



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Configuring TLS for iWD

intelligent Workload Distribution 9.0.0

12/29/2021

Table of Contents

Configuring TLS for iWD	3
List of connections and known limitations	4
Configuration summary	8
Configuring database connections	9
GAX Plugin	11
Connection to Web service from browser	12
Configuring Windows certificates	13
Configuration options	14
Specific connection configuration details	16
History Node connections	20
Data Mart connections	22
GAX plug-in connections	23
Stat Server Java Extensions connections	24
iWD Manager connections	25
iWD Web connections	26

Configuring TLS for iWD

This document provides iWD-specific information with respect to configuring TLS.

Genesys reference documents

Before attempting any detailed or iWD-specific configuration for supporting TLS, Genesys recommends that you completely familiarize yourself with the TLS-related content of the following Genesys documents:

- [Secure Connections \(TLS\)](#) in the *Genesys Security Deployment Guide*
- [Detailed TLS information](#) in the *Platform SDK Developer's Guide*

List of connections and known limitations

The table below lists all iWD component connections and their types.

Important

iWD supports TLS 1.2.

Please refer to the [eServices Integrated Capture Points Guide](#) for information about configuring secure Capture Point connections. See the following topics:

- [JMS Capture Point](#)
- [Web Services Capture Point](#)

Configuring secure connections between Interaction Server and the JMS Event Logger is done in the same way as described here:

- [Using the JMS logger with Apache Active MQ](#)

For TLS support of other Genesys components that iWD depends on, see the following topic:

- [TLS Protocol Support.](#)

Connections

iWD Component	Connection Type	Role	Connections	TLS Mode	Comments
iWD Manager	PSDK	Client	Configuration Server	mutual	The Configuration Server auto-upgrade port should be used for TLS.
	PSDK	Client	Interaction Server	mutual	
	PSDK	Client	UCS	mutual	
	PSDK	Client	Message Server	mutual	
	REST	Client	History Node	mutual	
	REST	Server	Web browser or custom desktops	mutual	

iWD Component	Connection Type	Role	Connections	TLS Mode	Comments
iWD Data Mart	PSDK	Client	Configuration Server	mutual	The Configuration Server auto-upgrade port should be used for TLS.
	JDBC	Client	iWD Data Mart database	tls	Configured via URL or JVM options or combination depending on database JDBC driver.
	JDBC	Client	ConfigServer database	tls	Configured via URL or JVM options or combination depending on database JDBC driver.
	REST	Client	iWD History Node	mutual	
	REST	Server	iWD Plug-in for GAX	mutual	
			LCA	no	LCA and product should be located on the same host, so TLS is not required.
	PSDK	Server	Message Server	mutual	Introduced in 9.0.005.
iWD History Node	PSDK	Client	Configuration Server	mutual	The Configuration Server auto-upgrade port should be used for TLS.
	JMS	Client	Interaction Server Event Log	mutual	
	Kafka	Client	Interaction Server Event Log	mutual	
	JDBC	Client	History Node database	tls	Configured via URL or JVM options or combination depending on database JDBC

iWD Component	Connection Type	Role	Connections	TLS Mode	Comments
					driver.
	REST	Server	iWD Data Mart and iWD Manager	mutual	
PSDK	Server	Message Server	mutual	Introduced in 9.0.005.	
Stat Server Extensions	JDBC	Client	iWD Data Mart database	tls	Configured via URL or JVM options or combination depending on database JDBC driver.
iWD GAX Plugin	JDBC	Client	Interaction Server DB	tls	Configured via URL or JVM options or combination depending on database JDBC driver.
	REST	Client	iWD Data Mart	mutual	
iWD Web	REST	Server	Web browser	mutual	
	PSDK	Client	Configuration Server	mutual	The Configuration Server auto-upgrade port should be used for TLS.
	PSDK	Client	Interaction Server	mutual	
	PSDK	Client	Message Server	mutual	
	REST	Client	WSCP	mutual	

Limitations

PEM and Windows (MSCAPI) certificates

iWD Manager, iWD Web, iWD Data Mart and iWD History Node REST APIs do not support PEM and Windows (MSCAPI) certificates. Data Mart and History Node are based on Dropwizard, which is Jetty-based. Dropwizard documentation refers to Jetty documentation which you can find at <http://www.eclipse.org/jetty/documentation/current/configuring-ssl.html>.

Jetty does not support PEM files directly, so when you get PEM certificates, you need to pack them into a keystore/truststore. There's more information at <http://www.eclipse.org/jetty/documentation/>

[current/configuring-ssl.html#loading-keys-and-certificates](#)

The iWD Manager and iWD Web REST server is based on Tomcat, which does not support PEM directly. There's more information at <https://tomcat.apache.org/tomcat-8.5-doc/ssl-howto.html>

Tomcat currently operates only on JKS, PKCS11 or PKCS12 format keystores.

iWD Stat Extensions shares database settings with Data Mart

iWD Stat Extensions has a limitation regarding TLS settings for JDBC connection. iWD Stat Extensions shares database settings with Data Mart. The Data Mart Stat Adapter job copies the JDBC URL from the Data Mart DAP to Stat Server options. So Stat Server must be configured in the same way as Data Mart.

- If Data Mart is set to use a TLS connection to the database via JVM arguments (the recommended way), then Stat Server must be provided with the corresponding JVM options and certificates.
- If Data Mart is set to use a TLS connection to the database via a JDBC URL which contains certificates and/or passwords, then Stat Server should be installed to the same host as Data Mart or use the same certificate paths and passwords.

iWD Manager and iWD Web client applications cannot be configured on HOST level

Client applications do not have a linked host value, so iWD cannot read host parameters while configuring such application connections. There are two client applications in iWD—iWD Manager and iWD Web— with connections to Configuration Server. These connections through the auto-upgrade port can be configured ONLY on the connection or the application level.

Mutual TLS for databases

Mutual TLS for databases is not supported.

Configuration summary

Levels

Connections between Genesys components that are defined using objects in Genesys Configuration Server can be configured at three different levels:

- Port (Connection)
- Application
- Host level

Levels are read by the application in the priority shown above, so port level has the highest priority. Genesys recommends setting all required options at the same level.

Recommendations

iWD follows the standard Genesys approach. Genesys recommends the following:

- **For PSDK connections, REST clients, JMS client**—pem on UNIX and Windows certificates on Windows (see [Limitations](#))
- **For JDBC clients, REST servers**—Java KeyStore (JKS)

Configuring database connections

SSL certificates and storage passwords can be provided in two ways: via a URL in DAP options, or through the JVM system properties. To pass options to JVM, use the relevant `.sh` file on Linux and `JavaServerStarter.ini` on Windows.

Sample connection URLs

MS SQL 2016

```
"jdbc:sqlserver://MSSQL2016TLS:1433;databaseName=iwd_dm;encrypt=true;trustServerCertificate=false;trustStore=/genesys/MSSQL2016.ts;trustStorePassword=storePassword;"
```

JVM properties example

```
-Djavax.net.ssl.trustStore=/genesys/MSSQL2016.ts  
-Djavax.net.ssl.trustStoreType=JKS  
-Djavax.net.ssl.trustStorePassword=storePassword
```

For more information about configuring your database, refer to the vendor's documentation for the database that you are using.

Important

When storage passwords are provided through the URLs, they will be stored in Configuration Server as plain strings without encryption and can be seen in GAX. To hide passwords, you must use JVM properties.

GAX Plugin

If multiple Data Marts are configured and TLS is used for the GAX plug-in client to Data Mart, all Data Mart's REST servers must have the same TLS configurations.

To enable TLS, the connection to Data Mart is created in the GAX applicaton itself. If you configure TLS on the Connection level, TLS parameters will be read from the first Data Mart connection.

Connection to Web service from browser

The server application should be configured as a Genesys server application through the corresponding configuration object

To configure the browser, do the following:

- For simple TLS—import into the browser a CA certificate to be used for the signing server certificate.
- For mutual TLS—along with a CA certificate for the server, import a client certificate into the browser as well.

Configuring Windows certificates

When the iWD application has Transport Layer Security (TLS) configured for any connection which supports Windows certificates, follow one of the two procedures below to enable it as a Windows Service:

Either: Import the certificate to the Local System Account using one of the two following commands:

- `psexec.exe -i -s mmc.exe` and then import the certificate for the user who is the local system account.
- `psexec.exe -i -s certutil -f -user -p [password] -importpfx [path to the certificate]`

Or:

1. Import the certificate for a local host user.
2. Select the Windows service related to the application.
3. Select the **Log On** tab. The default setting is Log on as local system account.
4. Select Log on as this account and provide the login/password of a local host user.

Important

The `psexec.exe` command with flag `-s` executes the specified program under the system account. `psexec` is part of the PS Tools which can be downloaded from <http://technet.microsoft.com/en-US/sysinternals>.

Configuration options

Standard Genesys TLS configuration options

Use the configuration options described in the [TLS List of Parameters](#) in the *Platform SDK Developer's Guide*.

Certificate password configuration (iWD specific)

There are two ways to set certificate passwords.

Application and Host level options (strongly recommended)

Section	Name	Client/Server Side	Default value	Description
tls-keystore	password	<ul style="list-style-type: none">Simple TLS: SERVERMutual TLS: BOTH	N/A	Keystore password
tls-keystore-entry	password	<ul style="list-style-type: none">Simple TLS: SERVERMutual TLS: BOTH	N/A	Keystore entry password
tls-truststore	password	<ul style="list-style-type: none">Simple TLS: CLIENTMutual TLS: BOTH	N/A	Trusted certificates storage password

In this case passwords are automatically encrypted in the configuration database and masked in the GAX UI.

Important

Passwords set at the Application and Host levels using the parameters above are

applied to all secured connections of these Applications and Hosts respectively. So every connection MUST use certificates with the same passwords.

Connection/port level options

Important

Passwords set using the parameters below are NOT encrypted in the configuration database and can be viewed in GAX UI. Genesys strongly recommends not using these parameters. The options are to be used only if connection level configuration cannot be avoided.

If it is necessary to define passwords at the connection/port level, use the following parameters.

Name	Client/Server Side	Default value	Description
keystore-password	Simple TLS—Server	N/A	Keystore password
	Mutual TLS—Both.		
keystore-entry-password	Simple TLS—Server	N/A	Keystore entry password
	Mutual TLS—Both.		
tls-truststore	Simple TLS—Client	N/A	Trusted certificates storage password
	Mutual TLS—Both.		

Specific connection configuration details

iWD follows the common Genesys approach and recommends using Host-level configuration with PEM-files on Linux and Windows certificates on Windows for all PSDK connections, REST clients and JMS client. For REST servers, Connection-level configuration with JKS-files is recommended. For JDBC clients the recommended way is to use JKS files and configure connection via JVM options.

Assumptions/Prerequisites

- You have generated certificates with associated private and public keys.
- You have Java keystore and truststore files with the same keys.

Please find more information about certificates [here](#).

ActiveMQ

To configure SSL in ActiveMQ, please refer to the vendor's ActiveMQ documentation.

Sample ActiveMQ SSL configuration

The following is an example of how you might configure SSL for ActiveMQ.

1. Copy **keystore.jks** and **truststore.jks** into the **<Apache ActiveMQ installation directory>/conf** folder.
2. Open the **<Apache ActiveMQ installation directory>/conf/activemq.xml** file and add the following lines:

```
<transportConnectors>
  ...
  <transportConnector name="ssl"
uri="ssl://0.0.0.0:61617?trace=true&edClientAuth=true"/>
  ...
</transportConnectors>
<sslContext>
  <sslContext keyStore="file:${activemq.base}/conf/keystore.jks"
keyStorePassword="YourKeyStorePassword"
trustStore="file:${activemq.base}/conf/truststore.jks"
trustStorePassword="YourTrustStorePassword" />
</sslContext>
```

3. Change keystore and truststore passwords accordingly.
4. Restart ActiveMQ.

JDBC configuration

Please read the [Configuring database connections](#) topic.

Windows certificates import

For the Windows certificates installation procedure please see the [Windows certificates](#) topic.

Framework configuration

Configuration Server

Configure the auto-upgrade port as described in [Securing core framework connections](#) topic.

Interaction Server, UCS, Message Server

Configure secured ports as described [here](#).

Host configuration

1. Open **GAX -> Hosts -> <your_host> ->** and fill the certificate fields with your PEM certificate files.

Home > Hosts > Hosts > Properties

General

Name *: [redacted].int.genesyslab.com

IP Address: [empty]

OS Type *: RedHat Enterprise Linux AS/Intel

Version: 1

LCA Port *: 4999

Solution Control Server: scs

Certificate: /genesys/TLS/02_[redacted].cert.pem

Certificate Description: [empty]

Certificate Key: /genesys/TLS/02_[redacted].priv_key.pem

Trusted CA: /genesys/TLS/ca_cert.pem

Tenant: Environment

☒ **State Enabled**

2. Open the **Options** tab and add the following options:

- **[security]/tls** = true
- **[security]/mutual-tls** = true

Home > Hosts > Hosts > Properties

Options

Name	Section	Key	Value
conf			
security			
security \ certificate	security	certificate	/genesys/TLS/02_[redacted].cert.pem
security \ certificate-key	security	certificate-key	/genesys/TLS/02_[redacted].priv_key.pem
security \ trusted-ca	security	trusted-ca	/genesys/TLS/ca_cert.pem
security \ tls	security	tls	true
security \ mutual-tls	security	mutual-tls	true

Troubleshooting

If you start receiving ESP Server is not connected interaction errors and tasks, do the following:

1. Go to the ErrorHeld queue, add the `tls=1` transport parameter to the connection between Interaction Server and GRE or the GRE cluster application and GRE nodes.
2. Add the `tls=1` transport parameter to the connection between Interaction Server and Universal Contact Server.

History Node connections

REST server

1. Open **GAX -> Applications -> <iWD History Node app> -> ports -> admin** .
 1. Change it to Secured listening mode.
 2. Fill Certificate and Trusted CA fields with JKS keystore and truststore files.
 3. Add Transport parameters **tls-mutual=1, keystore-password, truststore-password, provider=JKS**. The final string should look like this:
`tls=1;certificate=/genesys/TLS/keystore.jks;trusted-ca=/genesys/TLS/truststore.jks;keystore-password=KSPASSWD;truststore-password=TSPASSWD;tls-mutual=1;provider=JKS`
2. Open **GAX -> Applications -> <iWD History Node app> -> ports -> default** .
 1. Change it to Secured listening mode.
 2. Fill Certificate and Trusted CA fields with JKS keystore and truststore files.
 3. Add Transport parameters **tls-mutual=1, keystore-password, truststore-password, provider=JKS**. The final string should look like this:
`tls=1;certificate=/genesys/TLS/keystore.jks;trusted-ca=/genesys/TLS/truststore.jks;keystore-password=KSPASSWD;truststore-password=TSPASSWD;tls-mutual=1;provider=JKS`

JMS client

1. Open **GAX -> Applications -> <iWD History Node app> -> connections**.
2. Find a DAP which History Node uses to connect to ActiveMQ.
3. Open the DAPs Application Options and set the following:
 - **logger-settings / jms-initial-context-factory** =
`org.apache.activemq.jndi.ActiveMQSslInitialContextFactory`
 - **logger-settings / jms-provider-url** = `ssl://<activemq hostname>:61617`

Kafka consumer

1. Open **GAX -> Applications -> <iWD History Node app> -> connections**.
2. Find a DAP which History Node uses to connect to Kafka Event Logger.
3. You can choose one of two configuration methods:

-
1. Via standard Kafka options.
 1. Open the DAP's Application Options and modify the following sections:
 1. **[consumer-options]** section
Add in this section the Kafka consumer TLS configuration options in accordance with the official [Apache Kafka documentation](#).
 2. **[kafka-settings]** section
Check that the **servers** property has a TLS protected port and the hostname is fully qualified.
 2. Via iWD History Node Connection parameters:
 1. Make sure that connection to Kafka Event Logger DAP uses a secured port.
 2. Add Transport parameters **tls-mutual=1**, **keystore-password**, **truststore-password**, **provider=JKS**. The final string should look like this:
`tls=1;certificate=/genesys/TLS/keystore.jks;trusted-ca=/genesys/TLS/truststore.jks;keystore-password=KSPASSWD;truststore-password=TSPASSWD;tls-mutual=1;provider=JKS`

Data Mart connections

REST server

1. Open **GAX -> Applications -> <iWD Datamart app> -> ports -> webservice.**
2. Change it to Secured listening mode.
3. Fill Certificate and Trusted CA fields with JKS keystore and truststore files.
4. Add Transport parameters **tls-mutual=1, keystore-password, truststore-password, provider=JKS**. The final string should look like this:
 - `tls=1;certificate=/genesys/TLS/keystore.jks;trusted-ca=/genesys/TLS/truststore.jks;keystore-password=KSPASSWD;truststore-password=TSPASSWD;tls-mutual=1;provider=JKS`

GAX plug-in connections

REST client to Datamart

1. Open **GAX -> Applications -> <GAX app> -> connections**.
2. Create a connection to the iWD Data Mart application using the webservice port.

Stat Server Java Extensions connections

JDBC client

Regarding database settings, please read the documentation [DB connections](#) xx

Add the following Java parameters to the Stat Server applications **[jvm-options]** section:

- -Djavax.net.ssl.trustStore = /path/to/truststore.jks
- -Djavax.net.ssl.trustStoreType = JKS
- -Djavax.net.ssl.trustStorePassword = PASSWORD

Please refer to the [List of connections and known limitations](#) topic.

iWD Manager connections

PSDK client to Message Server, UCS, Interaction Server

1. Open **GAX -> Applications -> <iWD Manager Server app> -> connections**.
2. Make sure it has a secured connection to the Message Server application.
3. Change the connection to UCS to a Secured port.
4. Change the connection to Interaction Server to a Secured port.

REST server

1. Open **GAX -> Applications -> <iWD Manager Server app> -> ports -> default**.
2. Change it to Secured listening mode.
3. Fill Certificate and Trusted CA fields with JKS keystore and truststore files.
4. Add Transport parameters **tls-mutual=1, keystore-password, truststore-password, provider=JKS**. The final string should look like this:
 - `tls=1;certificate=/genesys/TLS/keystore.jks;trusted-ca=/genesys/TLS/truststore.jks;keystore-password=KSPASSWD;truststore-password=TSPASSWD;tls-mutual=1;provider=JKS`

PSDK client to Configuration Server

1. Open **GAX -> Applications -> <iWD Manager Client app>**
2. Make sure it has a connection to the Configuration Server application.
3. Open the connection to Configuration Server and add the following transport parameters:
 - For Linux:
 - `certificate=<path to PEM certificate file>;certificate-key=<path to PEM certificate key file>;trusted-ca=<path to PEM certificate truststore file>;tls-mutual=1`
 - For Windows:
 - `certificate=<certificate_thumbprint>;trusted-ca=<truststore_thumbprint>;tls-mutual=1`

iWD Web connections

REST server

1. Open **GAX -> Applications -> <iWD Web Server app> -> ports -> default**
2. Fill Certificate and Trusted CA fields with JKS keystore and truststore files.
3. Add Transport parameters **tls-mutual=1, keystore-password, truststore-password, provider=JKS**. The final string should look like this:
 - `tls=1;certificate=/genesys/TLS/keystore.jks;trusted-ca=/genesys/TLS/truststore.jks;keystore-password=KSPASSWD;truststore-password=TSPASSWD;tls-mutual=1;provider=JKS`

PSDK client to Interaction Server, Message Server

1. Open **GAX -> Applications -> <iWD Web Server app> -> connections .**
2. Change the connection to the Interaction Server application to a Secured port.
3. Make sure it has a secured connection to the Message Server application.

PSDK client to Configuration Server

1. Open **GAX -> Applications -> <iWD Web app> -> connections .**
2. Open the connection to Configuration Server and add the following transport parameters:
 - For Linux:
 - `certificate=<path to PEM certificate file>;certificate-key=<path to PEM certificate key file>;trusted-ca=<path to PEM certificate truststore file>;tls-mutual=1`
 - For Windows:
 - `certificate=<certificate_thumbprint>;trusted-ca=<truststore_thumbprint>;tls-mutual=1`