	JKS

For example, on Windows, uncomment and modify options similar to the following:

```
REM Setup the Trust Store
SET JVM_ARGS=-Djavax.net.ssl.trustStore
="C:\Oracle\Endeca\IAS\workspace\conf\truststore.ks" %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.trustStoreType=JKS %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.trustStorePassword=endeca %JVM_ARGS%
```

For example, on UNIX, uncomment and modify options similar to the following:

```
# Setup the Trust Store
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.trustStore=$IAS_WORKSPACE/conf/truststore.ks"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.trustStoreType=JKS"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.trustStorePassword=endeca"
```

5. If you enabled mutual authentication in `jetty.xml`, uncomment the Java options for the keystore location (`-Djavax.net.ssl.keyStore`), type (`-Djavax.net.ssl.keyStoreType`), and password (`-Djavax.net.ssl.keyStorePassword`).

6. If you uncommented the keystore options, replace the tokens listed below for the keystore location path, type and password values as appropriate for your environment.

Property	Token to replace
javax.net.ssl.keyStore	@KEYSTORE_FILE@
javax.net.ssl.keyStorePassword	@KEYSTORE_PASSWORD@
javax.net.ssl.keyStoreType	JKS

For example, on Windows you uncomment and modify options similar to the following:

```
SET JVM_ARGS=-Djavax.net.ssl.keyStore
="C:\Oracle\Endeca\IAS\workspace\conf\keystore.ks" %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.keyStoreType=JKS %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.keyStorePassword=endeca %JVM_ARGS%
```

For example, on UNIX you uncomment and modify options similar to the following:

```
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.keyStore=$IAS_WORKSPACE/conf/keystore.ks"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.keyStoreType=JKS"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.keyStorePassword=endeca"
```

7. Save and close either `ias-cmd.bat` (for Windows) or `ias-cmd.sh`.
8. Similarly, to enable SSL and mutual authentication in the CIM Command-line Utility (`component-manager-cmd`) and the Record Store Command-line Utility (`recordstore-cmd`):
 - (a) Open the batch or shell files.
 - (b) Uncomment the truststore and keystore properties listed above.
 - (c) Modify the property values as appropriate for your environment.
 - (d) Save and close the files.
9. If you disabled HTTPS redirects, navigate to `%ENDECA_TOOLS_CONF%\conf` on Windows or `$ENDECA_TOOLS_CONF/conf` on UNIX. (You do not need to perform this step if using HTTPS redirects.)
 - (a) Open `commandline.properties` in a text editor.
 - (b) Modify the following properties:

Property name	Description
com.endeca.eidi.ias.server.host	Specify the fully qualified name of the machine running the command-line utility.
com.endeca.eidi.ias.server.port	Specify the port for Endeca IAS Service.

Property name	Description
com.endeca.eidi.ias.server.isPortSsl	<p>Specify how to interpret the IAS port setting.</p> <p>A value of <code>true</code> means that <code>com.endeca.eidi.ias.server.port</code> is an SSL port and IAS uses HTTPS for connections.</p> <p>A value of <code>false</code> means that <code>com.endeca.eidi.ias.server.port</code> is a non-SSL port and IAS uses HTTP for connections. The default is <code>false</code>.</p> <p>Specify <code>false</code> if you enabled HTTPS redirects.</p>

(c) Save and close `commandline.properties`.

About enabling SSL in Jetty Web server installations

The following topics apply only to IAS installations in Jetty Web server.

[Enabling SSL for the Endeca IAS Service](#)

[Enabling SSL for the IAS Command-line Utilities](#)

[Enabling the Endeca Web Crawler to write to an SSL enabled Record Store instance](#)

Enabling SSL for the Endeca IAS Service

This procedure establishes a secure connection from any IAS client to the Web services running inside the Endeca IAS Service. Recall that the IAS Server, Component Instance Manager, and Record Store Web services run inside the Endeca IAS Service. This task also explains how to disable mutual authentication for clients of the Endeca IAS Service and explains how to disable HTTPS redirects (mutual authentication and HTTPS redirects are enabled by default).

This task requires that you have already created keystore and truststore files for the server running the Endeca IAS Service.



Note: For simplicity, the paths below use Windows syntax. The UNIX paths are equivalent.

To enable SSL for the Endeca IAS Service:

1. Stop the Endeca IAS Service.
2. Copy the keystore file you previously created to `<install path>\IAS\workspace\conf`.
3. Optionally, to enable mutual authentication, copy the truststore file you previously created to `<install path>\IAS\workspace\conf`.
4. In `<install path>\IAS\workspace\conf`, open `jetty.xml` in a text editor.

5. Uncomment the section with SSL configuration properties.
6. In the SSL configuration section, replace the tokens listed below for the fully qualified host, SSL port, truststore, and keystore properties on the machine running IAS.

Property	Token to replace
com.endeca.ias.fullyQualifiedHostName	@IASHOST@
com.endeca.ias.ssl.port	@IASSSLPORT@
javax.net.ssl.trustStore	@TRUSTSTORE_FILE@
javax.net.ssl.trustStorePassword	@TRUSTSTORE_PASSWORD@
javax.net.ssl.trustStoreType	JKS
javax.net.ssl.keystore	@KEYSTORE_FILE@
javax.net.ssl.keystorePassword	@KEYSTORE_PASSWORD@
javax.net.ssl.keystoreType	JKS

Ensure that host names are fully qualified. Also, Oracle recommends relative paths for truststore and keystore locations. The paths should be relative to IAS\<<version>.

For example:

```
<Configure id="Server" class="org.mortbay.jetty.Server">
  <!-- Redirect java.util.logging to slf4j -->
  <Call class="org.slf4j.bridge.SLF4JBridgeHandler" name="install"/>

  <!-- Set the com.endeca.ias.port with the port we will run on -->
  <Call class="java.lang.System" name="setProperty">
    <Arg>com.endeca.ias.port</Arg>
    <Arg><SystemProperty name="com.endeca.ias.port" default="8401"/></Arg>
  </Call>

  <!-- If the com.endeca.ias.ssl.port is set. IAS will default to using
  this port and use HTTPS to communicate with IAS Component instances.
  Make sure to also add a secure data source by uncommenting the proper
  section in this file.

  Make sure the trust store and key store are set properly.
  If using self-signed certificate, you need to also put the
  certificate in the truststore.-->

  <Call class="java.lang.System" name="setProperty">
    <Arg>com.endeca.ias.ssl.port</Arg>
    <Arg><SystemProperty name="com.endeca.ias.ssl.port" default="8402"/></Arg>
  </Call>

  <Call class="java.lang.System" name="setProperty">
    <Arg>com.endeca.ias.fullyQualifiedHostName</Arg>
    <Arg>hostname.eng.endeca.com</Arg>
  </Call>

  <Call class="java.lang.System" name="setProperty">
    <Arg>javax.net.ssl.trustStore</Arg>
```

```

        <Arg><SystemProperty name="jetty.home" default="." />../workspace/conf
        /truststore.ks</Arg>

        <Call class="java.lang.System" name="setProperty">
          <Arg>javax.net.ssl.trustStorePassword</Arg>
          <Arg>endeca</Arg>
        </Call>

        <Call class="java.lang.System" name="setProperty">
          <Arg>javax.net.ssl.trustStoreType</Arg>
          <Arg>JKS</Arg>
        </Call>

        <Call class="java.lang.System" name="setProperty">
          <Arg>javax.net.ssl.keyStore</Arg>
          <Arg><SystemProperty name="jetty.home" default="." />../workspace/conf/keystore.ks<
        /Arg>

        <Call class="java.lang.System" name="setProperty">
          <Arg>javax.net.ssl.keyStorePassword</Arg>
          <Arg>endeca</Arg>
        </Call>

        <Call class="java.lang.System" name="setProperty">
          <Arg>javax.net.ssl.keyStoreType</Arg>
          <Arg>JKS</Arg>
        </Call>

```

7. Also in `jetty.xml`, locate the data source section and enable the `SslSocketdata` source class by uncommenting the following:

```

<Call name="adddata source">
  <Arg>
    <New class="org.mortbay.jetty.security.SslSocketdata source">
      <Set name="Port"><SystemProperty name="com.endeca.ias.ssl.port"/></Set>
      <Set name="maxIdleTime">30000</Set>
      <Set name="keystore"><SystemProperty
name=" javax.net.ssl.keyStore"/></Set>
      <Set name="keyPassword"><SystemProperty
name=" javax.net.ssl.keyStorePassword"/></Set>
      <Set name="truststore"><SystemProperty
name=" javax.net.ssl.trustStore"/></Set>
      <Set name="trustPassword"><SystemProperty
name=" javax.net.ssl.trustStorePassword"/></Set>
      <!-- set this to false if you want mutual authentication to
be turned off -->
      <Set name="needClientAuth">true</Set>
    </New>
  </Arg>
</Call>

```

8. Optionally, to disable mutual authentication, set the `needClientAuth` shown above to `false`. The default is `true` (mutual authentication is enabled).
9. Optionally, to disable HTTPS redirects, comment out the `SelectChanneldata` source data source or delete the data source. By default, HTTPS redirects are enabled. (A non-secure HTTP request is redirected to a secure SSL port. The default SSL port for IAS is 8402.)

For example, to disable redirects, comment out the following:

```

<!-- <Call name="adddata source">
  <Arg>
    <New class="org.mortbay.jetty.nio.SelectChanneldata source">
      <Set name="port"><SystemProperty name="com.endeca.ias.port"/></Set>
      <Set name="maxIdleTime">30000</Set>
      <Set name="Acceptors">2</Set>
    </New>
  </Arg>
</Call>

```

```

        <Set name="statsOn">false</Set>
        <Set name="confidentialPort"><SystemProperty
            name="com.endeca.ias.ssl.port" default="8402"/></Set>
            <Set name="lowResourcesConnections">5000</Set>
            <Set name="lowResourcesMaxIdleTime">5000</Set>
        </New>
    </Arg>
</Call> -->

```

10. If you are using HTTPS redirects to a port other than the IAS SSL default (8402), modify default="8402" to the appropriate port for your environment.
11. Save and close jetty.xml.
12. In the same directory (<install path>\IAS\workspace\conf), open webdefault.xml in a text editor.
13. Uncomment the user-data-constraint section.

For example:

```

<security-constraint>
  <user-data-constraint>
    <transport-guarantee>
      CONFIDENTIAL
    </transport-guarantee>
  </user-data-constraint>
  <web-resource-collection>
    <url-pattern>*/</url-pattern>
  </web-resource-collection>
</security-constraint>

```

14. Save and close webdefault.xml.
15. Start the Endeca IAS Service.

You can confirm that SSL is enabled for the Endeca IAS Service by starting a Web browser and loading the IAS Server WSDL on the IAS SSL port, or if you have redirects enabled, loading it on the non-SSL port. For example, specify `https://hostname:8402/ias/?wsdl`, or if you have redirects enabled specify `http://hostname:8401/ias/?wsdl`. The following WSDL displays:

```

<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions name="IasCrawlerService" targetNamespace="http://endeca.com/eidi/ias
/2011-12" xmlns:ns1="http://schemas.xmlsoap.org/soap/http" xmlns:soap="http://schemas.xmlsoap.org
/wsdl/soap/" xmlns:tns="http://endeca.com/eidi/ias/2011-12" xmlns:wsdl="http://schemas.xmlsoap.org
/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <wsdl:types>
...

```

Enabling SSL for the IAS Command-line Utilities

This procedure establishes a secure connection between all of the IAS Command-line Utilities and the Web services running in the Endeca IAS Service. Recall that the IAS Command-line Utilities include the IAS Server Command-line Utility, the Component Instance Manager Command-line Utility, and the Record Store Command-line Utility.

To enable SSL in the IAS command-line utilities:

1. Navigate to <install path>\IAS\<version>\bin on Windows or <install path>/IAS/<version>/bin on UNIX.

2. To enable SSL in the IAS Server Command-line Utility, open either `ias-cmd.bat` (for Windows) or `ias-cmd.sh` (for UNIX) in a text editor.
3. Uncomment the Java options for the truststore location (`-Djavax.net.ssl.trustStore`), type (`-Djavax.net.ssl.trustStoreType`), and password (`-Djavax.net.ssl.trustStorePassword`).
4. Replace the tokens listed below for the truststore location path, type, and password values as appropriate for your environment.

Property	Token to replace
<code>javax.net.ssl.trustStore</code>	@TRUSTSTORE_FILE@
<code>javax.net.ssl.trustStorePassword</code>	@TRUSTSTORE_PASSWORD@
<code>javax.net.ssl.trustStoreType</code>	JKS

For example, on Windows, uncomment and modify options similar to the following:

```
REM Setup the Trust Store
SET JVM_ARGS=-Djavax.net.ssl.trustStore
="C:\Oracle\Endeca\IAS\workspace\conf\truststore.ks" %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.trustStoreType=JKS %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.trustStorePassword=endeca %JVM_ARGS%
```

For example, on UNIX, uncomment and modify options similar to the following:

```
# Setup the Trust Store
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.trustStore=$IAS_WORKSPACE/conf/truststore.ks"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.trustStoreType=JKS"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.trustStorePassword=endeca"
```

5. If you enabled mutual authentication in `jetty.xml`, uncomment the Java options for the keystore location (`-Djavax.net.ssl.keyStore`), type (`-Djavax.net.ssl.keyStoreType`), and password (`-Djavax.net.ssl.keyStorePassword`).
6. If you uncommented the keystore options, replace the tokens listed below for the keystore location path, type and password values as appropriate for your environment.

Property	Token to replace
<code>javax.net.ssl.keyStore</code>	@KEYSTORE_FILE@
<code>javax.net.ssl.keyStorePassword</code>	@KEYSTORE_PASSWORD@
<code>javax.net.ssl.keyStoreType</code>	JKS

For example, on Windows you uncomment and modify options similar to the following:

```
SET JVM_ARGS=-Djavax.net.ssl.keyStore
="C:\Oracle\Endeca\IAS\workspace\conf\keystore.ks" %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.keyStoreType=JKS %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.keyStorePassword=endeca %JVM_ARGS%
```

For example, on UNIX you uncomment and modify options similar to the following:

```
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.keyStore=$IAS_WORKSPACE/conf/keystore.ks"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.keyStoreType=JKS"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.keyStorePassword=endeca"
```

7. Save and close either `ias-cmd.bat` (for Windows) or `ias-cmd.sh`.
8. Similarly, to enable SSL and mutual authentication in the CIM Command-line Utility (`component-manager-cmd`) and the Record Store Command-line Utility (`recordstore-cmd`):
 - (a) Open the batch or shell files.
 - (b) Uncomment the truststore and keystore properties listed above.
 - (c) Modify the property values as appropriate for your environment.
 - (d) Save and close the files.
9. If you disabled HTTPS redirects, navigate to `%ENDECA_TOOLS_CONF%\conf` on Windows or `$ENDECA_TOOLS_CONF/conf` on UNIX. (You do not need to perform this step if using HTTPS redirects.)
 - (a) Open `commandline.properties` in a text editor.
 - (b) Modify the following properties:

Property name	Description
<code>com.endeca.eidi.ias.server.host</code>	Specify the fully qualified name of the machine running the command-line utility.
<code>com.endeca.eidi.ias.server.port</code>	Specify the port for Endeca IAS Service.
<code>com.endeca.eidi.ias.server.isPortSsl</code>	Specify how to interpret the IAS port setting. A value of <code>true</code> means that <code>com.endeca.eidi.ias.server.port</code> is an SSL port and IAS uses HTTPS for connections. A value of <code>false</code> means that <code>com.endeca.eidi.ias.server.port</code> is a non-SSL port and IAS uses HTTP for connections. The default is <code>false</code> . Specify <code>false</code> if you enabled HTTPS redirects.

- (c) Save and close `commandline.properties`.

Enabling the Endeca Web Crawler to write to an SSL enabled Record Store instance

This procedure establishes a secure connection between the Endeca Web Crawler and the Record Store Web service running in IAS.

This task requires that you have already configured the Endeca Web Crawler to write to a Record Store instance. If you have not done so, see "Configuring Web crawls to write output to a Record Store instance" in the *IAS Web Crawler Guide*.

To enable SSL in the Endeca Web Crawler:

1. Navigate to <install path>\IAS\<version>\bin on Windows or <install path>/IAS/<version>/bin on UNIX.
2. Open either `web-crawler.bat` (for Windows) or `web-crawler.sh` (for UNIX) in a text editor.
3. Uncomment the Java options for the truststore location (`-Djavax.net.ssl.trustStore`), type (`-Djavax.net.ssl.trustStoreType`), and password (`-Djavax.net.ssl.trustStorePassword`).
4. Replace the tokens listed below for the truststore location path, type, and password values as appropriate for your environment.

Property	Token to replace
<code>javax.net.ssl.trustStore</code>	@TRUSTSTORE_FILE@
<code>javax.net.ssl.trustStorePassword</code>	@TRUSTSTORE_PASSWORD@
<code>javax.net.ssl.trustStoreType</code>	JKS

For example, on Windows, uncomment and modify options similar to the following:

```
SET JVM_ARGS=-Djavax.net.ssl.trustStore
="C:\Oracle\Endeca\IAS\workspace\conf\truststore.ks" %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.trustStoreType=JKS %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.trustStorePassword=endeca %JVM_ARGS%
```

For example, on UNIX, uncomment and modify options similar to the following:

```
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.trustStore=$IAS_WORKSPACE/conf/truststore.ks"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.trustStoreType=JKS"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.trustStorePassword=endeca"
```

5. If you enabled mutual authentication in `jetty.xml`, uncomment Java options for the keystore location (`-Djavax.net.ssl.keyStore`), type (`-Djavax.net.ssl.keyStoreType`), and password (`-Djavax.net.ssl.keyStorePassword`).
6. If you uncommented the keystore options, replace the tokens listed below for the keystore location path, type and password values as appropriate for your environment.

Property	Token to replace
<code>javax.net.ssl.keyStore</code>	@KEYSTORE_FILE@

Property	Token to replace
javax.net.ssl.keyStorePassword	@KEYSTORE_PASSWORD@
javax.net.ssl.keyStoreType	JKS

For example, on Windows you uncomment and modify options similar to the following:

```
SET JVM_ARGS=-Djavax.net.ssl.keyStore
=C:\Oracle\Endeca\IAS\workspace\conf\keystore.ks" %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.keyStoreType=JKS %JVM_ARGS%
SET JVM_ARGS=-Djavax.net.ssl.keyStorePassword=endeca %JVM_ARGS%
```

For example, on UNIX you uncomment and modify options similar to the following:

```
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.keyStore=$IAS_WORKSPACE/conf/keystore.ks"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.keyStoreType=JKS"
JVM_ARGS="$JVM_ARGS -Djavax.net.ssl.keyStorePassword=endeca"
```

7. Save and close either `web-crawler.bat` (for Windows) or `web-crawler.sh` (for UNIX).
8. If you disabled HTTPS redirects, open either `default.xml` (for global configuration of the Endeca Web Crawler) or `site.xml` (for per-crawl property overrides) in a text editor. (You do not need to change this file or the `isPortSsl` property if you are using HTTPS redirects.)
9. Add the `isPortSsl` configuration property to specify how to interpret the IAS port setting. Setting the property to `true` instructs the Web Crawler to use the IAS port as an HTTPS port. Setting the property to `false` means the IAS port is treated as an HTTP port. If you enabled redirects for the Endeca IAS Service, you typically set the property to `false` and let the redirect forward the request to the SSL port.

For example:

```
<property>
  <name>output.recordStore.isPortSsl</name>
  <value>>false</value>
  <description>
    Whether or not to use the IAS port as an HTTPS port to
    communicate with the record store service.
    Default: false
  </description>
</property>
```



Chapter 5

Configuring SSL in the IKM SQL to Endeca Server

This section describes how to configure the IKM SQL to Endeca Server to connect to an SSL-enabled Endeca Server and the section also refers to documentation that describes how to secure optional knowledge modules within Oracle Data Integrator.

[About configuring SSL in the IKM SQL to Endeca Server](#)

[About securing the IKM SQL to Endeca Server within Oracle Data Integrator](#)

About configuring SSL in the IKM SQL to Endeca Server

The IKM SQL to Endeca Server has a number of configuration options that allow the module to connect to an SSL-enabled Endeca Server. The options specify paths to the same keystore and truststore files used by Endeca Server.

For details about these options and how to set them, see the "Configuration reference for the IKM SQL to Endeca Server" topic in the *IKM SQL to Endeca Server Installation and Usage Guide*.

About securing the IKM SQL to Endeca Server within Oracle Data Integrator

You install the IKM SQL to Endeca Server as an optional module into Oracle Data Integrator. Because ODI is hosting the module, ODI handles security issues such as access control, user management, SSL configuration, and so on.

For details about security in ODI, see "Managing the Security in Oracle Data Integrator" in *Oracle® Fusion Middleware Developer's Guide for Oracle Data Integrator*.



Chapter 6

Security Considerations for Integrator ETL Developers

This section contains security information useful to data developers working with Integrator ETL.

[Securing individual components](#)

[Encrypting database access passwords](#)

[SOAP authentication](#)

Securing individual components

All Information Discovery components support SSL connections to an SSL-enabled Endeca Server.

Information Discovery components that communicate with other applications such as the Endeca Server include an **SSL Enabled** configuration property. When the value of this property is set to *true*, the component uses SSL to communicate with the other application.

Integrator ETL Designer must also be configured for SSL. See [Configuring SSL in Integrator ETL Designer on page 12](#) for details about configuring SSL in Integrator ETL Designer. If SSL is not configured in Integrator ETL Designer, graphs including components configured to support SSL will fail.

[Configuring SSL for the Web Service Client component](#)

Configuring SSL for the Web Service Client component

The **Web Service Client** component requires special configuration to support secure communication with Endeca Server over SSL.

If you want to use the **Web Service Client** in an SSL environment, you must check the **Disable SSL Certificate Validation** box on the instance of the **Web Service Client** in your graph.

You must also ensure that SSL is enabled in Integrator ETL Designer and Integrator ETL Server. For details, see [Configuring SSL on page 12](#).

Encrypting database access passwords

Database access passwords should be encrypted for security.

Database access passwords are stored in project configuration files or in the graphs unless encrypted. To ensure secure access to any databases you connect to, you should encrypt all database access passwords. For details, see "Encrypting the Access Password" in the *Oracle Endeca Information Discovery Integrator ETL Designer Guide*.

SOAP authentication

The exposed SOAP service in Integrator ETL Server is stateless.

Any SOAP clients you implement must pass an authentication session token as a parameter of each operation. SOAP clients can obtain an authentication session token by calling the `login` operation.

Index

A

- additional information finding 10

C

- clover user 14
- components 35
- container 10

D

- database 14
- database access passwords 35
- default user 14
- deployment configurations 8

F

- file system 11

G

- graph parameters, securing 16
- groups 15

H

- HTTPS redirects 21

I

- IKM SQL to Endeca Server 34

L

- LDAP 15
- Linux 9

M

- master password, setting 16

O

- obtaining additional information 10
- ODI security 34
- Operating System 9

S

- sandbox 15
- secure graph parameters 16
- SOAP 36
- SSL 12
 - components 35
 - ETL and IAS 13
 - Integrator ETL Designer 12
 - Integrator ETL Server 13
 - Web Service Client 35

T

- Tomcat 10

U

- users 15

W

- WebDav 16
- WebLogic 10
- Windows 9