Oracle® Endeca Information Discovery Integrator

Security Guide for Integrator

Version 3.0.0 • May 2013

ORACLE

Copyright and disclaimer

Table of Contents

Preface

Oracle® Endeca Information Discovery Integrator provides a suite of products to load data from disparate source systems and store it for use in an Endeca Server data domain. The Integrator products include:

- Integrator ETL - Integrator ETL is a high-performance data integration platform that lets you extract source records from a variety of sources and sends that data to the Data Ingest Web Service, which in turn loads the records into the Oracle Endeca Server.
- Integrator Acquisition System - The Integrator Acquisition System, or IAS, is a set of components that crawl source data stored in a variety of formats including: file systems, delimited files, JDBC databases, and custom data sources. IAS transforms the data, if necessary, and outputs the data to an XML file or a Record Store instance that can be accessed by Integrator ETL for use in the Endeca Server.
- IKM SQL to Endeca Server - provides integration and loading modules that enable writing source data to an Endeca Server target within Oracle Data Integrator.

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.

Table 0.1: 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.

Table 0.2: 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 Designer, Integrator Server, Integrator Acquisition System, and their communication with Endeca Server.

Overview of security in Integrator

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

Integrator consists of the following parts:

- Integrator Designer (often referred to as simply Integrator) 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 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 systems, Content Management Systems, 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 for use in the Endeca Server.

Integrator Server is not required to load data to Endeca Server data domains. Integrator 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 Designer, and would need to run graphs manually. In an enterprise environment, it is more effective to use Integrator Server, which allows multiple users and groups of users to access and run graphs. Moreover, Integrator Server provides the ability to automatically schedule graphs to run at designated times. Integrator Server also provides tools for monitoring the progress and success of graph execution.

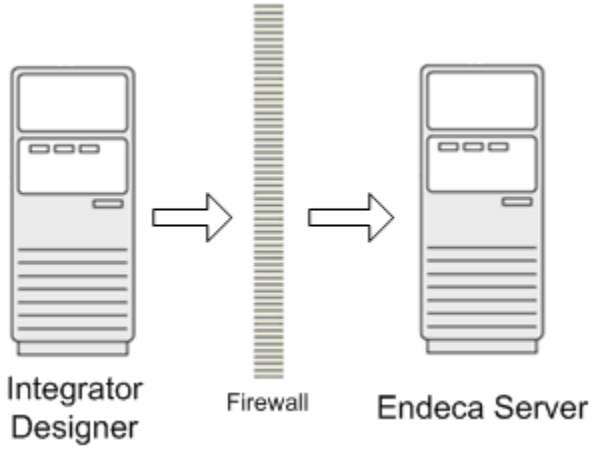
Integrator Acquisition System, Integrator Designer, and Integrator Server use HTTP/HTTPS to communicate. They also use HTTP/HTTPS to communicate with Endeca Server. These protocols work well with complex network setups and firewalls.

Deployment configurations

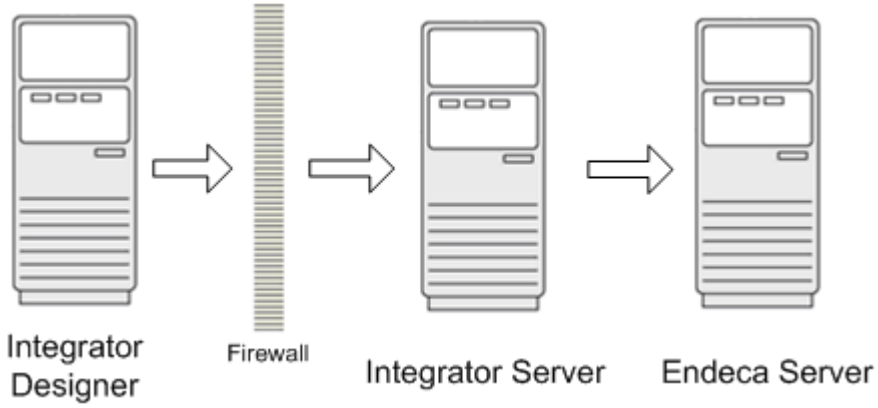
This topic illustrates typical deployment scenarios for Integrator.

The simplest deployment configuration is Integrator 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.

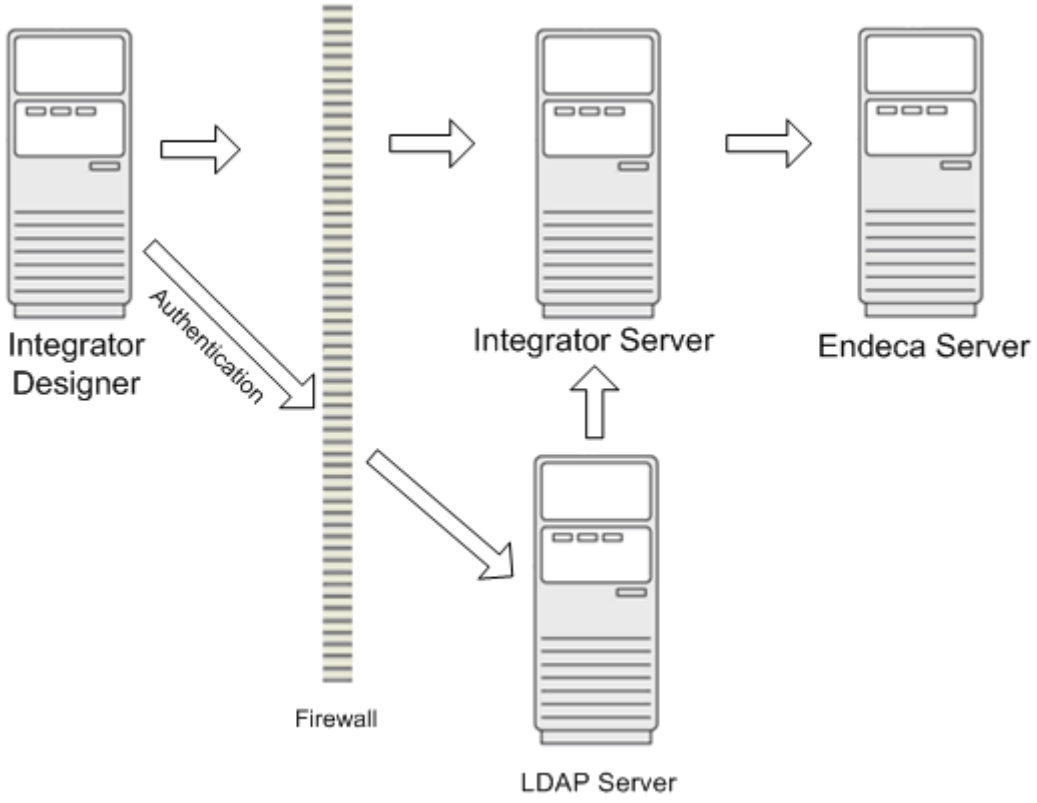
A more complex deployment consists of Integrator Designer deployed on one machine and Endeca Server deployed on another machine. In this scenario, Endeca Server may be deployed behind a firewall.



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

The preferred operating system for Integrator Designer and Integrator Server is Red Hat Enterprise Linux 5. 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*
- *http://www.nsa.gov/ia/_files/factsheets/rhel5-pamphlet-i731.pdf*

Note the following exceptions to the recommendations above:

- If you are installing Integrator 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 Server is 7001.
  - HTTPS

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

The preferred container for Integrator Server is WebLogic Server 10.3.6 . For details about securing and hardening WebLogic Server, see *http://docs.oracle.com/cd/E23943_01/web.1111/e13707/toc.htm*.

Integrator 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 User's Guide*, *Integrator Designer Guide*, and *Integrator Server Guide* describe how to use Integrator.

Chapter 2

Securing Integrator Installations

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

Securing the file system

This topic provides recommendations for securing the installed file system.

Securing the Integrator Designer file system

Integrator 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 Server file system

When installing Integrator 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 between Integrator Acquisition System, Integrator Designer, and Integrator Server. They can also use SSL to communicate with Endeca Server.

Copy the keystore and truststore files from the Endeca Server to the Integrator Acquisition System, Integrator Designer, and the Integrator Server.

Configuring SSL in Integrator Designer

This topic describes how to configure SSL in Integrator Designer.

To configure SSL in Integrator Designer:

1. Copy the keystore and truststore files from your Endeca Server installation to a location in your Integrator Designer installation.
2. Start Integrator 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 `integrator.ini` file in the root of your Integrator 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 26*.

Configuring Integrator Server to support SSL

To implement SSL on Integrator Server, copy the keystore and truststore files from the Endeca Server installation to the Integrator 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.

Database security

Integrator 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 *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 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 Server Guide*. For details about disabling a user, see "Disabling/enabling users" in the *Oracle Endeca Information Discovery Integrator Server Guide.*

Chapter 3

Implementing Integrator Security

This section describes security features of Integrator.

Managing users and groups

Integrator 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 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 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 Server. A sandbox may exist independently on the server, or may be to a project in Integrator 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 Server Guide*.

WebDAV authentication

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

You can configure Integrator 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 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 Server Guide*.

Chapter 4

Configuring SSL in the Integrator Acquisition System

This section describes how to configure the Integrator Acquisition System to use SSL.

About configuring SSL in the Integrator Acquisition System

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

To configure SSL in IAS, you need to do the following:

1. Enable SSL for the Endeca IAS Service. Optionally, enable mutual authentication and if desired disable HTTPS as part of this step.
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 system setup and then perform minimal configuration to redirect requests from the default port (HTTP) to a secure port (HTTPS). For this reason, the IAS configuration files have HTTPS redirects enabled by default.

Mutual authentication and server-only authentication

The Integrator Acquisition System supports both mutual authentication (client and server authentication) and server-only authentication. Oracle recommends configuring your environment for mutual authentication.

Mutual authentication requires a keystore and truststore for clients of the Endeca IAS Service. 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 for its communication endpoints.

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 the IAS Service. The fully qualified host name must match the 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 enecerts (included with MDEX Engine installation), followed by endeca-key-importer (included with the Platform Services installation), 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` for Windows and `usr/local/oracle/endeca/IAS/<version>/java/bin` 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.

Enabling SSL for the Endeca IAS Service

This procedure establishes a secure connection from any IAS client to the Web services running inside the IAS Service. Recall that the IAS Server, Component Instance Manager, and Record Store Web services run inside the 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 on which you are 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, Endeca 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.cas.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="8510"/></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.cas.ssl.port" default="8505"/></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.cas.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 8505.)

   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.cas.port"/></Set>
          <Set name="maxIdleTime">30000</Set>
          <Set name="Acceptors">2</Set>
          <Set name="statsOn">false</Set>
          <Set name="confidentialPort"><SystemProperty
             name="com.endeca.cas.ssl.port" default="8505"/></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 default (8505), modify `default="8505"` to the appropriate port for your environment.
11. Save and close `jetty.xml`.
12. In the same directory (`<install path>\IAS\workspace\conf` on Windows or `/ias/workspace/conf` on UNIX), 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 SSL port or if you have redirects enabled, loading it on the non-SSL port. For example, specify `https://hostname:8505/cas/?wsdl` or if you have redirects enabled specify `http://hostname:8510/cas/?wsdl`. The following WSDL displays:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions name="CasCrawlerService" targetNamespace="http://endeca.com/itl/cas
/2011-12" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://endeca.com/itl/cas
/2009-09" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <wsdl:documentation>IAS Crawler Service</wsdl:documentation>
- <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. Uncommment 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
`javax.net.ssl.trustStore`	`@TRUSTSTORE_FILE@`
`javax.net.ssl.trustStorePassword`	`@TRUSTSTORE_PASSWORD@`
`javax.net.ssl.trustStoreType`	`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.
`com.endeca.itl.cas.server.isPortSsl`	Specify how to interpret the IAS port setting. A value of `true` means that `com.endeca.eidi.ias.server.port` is an SSL port and IAS Console uses HTTPS for connections. A value of `false` means that `com.endeca.eidi.ias.server.port` is a non-SSL port and IAS Console uses HTTP for connections. The default is `false`. Specify `false` 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 the Endeca IAS Service.

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
`javax.net.ssl.trustStore`	`@TRUSTSTORE_FILE@`
`javax.net.ssl.trustStorePassword`	`@TRUSTSTORE_PASSWORD@`
`javax.net.ssl.trustStoreType`	`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
`javax.net.ssl.keyStore`	`@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>
```

IAS security support in the front-end application

The IAS Server supports the Endeca Access Control System, which you can use to make your front-end Endeca application secure. This topic explains the details involved in making your Endeca application secure based on the ACL properties generated from your CMS repository.

To make use of the ACL properties generated by the IAS Server in your Endeca front-end application, take into account the following considerations:

- The IAS Server tags each record with access control list (ACL) properties that it generates. The generated ACL properties are based on the corresponding properties for each entry in the CMS repository. In other words, the ACL properties generated by a crawl are based on ACL properties created by your CMS repository.

You can use manipulators to transform the generated ACL properties into the format for ACLs that is used by the Endeca Access Control System. You can then use the modified properties in conjunction with security login modules to limit access to records based on user or group login profiles. For details on using the Endeca Access Control System, see the *Endeca Security Guide*.

- Typically, the generated ACL properties, since they are based on the ACL information specific to a repository, can apply to either users or groups. If they apply to groups, the code for the Endeca front-end application has to map users to their corresponding groups.

Chapter 5

Security Considerations for Integrator Developers

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

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 Designer must also be configured for SSL. See *Configuring SSL in Integrator Designer on page 11* for details about configuring SSL in Integrator Designer. If SSL is not configured in Integrator Designer, graphs including components configured to support SSL will fail.

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 you 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 Designer and Integrator Server. For details, see *Configuring SSL on page 11*.

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 Designer Guide*.

SOAP authentication

The exposed SOAP service in Integrator 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