

# **GENESYS**

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.

# Deployment Guide

Transport Layer Security (TLS)

# Transport Layer Security (TLS)

## Contents

- 1 Transport Layer Security (TLS)
  - 1.1 Configuring TLS for Genesys Servers
  - 1.2 Configuring TLS for Web Engagement Servers

Genesys Web Engagement supports the Transport Layer Security (TLS) protocol to secure data exchanged with other Genesys components. For details about TLS, see the Genesys 8.1 Security Deployment Guide. You can configure TLS for Web Engagement by completing the procedures on this page.

## Configuring TLS for Genesys Servers

To configure the TLS parameters for Genesys servers like Configuration Server, Message Server, or Chat Server, see Configuring TLS Parameters in Configuration Manager.

### Configuring TLS for Web Engagement Servers

To enable TLS support for the Genesys Web Engagement Backend Server, you must do the following:

- 1. Have properly installed trusted certificates for the Genesys servers.
- 2. Configure TLS options for the Web Engagement Backend Server application.
- 3. Configure the appropriate connections between the Web Engagement Backend server application and the necessary Genesys servers through secure ports.

#### Configuring TLS Options

The Genesys Web Engagement Backend Server includes the following TLS-related configuration options in its [security] section.

Option	Default Value	Mandatory	Changes Take Effect	Description
trusted-ca-type	none	no	after restart	Type of trusted storage Valid values: MSCAPI, PEM or JKS. If empty, TLS support is disabled.
trusted-ca	none	no	after restart	Specifies the name of the trusted store file which holds the public certificate to verify the server. Applicable for PEM and JKS trusted storage types only. Valid values: valid file name (including path)
trusted-ca-pwd	none	no	after restart	Password for the JKS trusted

Option	Default Value	Mandatory	Changes Take Effect	Description
				storage.
				Valid values: any string

See Configuring Trusted Stores below for details about configuration for a specific type of store (PEM, JKS, MSCAPI).

#### Configuring Trusted Stores

#### PEM Trusted Store

PEM stands for "Privacy Enhanced Mail", a 1993 IETF proposal for securing e-mail using public-key cryptography. That proposal defined the PEM file format for certificates as one containing a Base64-encoded X.509 certificate in specific binary representation with additional metadata headers.

PEM certificate trusted store works with CA certificate from an X.509 PEM file. It is a recommended trusted store to work on Linux systems.

Complete the steps below to work with the PEM certificate trusted store:

#### Start

- 1. Configure TLS for Genesys servers to use certificates signed by CA certificate **certificateCA.crt**.
- 2. Place the trusted CA certificate in PEM format on the Genesys Web Engagement Backend Server application host. To convert a certificate of another format to .pem format you can use the OpenSSL tool. For example:
  - Convert a DER file (.crt .cer .der) to PEM: openssl x509 -inform der -in certificateCA.crt -out certificateCA.pem
  - Convert a PKCS#12 file (.pfx .p12) containing a private key and certificates to PEM: openssl pkcs12 -in certificateCA.pfx -out certificateCA.pem -nodes

You can add **-nocerts** to only output the private key or add **-nokeys** to only output the certificates.

- 3. In Genesys Administrator, navigate to **Provisioning > Environment > Applications** and open your Backend Server application.
- 4. Click the **Options** tab and navigate to the [security] section.
- 5. Set the **trusted-ca-type** option to PEM.
- 6. Set the **trusted-ca** option to the path and file name for your trusted CA in PEM format on the Genesys Web Engagement Backend Server application host.
- 7. Click Save & Close.

End

#### JKS Trusted Store

A Java KeyStore (JKS) is a repository of security certificates used, for instance, in SSL/TLS encryption. The Java Development Kit provides a tool named keytool to manipulate the keystore.

Complete the steps below to work with the JKS certificate trusted store:

#### Start

- 1. Configure TLS for Genesys servers to use certificates signed by CA certificate certificateCA.crt.
- 2. Import the CA certificate to an existing Java keystore using keytool:
  - Run the keytool command with option -alias set to root: keytool -import -trustcacerts -alias root -file certificateCa.crt -keystore /path/to/keysore/keystore.jks
  - Enter the keystore password in command line prompt for example: Enter keystore password: somepassword
- 3. In Genesys Administrator, navigate to **Provisioning > Environment > Applications** and open your Backend Server application.
- 4. Click the **Options** tab and navigate to the [security] section.
- 5. Set the **trusted-ca-type** option to JKS.
- 6. Set the **trusted-ca** option to the path and file name for your JKS trusted storage type on the Genesys Web Engagement Backend Server application host.
- 7. Set the **trusted-pwd** option to the password defined for your keystore in Step 2.
- 8. Click Save & Close.

#### End

**MSCAPI** Trusted Store

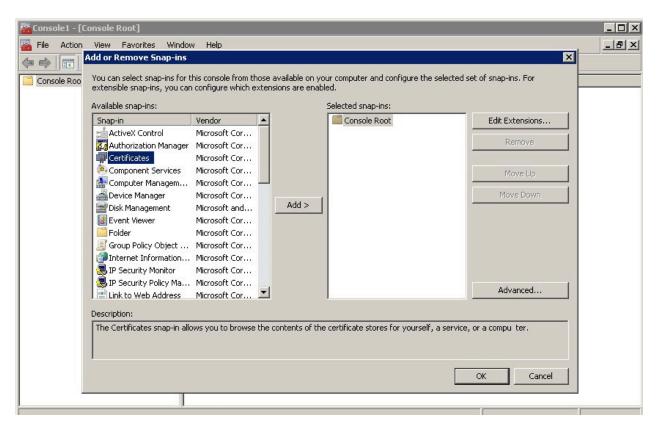
Complete the steps below to work with the MSCAPI certificate trusted store:

#### Start

- 1. Configure and tune TLS for Genesys servers to use certificates signed by the same CA.
- 2. If the Web Engagement Backend Server is running on a different host, copy the trusted CA certificate to this host.
- 3. Import the CA certificate to WCS via Certificates Snap-in on the Web Engagement Backend Server host by launching the MMC console. Enter mmc at the command line.
- 4. Select **File > Add/Remove Snap-in...** from the main menu.

🚟 Console1 - [Console Root]			
🚡 File Action View Favorites	Window Help		_8×
Vew Open	Ctrl+N Ctrl+O	1	
Save Save	Ctrl+5	I There are no items to show in this view.	
Add/Remove Snap-in Options	Ctrl+M		
1 C:\Windows\\services.msc 2 ServerManager.msc 3 C:\Windows\\compmgmt.ms 4 C:\Windows\system32\secpol.			
Exit			
Enables you to add snap-ins to or remov	re them from the snap-in console.		

5. Select **Certificates** from the list of available snap-ins and click **Add**.



6. Select the account to manage certificates for and click **Finish**. It is important to place certificates under the correct Windows account. Some applications are run as services under the Local Service or System account, while others are run under user accounts. The account chosen in MMC must be the same as the account used by the application which certificates are configured for, otherwise the application will not be able to access this WCS storage.

#### Transport Layer Security (TLS)

🔚 Console1 - [Co	onsole Root]	_ 🗆 🗵
	View Favorites Window Help	X
	Add or Remove Snap-ins	X
Console Roo		of snap-ins. For
	This snap-in will always manage certificates for:	
	My user account	Edit Extensions
	C Service account	
	C Computer account	Remove
		Move Up
		Move Down
		Advanced
	< Back Finish Cancel	r a compu ter.
		OK Cancel
-		
	,	

- 7. Click **OK**.
- 8. Import a certificate. Right-click the "Trusted Root Certification Authorities/Certificates" folder and choose All Tasks > Import... from the context menu. Follow the steps presented by the Certificate Import Wizard. Oonce finished the imported certificate appears in the certificates list.

<ul> <li>Console Root</li> <li>Certificates - Current User</li> <li>Personal</li> <li>Certificates</li> <li>Trusted Root Certification Authon</li> <li>Certificates</li> <li>Trusted Root Certification Authon</li> <li>Certificates</li> <li>Intern</li> <li>Active</li> <li>Active</li> <li>Intern</li> <li>View</li> <li>Active</li> <li>Truste</li> <li>New Window from Here</li> <li>Untru:</li> <li>New Taskpad View</li> <li>Certificates</li> <li>Other</li> <li>Certification Root Cartification</li> <li>Smart care mascer roots</li> <li>Microsoft Code Signing PCA</li> <li>Microsoft Corporation</li> <li>Microsoft Corporation</li> <li>Microsoft Corporation</li> <li>Microsoft Code Signing PCA</li> <li>Microsoft Code Signing PCA</li> <li>Microsoft Code Signing PCA</li> <li>Microsoft Time-Stamp Service</li> <li>Microsoft Time-Stamp Service</li> <li>Microsoft Time-Stamp Service</li> </ul>	Console1 - [Console Root\Certificate	
<ul> <li>Certificates - Current User</li> <li>Personal</li> <li>Certificates</li> <li>Trusted Root Certification Autho</li> <li>Enterr</li> <li>All Tasks</li> <li>Intern</li> <li>Yiew</li> <li>Active</li> <li>New Window from Here</li> <li>Truste</li> <li>New Taskpad View</li> <li>Truste</li> <li>Certificates</li> <li>Window from Here</li> <li>Truste</li> <li>Refresh</li> <li>Other</li> <li>Certification Roots</li> <li>Smart-caro masca roots</li> <li>Microsoft Corporation</li> <li>Microsoft Time-Stamp Service</li> <li>Microsoft Time-stamp Service</li> <li>Microsoft Time-stamp Service</li> </ul>	늘 🧼 🖄 📅 📔 🛍 🗎 🙆 😽 🚺	D.
	<ul> <li>Certificates - Current User</li> <li>Personal</li> <li>Certificates</li> <li>Trusted Root Certification Autho</li> <li>Certificates</li> <li>Trusted Root Certification Autho</li> <li>Certificates</li> <li>Intern</li> <li>Active</li> <li>Active</li> <li>Active</li> <li>Truste</li> <li>Other</li> <li>Certifi</li> <li>Certifi</li> <li>Help</li> </ul>	AddTrust External CA Root Baltimore CyberTrust Root Class 3 Public Primary Certifica Class 3 Public Primary Certifica Class 3 Public Primary Certifica Class 3 Public Primary Certifica T Microsoft C procert right Assurance EV Ro Entrust.net Certification Author Entrust.net Secure Server Cert Equifax Secure Certificate Aut GeoTrust Global CA GlobalSign Root CA Go Daddy Class 2 Certification GTE CyberTrust Global Root http://www.valicert.com/ Microsoft Authenticode(tm) Ro Microsoft Code Signing PCA Microsoft Corporation Microsoft Corporation Microsoft Root Authority Microsoft Root Authority Microsoft Root Authority Microsoft Time-Stamp Service Microsoft Timestamping PCA NO LIABILITY ACCEPTED, (c)9

- 9. In Genesys Administrator, navigate to **Provisioning > Environment > Applications** and open your Backend Server application.
- 10. Click the **Options** tab and navigate to the [security] section.
- 11. Set the **trusted-ca-type** option to MSCAP.
- 12. Click Save & Close.

#### End

#### **Next Steps**

• Return to the Genesys Web Engagement Security page.