Information Builders

iWay

Omni-Gen™ Security Guide

Version 3.6

September 13, 2018

Contents

Preface

This documentation describes how to configure security for Omni-Gen. It is intended for Omni-Gen solution development teams.

How This Manual Is Organized

This manual includes the following chapters:

	Chapter/Appendix	Contents
1	Introduction and Architecture Overview	Provides an introduction to Omni-Gen web services security.
2	Enabling HTTPS, Strong Encryption Support, and Password Encryption	Describes how to enable HTTPS, strong encryption support, and password encryption.
3	Updating Security Certificates	Describes how to update security certificates.
4	Use Case Scenarios and Considerations	Describes use case scenarios and considerations.

Documentation Conventions

The following table lists and describes the documentation conventions that are used in this manual.

Convention	Description

`THIS TYPEFACE`
or
`this typeface`
	Denotes syntax that you must type exactly as shown.

this typeface	Represents a placeholder (or variable), a cross-reference, or an important term. It may also indicate a button, menu item, or dialog box option that you can click or select.
underscore	Indicates a default setting.
Key + Key	Indicates keys that you must press simultaneously.
{}	Indicates two or three choices. Type one of them, not the braces.

Convention	Description
|	Separates mutually exclusive choices in syntax. Type one of them, not the symbol.
...	Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis (...).
. . .	Indicates that there are (or could be) intervening or additional commands.

Related Publications

Visit our Technical Documentation Library at *http://documentation.informationbuilders.com*. You can also contact the Publications Order Department at (800) 969-4636.

Customer Support

Do you have questions about this product?

Join the Focal Point community. Focal Point is our online developer center and more than a message board. It is an interactive network of more than 3,000 developers from almost every profession and industry, collaborating on solutions and sharing every tips and techniques. Access Focal Point at *http://forums.informationbuilders.com/eve/forums*.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our website, *http://www.informationbuilders.com*. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of *www.informationbuilders.com* also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

Call Information Builders Customer Support Services (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 A.M. and 8:00 P.M. EST to address all your questions. Information Builders consultants can also give you general guidance regarding product capabilities. Be prepared to provide your six-digit site code (*xxxx.xx*) when you call.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

Help Us to Serve You Better

To help our consultants answer your questions effectively, be prepared to provide specifications and sample files and to answer questions about errors and problems.

The following table lists the environment information that our consultants require.

Platform	
Operating System	
OS Version	
JVM Vendor	
JVM Version	

The following table lists additional questions to help us serve you better.

Request/Question	Error/Problem Details or Information
Did the problem arise through a service or event?	
Provide usage scenarios or summarize the application that produces the problem.	
When did the problem start?	
Can you reproduce this problem consistently?	
Describe the problem.	
Describe the steps to reproduce the problem.	
Specify the error messages.	

Request/Question	Error/Problem Details or Information
Any change in the application environment: software configuration, EIS/database configuration, application, and so forth?	
Under what circumstance does the problem *not* occur?	

The following is a list of error and problem files that might be applicable.

- ❑ Input documents (XML instance, XML schema, non-XML documents)
- ❑ Transformation files
- ❑ Error screen shots
- ❑ Error output files
- ❑ Trace files
- ❑ Custom functions and agents in use
- ❑ Diagnostic Zip
- ❑ Transaction log

User Feedback

In an effort to produce effective documentation, the Technical Content Management staff welcomes your opinions regarding this document. Please use the Reader Comments form at the end of this document to communicate your feedback to us or to suggest changes that will support improvements to our documentation. You can also contact us through our website, *http://documentation.informationbuilders.com/connections.asp*.

Thank you, in advance, for your comments.

iWay Software Training and Professional Services

Interested in training? Our Education Department offers a wide variety of training courses for iWay Software and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our website, *http://education.informationbuilders.com*, or call (800) 969-INFO to speak to an Education Representative.

Interested in technical assistance for your implementation? Our Professional Services department provides expert design, systems architecture, implementation, and project management services for all your business integration projects. For information, visit our website, *http://www.informationbuilders.com/consulting*.

Chapter 1

Introduction and Architecture Overview

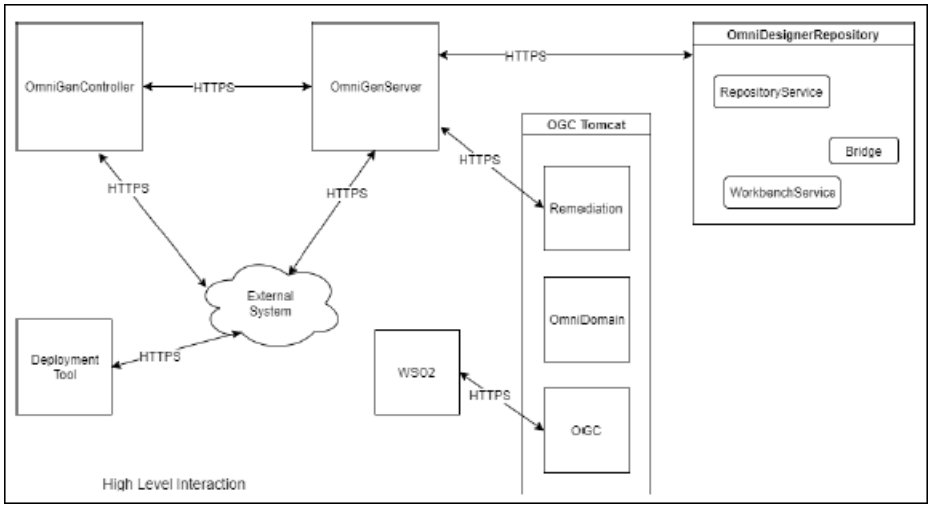
This section provides an introduction to Omni-Gen web services security.

In this chapter:

Overview

Omni-Gen™ consists of several applications that communicate with each other through web services. In addition, some of the web services are exposed to external systems. Therefore, the access to these web services needs to be secure and the data being transmitted must be encrypted. This document describes at a high level, the approach taken to secure web services using TLS or SSL, enforce strong encryption on the server side, and handle passwords. It also describes steps needed to import a CA-approved SSL certificate or create a self-signed certificate and import them to the keystores and truststores used by the application.

The following diagram illustrates the implementation.



Chapter 2

Enabling HTTPS, Strong Encryption Support, and Password Encryption

HTTPS is enabled on all the Omni-Gen applications with a web front end and/or those exposing RESTful web services.

To enable HTTPS, a signed SSL certificate must be used. Certificates are data files that digitally bind a cryptographic key to the details of an organization and to ensure that the content provided is from the correct (verified) sender. The following procedure describes the steps needed to enable HTTPS on the various Spring Boot applications. The TLS 1.2 protocol is enabled, by default, in the version of Spring Boot. Currently, 1.4.x is used.

Omni-Gen creates and uses a self-signed SSL certificate by default. The installation software captures the required parameters to create the certificate and configure the application software to use the certificate. This can be replaced with a CA-approved certificate.

In this chapter:

Understanding the Steps Required to Enable HTTPS

This section describes the steps that are required to enable HTTPS on the various Spring Boot applications that make up Omni-Gen.

1. Using a self-signed SSL certificate.

   A self-signed certificate is used, by default, and created at installation time. The parameters used depend upon the input provided during installation. The following syntax generates an omnigenstore keystore using the RSA algorithm with a key size of 2K with a new certificate. The application that needs to enable HTTPS references the keystore in its configuration.

```
keytool -genkey -alias boot -storetype PKCS12 -keyalg RSA -keysize 2048 -
keystore omnigenstore.p12 -storepass omnigen -noprompt -keypass omnigen -
validity 3650 -dname "cn=sr14386.ibi.com, ou=Omni, o=IBI, l=Rochester,
st=NY, c=US"
```

where:

`alias`

Specifies the certificate alias. By detault, this is set to *boot*.

`keystore`

Specifies the location or name of the keystore. This can be the file name with a fully qualified path.

`keypass`

Password used to protect the private key.

`dname`

Distinguished name associated with the alias and contains the server name.

`storepass`

Password used to protect the keystore.

The Omni-Gen installation will invoke this command (and commands in the following steps), with the appropriate arguments.

2. Using a CA-approved certificate.

   The CA-approved certificate can be imported into the omnigenstore keystore and the Omni-Gen applications reference the keystore. You can then import the certificate, which is described in *Importing an External Certificate* on page 16.

3. Exporting the certificate into a PEM file.

   You must create the actual certificate for the client applications using the keytool. The intermediate encoded file is created in order to create the truststore for the client applications (external or internal Omni-Gen applications). For example:

   ```
   keytool -export -alias boot -keystore omnigenstore.p12 -storepass
   omnigen -noprompt -file omnigenstore.pem
   ```

4. Enabling HTTPS in Spring Boot.

   To enable HTTPS, the Spring Boot applications need to be configured by setting the SSL parameters and pointing them to the keystore (created in step 3). The following properties need to be set:

   ```
   server.port = 9500
   server.ssl.enabled=true
   server.ssl.key-store = omnigetstore.p12
   server.ssl.key-store-password = omnigen
   server.ssl.keyStoreType = JKS
   server.ssl.keyAlias = boot
   ```

   Note: The Spring Boot application understands these properties, which are exposed through the installation software and its associated configuration file differently.

5. Redirecting HTTP to HTTPS.

   This is done by adding another Tomcat connector programmatically. It is configured as an HTTP connector that redirects all the traffic to the earlier configured HTTPS connector and entails adding a TomcatEmbeddedContainerFactory bean to one of the @Configuration classes. This allows supporting both HTTP and HTTPS or enabling the redirect.

   These steps ensure the web services exposed by the application can be accessed over HTTPS.

Consuming HTTPS

The applications must be able to consume the web services over HTTPS. When acting as a client, the certificate created or used earlier must be added to the Java truststore. This requires importing the certificate into the truststore using the keytool, as shown below:

```
keytool -import -alias boot -keystore ibi-cacerts -storepass boot -noprompt
-file omnigenstore.pem
```

The application is made aware of the certificate by setting the javax.net.ssl.trustStore property. This is added as a Java argument when invoked. For the applications running on Apache Tomcat (OGC, OmniDesignerRepository), this is added to CATALINA_OPTS.

UI/Configuration

The Common Name (CN) for the self-signed certificate is the fully qualified host name. The Omni-Gen installer UI captures the host name and domain, along with all the elements of the distinguished name, the keystore, and truststore locations, as part of the configuration. This is then used to build the self-signed certificate.

Importing an External Certificate

Scripts for Linux and Windows are included (in the scripts folder) to import a CA-approved certificate into the omnigenstore keystore. The following syntax shows the format.

```
importCert <certificate> <password> <key_alias>
```

Supporting Strong Encryptions

In addition to the basic privacy, integrity, and protection for the data that is transmitted between the client and the server, strong encryptions refer to a TLS implementation which provides all of the following:

- Perfect Forward Secrecy, which ensures that a compromise to the private key of a server in the present does not compromise the confidentiality of past TLS communications.
- Protection from known attacks on older SSL and TLS implementations, such as POODLE and BEAST.
- Support for the strongest ciphers available to modern web browsers and other HTTP clients.
- Rejection of clients that cannot meet these requirements.

The following configuration is in place to support this.

```
SSLProtocol=TLSv1.2
# Supported Ciphers
SSLCipherSuite=ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-
SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-
ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-
SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-
SHA256
SSLHonorCipherOrder=on
SSLCompression=off
SSLSessionTickets=off
```

Password Encryption

Passwords used by Omni-Gen are stored in property and configuration files. The files are in plain text, but the password fields are encrypted using 128-bit AES encryption.

The following syntax shows a sample password field entry.

```
server.runtime.ssl.keystore-file = ${server.runtime.dataDirectory}/
omnigenstore.p12
server.runtime.ssl.keystore-password =
yc4QxoL5oCAqqEHn11e91Q==:NqG0dkZKuVW2RSgxqsi/eQ==
```

The application is responsible for encrypting and decrypting the password fields, prior to use.

Chapter 3

Updating Security Certificates

This section describes how to update security certificates.

In this chapter:

Overview

If you need to update the default security certificate with a different certificate (for example, a certificate approved by a Certificate Authority), then you must import the certificate along with the private key into the keystore. Sample scripts for Windows and Linux are available below for reference.

If you are copying the script directly from this document, consider the fact that whitespace characters might be distorted, requiring you to reformat the script. This will be streamlined in future releases.

To update the security certificates:

1. Copy the security certificate file and paste it in the \OmniGenData folder.

   This file must be in PKCS#12 (or PFX) format. If it is in PEM format, then it must be converted.

2. Create the script and copy it to the \OmniGenData folder.

   The exact location of the script will change in future releases.

3. Run the script, which takes the following three arguments:

   - Source keystore (certificate)
   - Keystore password
   - Source alias

Sample Script for Windows

For your convenience, the sample script for Windows is also attached to this PDF.

```
@set KT="%JAVA_HOME%\bin\keytool"
@set OMNIGENDATA=..\OmniGenData

@if "%2" == "" goto args_count_wrong
@if "%3" == "" goto args_count_wrong
@if "%4" == "" goto args_count_ok

:args_count_wrong
@echo Invalid parameters. Usage: import.cmd srckeystore srcstorepass
srcalias
@exit /b 1

:args_count_ok

cd %OMNIGENDATA%
@del /Q omnigenstore.p* ibi-certs

%KT% -importkeystore ^
-srckeystore %1 -destkeystore omnigenstore.p12 ^
-srcstorepass %2 -deststorepass omnigen ^
-srcalias %3 -destalias boot ^
-srcstoretype pkcs12 -deststoretype JKS ^
-destkeypass omnigen ^
-noprompt

%KT% -exportcert -alias boot -keystore omnigenstore.p12 -storepass omnigen -
keypass omnigen -noprompt -rfc -file omnigenstore.pem
%KT% -importcert -alias boot -keystore ibi-certs -storepass changeit -
noprompt -file omnigenstore.pem

%KT% -delete -alias boot -keystore OmniGovConsole\data\security\client-
truststore.jks -storepass wso2carbon -noprompt
%KT% -importcert -alias boot -keystore OmniGovConsole\data\security\client-
truststore.jks -storepass wso2carbon -noprompt -file omnigenstore.pem

cd ..\scripts
```

Sample Script for Linux

For your convenience, the sample script for Linux is also attached to this PDF.

```
#!/bin/sh

KT=$JAVA_HOME/bin/keytool
OMNIGENDATA=../OmniGenData

EXPECTED_ARGS=3
E_BADARGS=65

if [ $# -ne $EXPECTED_ARGS ]
then
  echo "Invalid parameters. Usage: `basename $0` srckeystore srcstorepass
srcalias"
  exit $E_BADARGS
fi

cd $OMNIGENDATA
rm -rf omnigenstore.p* ibi-certs

$KT -importkeystore \
-srckeystore $1 -destkeystore omnigenstore.p12 \
-srcstorepass $2 -deststorepass omnigen \
-srcalias $3 -destalias boot \
-srcstoretype pkcs12 -deststoretype JKS \
-destkeypass omnigen \
-noprompt

$KT -exportcert -alias boot -keystore omnigenstore.p12 -storepass omnigen -
keypass omnigen -noprompt -rfc -file omnigenstore.pem
$KT -importcert -alias boot -keystore ibi-certs -storepass changeit -
noprompt -file omnigenstore.pem

$KT -delete -alias boot -keystore ./OmniGovConsole/data/security/client-
truststore.jks -storepass wso2carbon -noprompt
$KT -import -alias boot -keystore ./OmniGovConsole/data/security/client-
truststore.jks -storepass wso2carbon -noprompt -file omnigenstore.pem

cd ../scripts
```

Chapter 4

Use Case Scenarios and Considerations

HTTPS requires the creation and installation of signed certificates. For Omni-Gen applications, self-signed certificates are used. The steps are the same using a signed certificate from a Certificate Authority (CA). Depending on whether the individual applications run locally or on different machines, the certificate may need to be installed on one or more machines.

In this chapter:

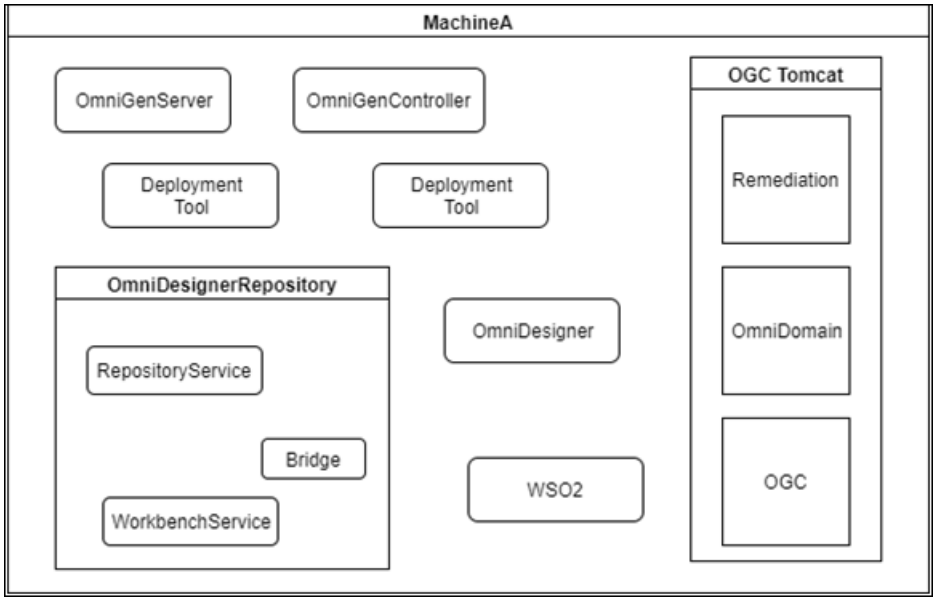
Basic Requirements

The following list describes the basic requirements for the use case scenarios and considerations.

- All Omni-Gen applications use the omnigenstore keystore, which is created by the installation software.
- Omni-Gen applications that need to communicate with HTTPS-enabled applications use the ibi-certs truststore.
- The keystore, pem, and truststore files are in the OmniGenData directory.

Installing the Application on a Single Host

All applications are running on a single host. The following image shows the workflow example behind the application running on Machine A.

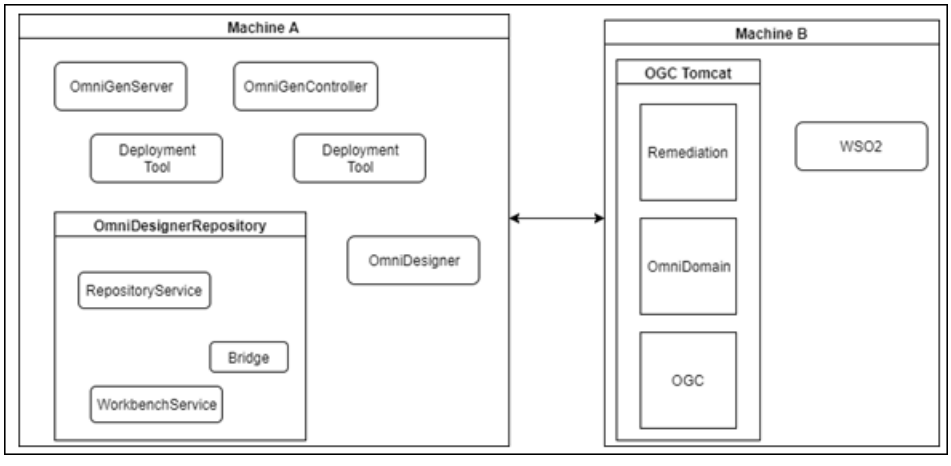


The following list describes how the application is installed on a single host.

- The installation process captures the installation type (single host and multiple hosts), along with the fully qualified host names.
- It collects the locations of the keystore and truststore.
- By default, the cert directory is the OmniGenData directory.
- Installation software creates the omnigenstore keystore, omnigenstore cert file, and the ibi-certs truststore in the cert directory.
- If multiple certificates need to be included (for example, in OGC), separate truststores are used. This is handled by the installation software.
- The fully qualified host name is used when accessing each web service.

Deploying on Multiple Hosts

Applications must be run on different hosts. In the following example, OGC Tomcat and WSO2 are running on Machine B while the Omni-Gen Server and the other applications are running on Machine A.



The following list describes how the application is deployed on multiple hosts.

- The installation process captures the installation type (single host and multiple hosts), along with the fully qualified host names.
- It collects the locations of the keystore and truststore.
- By default, the cert directory is the OmniGenData directory.
- Installation software creates the omnigenstore keystore, omnigenstore cert file, and the ibi-cert truststore in the cert directory on Machine A.
- Installation software remote copies the certificate files to the cert directory on Machine B and creates the required truststore locally (on Machine B).
- The invoker software that starts up the application references the local truststore.
- The fully qualified host name is used when accessing web services.

The same model is followed for any other variations.

Feedback

Customer success is our top priority. Connect with us today!

Information Builders Technical Content Management team is comprised of many talented individuals who work together to design and deliver quality technical documentation products. Your feedback supports our ongoing efforts!

You can also preview new innovations to get an early look at new content products and services. Your participation helps us create great experiences for every customer.

To send us feedback or make a connection, contact Sarah Buccellato, Technical Editor, Technical Content Management at *Sarah_Buccellato@ibi.com.*

To request permission to repurpose copyrighted material, please contact Frances Gambino, Vice President, Technical Content Management at *Frances_Gambino@ibi.com.*

Information Builders

iWay

Omni-Gen™ Security Guide

Version 3.6

DN3502331.0818

Information Builders, Inc.
Two Penn Plaza
New York, NY 10121-2898