Genesys Security Deployment Guide

System Level Guides 8.5.x

10/30/2023

Table of Contents

Genesys Security Deployment Guide

Use this guide to introduce you to security features offered by Genesys software, and how to install, configure, and run them.

This Guide applies to all releases of Genesys software, and is updated regularly.

Introduction

Overview

New in This Release

Document Change History

Authentication and Authorization

User Authentication and Authorization

Passwords

Access Control

No Default Access for New Users

Inactivity Timeout

Security Banner

Protection of Data at Rest

Encryption of Configuration Database Password

Encryption of Data in Databases

Encryption of Call Records

Hiding (Masking) Data

Service Availability

Application Redundancy

Proxy and Parallel Servers

Client-side Port Definition

Protection of Data in Transit

Secure Connections using TLS

Federal Information Processing Standards

Web Application Security

Open Web Application Security Project

RESTful Web Services

(FIPS)

Secure HTTP (HTTPS)

Secure Real-Time Transport Protocol (SRTP)

Lightweight Directory Access Protocol Secure (LDAPS)

Data Privacy

Overview

General Data Protection Regulation (GDPR)

Genesys Engage cloud Support for GDPR

Genesys Engage Premise Support for GDPR

Introduction

This Guide provides an overview of the security risks and requirements inherent in a contact-center environment, and describes how Genesys addresses those risks.

> **Warning**
> Genesys software is not intended to be used in an unrestricted Internet-facing environment. Genesys strongly recommends that you use security features described in this document and elsewhere in combination with good system-security practices—including secure and/or encrypted file storage and the use of firewalls where appropriate.

Overview

The risks and threats inherent to data networks also apply to contact centers. In general, the risks common to contact center solutions can be broken down into the following categories:

- Authentication and authorization
- Protection of data at rest
- Service availability
- Protection of data in transport
- Web application security

This Guide is not an exhaustive study of all of the security features that Genesys offers. Many security features are documented elsewhere in the Genesys documentation suite. As these features evolve, so too will this document—to provide a concise one-stop reference for all of your security needs.

Security Deployment

This Guide describes each of the Genesys security features mentioned in the preceding sections. It also includes detailed deployment instructions for those features that can be installed either system-wide, or in a manner that is consistent for all products. If the deployment process differs between components or products, you are referred to appropriate product documentation for the specific steps.

Where part of the deployment of a feature is performed as part of another procedure, this document provides an overview of that part. For detailed instructions, you are referred to the appropriate product documentation.

> **Tip**
> If you are considering deploying Genesys in a complex environment with multiple users, roles, and credentials, Genesys strongly recommends that you retain an experienced security consultant or a Genesys Customer Care representative to review your configuration and security plan.

In Case of Emergency

If you have a problem or emergency related to the security of your Genesys system, do not hesitate to contact Genesys Customer Care at 1-888-GENESYS (436-3797) or customercare@genesys.com. Do not further jeopardize the safety of your system by discussing the situation in online message boards or applying any unapproved remedial software.

Security and Standards Compliance

The Genesys suite of products is designed to make up part of a fully functioning contact center solution, which may include certain non-Genesys components and customer systems. Genesys products are intended to provide customers with reasonable flexibility in designing their own contact center Solutions. As such, it is possible for a customer to use the Genesys suite of products in a manner that complies with the security-related business standards such as General Data Protection Regulation, ISO 27001/27002 (formerly 17799), HIPAA, PCI DSS etc. However, the Genesys products are merely tools to be used by the customer and cannot ensure or enforce compliance with these standards. It is solely the customer's responsibility to ensure that any use of the Genesys suite of products complies with these business standards. Genesys recommends that the customer take steps to ensure compliance with these business standards as well as any other applicable local security requirements.

Our Pure Engage Cloud infrastructure is compliant with industry standards such as PCI DSS, SOC2 Type II, ISO 27001, and HIPAA.

New in This Release

The following new security features and functions have been introduced in release 8.5:

- Kerberos authentication is supported by some components for user authentication.
- Call recordings can be encrypted, then decrypted for feedback. See Encrypted Call Recordings.
- When configuring TLS, you can specify the version of TLS protocol to use to secure connections.

Supporting Components information for all features has been updated as required.

Authentication and Authorization

Unauthorized data access and the abuse of user privileges are common concerns for multi-user environments. Ensuring data correctness and its instant availability over the course of its lifecycle is critical for the business. Data, software, or the configuration must not be corrupted or modified by an unauthorized party.

Genesys provides the following security features to address data confidentiality:

- User Authentication and User Authorization
- User Passwords
- SNMPv3 Passwords
- Object-Based Access Control
- Role-Based Access Control
- No Default Access for New Users
- Inactivity Timeout
- Security Banner at Login
- Last Logged In Display

> **Tip**
> Genesys strongly recommends careful consideration of network, file system, database, and operating system permissions to complete the protection afforded by these features.

User Authentication and User Authorization

Secure access to the resources of an interaction-management system plays an important role in ensuring trouble-free operation of all system parts and functions. Changes made by unqualified users can adversely affect system availability and the quality of service.

Secure access to a system requires that each user pass the following tests:

- User authentication—This test checks to see that the user is actually who he or she claims to be, and is usually carried out using a system of passwords or other unique and confidential (or unalterable) identifiers.
- User authorization—After the user is authenticated, this test determines that the user is entitled to access the system, either all or parts thereof, and defines what the user can do to or with the data that they can access. This is usually carried out using a system of permissions or similar access rules.

The data a Genesys solution requires for operating in a particular environment, as well as the applications and the solutions, are described in the form of Configuration Database objects. To be authenticated, any person who needs access to this data or these applications must have an account in this database.

User authorization is provided by the security mechanism implemented in Configuration Server, which allows the system administrator to define separately a level of access for any account with respect to any object.

> **Important**
>
> In the context of user authentication and authorization as described in this Guide, the term *object* refers to an instance of an object type, not the object type itself.

User Authentication

User authentication determines that a user is actually who he or she claims to be. In a physical environment, this is often implemented by photo identification cards. In a computer system, this is often accomplished by a password system—the user must enter the correct username and password combination before being authenticated.

Genesys software uses a password system. Each user is assigned a unique username and a confidential password. When logging in to any Genesys interface, the user must enter both of these identifiers before they can be authenticated. User authentication is carried out by one of the following:

- Configuration Server, as described in User Passwords.
- An external authentication module, to which Configuration Server sends the login credentials. The external authentication module performs the actual authentication. For more information about

external authentication, refer to the *Framework External Authentication Reference Manual*.

Kerberos Authentication

Some Genesys components (Management Framework, Platform SDK, and Workspace Desktop Edition) also support the use of Kerberos external authentication to authenticate users. This enables authentication to be done on the client side before a connection to Configuration Server is made. For more information, refer to the "Kerberos External Authentication" chapter in the *Framework External Authentication Reference Manual*.

User Authorization

After the user is authenticated, user authorization determines that the user is entitled to access the system, either all or parts thereof, and defines what that user can do to or with the data that they can access. In a physical environment, this could be implemented by a series of locked doors - only certain people are authorized to access what lies behind each door, and only authorized people carry the keys to the doors to which they are authorized to enter. Similarly, in a computer system, this is often accomplished with a permissions system, in which only authorized users can see (in some cases) only specific data and can perform only certain tasks on that data.

Genesys software uses two levels of permissions to implement user authorization:

- Object-Based Access Control—What the user can see and do to an object is controlled by a set of permissions.
- Role-Based Access Control—Provides an additional layer of protection of your data from unauthorized users by defining what is displayed in the interface and therefore limiting the data to which a user has access.

Supporting Components

Most Genesys components support authentication and authorization as described in this document. The following components support authentication and authorization, but do not use Genesys Configuration Server:

- Genesys Interactive Insight (GI2)
- Genesys Enterprise Telephony Software (GETS)
- Genesys Quality Management (GQM) OEM products from Zoom
- Workforce Management-related OEM products from:
  - SilverLining
    - Genesys Training Manager
    - Genesys Skills Assessor
  - Aria
    - *Gplus* Adaptor for Aspect WFM

- *Gplus* Adapter for IEX WFM
- *Gplus* Adapter for Teleopti WFM
- *Gplus* Adapter for Verint WFM

User Passwords

In Genesys, user authentication is provided by the use of passwords stored in the Configuration Database. Any person who needs access to Genesys data or applications must have an account in this database.

Logging In

At startup, every Genesys GUI application opens a Login dialog box for users to supply a User Name and Password, which are used for authentication. The authentication procedure succeeds if both of the following conditions are true:

- The password specified by the user is a valid password. That is, it meets the criteria of a valid password as described in this chapter.
- A person with the specified User Name and Password is registered in the Configuration Database.

Otherwise, the working session is stopped.

The date and time at which a user last logged into a specified Configuration Server or Configuration Server Proxy Application object via a GUI can be displayed when the user logs into the same server. This feature enables an individual user to recognize possible misuse of their account. For information about this feature, see Last Logged In.

Passwords in a Multi-Tenant Configuration

In multi-tenant configurations, the inheritance rule applies for many of the password-related features listed in this chapter. If a feature is not configured for a particular tenant, rules for ancestor tenants are used, up to the ENVIRONMENT tenant (assuming there is no termination of inheritance otherwise). If no rule is set in the ancestor tree, no limits exist.

If a particular tenant requires different settings from its ancestors, you can configure this in two ways:

- Configure only those settings that are to be changed. Use this method only if you want to change a few specific settings, but otherwise use the inherited value for the other settings. This will override the inherited values for those settings, and leave the values of other settings unchanged, including those inherited from ancestor tenants. Where applicable, child tenants of this tenant will inherit the new values of the changed settings.
- Reset all options to their default values and then customize the values as required for this tenant. Use this method only if you want to reset or change multiple settings for this and descendent tenants. To set all options in the **[security-authentication-rules]** section to their default settings, set the **tenant-override-section** option to `true`. This option breaks the inheritance chain, effectively making this tenant a new inheritance node for all child tenants, and is easier than manually changing each option. Then, for this and its child tenants, you can set appropriate values for any individual option for which you do not want the default value to apply. For detailed descriptions of these configuration options, refer to the *Framework Configuration Options Reference Manual*.

This override is available for all options in the **[security-authentication-rules]** section.

Password Properties

A generic password for most Genesys applications has very basic properties, as follows:

- Has a maximum length of 64 characters
- Contains any combination of the following characters:
  - Alphanumeric characters of any case, such as A, a, Φ, ϕ, 1, and 2
  - Punctuation characters, such as comma (,), period (.), colon (:), and semi-colon (;)
  - Parentheses (), curly brackets {}, and braces []
  - Other characters found on a standard keyboard, such as %, &, @, and #

This section describes how to set customized properties of a password. These properties provide additional security to a password system by defining specific criteria that a valid password must meet.

For detailed descriptions of the configuration options used to define properties of a valid password, refer to the *Framework Configuration Options Reference Manual*.

Password Length

Passwords can be anywhere from 0 to 64 characters long. They cannot exceed 64 characters, but if required, you can set the minimum length to as little as zero (0), which would indicate that empty, or blank, passwords are acceptable to the Configuration Server.

To set a minimum password length, use the configuration option **password-min-length**. This option is defined at the Tenant level, and applies to all users in the Tenant.

> **Important**
> This feature does not apply if you are using external authentication.

Empty (Blank) Passwords

If you are not using external authentication, empty passwords in a client request are permitted or rejected based on the value of the Configuration Server option **password-min-length**, or in its absence, **allow-empty-password**, as follows:

- If the **password-min-length** option is used in a Tenant, **allow-empty-password** does not apply to any user in that Tenant.
- If **password-min-length=0**, empty passwords are permitted; any other value means empty passwords are not permitted. The value of **allow-empty-password** is ignored.

- If **password-min-length** is not set, and **allow-empty-password=true**, empty passwords are permitted; if **allow-empty-password=false**, empty passwords are not permitted.

> **Important**
>
> Genesys strongly recommends that you use the **password-min-length** option, instead of **allow-empty-password**. The latter is provided only for purpose of backward compatibility.

If you are using external authentication, Genesys strongly recommends that you do not allow empty passwords at all by setting **allow-empty-external-password** option to `false`. There might be instances in which an LDAP server, instead of rejecting a blank password, might (depending on the LDAP Server configuration) interpret this to mean that it should make an unauthenticated connection, giving the false impression that authentication has succeeded. To allow empty passwords in Configuration Server and still avoid this, set the **allow-empty-external-password** option to `false` so that configuration will enforce at least one character in a password sent to an external system.

Password Characters

You can define the type and case of characters that a password must contain by using the configuration options in this section. Configuration Server uses these options to validate a new password when it is being created or changed. Any password that does not satisfy the requirements is not a valid password and will be rejected by Configuration Server.

You can configure any combination of the character type and case requirements listed in the table below. To implement these requirements, set the corresponding configuration option to `true` in the **[security-authentication-rules]** section of the Tenant's options.

> **Important**
>
> Any characters that are enforced by these requirements must be ASCII characters. Other characters in the password do not have to be ASCII characters, and can be as shown above.

Character Type or Case	Description	Examples	Configuration Option
Alphabetic	A password must contain at least one alphabetic character (a–z, A–Z).	abcde, ab8de, a1234, a12фи	**password-req-alpha**
Mixed Case	A password must contain at least one upper-case (A-Z) and one lower-case (a-z) character.	pAssWoRD, MyName, MyTфьy	**password-req-mixed-case**
Numeric	A password must	password123,	**password-req-**

Character Type or Case	Description	Examples	Configuration Option
	contain at least one numeric character (0–9).	myname8, кгыышф7	**number**
Punctuation	A password must contain at least one punctuation character from the set: !\"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~).	password!, my-name, ьн-тфьу	**password-req-punctuation**

Password Expiration

You can define a time interval for passwords, after which a password will be considered expired. Set the time interval in the **password-expiration** option. A user with an expired password can log in using their expired password only:

- by using a legacy application version 8.0 or earlier that does not support this feature, or
- if the **override-password-expiration** option is set to `true` in the user's Person configuration object.

> **Important**
> Either one of these can be overridden by using the **force-password-reset** option.

Therefore, a current application version that does not provide the password change feature will lock out users with expired passwords unless they are explicitly configured as described in the second point above. However, an earlier version (for example, 8.0) of the same application will not lock out users because it is considered a lower-level application.

To configure a notice to be displayed to a user giving them advanced notice that their password will be expiring, set **password-expiration-notify** to `true`.

You can also configure the password of an individual user to be exempt from the expiry time set at the Tenant level. In the Annex of the particular Person object, set **override-password-expiration** to `true`.

Resetting Passwords

Password reset can occur in one of two ways:

- The user can change their password once they have logged in to an interface.
- The system administrator can force the user to reset their password the next time that they log in.

For detailed descriptions of the configuration options used to control the resetting of passwords, refer to the *Framework Configuration Options Reference Manual*.

Password Reset by User

By default, a user can change his or her own password at any time, once he or she has successfully logged in to one or more interfaces. He or she does not need to have Change permission to their own User object. To restrict user-initiated password changes to only users with Change permissions to their own Person objects, set **password-change** to `false` in the configuration file of the master Configuration Server.

> ### Important
> Genesys recommends that you not activate password expiration if users are unable to change their passwords themselves. If password expiry is enabled, a user whose password will be expiring will be unable to change their password when they receive the warning notice that their password will be expired. However, once the password has expired, the user can change their password by logging in via an interface which supports changing password at next login (see Password Reset Forced by System Administrator). In this case, the password reset forced by the system administrator overrides the user's ability to change their password at any time.

Password Reset Forced by System Administrator

Starting in release 8.1.1, a system administrator can force users to reset their password at the user's next login. If supported by the application, the user must reset his or her password at this point, or he or she cannot gain access.

In Genesys Administrator, the System Administrator initiates password reset by selecting the **Reset password** checkbox on the **Configuration** tab of each new or existing User object. This must be done individually for each User object. As a result, password reset is now required in the system, and the user can log in only using the following applications:

- An application that supports the password change feature.
- A legacy (pre-8.1.1) application for which the feature is not enforced.
- A version 8.1.1 or later application that is supposed to support password change but does not (such as Configuration Manager), in which **no-password-change-at-first-login** is set to `true`. This option enables the application to be treated as a legacy feature that does not support password change if the user logs in through that application.

If your security policies do not allow for these exceptions to exist, set **force-password-reset** to `true` at the Tenant level. In addition to forcing all users to change their passwords when they next log in, this option will cause the enforcement of password change at first login regardless of whether applications are legacy or configured with the **no-password-change-at-first-login** option. However, this means that these applications will not be usable by the user unless he or she first changes his or her password using a compliant application.

> ### Important
> The Password Reset feature is supported by Genesys Administrator starting with

> version 8.1.2. Configuration Manager and Solution Control Server do not support this feature.

Re-using Passwords

You can define the frequency with which passwords can be re-used. That is, they can re-use a password only after they have used a specified number of different passwords. Set the number of unique passwords that must be used in the **password-no-repeats** option.

Account Lockout After Failed Connection Attempts

You can configure your system to lock out a user account after a specified number of unsuccessful connection attempts to Configuration Server.

Configuration Server tracks connection attempts for a user account when the first unsuccessful connection attempt is made. If one user account is unsuccessful when trying to connect to Configuration Server, and further attempts to connect are also unsuccessful, the user with that account will be unable to connect (or try to connect) until it is unlocked by an administrator or the lockout expired. However, if the user successfully connects before the number of unsuccessful attempts has been reached, the account is not locked out.

Failed connection attempts are tracked individually and independently on each Configuration Server instance. In other words, an account that is locked out at one Configuration Server may not be locked out at another Configuration Server, unless it has also exceeded the number of failed attempts at that server.

A failed connection attempt is defined as one of the following:

- A new connection to Configuration Server cannot be completed because of incorrect authentication credentials.
- Authentication of the user account fails on an existing connection because of incorrect authentication credentials.

Connection attempts for a given account are not tracked if the account is disabled (in Genesys Administrator or Configuration Manager), or if the account is configured to override the lockout.

To configure basic account lockout functionality, you need to define the following parameters at the Tenant-level:

- The number of unsuccessful connections allowed before lockout takes effect. Set this using the **account-lockout-threshold** option.
- The length of time that the lockout will last until the account can then attempt to connect again. Set this using the **account-lockout-duration** option.
- The length of time since the last failed connection attempt in which another failed attempt will count towards the number of allowed connections before lockout. Set this using the **account-lockout-attempts-period** option. This parameter enables lockouts to occur only if unsuccessful attempts are

made in quick succession.

- Optionally, you can specify that the account will be locked out until an administrator explicitly unlocks it by setting the **account-lockout-mode** option to 1.

> **Important**
>
> For detailed descriptions of the configuration options used to control account lockouts, refer to the *Framework 8.5 Configuration Options Reference Manual*.

When an account is locked out, the following occurs:

- The account's status changes to Locked.
- Configuration Server generates log event `21-22140`.
- The date and time of lockout, and the instance of the Configuration Server to which the client application, used to review Person object options, is currently connected, is recorded in the read-only **last-locked-at** option in the user's Person object.

The administrator can unlock an account manually by:

- Changing the password for the user.
- Enabling force password reset for the user.
- Setting the **account-override-lockout** option to `true` at User-level, which overrides the account lockout for that user.

This basic configuration applies to all user accounts in the Tenant. In a multi-tenant configuration, the inheritance rule also applies (see Passwords in a Multi-Tenant Configuration).

Account Expiry after Inactivity

You can configure a time interval after which an account can be disabled (that is, expire) if the password for that account has not been used. After the time interval has expired, the account will be considered expired and the user will not be able to log in until the account has been reactivated by the system administrator. Configuration Server checks for expired accounts when an account belonging to the Tenant tries to log in or authenticate, or when a User object belonging to this Tenant is retrieved or changed.

> **Important**
>
> - This feature does not work correctly if the Last Logged In feature is not configured on the master Configuration Server and all Configuration Server Proxies. Calculations for

the expiration of a particular account starts after the first login is recorded as a part of the Last Login feature; if the last login is not available, account expiration does not apply.

- This feature does not apply to accounts that are externally authenticated, if an external authentication Domain is configured.

This setting can be overridden for individual users using the **override-account-expiration** option.

Account expirations are tracked individually and independently on each Configuration Server instance. In other words, an account that is expired at one Configuration Server may not be expired at another Configuration Server, unless it has also been disabled because of inactivity.

In a multi-tenant configuration, the inheritance rule also applies (see Passwords in a Multi-Tenant Configuration, above).

Password Encryption

Passwords are encrypted automatically within the system, as follows:

- During transit between servers and Configuration Server, Genesys passwords are encrypted using the AES128 encryption algorithm, whether or not Transport Layer Security (TLS) is used.
- In the Configuration Database, user passwords are stored with a one-time-use SALT that is encrypted with TEA. This combination is then hashed using the SHA256 algorithm before storage.
- Passwords in configuration files are encrypted using TEA.

Important

If a password to be encrypted contains one or more UNIX shell special characters, the password must be enclosed in single quotes if it is provided on the command line. For example, if the password is $Montana, enter the following at the command line:

```
confserv -p confserv '$Montana'
```

Passwords that were hashed in a version of Management Framework prior to 8.1.2 use MD5 until they are changed. In Management Framework 8.1.2 and later, SHA256 is used for new Person objects and for existing Person objects when they change their password.

The algorithm used to hash a password is stored internally by Configuration Server so it can know when processing an authentication request to hash the submitted password using MD5 or SHA256 before comparing it to the stored password. All new or updated passwords will be updated to be hashed with SHA256 before storage.

Passwords must be hashed using the same algorithm. This creates the one case in which you must use the MD5 algorithm. If you are running Configuration Server Proxy 8.1.0 or earlier, that supports only MD5, and a master Configuration Server 8.1.1 or later, that can support SHA256, the two servers may be running together long enough to encounter password requests. Because they use two different hashing algorithms, the master Configuration Server will be unable to process the requests. You must force Configuration Server to use MD5 by setting **force-md5** to `true` in the confserv section of the master Configuration Server. Refer to the *Framework Configuration Options Reference Manual* for a detailed description of this option.

> **Important**
>
> Genesys does not recommend running a newer version of Configuration Server with an earlier version of Configuration Server Proxy. However, this situation is allowed for a short time during migration.

Hiding Passwords in Log Files

Genesys user passwords are never written to log files, and therefore do not need to be encrypted or otherwise hidden. To prevent a non-user password from appearing in plain text in log files and attached data, you can encrypt them in logs as follows:

- Hide the password used to access the Configuration Database. Refer to Encrypted Configuration Database Password. This password encryption does not use the SALT used with user passwords.
- If passwords appear in the UserData, Reasons, or Extensions attributes of a log, you can hide all or part of them with a string of asterisks or other characters. Refer to Hide Selected Data in Logs.

Restrictions on User Connections

In addition to the access rights provided by Object-Based and Role-Based Access Control, Configuration Server also provides some basic restrictions on user connections. This section describes these restrictions.

Number of Concurrent Connections

You can configure the maximum number of simultaneous connections that each account can have with a single instance of Configuration Server. If an account tries to exceed the number of connections, login is denied.

To specify the maximum number of connections, use the Tenant-level **max-account-sessions** option. Refer to the *Framework Configuration Options Reference Manual* for detailed information about this option.

> **Important**

> Sessions that are restored and authenticated through existing sessions are not included in the count of sessions for this feature.

In a multi-tenant configuration, the inheritance rule also applies (see Passwords in a Multi-Tenant Configuration).

Control over Linked Connections

In a situation where a user is editing an object that is linked to other objects, only a user with access to one or more of those linked objects can change the link between their linked objects and the object being edited.

Control over HA pairs

Configuration Server restricts two applications created with different accounts from being linked (configured) as a redundant HA pair. This ensures that the two applications must be started from the same account.

SNMPv3 Passwords

Starting in release 8.1, you can configure your SNMPv3 passwords for both authentication and data privacy so the passwords are:

- masked when you type them into Genesys Administrator, and
- encrypted by Configuration Server in the Configuration database.

Feature Description

There are two SNMPv3 passwords: one for authentication, and one for data privacy. Prior to Genesys release 8.1, these passwords were not masked (displayed as a string of asterisks, for example) when a user was entering them in the interface. They were also stored as plain text in the Configuration Database.

Starting in release 8.1, this feature masks the passwords when a user is entering them in Genesys Administrator, and encrypts them in the Configuration Database.

Feature Configuration

To configure this feature, set the following options in the options of the SNMP Master Agent Application object:

- In the **[snmp-v3-auth]** section, set the **password** option to the password used for authentication by the SNMPv3 system.
- In the **[snmp-v3-priv]** section, set the **password** option to the password used for data privacy in the SNMPv3 system.

The **password** option masks and encrypts the SNMPv3 user's password used for authentication or data privacy, depending on the section (**[snmp-v3-auth]** or **[snmp-v3-priv]**) in which the option is configured.

For more information about this option and the related sections, refer to the *Framework Configuration Options Reference Manual*.

Object-Based Access Control

Object-Based Access Control implements user authorization by using permissions to define what each user can do to the objects to which he or she has access.

In general, any object for which permissions is not explicitly granted is forbidden.

Elementary Permissions

User authorization is provided by the combination of a set of elementary permissions, shown in the following table. This security mechanism implemented in Configuration Server allows the system administrator to define separately a level of access for any account with respect to any object.

Permission	Description
Read	Permission to read information and receive updates about the object.
Create	Permission to create objects in this folder.
Change	Permission to change the properties of the object. The Change permission is the same as allowing “Write” access.
Execute	Permission to perform a predefined action or set of actions with respect to the object. This is also required for a user to log in to a Graphical User Interface (GUI) application.
Delete	Permission to delete the object.
Read Permissions	Permission to read the access control settings for the object.
Change Permissions	Permission to change the access control settings for the object.
Read & Execute	• Permission to read information and receive updates about this object. • Permission to perform a predefined action or set of actions with respect to this object.
Propagate	For container objects (such as Tenants, Folders, Switches, IVRs, and Enumerators). The `Propagate` check box controls whether to propagate this set of elementary permissions to the child objects. By default, the check box is selected).

Access Privileges

The access privileges of authenticated user accounts define what the user can and cannot do within this application. The Execute permission is used to control access to applications, solutions, and other configuration objects. Without such permission, the user cannot work with a given application or execute control over a given object. Combinations of the Read, Create, Change, and Delete permissions define the level of access to configuration data. For example, users might have access to a real-time reporting solution but will get reports only about objects they have permission to read.

Access control for daemon applications is different from that for GUI applications. Access permissions for GUI applications are determined by the profile of the person who is currently logged in. Daemon applications do not have an explicit login procedure. Instead, their access permissions are determined by the permissions of the account with which they are associated: a personal account or the SYSTEM account. Any personal account registered as a Person object in the Configuration Database can be used as an account for any daemon application. By default, every daemon application is associated with a special account SYSTEM that has Read and Execute permissions for all objects in the Configuration Database except Access Groups.

Access Groups and Default Security Settings

Access Groups are groups of Person objects who must have the same set of permissions with respect to Configuration Database objects. By adding individuals to Access Groups—and then setting permissions for those groups—access control is greatly simplified.

Genesys offers these preconfigured Default Access Groups:

- Users: Members have Read and Execute permissions with respect to all objects except Access Groups.
- Administrators: Members have a full set of permissions with respect to all objects except the Super Administrators Access Group.
- Super Administrators: Members have a full set of permissions with respect to every object in the Configuration Database. No person is added to this group by default.

In addition, in a hierarchical multi-tenant configuration, Configuration Server creates these Default Access Groups for each new Tenant object:

- Users: Members have Read and Execute permissions with respect to all objects under this Tenant except Access Groups.
- Administrators: Members have a full set of permissions with respect to all objects under this Tenant.

> **Important**
>
> You cannot delete or rename Default Access Groups, although you can change their default privileges.

New Users

By default, Configuration Server considers a new user to be a member of the EVERYONE group. It does not assign that user to any Access Group when he or she is created. Likewise, the new user is not automatically assigned any permissions by default. In effect, the new user has no privileges, and cannot log in to any interface or use a daemon application. The new user must be explicitly added to appropriate Access Groups by an Administrator or by existing users with access rights to modify the user's account. Refer to *Genesys Administrator 8.1 Help* for more information about adding a user to an Access Group.

By default, this behavior applies to all new users added by Configuration Server release 7.6 or later. Users created before release 7.6 keep their existing set of permissions and Access Group assignments. If you want new users to be added automatically to predefined Access Groups, as was the behavior prior to release 7.6, you must manually disable this feature by using the Configuration Server **no-default-access** configuration option.

For more information about this feature, including how it works and how to modify it, see No Default Access.

Master Account and Super Administrators

The Configuration Database contains a predefined user object, otherwise known as the *Master Account* or *Default User*. This account, named **default** and with a password of **password**, is not associated with any Access Group. The Master Account always exists in the system and has a full set of permissions with respect to all objects in the Configuration Database. You must use this account when you log in to the Configuration Layer for the first time after Configuration Database initialization.

Important

- In addition to emergency situations, you still must use the Master Account for some specific administrative tasks, especially during migration. Refer to the description of the specific tasks throughout this and other documents, including the *Genesys Migration Guide*, to determine whether you need to use the Master Account, or whether you can use another account that has the required permissions.
- Genesys recommends that you change the default name and password of the Master Account, store it securely, and use this account for only emergency purposes or whenever specifically required.

During one of your first working sessions, create non-agent accounts for everyone who needs full access to all objects and add these accounts to the Super Administrators group. By default, every member of the Super Administrators group has the same permissions as the Master Account.

EVERYONE Group

Think of the EVERYONE group as an Access Group that includes every user registered in the Configuration Database. You cannot delete or modify this group, which, by default, has no permissions set for any configuration objects.

Multiple Permissions

Multiple (and unequal) permissions can affect a User's access to an object. If a User belongs to multiple Access Groups and those Access Groups have different permissions for the object, the User gets the logical union of privileges from the set of access privileges with one exception: the No Access access privilege supersedes all others.

Examples

Assume that:

- User John is a member of Access Group A and Access Group B.
- Access Group A has Read-only access to the Host Friday, but Access Group B has Read/Write access to the Host Friday.

As a result, John has Read/Write access to the Host Friday.

To understand the exception to this rule, now assume that:

- User John joins Access Group C, which has No Access privileges to the Host Friday.

As a result, User John now has no access to the Host Friday.

Setting and Changing Permissions

Permissions are set and changed in Genesys Administrator on the **Permissions** tab of the appropriate object.

> **Important**
> Use caution when assigning permissions. Remember, the more complex the security system is, the more difficult it becomes to manage the data and the more it affects the performance of the Configuration Layer software.

Granting Permissions

To grant permissions, use the following steps. **[+] Show steps**

1. In Genesys Administrator, open the **Permissions** tab of the object for which permissions are to be granted.
2. Click **Add User** or **Add Access Group**, as applicable. A list of configured Users or Access Groups is displayed in a dialog box.
3. Select the User or Access Group to be granted permission.
4. Click **OK**. The dialog box closes, and the selected User or Access Group appears in the list on the

Permissions tab, with default Read permission.

5. Click **Save** to save your configuration changes.

Modifying Permissions

To modify permissions, use the following steps: **[+] Show steps**

1. In Genesys Administrator, open the **Permissions** tab of the object for which you want to modify permissions.
2. Do one of the following:
   - Double-click on the name of the User or Access Group for whom you want to change permissions. A list of permissions appears in the **Access** dialog box; those permissions with a checkmark are currently assigned to that User or Access Group. Check those permissions that you want the User or Access Group to have. Clear the checkbox for those permissions that you do not want the User or Access Group to have.
   - Click the corresponding entry in the Access column and select an Access Level from the drop-down list. These Access Levels are pre-defined sets of permissions.
3. When you have checked the required permissions and cleared the permissions not required, do one of the following:
   - Click **OK** to save the changes. (If no changes were made, the **OK** button is not active.)
   - Click **Cancel** to save the permissions with no changes.
4. Click **Save** to save your configuration changes.

Removing Permissions

To remove permissions previously granted to a user or group of users, use the following steps: **[+] Show steps**

1. In Genesys Administrator, open the **Permissions** tab of the object from which the User or Access Group is to have permissions removed.
2. Select the User or Access Group.
3. Click **Remove**. The User or Access Group no longer appears on the `Permissions` tab for this object.

> **Important**
>
> This action does not remove the User or Access Group from the Configuration Database. It only removes the User or Access Group from the list of User or Access Group objects that have access to this particular object. To remove the Access Group or User from the Configuration Database, refer to instructions in *Genesys Administrator 8.1 Help*.

4. Click **Save** to save your configuration changes.

Changing Permissions Using Propagation

The **Propagate** check box in the properties of so-called container objects (such as Tenants, Folders, Switches, and IVRs) allows you to manage access permissions to both the container object and those objects that they contain—the so-called child objects—without affecting the permissions of other Users or Access Groups.

When the **Propagate** check box is selected (the default setting) for a container object, any changes to permissions to the container object will be propagated to (that is, also made to) the permissions to each child object.

Use propagation when you want to set identical permissions for a user to a container object and all its child objects. For example, if you are setting up a new user or Access Group, and that user or group is to have identical permissions to a container object and all the objects that it contains, you have to add permissions for that user or groups only once—in the container object.

If you want to change the permissions to the container object without changing those of the child objects, clear the **Propagate** check box before changing the object's access permissions.

The setting of the **Propagate** check box (checked or unchecked) is saved between propagations. This enables you to ensure that subsequent changes to permissions settings are consistently propagated or not.

If you want to set permissions for only the child objects without changing those of the parent object, set the child permissions as required. If the consistently check box in the parent is checked for the users whose permissions were changed, any changes for the child will last only until the next propagation. However, if you then change permissions for another user at the parent level, the resulting propagation will not overwrite the earlier manual change to the first user.

Changing Permissions Recursively

If the **Propagate** and **Replace permissions recursively** check boxes are selected for a container object, all permission settings for its child objects are removed and replaced with all permission settings configured for the parent object. Recursion is basically propagation on a clean slate—removing any access rights to the child objects for any users and groups except those propagated from the parent object.

The **Replace permissions recursively** check box is unchecked by default, and must be selected explicitly each time that you want to propagate recursively.

Hierarchical Multi-Tenant Environments

Generally, permissions function in a hierarchical multi-tenant environment in the same way as they do in an enterprise environment. However, there are some exceptions. This section identifies the issues related to using object permissions in a hierarchical multi-tenant environment, and provides workarounds where available.

Accessing Tenants and Objects in Other Tenants

By default, and with one exception, users in one tenant cannot create another tenant, nor can they

access any objects in another tenant. Generally, the only exception to this situation is that the Default User (using the Master Account) and members of the SuperAdministrators Access Group can create new tenants and access objects in other tenants.

The details of default behavior in a hierarchical multi-tenant environment, and recommendations to work around the limitations imposed by that default behavior, are given in the following sections.

Creating New Tenants

A new Tenant object can be created only by the Default User or a user who is a member of the Super Administrators Access Group.

When a tenant is created, permissions to it are granted to the following Access Groups, as follows:

- Environment/default (the Default user)—Full control
- Super Administrators (from the Environment Tenant)—Full control
- SYSTEM (from the Environment Tenant)—Read & Execute (RX)
- [new Tenant]\Administrators—Read & Execute (RX)
- [new Tenant]\Users—Read & Execute (RX)

Resolution
Add users as necessary to the Super Administrators group to enable them to create tenants. Refer to *Genesys Administrator 8.1 Help* for instructions about adding users to Access Groups.

Providing Users Access to Objects in Other Tenants

By default, a new user is not granted access to any objects. As in an enterprise environment, each new user must explicitly be granted permissions and/or added to an Access Group with permissions, to access any objects. See No Default Access for New Users for more information.

To log in to an Application, a user must have at least Read & Execute permissions for that Application. After he or she is logged in, the user can access only those objects in his or own Tenant; he or she cannot access any objects in another Tenant.

Resolution
To gain access to objects in another Tenant, the user must be granted permissions to those other objects by one of the following:

- the creator of the other Tenant
- another member of the Super Administrators Access Group

Providing Users in Parent Tenant Access to Objects in Child Tenants

A user in a parent tenant has no default access to the objects in the child tenants.

> Tip

To work around this limitation, do one (or both) of the following:

- Explicitly grant at least Read access to all child tenants.
- Explicitly add the user to one of the two built-in Access Groups in each child tenant—Administrators or Users.

Voice Platform Solution Limitation

When a hierarchical multi-tenant configuration is used with the Voice Platform Solution in a managed server setting, a major limitation arises when creating Tenants and Direct Inward Dialing (DID) numbers. In essence, this limitation forces the system owner to create and maintain all Tenants and DIDs for all tenants.

In the Voice Platform Solution, DIDs must be unique across the entire system. The software is designed to validate this uniqueness when DIDs are created. This requires that the user who inputting this information must have at least Read access to all DID objects in all Tenants, and therefore access to the Tenants themselves. However, in a managed server environment, it is highly unlikely that the Service Provider wants one tenant to see, or even know about, other tenants. Therefore, the only user that could input this information would be a member of the Super Administrators Access Group, namely, the system owner. The current model of access permissions does not permit any workaround to this situation at this time.

Role-Based Access Control

> **Warning**
> Role-Based Access Control is complementary to Object-Based Access Control. Appropriate object permissions should be defined before setting up role-based access privileges.

Role-Based Access Control provides an additional layer of protection of your data from unauthorized users by defining what is displayed in the interface and therefore limiting the data to which a user has access.

> **Important**
> In this section, the term user is intended to mean both an individual user and Access Groups. This feature applies to both object types.

Roles enhance object-based access control by limiting the visibility of sets of configuration objects, and allowing you to tune elementary permissions to those objects to a finer level. For example, elementary permissions might indicate that you can write to an object, but roles can be used to restrict writing to an individual property of that object, such as **Name**.

Roles can also be used to protect access to entities that are not represented by configuration objects, such as tracking and troubleshooting information. Elementary permissions do not protect these entities, but it is logical to expect that unlimited access to them is not desirable.

Security Benefits

Permissions alone protect access to all parts of individual objects. In other words, once a user has access to an object, he or she has access to all properties of that object. Role-Based Access Control enables you to fine tune access to your data so that individual properties of objects are also protected. A user's permissions might allow that user to access an object, but roles limit what properties of the object the user can see and what the user can do to those properties. Roles also limit access to resources and functionality beyond configuration. In other words, access to an object can be modified without reconfiguring the object.

Furthermore, roles limit access to resources and functionality. Because roles affect what is displayed to the user, a user will not be made aware of functionality unless it is appropriate to their responsibilities.

Supporting Components

Role-Based Access Control is supported by the following components:

- Management Framework
- Genesys Administrator
- Genesys Administrator Extension

This feature is used by the following components:

- Genesys Administrator, on behalf of Management Framework and Outbound Contact
- Interaction Workspace
- Universal Contact Server
- Knowledge Manager

In addition, Platform SDK provides access to configuration objects needed to implement Role-Based Access Control in an application. For details about how this feature can be used in custom-built applications, refer to the appropriate API Reference for your development platform.

Feature Description

The major component of Role-Based Access Control is a *role*. Roles define what facilities are provided to users to review and manipulate various types of data. These include which property controls are available for items permitted by object permissions, what modules are visible, and access control for entities not represented by configuration objects. A role is assigned to a user, and that user is then able to do only what that role permits.

> **Important**
> One user can be assigned multiple roles, and one role can be assigned to multiple users.

Roles consist of a set of *role privileges*. Role privileges are tasks that can be performed on a given type of data. They are pre-defined in Genesys Administrator and are unique to each product. By default, any role privilege is not assigned to any role, so you must explicitly assign privileges to roles. Role privileges range from general to very specific tasks. An authorized user, normally a System Administrator, bundles these tasks into roles. These roles are then assigned to users. As a result, each user can perform only those tasks for which they have the privileges.

Role-Based Access is enforced primarily by visibility in the interface. When a user logs into an interface that supports roles, what that user sees is determined by the roles which have been assigned. If the user is not assigned a role that grants them access to a piece of functionality, that functionality will not be displayed to the user.

Roles vs. Permissions

Roles are intended to work with permissions to more finely tune what a user in your system can access.

Elementary permissions protect access to a whole object. That is, the permissions applied to the object apply equally to all properties of the object. There is no way to limit access to an individual property of that object. In addition, permissions do not restrict access to any parts of the object - if you have access permissions, you see the entire object.

Roles serve to protect properties of an object by hiding or disabling those properties for which a user should not have access. Different roles can define different access and allowed functionality for the same objects. In essence, roles resolve both problems with using permissions alone—the user can access and work with only those parts of the object to which that user is allowed.

Roles can also be used to protect access to entities that are not configured as configuration objects, such as logs.

In general, when determining the accessibility of an object to a user, the user session cannot retrieve objects if they are not among those objects to which the user has access (as defined by object-access permissions). Then, for that data that is available in the session, role privileges refine what can be done with the data. For example, if the user's permissions do not allow any Change permissions for a set of objects, that user cannot make any changes to those objects regardless of what his or her role privileges are for tasks for properties of those objects.

Multiple Roles

You can assign more than one role to a user. In such cases, the user will have the combined set of privileges granted by each role. In other words, the user is granted any privilege that is granted by at least one of the assigned roles. This ensures that the user is able to perform the tasks of all roles in which they participate.

New Users

By default, new users are not assigned any default roles. They must be assigned roles by a System Administrator or by an existing user with appropriate privileges.

Feature Configuration

> **Important**
> To determine if this section applies to you, see Supporting Components.

Role-Based Access Control is configured in Genesys Administrator. You can create a role, give it a name, and assign it to users in Configuration Manager, but the role privileges can be defined only in Genesys Administrator. Configuration Manager itself does not support the feature.

Configuring Role-Based Access Control

To configure Role-based Access Control, use the following steps: **[+] Show steps**

1. In Genesys Administrator, go to **Provisioning** > **Accounts** > **Roles**.
2. If required, navigate to the folder in which you want to store the new Role.
3. Click **New**.
4. In the **General** section of the **Configuration** tab, enter information in the following fields:
   a. **Name**—The name of this Role. You must specify a value for this property, and that value must be unique within the Configuration Database (in an enterprise environment) or within the Tenant (in a multi-tenant environment).
   b. **Description**—(Optional) A description of this Role.
   c. **Tenant**—This field appears only in a multi-tenant environment, and indicates the Tenant to which this Role belongs. This value is set automatically, and you cannot change it.
   d. **State**—This field is enabled by default.
5. In the **Members** section of the **Configuration** tab, enter the Users and/or Access Groups to whom the Role is to be assigned.

   > **Important**
   > You can complete this step either now or later. If you decide to complete it later, use the steps in Assigning Existing Roles to Existing Users and Existing Access Groups.

6. On the **Role Privileges** tab, define the privileges to be granted by this Role, as follows:
   a. Select the products for which you want to include privileges in the Role. Only installed products that support Role-Based Access Control are listed.
   b. For each privilege, set its value to one of the following:
      - **Unassigned**—(Default) This privilege is not granted by this Role. However, if multiple Roles are assigned to the same User or Access Group, this setting is overridden if another Role sets this privilege as Allowed.
      - **Allowed**—This privilege is explicitly granted by this Role.
7. To save the new Role and register it in the Configuration Database, do one of the following:
   - Click **Save and Close** to return to the list of Roles.
   - Click **Save** to continue configuring the Role.
   - Click **Save and New** to save the new Role and start creating another one.

   If you have assigned this Role to any Users or Access Groups, a configuration dialog box will appear notifying you that Read access for this Role object will be granted to those Users and Access Groups.
8. Click **Yes**.

Assigning Roles

To assign roles to users and Access Groups, use the following steps: **[+] Show steps**

Prerequisites

- The Roles to be assigned, and the Users or Access Groups to which they are to be assigned must exist in the Configuration Database.

Start of procedure

1. Log in to Genesys Administrator, if necessary. You can assign Roles to Users and Access Groups from three locations: Roles, Users, and Access Groups. The following steps describe each of these approaches.
2. Starting from Role objects: To assign one or more Roles to one or more Users or Access Groups, do the following:
   a. Go to **Provisioning** > **Accounts** > **Roles**.
   b. If necessary, navigate to the folder that contains the Roles you want to assign.
   c. Select one or more Roles.
   d. Open the **Tasks** panel, if necessary, and click **Assign Users** or **Assign Access Groups**, as appropriate, in the **User Access** section.
   e. Follow the steps in the **Role Management Wizard** to select the required Users or Access Groups and assign the Roles to them.
3. Starting from User objects: To assign one or more Roles to one or more Users, do the following:
   a. Go to **Provisioning** > **Accounts** > **Users**.
   b. If necessary, navigate to the folder that contains the Users to whom you want to assign Roles.
   c. Select one or more Users.
   d. Open the **Tasks** panel, if necessary, and click **Assign Roles** in the **User Access** section.
   e. Follow the steps in the **User Management Wizard** to select and assign the Roles.
4. Starting from Access Group objects: To assign one or more Roles to one or more Access Groups, do the following:
   a. Go to **Provisioning** > **Accounts** > **Access Groups**.
   b. If necessary, navigate to the folder that contains the Access Groups to whom you want to assign Roles.
   c. Select one or more Access Groups.
   d. Open the **Tasks** panel, if necessary, and click **Assign Roles** in the **User Access** section.
   e. Follow the steps in the **User Management Wizard** to select and assign the Roles.

End of procedure

Removing Roles

To remove (unassign) Roles from Users or Access groups, use the same steps as in Assigning Roles, but select the corresponding **Unassign** option in the Tasks panel.

Example

The scenario for this example is two office clerks responsible for updating information in the Genesys configuration, as follows:

- Clerk A is responsible for update the records for all employees, or User objects (both agents and non-agents).
- Clerk B is responsible for updating the list of skills, or Skill objects, that can be assigned to agents.

You want to use permissions and roles to ensure that each clerk has access to only the data they need to perform their job.

Permissions

Both clerks require Read/Write access permissions to their respective objects—Clerk A to Users, and Clerk B to Skills. Read access enables them to see the complete lists of objects, from which they can choose the specific object to be updated. Write access (the Change permission) enables them to update the objects.

Roles

Define specific roles as follows:

- HR_Clerk: Update information for all employees.
- Operations_Clerk: Update information for all skills that can be assigned to employees who are agents.

Create and configure each Role object with the appropriate role privileges, then assign each role to appropriate users as indicated in the following table:

Role	Role Privileges (as provided in Genesys Administrator)
HR_Clerk	Genesys Administrator - Modules > Provisioning = Allowed Genesys Administrator - Provisioning > Accounts = Allowed Genesys Administrator - Account Provisioning > Agent Info = Allowed Genesys Administrator - Account Provisioning > Users = Allowed
Operations_Clerk	Genesys Administrator - Modules > Provisioning = Allowed Genesys Administrator - Provisioning > Accounts = Allowed Genesys Administrator - Account Provisioning > Skills = Allowed

After the roles are assigned to users, only certain parts of the Genesys Administrator interface will be visible or available for use. The permissions assigned to each user determine what the user can do to or with the data displayed in those visible sections. In addition to the Provisioning tab, each clerk can see and do only the following:

- Clerk A:
  - View the Accounts section with only one item, Users.

- View the full list of Users, from which he or she selects the User to be modified.
- View and modify any property of the selected User.

> **Important**
>
> The Genesys Administrator > Account Provisioning > Agent Info = Allowed privilege enables the clerk to also modify information for agents.

- Clerk B:
  - View the Accounts section with only one item, Skills.
  - View the full list of Skills, from which he or she selects the Skill to be modified.
  - View and modify any property of the selected Skill.

Precautionary Notes

When configuring and using Role-Based Access Control, take note of the information in this section.

Searching for Objects

The Search facility in Genesys Administrator ignores any restrictions placed by roles, meaning that a user can view any object regardless of what roles they have been assigned. Therefore, in addition to roles, it is imperative that you also use permissions to prevent a user seeing objects for which they have no role privileges.

Hierarchical Access

When assigning a role to users, you must ensure that the lowest level object to which the role is intended to provide access is visible. In other words, if you grant access to an object inside one or more of the functional modules in Genesys Administrator (Monitoring, Provisioning, Deployment, and Operations), you must ensure that you also grant access to the appropriate modules themselves. See the table above to see how this is applied in the example.

For example, if you want to create a role that provides access to Places on the **Provisioning** tab, you must ensure that the users to whom this role will be assigned also have access to the Provisioning module. This can be done by defining and assigning two separate roles (one that grants access to the Provisioning module, and one that grants access to Places), or combined into one Role (one that grants access to both the Provisioning module and access to Places).

Assigning Roles to Individuals vs. Access Groups

Genesys strongly recommends that you avoid assigning a role to a large number of individual users directly. Instead, add the users to an access group and then assign the role to the access group. Assigning a role to a user directly is meaningful only if there are few administrative users for the role, for which it makes no sense to have an access group.

No Default Access for New Users

New users created in release 7.6 or later applications are, by default, not automatically assigned any default privileges—either access permissions or role privileges. In effect, the new users cannot log in to any interface or use a daemon application. Each new user must have the appropriate access privileges and roles assigned by either a system administrator or another existing user with appropriate access rights.

This feature is enabled by default, and applies only to new users created in release 7.6 or later. You can disable the feature if required.

> **Important**
>
> In this chapter, the term *privileges* is intended to mean both access permissions and role privileges.

Security Benefits

New users can be created in multiple ways—directly in a graphical user interface (GUI) or by using the Software Development Kit (SDK). This feature ensures that no user is assigned default privileges, regardless of how the user is created.

Supporting Components

This feature is configured in Genesys Administrator or Configuration Manager. It is not supported by Configuration Server 7.5 or earlier.

Genesys Desktop

Genesys Supervisor Desktop supports a complementary feature. For more information, see the *Genesys Desktop 7.6 Deployment Guide*.

Feature Description

New users created in release 7.6 or later are not automatically assigned any default privileges. In effect, the new users have no privileges and cannot log in to any interface or use a daemon application. Each new user must be explicitly assigned Roles and added to appropriate Access Groups by either a system administrator or by an existing user with access rights to modify the new user's account.

By default, this new feature applies only to new users created in release 7.6 or later. If required, it can be disabled.

Compatibility with Previous Releases

New users created for release 7.5 or earlier Configuration Server Application objects imported into Configuration Server 7.6 or later are also subject to this feature unless the feature is manually disabled in each 7.5 or earlier Configuration Server Application object.

Feature Configuration

> **Important**
> To determine if this section refers to you, see Supporting Components above.

By default, this feature is enabled for all new users created in release 7.6 or later with the **no-default-access** configuration option in the **[security]** section. The Configuration Server application template contains this option set to its default value of zero (`0` - No default access privileges). To disable this feature, set the option to one (1 - Default access privileges).

This feature is also enabled automatically for release 7.5 or earlier Configuration Server Application objects imported by Configuration Server release 7.6 or later. To maintain backward compatibility, you must manually add the **no-default-access** option in the **[security]** section to the options of each imported Configuration Server Application object, and disable the feature by setting the option to 1 (Default access privileges). This will ensure that new users created for those imported applications are assigned default permissions based on the rules present in the original release.

For a detailed description of this option, refer to the *Framework Configuration Options Reference Manual*.

To assign permissions to those new users who are subject to this feature, see Setting and Changing Permissions.

Inactivity Timeout

The inactivity timeout is a configurable period of time during which a user can be inactive (that is, not interact with the system in any way) without any impact on their session. After the timeout expires, the user is locked out of the session, and in some cases, all session displays are minimized. The user must log back in to continue with the session. Alternatively, anyone (not just the owner of the session) can close the session completely, without logging back in.

> **Important**
>
> For purposes of this feature, *activity* is defined at screen level, regardless of the application in focus, and includes: using the mouse (clicking, moving, or scrolling), pressing a key, changing the state of a window between active and inactive, or acknowledging any warning that might be generated by the operating system's own timeout functionality. Watching the progress of an activity, as when a progress indicator appears on the screen, for example, is not interpreted as inactivity. Therefore, the inactivity timeout is not triggered in this case.

Security Benefits

If a user is distracted while logged in to a session, causing them to either turn away or walk away from their computer, that session is available for anyone (authorized or not) to access. The Inactivity Timeout feature minimizes the possibility of that second party viewing or accessing the system. It is a best effort because the length of the timeout is a trade-off between the inconvenience to the logged-in user of having to log in repeatedly, and the risk of exposing the system to other people.

Supporting Components

The following components support this feature:

- Configuration Manager
- Genesys Administrator Extension
- Solution Control Interface
- Genesys Rules System - Genesys Rules Authoring Tool (GRAT)
- Interaction Routing Designer
- Outbound Contact Manager
- Pulse - See Genesys Pulse Configuration Options for more information.
- Workspace Web Edition

- Workspace Desktop Edition (formerly known as Interaction Workspace) also supports this feature, but configures it differently than described in this section. For configuration details of this feature in Workspace Desktop Edition, refer to the *Workspace Desktop Edition Deployment Guide*.
- Genesys Customer Experience Insights. (GCXI) - See KB33832: How does the User Session Idle Timeout work in MicroStrategy for more information.
- SIP Feature Server
  - (For Web based access inactivity timeout (when no user action is there), the session will be disconnected after 10 minutes. This is a configurable value of GAX. For Telephony User Interface inactivity timeout (when no input is given by user), the session will be disconnected after 30 seconds (after 3 attempts of 10 seconds each).

Feature Description

When a user is inactive for the period of time equal to the inactivity timeout, all display screens are minimized (with the exception of some modal dialog screens), and a re-login dialog box is displayed. The connection to the server should be preserved. However, if the connection is lost for some reason, the High Availability (HA) functionality of the application will attempt to reestablish it automatically.

In the re-login dialog box, the user can do one of the following:

- Enter their password, and click **OK**. The user is then authenticated. One of two situations occurs:
  - If this user is not the original user, access will not be permitted.
  - If this user is the original user, that user will be logged back in, and the session state will be restored as much as possible.
- Click **Cancel** to close the application. A confirmation dialog box appears, requesting that the user verify that the application is to be closed.

In any case, the user must be re-authenticated before accessing the current session.

Password Changes

Genesys Administrator, Configuration Manager, and Interaction Routing Designer permit an authorized individual to change a user's password for that Application. If this occurs while the user is logged in, and before the inactivity timeout expires should the user become inactive, the user must use the new password in the re-login dialog box. The old password will be interpreted as an invalid password and access will not be permitted.

In Genesys Administrator or Configuration Manager, a system administrator can also change a user's password for another Application. If this occurs while the user is logged in, and before the inactivity timeout expires should the user become inactive, the user must use the old password in the re-login dialog box. The new password will be interpreted as an invalid password and access will not be permitted.

Feature Configuration

> **Important**
>
> This section describes a standard configuration method for this feature, as used by most components. Some components, such as those identified in Supporting Components, might implement this feature differently. In this case, see the product documentation for details.

The inactivity timeout is configured at the Application level, so can differ between applications. By default, the feature is disabled, and the timeout must be set to a non-zero value to enable the feature.

The inactivity timeout is specified by setting the **inactivity-timeout** option in the **[security]** section of the options of the GUI Application object. Application templates, if they exist, contain this option set to the default value. Refer to the *Framework Configuration Options Reference Manual* for a full description of this option.

Security Banner at Login

The security banner is a separate window that is displayed to a user when logging in to an application. The content of this window is defined by the system administrator, and can include such items as Terms of Use of the application or some kind of disclaimer. One security banner can be used by more than one application, and different applications can use different security banners.

The security banner can be enabled and configured in one of two ways:

- During application setup
- Before or after installation of the application, by creating specific registry entries in the application's host registry

The security banner can be configured differently for each application, to support a variety of corporate policies.

Security Benefits

The security banner does not actually provide true physical or virtual protection for your system. However, it can provide legal protection if an unauthorized user violates any access restrictions, such as Terms of Use, and accesses the system anyway.

Under the strictest configuration of the security banner, a user is not allowed to log in to an application without first accepting the contents of the banner. The various degrees of security depend on the options selected during installation.

Supporting Components

The following components support the implementation of the security banner as described in this chapter:

- Configuration Wizards
- Genesys Administrator
- Configuration Manager
- Solution Control Interface
- Interaction Routing Designer
- Outbound Contact Manager
- CCPulse+

Similar functionality can be achieved using customization features in the following components:

- Workspace Desktop Edition (formerly known as Interaction Workspace)
- Performance Management Advisors

For more information, refer to component-specific documentation.

Genesys Composer

Genesys Composer supports the basic concept of specifying and displaying a security banner. However, it implements a security banner differently than described in this document. Refer to Genesys Composer documentation for more information.

Genesys Desktop

Genesys Desktop supports the security banner in concept, but implements it differently from the way described in this document. In addition to a different installation procedure, all URLs related to the security banner must be in HTTP format (**http://**). Refer to the *Genesys Desktop 7.6 Deployment Guide* for more information.

Feature Description

The security banner is intended to display a user-defined security message prior to the login to a Genesys application, and provide the user with the means to confirm acceptance of the message. The message content is specified as an arbitrary URL, pointing to a document that can be displayed as an active document by Microsoft Internet Explorer 4.0 or later. Multiple URLs can be configured for redundancy.

The following characteristics of the security banner are configurable by the user, and can be configured differently for each application:

- Regularity with which the security banner is displayed. For example, it can be displayed only once for each user, only once for each user for each type of application, or for all logins.
- Whether the security banner is to be displayed, or if user acknowledgement is required.
- Behavior if the target URL of the security banner is not available.
- Title and dimensions of the security banner window.
- The timeout within which the security banner must be loaded and displayed on the screen. If this timeout expires, an intermediate message (**Downloading terms of use... Please wait...**) is displayed while the security banner loads.

By default, the security banner window contains user-defined text, two buttons (**Accept** and **Reject**) and a check box (**I Accept. Do not show this again**). The user logging in to the application must click **Accept** to proceed to the login dialog box. If the user clicks **Reject** or closes the security banner window without accepting the window contents, the application closes.

As previously described, an intermediate message (**Downloading terms of use... Please wait...**) is displayed whenever the security banner is not retrieved and displayed before the timeout expires. During this time, the user can close the window by clicking **Cancel**; the terms can only be accepted when the content is fully displayed.

You must also specify whether you allow a user to log in to the application if the security banner cannot be displayed; if you do not allow it, the application closes if the security banner cannot be displayed.

If the security banner cannot be retrieved at all, an error message is displayed. Error messages contain an **Exit** button instead of **Accept** and **Reject** buttons. The software includes a default error page, but you can also configure your own. The behavior of the error page depends on whether you have chosen to allow a user to log in to the application if the security banner is not displayed, as follows:

- If you have chosen to allow the user to log in, the error page closes automatically (if it is open) and the login dialog box appears. The user can then log in to the application.
- If you have chosen not to allow the user to log in, the error page included with the software is displayed, showing the error code. The login dialog box is not displayed, and the user cannot log in. For HTTP errors, refer to the HTTP specification. For system errors, refer to Microsoft technical documentation.

> ### Warning
> Genesys recommends that you use multiple redundant URLs, including a local file as appropriate, to minimize the risk that the security banner will not load.

Deploying the Security Banner for Multiple Applications on the Same Host

If, on a single host, you are installing two or more applications that support and will be using a security banner, you can choose to do one of the following:

- Provide individual settings for each type of application.
  In this case, if you choose to configure the security banner for just one (for this) application, all other applications will be deemed to have the security banner disabled. If you want any other applications to use a banner, you must enable and configure it for each of those other applications. In subsequent application installations, you can chose the for all option, but this will only set default values for subsequent installations; it will not impact the values for previous installations.
- Configure one security banner for all applications.
  In this case, the security banners for all applications on this host will have the same content and behavior. In effect, these settings become the default settings. You do not have to enable and configure the security banner for each application. Having done this, for each application with security banner that you subsequently install, you can choose to do one of the following:
  - Provide individual settings for this application only, while not impacting the default settings.
  -
  - Override the default settings by choosing to configure the security banner for all applications, and modifying the settings as required. The default values will appear in the installation interface, and can be overwritten or kept as is. If you change any of these values, all applications that use the default values, both those installed previously and subsequently, will be impacted.

In general, when setting up an application, the setup program looks first for a security banner configuration specific to this application. If one is not found, it then looks for a configuration common to all applications. In either case, it inherits the security banner attributes already defined. If it is unable to find any security banner configuration, it defaults to a disabled security banner, and you must then enable and configure the security banner from the beginning.

Feature Deployment

> **Important**
> To determine if this section applies to your component, see Supporting Components.

Deployment of the security banner consists of three steps:

1. Design and create the required security banners and optional customized error pages, using the editor of your choice.
2. Deploy security banner documents as files or as web content, and record the URLs. Each URL must be able to be resolved by the installed Microsoft Internet Explorer (IE) and displayed as an active page within the IE window.
3. Configure the URLs in one of the following ways:

As directed during installation of the GUI application **[+] Show steps**

The installation and configuration of the security banner is part of the application installation procedure for the following applications:

- Configuration Wizards
- Genesys Administrator
- Configuration Manager
- Solution Control Interface
- Interaction Routing Designer
- Outbound Contact Manager

The security banner can also be installed after the application has been installed.

Refer to documentation for your application for detailed instructions about installing the application. Use the following procedure only if you select the **Enable Security Banner** option when installing the application.

Prerequisites

- You are installing one of the supporting components listed above, and have reached the **Security Banner Configuration** page of the installation wizard.
- You have created, and have the URLs of, the security banner and any custom error pages that will be used.

Start of procedure

1. On the **Security Banner Configuration** page of the installation wizard for the application that you are installing:
   a. In the **Select Security Banner behavior and configuration** section, select whether you want the security banner that you are about to define to be used by all applications that support the security banner feature, or just by applications of this type.
   b. Click **Next**.

2. On the **Security Banner Parameters** page, specify the parameters for the security banner as follows:

   a. Select how the security banner is displayed to the user the next time an application of this type is started:

   - **Until each user chooses to turn it off**—The security banner includes an **I Accept. Do not show this again.** check box that, by default, is not selected. If the user selects this option and clicks **Accept**, the security banner will not be displayed again to that user, regardless of the application that the user is starting. Each Windows user account must explicitly select this option and click **Accept** to disable the security banner for all applications.
   - **Until each user chooses to turn it off once for each application type**—The security banner includes an **I Accept. Do not show this again.** check box that, by default, is not selected. If the user selects this option and clicks **Accept**, the security banner will not be displayed again to that user when starting any application of this type. However, each time the user starts another type of application for which the security banner is active, the security banner will be displayed. Each Windows user account must explicitly select this option and click **Accept** to disable the security banner for this type of application.
   - **Every time the application starts**—The security banner does not include an **I Accept. Do not show this again.** check box. The security banner is displayed to every user every time any Genesys application is started.

> **Important**
>
> If you select **Until each user chooses to turn it off** or **Until each user chooses to turn it off once for each application type**, and the user logging in selects **I Accept. Do not show this again.** in the security banner window, this setting will apply for all subsequent installations of the one or multiple applications. It (the **AckMandatory** registry variable) must manually be removed or reset to zero (**0**) in the registry by an authorized person.

   b. Select how to proceed if the security banner message at the specified URL cannot be displayed:

   - **Proceed to login without banner**—The user can log in to the application anyway.
   - **Exit, no login dialog box is displayed**—The user is not permitted to log in.

> **Warning**
>
> Selecting the **Exit, no login dialog box is displayed** option effectively disables access to the application when the document specified by the URL cannot be retrieved or rendered for any reason.

   c. (Optional) Specify the title that appears in the title bar of the security banner window. If you do not specify a title, the window title is derived from the following:

   - If the security banner is an HTML file, the **<title>** element.
   - If the security banner is an HTML file but has no **<title>** element, the URL address.

- If the security banner is not an HTML file, the URL address.

In all cases, the application name follows the title in the title bar.

> **Important**
> If rebranding resources are present, the corresponding rebranding resource overrides this entry.

d. Specify the timeout, in milliseconds, within which the security banner must be displayed. The default is 3000. If the entire document is not available for display within this time, an intermediate message, **Downloading terms of use ... Please wait ...,** is displayed until the security banner itself can be displayed.

e. Specify the height and width, in pixels, of the security banner window, intermediate window, and any error window, if defined. The default values are 180 and 360 pixels, respectively. If neither of these values is specified (the default), the window is sized to fit the complete content of the document at the specified URL. At no time does the window exceed the work area of the screen. The document retains its size between logins, and once displayed, can be resized using standard IE tools.

> **Important**
> If the exact screen size for the security banner documents cannot be determined or estimated, Genesys recommends that the height and width parameters be specified.

f. Click **Next**.

3. On the **Security Banner Documents** page, for each document containing text that will be displayed in the security banner, specify the URL of the document and click **Add**. When you have added all the URLs, click **Next**.
   If this URL is not specified, all of the other options are ignored, and:
   - If an older security banner bitmap is configured, it is displayed.
   - Otherwise, no security banner is displayed.
4. On the **Security Banner Error Documents** page, do one of the following:
   - If you selected **Proceed to login without banner**, click **Next**. Do not enter any URLs on this page.
   - Otherwise, specify the URL of an error document—either the default error page or one that you specifically created—and click **Add**. When you have added all the URLs, click **Next**.

End of procedure

Next Steps

- Finish installing your application, as required. Refer to product-specific documentation for detailed instructions.

By modifying registry entries directly. **[+] Show steps**

> **Warning**
>
> Editing a registry incorrectly can cause serious, system-wide problems, and correcting them might require you to reinstall your operating system. Genesys cannot guarantee that any problems resulting from editing the registry can be solved. Edit your registry at your own risk. If you do decide to edit the registry, Genesys strongly recommends that you back up the registry file before editing it.

The Security Banner feature and URLs are defined in the registry of the application's host. Only someone with Write access (the Change permission) to the **HKEY_LOCAL_MACHINE** registry key—normally the system administrator—can set up and maintain the security banner.

This authorized person should:

- Specify the target URLs of the security banners and any customized error pages.
- Customize the windows as required.
- Subsequently modify the behavior as required, by changing the listed registry entries. This can be done either locally or remotely.

Configuring Security Banner Functionality

Configure the security banner functionality by using the following registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\GCTI\Unilogin\Banner

The values in this key specify the default behavior for all applications. Each entry can be redefined for specific applications in the subkeys, as follows:

HKEY_LOCAL_MACHINE\SOFTWARE\GCTI\Unilogin\Banner\<CfgAppType>

where **<CfgAppType>** is the numeric value of the application type, as defined in the following table.

CfgAppType	Application
13	Outbound Contact Manager
19	Configuration Manager Wizard Manager
44	Solution Control Interface
51	Interaction Routing Designer
165	Genesys Administrator

For example, to specify values specific to Genesys Administrator, which has application type 165, define the registry subkey as follows:

HKEY_LOCAL_MACHINE\SOFTWARE\GCTI\Unilogin\Banner\165

When selecting the security banner to display and use, the library first looks for a corresponding subkey, and then uses the default key if the subkey does not exist.

String entries can be entered as STRING or EXPANDABLE_STRING registry values. If they are entered as EXPANDABLE_STRING, environment variable strings enclosed in percent signs (%) are replaced with their values defined by the environment variables (located in **%HOMEDRIVE%%HOMEPATH%\default.htm**). Integer entries can be entered either as DWORD or STRING registry values, representing decimal numbers.

Configuring URLs

The URLs for the security banner and any associated error pages are configured in the following registry keys:

- For all applications:
  HKEY_LOCAL_MACHINE\SOFTWARE\GCTI\Unilogin\Banner\URLs\<seq_number>
- For specific applications:
  HKEY_LOCAL_MACHINE\SOFTWARE\GCTI\Unilogin\Banner\<CgfAppType>\<seq_number>

  where **<seq_number>** is the sequence number for multiple URLs. Multiple URLs are tried in the order of their sequence number.

  The URLs are specified by the registry options Error Page and URL.

Example

The following sample registry entries:

```
KEY_LOCAL_MACHINE\SOFTWARE\GCTI\Unilogin\Banner\URLs\1\URL=http://MyServer1/
Banner.htm
HKEY_LOCAL_MACHINE\SOFTWARE\GCTI\Unilogin\Banner\URLs\2\URL=http://MyServer2/
Banner.htm
HKEY_LOCAL_MACHINE\SOFTWARE\GCTI\Unilogin\Banner\URLs\3\
URL=%SystemRoot%AdminContacts.htm
HKEY_LOCAL_MACHINE\SOFTWARE\GCTI\Unilogin\Banner\URLs\3\ErrorPage=1
```

specify the following behavior:

The dialog attempts to retrieve **Banner.htm** from **MyServer1**. If it cannot retrieve that file, the dialog attempts to retrieve **Banner.htm** from . If it cannot retrieve that file, the dialog attempts to retrieve the custom error page located in the **system32** directory. And if that page cannot be retrieved, the default error page is displayed.

Security Banner Registry Entries

This section describes the registry options used to specify and customize the appearance and behavior of the security banner. These options are intended only for advanced users with registry access.

> **Important**
>
> Unless otherwise noted, the registry entries in this section are equivalent to the options presented when installing the security banner during application setup.

AckMandatory
Default Value: **0**
Valid Values: One of the following:

0	Proceed with the login, without acknowledgement of the contents of the security banner. The login dialog box is displayed.
1	Exit the application. The login dialog box is

	not displayed.

Changes Take Effect: After the application restarts

Specifies whether login to the application will be allowed if the document specified in the URL cannot be displayed for any reason.

> ## Warning
>
> Setting this option to **1** effectively disables access to the application when the document specified by the URL cannot be retrieved or rendered for any reason.

AckMode
Default Value: **0**
Valid Values: One of the following:

0	User can choose to hide the security banner for all subsequent logins, for all applications.
1	User can choose to hide the security banner for all subsequent logins to the current application only.
2	User cannot choose to hide the security banner; user must accept content of the banner whenever logging in to any application.

Changes Take Effect: After the application restarts

Specifies whether the user is presented with the option to hide the security banner, and therefore does not need to accept the security banner content, the next time an application is launched.

If this option is set to **0** or **1**, the **I Accept. Do not show this again.** check box appears in the security banner window. If the user selects this check box, they will not see the security banner at subsequent attempts to access either this (**0** or **1**) or any application (**1**) for which the security banner is configured.

> ## Important
>
> If option **0** or **1** is selected, the only way to have the security banner displayed again when logging in to this (**0** or **1**) or any application (**1**) is to manually remove this entry from the registry. This registry entry and its value is persistent across installations—it is not removed when uninstalling the application, nor is it cleared or reset when reinstalling the application.

If this option is set to **2**, the **I Acknowledge. Don't show this again.** check box does not appear in the security banner window, and the security banner is displayed every time anyone tries to access the application.

ErrorPage
Default Value: **0**
Valid Values:

0	The security banner is displayed.
1	An error page is displayed.

Changes Take Effect: After the application restarts

Required if you are using a custom error page. Specifies that the URL points to an error page or the security banner. If the error page is displayed, the window displays the **Exit** button in place of the **Accept** and **Reject** buttons. Use this setting to substitute the default error page with a customized error page.

Height
Default Value: No default value
Valid Values: Any positive integer greater than **180**
Changes Take Effect: After the application restarts

Width
Default Value: No default value
Valid Values: Any positive integer greater than **359**
Changes Take Effect: After the application restarts

Optional; these two options specify the dimensions (in pixels) of the document area of the security banner and error page window. If neither of these values is specified (the default), the window is sized to fit the complete content of the document specified by the URL. At no time does the window exceed the work area of the screen. The document retains its size between logins, and once displayed, can be resized using standard IE tools.

> ## Important
>
> If the exact screen size for the security banner documents cannot be determined or estimated, Genesys recommends that the height and width parameters be specified.

NoCompleteTimeout
Default Value: **2000**
Valid Values: Any non-negative integer
Changes Take Effect: After the application restarts

Specifies the timeout (in milliseconds) for receiving download progress or status notifications from the WebBrowser control. To download and render the document, the security banner dialog uses components of IE in the form of WebBrowser control. In some cases, for security reasons, the WebBrowser control does not provide the client with the means to detect navigation cancellation. This timeout is used to detect and properly process these cases.

The absence of progress or status notifications from the WebBrowser control for a period exceeding this timeout is considered a failure to retrieve the document. If this timeout expires, the attempt to retrieve the document specified by the current URL is aborted, and the dialog attempts to retrieve the next URL from the URLs list. If this happens with the last URL in the list, the System error **0x80004004: Operation aborted** error message is reported to the user.

If this option is set to zero (**0**), progress and status notifications are not used to detect download failure or cancellation.

> ## Important
>
> **NoCompleteTimeout** is intended only for advanced users with access to the registry. It has no equivalent option in the process of installing the security banner during application setup, and its default value is considered adequate in these situations.

ShowUpTimeout
Default Value: **3000**
Valid Values: Any non-negative integer
Changes Take Effect: After the application restarts

Specifies the timeout (in milliseconds) within which the security banner window attempts to load the document specified

by the URL. If the timeout expires before the content is displayed, an intermediate window (**Downloading terms of use... Please wait...**) is displayed. During this time, the user can close the window by clicking **Cancel**; the terms can only be accepted when the content is fully displayed.

If the document cannot be retrieved, the behavior of the window depends on the value of AckMandatory, as follows:

- If **AckMandatory=0**, the window closes automatically (if it is open), and the login dialog box appears. The user can then log in to the application.
- If **AckMandatory=1**, the error page included with the software is displayed, showing the error code. For HTTP errors, refer to the HTTP specification. For system errors, refer to Microsoft technical documentation. The login dialog box is not displayed, so the user cannot log in.

Title
Default Value: No default value
Valid Values: Any string, or blank
Changes Take Effect: After the application restarts

Optional; specifies the title that appears in the title bar of the security banner window. If no value is specified for this option, the title is derived from the following:

- If the security banner is an HTML file, the **<title>** element.
- If the security banner is an HTML file but has no **<title>** element, the URL address.
- If the security banner is not an HTML file, the URL address.

In all cases, the application name follows the title in the title bar.

> **Important**
>
> Note: If rebranding resources are present, the corresponding rebranding resource overrides this entry.

URL
Default Value: No default value, for backward compatibility
Valid Values: A URL address that can be resolved by the installed IE application and displayed as an active page within the IE window
Changes Take Effect: After the application restarts

Required; specifies the URL of the document displayed in the security banner window. If this value is not specified, all other options are ignored, and:

- If an old security banner bitmap is configured, it is displayed.
- Otherwise, no security banner is displayed.

> **Important**
>
> If you uninstall an application for which the security banner was configured, the configuration parameters of its security banner are not removed from the registry. To clear these parameters, you must reinstall the application without enabling the security banner.

Last Logged In

The Last Logged In feature enables a user logging into Configuration Server or Configuration Server Proxy via a Genesys graphical user interface (GUI) to see the date and time at which the user's account and credentials last logged into that Configuration Server or Configuration Server Proxy. The Last Logged In information relates to the Configuration Server or Configuration Server Proxy, not the GUI that was used to log in.

Security Benefits

This feature by itself does not proactively prevent unauthorized users from gaining access to the system. However, it does provide a method for each authorized user to monitor their own account, and take necessary action if someone other than that user has used the account to access the system.

Supporting Components

This feature is implemented by Configuration Server and Configuration Server Proxy. The display of the information is supported by the following Genesys GUI applications:

- Configuration Manager
- Solution Control Interface
- Outbound Contact Manager
- Workspace Desktop Edition (formerly known as Interaction Workspace)
- intelligent Workload Distribution
- Platform SDK
- Genesys Administrator Extension

Feature Description

When a user logs in to Configuration Server or Configuration Server Proxy through a Genesys GUI, the date and time when this account was last used to log in to the same Configuration Server or Configuration Server Proxy (regardless of the GUI used) is displayed in the bottom right-hand corner of the display. In case of Workspace Desktop Edition, the last logged-in information is displayed in the ToolTip for the Global Status icon. Place your mouse pointer over the Global Status icon to open its ToolTip.

If the user notes a difference between when they last logged in, and the date and time shown by the system, it is the responsibility of the user to take appropriate action, such as notifying the

appropriate authorities.

Feature Configuration

This feature is implemented by defining the following two configuration options in the **[confserv]** and **[csproxy]** sections of the Configuration Server and Configuration Server Proxy Application objects, respectively:

- The **last-login** option defines whether this feature is to be used.
- The **last-login-synchronization** option defines whether Last Login information is to be synchronized between all Configuration Servers and Configuration Server Proxies.

For more information about these options, refer to the Configuration Server Section topic in the *Framework Configuration Options Reference*.

Protection of Data at Rest

Disclosure of confidential customer information can result in serious legal consequences for a contact center, as well as the loss of a customer. Privacy includes protecting not only the customer's proprietary data, but also transaction and call statistics and sometimes, their identification as a customer of a particular contact center.

Genesys provides the following security features to protect data at rest:

- Encrypted Configuration Database Password
- Encrypted Data in Databases
- Encrypted Call Recordings
- Hide Selected Data in Logs

Encrypted Configuration Database Password

You can encrypt the password used to access the Configuration Database so that it appears in the Configuration Server logs as an encrypted string of characters.

> **Important**
>
> This encryption does not use the SALT used when encrypting user passwords. See Password Encryption.

Security Benefits

Once encrypted, the password to the Configuration Database is written as an encrypted string of characters into Configuration Server logs. This feature ensures that anyone reading the log cannot obtain the password and use it to access the Configuration Database directly through the DBMS.

Supporting Components

This feature is configured on the Configuration Server accessing the Configuration Database.

Feature Description

All entries in configuration files and logs are readable in plain text, unless explicitly configured to be hidden in some way. You can encrypt your password for accessing the Configuration Database. After password encryption, Configuration Server decrypts the value when reading its configuration file at subsequent startups. It accesses the Configuration Database using the decrypted value, and prints an encrypted string of characters as the password value into the log. In this way, the password does not explicitly appear in the Configuration Server logs.

Feature Configuration

To encrypt the Configuration Database password, do the following:

1. Force Configuration Server to encrypt the password. **[+] Show steps**

Important

Starting in release 8.5.1, the Configuration Server configuration file optionally supports an asymmetric encryption algorithm using separate encryption and decryption (private) keys that are not hardcoded. In this case, the keys are generated by Configuration Server and stored in separate files. The password is encoded using the key in the encryption file. Upon subsequent restarts of Configuration Server, it uses the key in the decryption file to decrypt and the password. See Encrypting the Configuration Database Password

Prerequisites

- Configuration Server is not running.
- Configuration DB Server is not running.

Start of Procedure

Force Configuration Server to encrypt the password, by starting Configuration Server with the following command line:

```
confserv -p <section name> <password value>
```

where:

-p	The command-line parameter that forces an instance of Configuration Server to start, encrypt the database password in the configuration file, and terminate.
<section name>	The section name in the Configuration Server configuration file that describes the Configuration Database whose access password is being encrypted.
<password value>	The password used for accessing the specified Configuration Database.

Important

- If the configuration file name differs from the default name (**confserv.conf** on UNIX or **confserv.cfg** on Windows), the command line should also contain the **-c** parameter followed by the file name. For a description of command-line parameters specific to Configuration Server, refer to the *Framework Deployment Guide*.
- If a password to be encrypted contains one or more UNIX shell special characters, the password must be enclosed in single quotes in the command line. For example, if the password is `$Montana`, enter the following at the command line:

```
confserv -p gauth_ldap '$Montana'
```

- When using Windows, if a password to be encrypted contains one or more special characters, the password must be enclosed in double quotes in the command line. For example, if the password is `p&ssword`, enter the following at the command line:

```
confserv -p dbserver "p&ssword"
```

Repeat this step for each Configuration Database section listed in the configuration file of Configuration Server.

2. Configure the **encryption** option in the Configuration Server configuration file. **[+] Show steps**

Prerequisites

Any primary and backup Configuration Servers associated with this Configuration Server have encrypted the password.

Start of Procedure

1. In a text editor, open the Configuration Server's configuration file.
2. In the **[confserv]** section, set **encryption** to `true`. This value applies to all Configuration Database sections specified in the configuration file. Refer to the *Framework Configuration Options Reference Manual* for a full description of this option.
3. Save and close the file.

Now, Configuration Server is ready to operate with the encrypted password.

3. Restart Configuration Server as for a regular operation. Refer to the *Framework Deployment Guide* for detailed information about starting Configuration Server.

Encrypted Data in Databases

This feature uses the data transparency and encryption functionalities provided by a Database Management System (DBMS) to encrypt data contained in a database.

> **Important**
>
> If database encryption is not in place, database passwords used by Database Access Points, and the values of any configuration options named **password**, such as those used with SNMPv3, will be automatically encrypted using AES 128. This is a failsafe measure only; Genesys strongly recommends that you encrypt all data in the database.

Security Benefits

By default, data in a database is stored as plain text, and is easily read by anyone (or anything) accessing it. Encrypting this data makes that data nearly impossible to read without the corresponding decryption mechanism. In effect, this feature provides a second level of protection should an unauthorized user get access to the database itself.

Supporting Components

Databases in the following Genesys products support this feature:

- Management Framework
- Outbound Contact
- eServices
- Universal Contact Server
- Genesys Voice Platform
- Genesys Interactive Insights
- Performance Management Advisors
- Genesys Info Mart
- Interaction Concentrator
- Workforce Management
- Pulse

Feature Description

Data in a database is stored as plain text by default, and therefore is easily read by anyone (or anything) accessing it. This feature uses the data transparency and encryption functionalities provided by a DBMS to encrypt that data, so that it cannot be read or understood without the corresponding decryption capabilities.

Feature Configuration

This feature is supported by Genesys only if the DBMS also supports encrypted data. Currently, only the data encryption mechanisms of the following DBMS are supported:

- MS SQL 2008 R2 and later
- Oracle 11g R1 and later
- Oracle 10.2 and later

MS SQL 2008 R2 and Later

Genesys provides transparent access to databases based on MS SQL 2008 R2 and later with the Transparent Data Encryption (TDE) feature enabled. The TDE feature is fully described, with implementation instructions, on the Microsoft Developer Network website at http://msdn.microsoft.com/en-us/library/bb934049.aspx.

Deployment of TDE must follow MS SQL documentation, and basically consists of the following steps:

1. Create a master key.
2. Create or obtain a certificate protected by the master key.
3. Create a database encryption key (in the existing Genesys database) and protect it with this certificate.
4. Configure the database to use encryption.

Before implementing this feature, first create (or convert) your database with a schema compatible with the release of your Genesys software that supports this feature. Then deploy encryption.

Oracle 11g R1 and Later

Genesys provides transparent access to Oracle 11g R1 and later encrypted tablespaces.

Deployment of TDE must follow Oracle 11g documentation. It includes the following steps:

1. Set up Oracle 11g encryption:
    a. Create a system encryption key.
    b. Load the master key at database restart.
    c. Initialize the autologin wallet to keep the master key accessible across restarts of this instance of the database.

2. Set up the Genesys database:
    a. Create the tablespace with encryption.
    b. Make it the default tablespace with [unlimited] quote for a user account used by Genesys applications.
    c. Create database schemas in the encrypted tablespace.

For more details and examples, see the Oracle-Base article at http://www.oracle-base.com/articles/11g/TablespaceEncryption_11gR1.php.

Deploy encryption on the Genesys tablespace before creating the database schema compatible with the release of your Genesys software that supports this feature. If the database tables already exist and reside in unencrypted tablespace, move the tables to an encrypted tablespace using tools provided by Oracle.

Oracle 10.2 and Later

Genesys provides transparent access to databases based on Oracle 10.2 and later with the Transparent Data Encryption (TDE) feature enabled on database columns that support it.

> **Important**
>
> - A list of column types that are supported by Oracle 10.2 and later is included in the Oracle documentation.
> - Columns that contain BLOB and CLOB data cannot be encrypted.

For details about the schema of the databases for which you want to encrypt the data, contact your Genesys representative. For example, the help file Framework Configuration Database Schema Reference contains the database schema for the Configuration Database.

Deployment of TDE must follow Oracle 10 documentation. It includes the following steps:

1. Set up the TDE feature:
    a. Create a system encryption key.
    b. Load the master key at database restart.
    c. Initialize the autologin wallet to keep the master key accessible across restarts of this instance of the database. For details, see the Oracle-Base articles starting at http://www.oracle-base.com/articles/10g/transparent-data-encryption-10gr2.php.
2. In the Genesys database, alter the database tables by setting up columns with transparent encryption to encrypt the data in those columns, as follows:
    a. Stop the server that uses the database that you want to alter. For example, if you are going to encrypt data in the Configuration Database, stop Configuration Server.
    b. Run the script to alter the table. For example, to add encryption to the password column of a Person object, alter the Configuration Server 8.1.1 database table definition as follows:

```
ALTER TABLE cfg person MODIFY (password ENCRYPT);
```

c. Restart the server.

Encryption of Call Recordings

This feature uses the Genesys Interaction Recording Solution to record calls and play them back. The solution includes a key management system that creates public and private keys. These keys are used to encrypt the recorded calls and decrypt the calls for playback. Encryption is performed with a session key using AES. The session key is encrypted using a public key and stored in a PKCS7 envelope. During decryption, a private key is used to decrypt the PKCS7 envelope and extract and return the AES key which will be used for decrypting the recording.

For more information about the Genesys Interaction Recording Solution, refer to Genesys Interaction Recording.

Hide Selected Data in Logs

This feature enables you to hide all or part of selected key-value (KV) pairs in the User Data, Extensions, and Reasons attributes of log messages generated by a Genesys component. The data can be masked completely or partially, or identified by specified characters (called *tags*).

Security Benefits

This feature prevents unauthorized users from seeing particular data in the output of log messages. Where logs are distributed to another party, such as for troubleshooting purposes, this feature enables you to hide confidential data that you do not want the other party to see. This feature is also useful for preserving the confidentiality of data provided to you by third parties, which might be attached to the logs.

Supporting Components

This feature is supported by the following Genesys components:

- Management Framework
- Media and Network T-Servers
- Load Distribution Server
- Outbound Contact Server
- Interaction Concentrator (ICON)
- Federated
- Universal Contact Server
- Universal Routing Server
- Orchestration Server
- eServices (partial)
- Enterprise SDK
- Interaction SDK
- Real Time Metrics Engine
- Genesys Mobile Services

This following Genesys components support this feature in part or in a similar manner:

- SIP Server, except for data that appears in a SIP header
- Stat Server, in a non-standard way. For more information, refer to Stat Server-specific documentation.

- Platform SDK (PSDK), in a non-standard way. For more information, refer to PSDK-specific documentation.
- Workspace Desktop Edition (formerly known as Interaction Workspace). For configuration details of this feature in Workspace Desktop Edition, refer to the *Workspace Desktop Edition Deployment Guide*.
- IVR Connector handles logging of attached data on T-Library events and messages using standard T-Server configuration as described in T-Server-specific documentation. However, to avoid logging attached data in XML messages within the XML interface, use the **hide-xml-udata** configuration option as described in IVR Connector-specific documentation.
- Genesys Voice Platform (GVP) supports hiding Voice XML (VXML) variables by using PRIVATE variables. For more information, refer to GVP documentation.

Feature Description

This feature enables you to hide selected KV pairs in the User Data, Extensions, and Reasons attributes of log messages generated by a Genesys component. You can choose to hide just the value itself by replacing it with a series of asterisks (*), or you can remove the whole KV pair from the log output.

Starting in release 8.0, you can also hide only part of the value in a particular KV pair. This provides the intended security, but with enough data to use for tracking field values, if necessary.

Starting in release 8.1, you can mark the selected KV pairs with specific characters (called *tags*), which enable the log message to be parsed by downstream applications and the marked data hidden. Default tags are provided (<# for a prefix and #> for a postfix), and you can define your own custom tags of up to 16 characters, if required.

Feature Configuration

This section describes how to configure this feature, along with some examples of hiding data in the different ways made possible by the feature. For detailed descriptions of the configuration options used to configure this feature, refer to the *Framework Configuration Options Reference Manual*.

This feature can be used to hide information in the User Data, Extensions, and Reasons attributes of the log. The implementation is the same for all three attributes.

This feature is implemented by defining the following configuration options in the server Application object:

- **default-filter-type** in the **[log-filter]** section defines the treatment for all KV pairs in the User Data, Extensions, and Reasons attributes. This setting will be applied to the attributes of all KVList pairs in the attribute except those that are explicitly defined in the **[log-filter-data]** section.
- One or more **<key-name>** options in the **[log-filter-data]** section define the treatment for specific keys in the log, overriding the default treatment specified by **default-filter-type**. If no value is specified for this option, no additional processing of this data element is performed.

The default settings of the options enable all data to be visible in the log.

> **Important**
>
> For T-Server Application objects, if the T-Server common option **log-trace-flags** is set to **-udata**, it will disable writing of user data to the log regardless of the settings of any options in the **[log-filter-data section]**. Refer to the documentation for your particular T-Server for information about the **log-trace-flags** option.

Examples

This section provides examples of using the options to define settings (**default-filter-type**) for the entire log, and settings (**<kv-pair>**) specific to a KV pair. For simplicity, the examples show only the use of the feature to hide information in the User Data attribute.

Default Settings

This example uses the default settings. Note that all data is visible in the log.

```
[log-filter]
default-filter-type=copy

message RequestSetCallInfo
   AttributeConsultType       3
   AttributeOriginalConnID    008b012ece62c8be
   AttributeUpdateRevision    2752651
   AttributeUserData          [111] 00 27 01 00
          ‘DNIS'              '8410'
          ‘PASSWORD'          '111111111'
          'RECORD_ID'         '8313427'
   AttributeConnID            008b012ece62c922
```

Masking Partial Values

This example replaces the first three characters of every key value with three asterisks (***).

```
[log-filter]
default-filter-type=hide-first,3

message RequestSetCallInfo
   AttributeConsultType       3
   AttributeOriginalConnID    008b012ece62c8be
   AttributeUpdateRevision    2752651
   AttributeUserData          [111] 00 27 01 00
          ‘DNIS'              '***0'
          ‘PASSWORD'          '***111111'
          'RECORD_ID'         '***3427'
   AttributeConnID            008b012ece62c922
```

Using Default Tags

This example uses the default tags **<#** and **#>**. Note that all KV pairs in the User Data attribute are

identically tagged.

```
[log-filter]
default-filter-type=tag()

message RequestSetCallInfo
   AttributeConsultType       3
   AttributeOriginalConnID    008b012ece62c8be
   AttributeUpdateRevision    2752651
   AttributeUserData          [111] 00 27 01 00
          ‘DNIS'              <#'8410'#>
          ‘PASSWORD'          <#'111111111'#>
          'RECORD_ID'         <#'8313427'#>
   AttributeConnID            008b012ece62c922
```

Using User-defined Tags for All Attributes

This example uses the user-defined tags **<**** and ****>**. Note that all KV pairs in the User Data attribute are identically tagged.

```
[log-filter]
default-filter-type=tag(<**,**>)
message RequestSetCallInfo
   AttributeConsultType       3
   AttributeOriginalConnID    008b012ece62c8be
   AttributeUpdateRevision    2752651
   AttributeUserData          [111] 00 27 01 00
          ‘DNIS'              <**'8410'**>
          ‘PASSWORD'          <**'111111111'**>
          'RECORD_ID'         <**'8313427'**>
   AttributeConnID            008b012ece62c922
```

Masking Individual Values in Selected KV Pairs

This example replaces the value of the PASSWORD key with a series of asterisks (****).

```
[log-filter-data]
PASSWORD=hide

message RequestSetCallInfo
   AttributeConsultType       3
   AttributeOriginalConnID    008b012ece62c8be
   AttributeUpdateRevision    2752651
   AttributeUserData          [111] 00 27 01 00
          ‘DNIS'              '8410'
          ‘PASSWORD'          '****'
          'RECORD_ID'         '8313427'
   AttributeConnID            008b012ece62c922
```

Masking Partial Values in Selected KV Pairs

This example replaces all but the last five characters of the PASSWORD key with a series of asterisks (****).

```
[log-filter-data]
PASSWORD=unhide-last,5

message RequestSetCallInfo
```

```
   AttributeConsultType       3
   AttributeOriginalConnID    008b012ece62c8be
   AttributeUpdateRevision    2752651
   AttributeUserData          [111] 00 27 01 00
          ‘DNIS'              '8410'
          ‘PASSWORD'          '****11111'
          'RECORD_ID'         '8313427'
   AttributeConnID            008b012ece62c922
```

Tagging Specific KV Pairs with Default Tags

This example tags the value of the PASSWORD key with the default tags **<#** and **#>**. Note that the values of the other keys are not tagged.

```
[log-filter-data]
PASSWORD=tag()

message RequestSetCallInfo
   AttributeConsultType       3
   AttributeOriginalConnID    008b012ece62c8be
   AttributeUpdateRevision    2752651
   AttributeUserData          [111] 00 27 01 00
          ‘DNIS'              '8410'
          ‘PASSWORD'          <#'1234'#>
          'RECORD_ID'         '8313427'
   AttributeConnID            008b012ece62c922
```

Tagging Specific KV Pairs with User-defined Tags

This example tags the value of the PASSWORD key with the user-defined tags **<!--** and **-->**. Note that the values of the other keys are not tagged.

```
[log-filter-data]
PASSWORD=tag(<!--,-->)

message RequestSetCallInfo
   AttributeConsultType       3
   AttributeOriginalConnID    008b012ece62c8be
   AttributeUpdateRevision    2752651
   AttributeUserData          [111] 00 27 01 00
          ‘DNIS'              '8410'
          ‘PASSWORD'          <!--'1234'-->
          'RECORD_ID'         '8313427'
   AttributeConnID            008b012ece62c922
```

Tagging Individual KV Pairs with Different Tags

This example tags the value of the PASSWORD key with user-defined tags **<!--** and **-->**, and the value of the RECORD_ID key with default tags **<#** and **#>**. Note that the values of the other keys are not tagged.

```
[log-filter-data]
PASSWORD=tag(<!--,-->)
RECORD_ID= tag()

message RequestSetCallInfo
   AttributeConsultType       3
   AttributeOriginalConnID    008b012ece62c8be
```

```
AttributeUpdateRevision    2752651
AttributeUserData          [111] 00 27 01 00
       ‘DNIS'              '8410'
       ‘PASSWORD'          <!--'1234'-->
       'RECORD_ID'         <#'8313427'#>
AttributeConnID            008b012ece62c922
```

Service Availability

Contact Center service interruption or unavailability can lead to direct revenue loss and customer dissatisfaction. Minimizing downtime and maintaining full performance capability are of the highest priority for any online service.

Availability provisioning implies using robust and quality software, preventing network intrusion and denial-of-service attacks, and protecting network and computational resources using redundant server configuration.

Genesys provides the following security features to maintain service availability, and to prevent or minimize the impact of Denial of Services (DoS) attacks:

- Redundancy
- Proxy and Parallel Servers
- Client-Side Port Definition

> **Tip**
>
> Genesys recommends using 3rd party network systems, such as firewalls, network zone partitioning, network address traversal, and network intrusion detection systems to enhance protection.

Application Redundancy

Redundant applications, normally server applications, provide backup capability in the event that an application fails. That is, if one server (the primary server) goes out of service for some reason, such as lost connectivity, the other server (the backup server) can act as the primary server, with little or no loss of service.

Security Benefits

The use of redundant applications greatly reduces the loss of functionality and data if an application is out of service because of a security-related attack, such as a denial-of-service attack.

Supporting Components

Refer to documentation for your product to determine if it supports redundancy, and the redundancy types that it supports.

Feature Description

Redundant applications address the potential loss of functionality and data in the event of an application failure.

A complete application failure can be the result of either an internal defect (for example, an infinite loop) or an external event (for example, a power failure). It can manifest itself either as no response from a process, or as termination. Typically, if a solution component stops working, the solution is no longer available to process customer interactions.

Because the application that fails cannot perform any functions, you must employ an external mechanism for both detection and correction of faults of this type. The Management Layer serves as such a mechanism. To detect an application failure, the Management Layer employs a simple monitoring component called Local Control Agent (LCA), which continuously maintains a connection with the application, confirming both its existence and its ability to communicate. To ensure that an application failure is never confused with a connection failure, the LCA that monitors a specific application always resides on the computer where the application itself is running.

LCA is installed on a one-per-host basis, and can connect to all Genesys applications located on the host. When a connection is broken, LCA generates a message to Solution Control Server (SCS), where an appropriate recovery action is chosen and executed according to the system configuration. SCS uses the Advanced Disconnect Detection Protocol (ADDP) to recognize a loss of connection with LCA. A loss of connection is interpreted as a failure of the host (that is, as failures of all Genesys components running on that host).

If a backup application is configured and running, the Management Layer automatically switches operations over to that application, provided that you have a so-called *high-availability (HA) license* . If the application is a server, the clients automatically connect to the backup server.

The Management Layer provides more robust switchover capabilities. In particular, it enables detection of situations when a running application is unable to provide service, and treats this situation as an application failure. The Service Unavailable application status serves this purpose.

When an application reports that its status has changed to Service Unavailable, if a backup server for this application is configured and running, the Management Layer automatically switches operations over to the backup server. When both the primary and backup applications are running with the Service Unavailable status, the backup application might report that it can now provide the service (that is, the status of the backup application changes to Started). In this case, the Management Layer automatically switches operations over to the backup application. As with a switchover resulting from an application failure, you must have an HA license to perform a switchover related to service unavailability.

> **Important**
>
> Although some applications support the Service Unavailable status and report it under appropriate circumstances, others do not. (For example, when T-Server loses its connection to the CTI Link, T-Server changes its status to Service Unavailable). The Management Layer bases its operation on the information supplied by an application, and cannot automatically detect an application's inability to provide service. Refer to application-specific documentation to determine whether the Service Unavailable status is supported on the application side.

Redundancy Types

There are two types of redundancy in Genesys software-warm standby and hot standby.

Warm Standby

Genesys uses the term *warm standby* to describe the redundancy type in which a backup server application remains initialized and ready to take over the operations of the primary server. The Warm standby redundancy type minimizes the inability to process interactions that might have originated during the time it took to detect the failure. It also eliminates the need to bring a backup server online, thereby increasing solution availability.

The backup server does not process client requests until its role is changed to primary by the Management Layer. When a connection is broken between the primary server and the LCA running on the same host, a failure of the primary process is reported. As a result, the Management Layer instructs the backup process to switch its role from backup to primary, and the former backup starts processing all new requests for service.

> **Important**
>
> To switch to Primary mode, the backup Configuration Server must have an active connection to the Configuration Database during the failure of the primary Configuration Server.

Although normal operations are restored as soon as the backup process takes over, the fault management effort continues. This consists of repeated attempts to restart the process that failed. Once it is restarted successfully, the process is assigned the backup role.

If SCS detects a loss of connection with the LCA of a host, it performs switchover for all applications located on the host, provided that backup applications are configured. There are two exceptions to this:

- A Configuration Server in backup mode ignores the switchover command if it detects another Configuration Server in primary mode. In other words, if the LCA residing on a host with a Configuration Server in primary mode goes down, the SCS requests that a Configuration Server in backup mode, on another host with an available LCA, switch over to primary mode. When it receives the request, this Configuration Server checks whether the Configuration Server in primary mode is down, as indicated by a lost connection between the two Configuration Servers. The Configuration Server in backup mode switches over to primary mode only if this connection is down. If the connection still exists, no switchover occurs.
- An SCS in backup mode does not try to switch itself over if it can still detect the SCS that is in primary mode. In other words, if an SCS in backup mode loses its connection to an LCA residing on a remote host with an SCS in primary mode—either because the LCA went down or a network timeout caused the SCS to drop its connection—the SCS in backup mode checks whether it is still connected to the remote SCS in primary mode. If that connection is also lost, the SCS switches over to primary mode.

Hot Standby

Genesys uses the term *hot standby* to describe the redundancy type in which a backup server application remains initialized, clients connect to both the primary and the backup servers at startup, and the backup server data is synchronized with the primary server. Data synchronization and existing client connections to the backup guarantee higher availability of a component.

Feature Configuration

The configuration of redundant Genesys applications can vary, depending on application type, and requires that you do the following steps:

1. Create an Application object for the primary application.
2. Install the primary application.
3. Configure an Application object for the backup application.
4. Install the backup application.
5. In the primary Application object, add the backup Application object and specify the supported redundancy type.

For more information, and for detailed instructions for setting up redundant applications in your environment, refer to product-specific documentation.

Proxy and Parallel Servers

Proxy and parallel servers add efficiency to large configurations, and can limit the damage caused by an outage.

Both configurations are a type of distributed configuration, but they differ in how the workload is distributed between the servers:

- In a proxy environment, each proxy server takes a portion of the workload and works on that portion exclusively.
- In a parallel environment, the workload is distributed among all of the servers, with one server attempting to keep the distribution as balanced as possible.

Proxy servers are particularly useful for systems that are widely dispersed over a large geographic area. In a proxy environment, the number of clients attached to a server is distributed across a set of servers (running in proxy mode), all of which funnel down to a central server (the Master).

Parallel servers enable load sharing. That is, multiple instances of a server run in parallel, and the load is distributed among them.

Security Benefits

The use of proxy and parallel servers greatly reduces the loss of functionality and data if a server goes out of service.

- If a proxy server fails, you lose only the clients associated with that proxy server. In a non-proxy environment with only one server instance, if that single server goes down, all the clients are lost.
- If a server in a parallel configuration fails, new requests are distributed to the remaining servers.

Supporting Components

Refer to documentation for your product to determine if it supports some variation of proxy and/or parallel configuration, redundancy, and how to implement it for your system.

Feature Description

Proxy and parallel servers address the efficiency issue inherent in large configurations, and also minimize the loss of functionality and data in the event of an application failure.

Proxy servers are particularly useful for systems that are widely dispersed over a large geographic area. In a proxy environment, the clients that require a connections to a server are distributed across

a set of servers (running in proxy mode), all of which down to a central server (the Master). If one proxy server fails, only the clients connected to that server are lost. Compare this to a single server environment, where, if the single server fails, the whole system is lost.

Parallel servers enable load sharing. That is, multiple instances of a server run in parallel and the load is distributed among them. If one of the parallel servers fails, new requests are distributed to the remaining servers so there should be no loss of service.

Each server in a Proxy or Parallel configuration can also be set up with a backup server, enabling each to take advantage of the benefits of redundancy. Refer to Application Redundancy.

Feature Configuration

The configuration of proxy or parallel Genesys servers can vary, depending on the server type. For more information, and for detailed instructions for setting up proxy or parallel servers in your environment, refer to the appropriate product documentation.

Client-Side Port Definition

The client-side port definition feature enables a client application (of server type) to define its connection parameters before connecting to the server application. This enables the server application to control the number of client connections. In addition, if the client application is located behind a firewall, the server application will be able to accept the client connection by verifying its predefined connection parameters.

Security Benefits

The client-side port definition feature enables a customer to better control the data connections through their firewalls, by enabling them to precisely define the connections that can tunnel through the firewalls. This reduces the susceptibility to denial-of-service (DoS) attacks, where an excessive number of malicious application-level requests arrive at the same server-side port. This can result in the server application dropping its performance or even becoming unstable. It also affects the other applications on the same server or in the network.

Supporting Components

This feature applies to the following components:

- Configuration Server Proxy on all of its connections, except to its HA partners
- License Resource Manager when connecting to Configuration Server/Configuration Server Proxy
- Media T-Servers when connecting to Configuration Server/Configuration Server Proxy
- Network T-Servers when connecting to Configuration Server/Configuration Server Proxy
- Load Distribution Server on all of its connections with T-Server and Configuration Layer.
- Universal Router Server when connecting to Configuration Server/Configuration Server Proxy, T-Server, Custom Server, Stat Server, and DB Server
- Custom Server when connecting to Configuration Server/Configuration Server Proxy
- Outbound Contact Server when connecting to Configuration Server/Configuration Server Proxy, T-Server, Stat Server, and DB Server
- CPD Server and CPD Proxy Server when connecting to Configuration Server/Configuration Server Proxy and T-Server
- IVR Server and IVR Drivers for WVR for AIX, and for MPS when connecting to Configuration Server/Configuration Server Proxy
- Stat Server when connecting to Configuration Server/Configuration Server Proxy, T-Server, DB Server, and Interaction Server
- Genesys Voice Platform (GVP) when connecting to Configuration Server/Configuration Server Proxy

- Interaction Server when connecting to Universal Contact Server, Interaction Server, Email Server Java, Chat Server, SMS Server, Social Messaging Server, Classification Server, Stat Server, Message Server, and Configuration Server/Configuration Server Proxy
- Chat Server when connecting to Message Server, Configuration Server/Configuration Server Proxy, Interaction Server, and Universal Contact Server
- Web API Server Java when connecting to Configuration Server/Configuration Server Proxy, Solution Control Server, and Message Server
- Web API Server .NET when connecting to Configuration Server/Configuration Server Proxy, Solution Control Server, and Message Server
- SMS Server when connecting to Protocol Adapter, Interaction Server, Message Server, Configuration Server/Configuration Server Proxy, and Solution Control Server
- Classification Server when connecting to Configuration Server/Configuration Server Proxy and Message Server
- Social Messaging Server when connecting to Message Server, Configuration Server/Configuration Server Proxy, and Interaction Server
- Email Server Java when connecting to Configuration Server/Configuration Server Proxy, Message Server, Interaction Server, and Universal Contact Server
- Genesys Info Mart when connecting to Configuration Server/Configuration Server Proxy and Message Server
- CCPulse+ when connecting to Configuration Server/Configuration Server Proxy

> **Important**
>
> For CCPulse+ connections to Configuration Server, refer to the *Reporting 8.0 CCPulse+ Administrator's Guide*.

- Workspace Desktop Edition (formerly known as Interaction Workspace) when connecting to Configuration Server, Stat Server, Universal Contact Server, Interaction Server, and T-Server/SIP Server.

> **Important**
>
> For Workspace Desktop Edition connections to Configuration Server, please refer to the *Workspace Desktop Edition Deployment Guide*. For the other connections, the procedures described in this guide are applicable.

- Genesys Rules Engine and Genesys Rules Authoring Tool when connecting to Configuration Server/ Configuration Server Proxy
- Genesys Interactive Insights on its connections between server components. Refer to Genesys Interactive Insights documentation for more information.

In addition, Enterprise SDK and Platform SDK support client-side port definition for Genesys components that support this feature. For details about how client-side port definition can be used in custom-built applications, refer to the appropriate API Reference for your development platform.

Known Issues and Recommendations

Several known issues exist in the current client-side port definition feature implementation:

- Activation of this feature requires you to supply client parameters, which Genesys recommends that you do through the Genesys Installation Wizard.
- The Media Configuration Wizard does not support the client-side port definition feature configuration. When installing T-Server in an environment where there will be a port-restricted firewall between T-Server and Configuration Server, you must initially configure and install such a T-Server manually.
- If the client-side port definition feature is enabled during T-Server installation, when T-Server starts, it will report warning messages in its log about command-line parameters related to this feature. Ignore these messages.
- If a client's connection parameters to Configuration Server are defined manually in several different places, make sure that those entries are identical.
- If you add this feature to configured redundant components, the port number (and, optional, IP address) specified in the primary server Application object are automatically propagated to the backup server Application object. Correct these parameters in the backup server Application object manually.
- Genesys licensing functionality does not support the client-side port definition feature configuration.

Feature Configuration

To configure client-side configuration, do the following steps:

1. Specify the client's connection parameters (the port number and optionally, the IP address). These parameters will be used for the initial connection to Configuration Server. **[+] Show steps**

You can specify the parameters while using the Genesys Installation Wizard to install the client or specify them manually.

> ### Important
>
> Genesys recommends that you specify the port number (and, optional, IP address) of a client when you install it by using the Genesys Installation Wizard. If you decide to enable this feature later, you can either re-install the component and define the client's connection parameters during the component installation, or specify the parameters manually.

<tabber> Using Wizard on UNIX=

a. In the directory to which the component installation package was copied during Wizard configuration, locate a shell script called **install.sh**.

b. Run this script from the command prompt by typing **sh** and the file name. That is: **sh install.sh**.

c. Proceed with the installation according to the instructions in the component's product documentation.

d. At the prompt:
Client Side Port Configuration
Select the option below to use a Client Side Port. If you select this option, the application

can use Client Side Port number for initial connection to Configuration Server.
Do you want to use Client Side Port option (y/n)?
Type y for yes, then press **Enter**.

e. At the prompt:
Client Side Port port
Enter the port number that the client application will use for its TCP/IP connection to the Configuration Server, and press**Enter**. Note that the installation script will not verify the availability of the component's port number. You must specify a unique port number that is dedicated to this connection.

f. At the prompt:
Client Side IP Address (optional), the following values can be used:
(Optional) Enter the IP address that the client application will use for its TCP/IP connection to the Configuration Server, and press **Enter**.

g. Complete the component installation as specified in the component product documentation. During the installation, the client's predefined port number (**- transport-port <port number>**) and IP address (**- transport-address <IP address>**) (if specified) will automatically be added to:

- The **Command-Line Arguments** text box on the **Start Info** tab of the server's **Application Properties** dialog box, so that the application can be started with the Management Layer.
- The server application's **run.sh** file, so that the application can be started by the startup files.
- The **ImagePath** in the Application folder in the Registry Editor, so the application can be started as a Windows Service.

|-|

Using Wizard on Windows=

a. Launch the component's Genesys Installation Wizard according to the instructions in the component's product documentation.

b. On the **Client Side Port Configuration** page, do the following:

i. Select the **Use Client Side Port** check box.

ii. Specify the component's (the client's) parameters for connecting to the Configuration Server associated with this client application, as follows:

- **Port**: Enter the port number that the client application will use for its TCP/IP connection to the Configuration Server. Note that the installation script will not verify the availability of the component's port number. Make sure that you specify a unique port number that is dedicated to this connection.
- (Optional) **IP Address**: Enter the IP address that the client application will use for its TCP/IP connection to the Configuration Server.

> **Important**
>
> Genesys recommends that you specify the port number (and, optional, IP address) of a client when you install it by using the Genesys Installation Wizard. If you decide to enable this feature later, you can either re-install the component and define the client's connection parameters during the component installation, or specify the parameters manually.

iii. Click **Next**.

c. Complete the component installation as specified in the component product documentation. During the installation, the client's predefined port number (**- transport-port <port number>**) and IP address (**- transport-address <IP address>**) (if specified) will automatically be added to:

- The **Command-Line Arguments** text box on the **Start Info** tab of the server's **Application Properties** dialog box, so that the application can be started with the Management Layer.
- The server application's **run.sh** file, so that the application can be started by the startup files.
- The **ImagePath** in the Application folder in the Registry Editor, so the application can be started as a Windows Service.

|-|

Manually= You configure a client's connection parameters by adding them as command-line parameters that are be used during component startup. You can start Genesys components by using the Management Layer, a startup file, a manual procedure, or the Windows Services Manager. For a server application, all these methods usually require command-line parameters in addition to an executable file name.

a. Add one or both of the following parameters to the application's command line depending on the method (see below) that will be used for starting the client application:

- **-transport-port <port number>**
- **-transport-address <IP address>** (if specified)

Where:

- **<port number>** is the port number that a client will use for its TCP/IP connection to Configuration Server.
- **<IP address>** is the IP address that a client will use for its TCP/IP connection to Configuration Server.

b. To start the application manually, add the client's connection parameters to the application's command line. For example:

```
<switch>_server.exe -host <Configuration Server host> -port <Configuration Server
port> -app <T-Server Application> -l <license address> -nco [X]/[Y] -transport-port
<port number> -transport-address <IP address>
```

For more information about starting and starting Genesys components, see the product documentation for the component.

<tabber>

2. Add a Configuration Server Application object to the client's Connections. **[+] Show steps**

a. In Genesys Administrator, open the **Provisioning** tab and navigate to the folder containing the client application.

b. Select the client application and open the **Configuration** tab.

c. If the Configuration Server Application object to which the client will connect is not displayed in the **Connections** table in the **General** section, do the following:

i. Above the table, click **Add**.

ii. In the **Browse** window, navigate as necessary and select the Configuration Server to which this client will connect.

iii. Click **OK**.

d. In the **Connections** table of the **General** section section, select the Configuration Server Application object to which the client will connect, and click **Edit** above the table.

e. In the **Connection Info** dialog box, open the **Advanced** tab.

f. In the **Transport Protocol Parameters** text box, enter one or both of the following parameters:

```
port=<port number>
address=<IP address>
```

Where:

- **<port number>** is the port number that a client will use for its TCP/IP connection to the server.
- **<IP address>** is the IP address (or host name) that a client will use for its TCP/IP connection to the server.

If you specify both of these parameters, use a semicolon as the delimiter. For example:

```
port=<port number>;address=<IP address>
```

> **Important**
>
> The parameters that you specify here must be the same as the parameters that you specified when installing the client.

g. Click **OK** to save the new connection configuration.

3. (Optional) Add a client's connection parameters to the server's connections properties. **[+] Show steps**

Use these steps to specify a client's parameters for connecting to a server application other than Configuration Server.

a. In Genesys Administrator, open the **Provisioning** tab and navigate to the folder containing the client application.

b. Select the client application and open the Configuration tab.

c. If the server-type Application object to which the client will connect is not displayed in the **Connections** table in the **General** section:

i. Above the table, click **Add**.

ii. In the **Browse** window, navigate as necessary and select the server to which this client will connect.

iii. Click **OK**.

d. In the **Connections** table of the **General** section, select the server Application object to which the client will connect, and click **Edit** above the table.

e. In the **Connection Info** dialog box, open the **Advanced** tab.

f. In the **Transport Protocol Parameters** text box, enter one or both of the following parameters:

```
port=<port number>
address=<IP address>
```

Where:

- **<port number>** is the port number that a client will use for its TCP/IP connection to the server.
- **<IP address>** is the IP address (or host name) that a client will use for its TCP/IP connection to the server.

If you specify both of these parameters, use a semicolon as the delimiter. For example:

```
port=<port number>;address=<IP address>
```

> **Important**
>
> When you add this feature to configured redundant components, the port number and IP address specified in the primary server Application configuration object are automatically propagated to the backup server Application configuration object. Correct these parameters in the backup server Application object manually.

g. Click **OK** to save the new connection configuration.

Protection of Data in Transit

In addition to the protection of data where it resides, as described in Protection of Data at Rest, data must also be protected when it is sent over communication channels.

Genesys provides the following security features to address data and service integrity:

- Transport Layer Security (TLS)
- Federal Information Processing Standards (FIPS)
- Hypertext Transport Protocol Secure (HTTPS)
- Secure Real-Time Transport Protocol (SRTP)
- Lightweight Directory Access Protocol Secure (LDAPS)—in Management Framework, this is implemented between Configuration Server and the External Authentication LDAP directory.

As a backup mechanism, passwords are also encrypted during transit.

Secure Connections (TLS)

Genesys products handle consumer data for customers in sensitive and regulated industries (banking, insurance, retail, government, and so on). In many cases, this data should be protected while being transmitted via network media. Such protection can be implemented for connections that are established across public networks, as well as corporate internal networks.

This section of the Genesys Security Deployment Guide is intended to present a user-level guide on how to utilize the Transport Layer Security (TLS) protocol to secure network connections in a Genesys deployment, both between Genesys components or between Genesys components and 3rd-party software. It is intended to be a complete source of information on how to configure secure connections.

Security Benefits

TLS provides strong authentication, message confidentiality, and integrity capabilities. TLS secures data transmission by using a variety of encryption options. TLS can authenticate one or both parties engaged in secure communication. It also provides data integrity through an integrity check value. In addition to protecting against data disclosure, the TLS protocol can be used to help protect against masquerade attacks, man-in-the-middle attacks, bucket brigade attacks, rollback attacks, and replay attacks. TLS, as implemented by Genesys, is in most cases considered to be consistent with Federal Information Processing Standards (FIPS).

Securing Connections – An Overview

Securing connections in an existing Genesys environment should be done iteratively. The following is a high-level summary of how to accomplish this.

1. Enumerate all connections that are to be secured. Then categorize those connections based on the following parameters:
   - Whether the connection is between Genesys components or a Genesys component and a 3rd-party component, and which Genesys components are linked by the connection.
   - The payload data protocol used, namely TLib, a Genesys internal protocol, or an application data protocol (such as SIP).
   - The types of operating systems used by the connection peers, namely Windows or *nix (Linux, AIX, Solaris).
   - Details of the connection peer implementation, such as a Java application, a Web application, a C++ application based on a common library, and so on.
2. Having identified and categorized the connection, do the following:
   a. Generate the entities required to secure the connection. Note that this is the minimum required, your implementation may require more.

- For the certificate presenter:
  - A certificate (or certificate chain), which represents the entity presenting it; typically digitally signed by a CA and set with an expiration date. This is not required on the client side of a simple TLS connection.
  - A private key, generated with the certificate as part of a public/private key pair. This is also not required on the client side of a simple TLS connection.
- For the receiver of the certificate:
  - A certificate list of the trusted CAs, including the CA that signed the certificate to be received.

b. Install the certificates on the host where the component is running.

c. Use Genesys Administrator to configure the certificates.

d. If running in the *nix environment, configure an environment variable to point to the security libraries.

e. For third-party components, determine the source of the certificates (such as a Java keystore, an Oracle `cacerts` file, web servers for https connections, and so on) and then configure them appropriately.

About this Section

This section of the Security Guide describes how to secure connections using TLS, and is intended to be a structured source of information. It describes how to secure certain types of connections that have been identified in the deployment and categorized as described above. Decisions on whether to protect or not to secure a connection are outside the scope of this guide.

A complex Genesys deployment can have many different network connections, utilizing many types of payload protocols. Most of these connections are configured in (almost) the same way. Some connections require specific details in their security configuration. Because it is impossible to describe every type of connection between every type of Genesys component, this guide is organized as follows. All information is divided into general sections, corresponding to general types of connections. Inside each general section, subsections describe specific connections of that category of connection type.

Before starting, note the following:

- The instructions in this document assume that you are adding Genesys Transport Layer Security (TLS) to existing connections of your Genesys 8.x environment—that is, that your applications have already been installed, properly configured, and associated with hosts and with each other. For information about configuring new hosts, applications, and associations between them, see the *Framework Deployment Guide*.
- If you are using pre-8.x Genesys components, you must upgrade them to release 8.x before you can configure secure connections between them.
- Some components require additional steps to complete the configuration of secure connections. These steps are provided in the Deployment Guide for the particular product or component.

What is TLS

For TCP connections, the industry standard mechanism is Transport Layer Security (TLS), the modern version of the old Secure Sockets Layer (SSL) protocol. TLS is also the underlying mechanism for many higher level protocols, such as HTTPS, SIPS, LDAPS, and so on.

TLS provides encryption of data and connection peer authentication. TLS protocol has been evolving with security in mind. The most recent version of TLS is v1.2, currently considered to be the most secure.

For more information, refer to Transport Layer Security (TLS).

Benefits of TLS

There are three benefits of using TLS, as follows:

- Authentication: Certificates are used to exchange information that validates the two parties.
- Confidentiality: Encryption is used to keep the contents of the transmission private.
- Integrity: Verification that data was not manipulated in transit.

Follow the links to read more about each benefit.

How TLS Works

TLS provides encryption of data and authentication of connecting peers. It utilizes a set of helper cryptographic algorithms, called ciphers, chosen from a vast number of supported cryptographic algorithms. For each session, a defined set of algorithms, called a cipher suite, are used to provide the hashing, key exchange, encryption, message verification, and other parts of the protocol. The algorithms chosen depend on the protocol version, implementation version and configuration, and are selected during the setup phase of the protocol, known as the handshake.

For more information, refer to Transport Layer Security (TLS).

Ciphers and Cipher Suites

A cipher suite represents a combination of encryption ciphers to achieve each of the three benefits of using TLS during handshaking, integrity checks and data exchange. TLS implementations support a variety of cipher suites. Generally, the set of available ciphers can be configured to the preferences of the user. See Tuning Available Cipher Lists for information about how to customize cipher lists.

The Handshake

The handshake is the way in which the two parties determine a mutually agreed cipher suite to use

for communication. Either party can shorten the list of acceptable cipher suites.

Until the secure connection is established, all exchanged data is not encrypted and therefore available to any party that might intercept handshake traffic. Asymmetric (public key) cryptography is used in the handshake phase to allow secure operations over a clear text connection. Well-defined key exchange algorithms (such as RSA and Diffle-Hellman provide the means to agree on an encryption key that will be used for secure communication, without providing a possible third-party intercepting party any real means of identifying that key. See Asymmetric cryptography explained for additional information about how that is possible.

At any stage of the handshake, if any party identifies any problem with the data, protocol versions, or keys provided or requested (for example, there is an encryption validation failure in steps t and u of the handshake procedure, below), the party drops the TCP connection indicating that the handshake cannot be continued. A new attempt to secure the connection can only be made by restarting the process with a new connection and a new handshake.

Important

During the handshake, the term protocol *version* has two meanings:

- The format of messages are tied to the agreed-upon TLS version, but can be upgraded. For example, an SSLv2 message can ask for TLS1.2 communication. If this is the case, communication will fall back to the agreed version
- Available cipher suites are tied to actual communication protocol version agreed-upon during the handshake.

In the simplest case, the client requests the highest version of protocol that it supports. The server responds with a protocol version equal to or lower than that requested; that is, the version that the server is willing to use for this session. If the client is not happy with the server choice (for example, the server chooses a very low version, like SSL 2), the client silently disconnects the TCP connection, and the connection is not made.

In the more complex case, specific protocol versions are specified by the client and server at the start of the negotiation. See Protocol Versions Compatibility for more information about how Genesys implements this.

Tip

99% of all errors in TLS communication occur during the handshake phase.

The handshake procedure consists of these steps. Note that this is the handshake as carried out by Genesys components; third-party components may perform the handshake differently.

[+] Show steps

1. Client establishes TCP connection to server.

2. Client requests TLS to be used.
3. Client provides acceptable protocol version and cipher lists for server to choose from. It determines its acceptable protocol version as follows:
   - If the default protocol version is configured on the client side, the client requests its highest available protocol version.
   - If a specific protocol version is configured on the client side and is available, the client requests the configured protocol version.
4. Client sends other cryptographic information (key exchange algorithm data).
5. Client indicates that server response is expected.
6. Server confirms TLS protocol is to be used.
7. If the version requested by the client is lower than any version supported by the server; the server drops the connection. For example, if the server is configured with TLSv12 and the client requests TLSv11, the server drops the connection. Otherwise, the server responds with the protocol version and cipher list that is to be used for the session. It determines the protocol version as follows:
   - If the default protocol version is configured on the server side, the server responds with the highest available protocol version equal to or lower than the version requested by the client. For example, if the client requests TLSv12, but the server has TLSv12 disabled and TLSv11 enabled, the server responds with TLSv11.
   - If a specific protocol version is configured on the server side and is available, the server responds with this version if it is equal to or lower than the version requested by the client. For example, if the server is configured with SSLv3 and the client requests TLSv11, the server responds with SSLv3.
8. The client receives the server's response with the negotiated version. If one or both of the following conditions are false, the client drops the connection silently and the connection is not made. If both conditions are true, the handshake continues.
   - The client has this version available; that is, the version is not disabled in the client's registry, and where applicable, is allowed by the value of the **sec-protocol** configuration option.
   - The client is configured to accept this version; for example, as specified by the value of the **sec-protocol** option.
9. Server sends its certificate for verification.
10. Server sends other cryptographic information (to complete key exchange).
11. If Mutual TLS is being used; the server explicitly requests client certificate for verification. Server sends a list of its own trusted CAs for client to select appropriate certificate for presentation.
12. Server indicates that client response is expected.
13. Client validates server certificate.
14. If server requested client certificate, client sends it.
15. Client completes key exchange calculations.
16. Client uses the resulting negotiated key and cipher list to encrypt and sign a test message (consisting of all data that was exchanged from the start of the session) and sends it to the server for verification.
17. If requested, server validates client certificate.
18. Server completes key exchange calculations.
19. Server uses the resulting negotiated key and cipher list to encrypt and sign a test message (consisting of all data that was exchanged from the start of the session) and sends it to the client for verification.

20. Server verifies the test message received from client.
21. Client verifies the test message received from server.
22. TLS session is established.

The handshake is finally verified at steps 16, 19, 20, and 21, in which both parties use the derived keys and agreed ciphers to encrypt and sign a test message consisting of all data that was previously exchanged during the handshake, and then exchange the test messages. To verify the result, the opposite party uses its own derived key data and ciphers to validate and decrypt the received message, and compares the decrypted text with known data that was exchanged during the handshake. As a result, both parties know that they may communicate in a safe, authenticated, and encrypted way.

After the handshake is complete, data exchange starts. Data is encrypted using the (symmetric) key that was derived and verified during the handshake. Messages are encrypted and signed for authentication using the cipher list agreed on in the Handshake. Since symmetric encryption is used, communication can occur much faster than the handshake process itself.

For more information about the TLS handshake, see SSL handshake explained.

Authentication

TLS authentication uses TLS certificates in X.509 format to tell the receiving party who the sending party is (in the **Subject** or **Subject Alternate Name** field), and includes the public key. The X.509 format is standard in the industry, but there are different storage formats, such as RSA, PKSC#12, PEM, DER, and so on. The certificate must be issued and signed by a Certificate Authority (CA) that both parties trust, and therefore the information in the certificate can also be trusted. The certificates themselves are stored in a keystore that the sender application can access.

Modes of Operation

TLS operates in one of three different modes:

1. **Encryption Only:** Neither party does any validation of the other's certificate, essentially removing the authentication part. This mode can be selected by explicitly specifying anonymous, non-authenticating ciphers. Genesys strongly recommends that you use this mode only for purposes of debugging.
2. **Simple TLS:** Only the server provides a certificate to the client, which the client validates for authentication. The client does not provide a certificate for server verification. This mode is used most often for HTTP(s) websites where consumers (the clients) want to know they are communicating with the real website (the server), but typically do not have a certificate to pass to the server.
3. **Mutual TLS (also known as Mutual Certificate Exchange):** Both the client and server pass certificates to each other and each validates (authenticates) the other party, establishing a two-way trust. See steps j, m, and p in the handshake procedure.

TLS Certificates

A TLS certificate is a stored and cryptographically signed entity that authenticates the presenter of the certificate. It is signed by the issuer certificate authority (CA). Certificate signature can be easily verified by any party having only the issuer and issued certificate. CAs can have a nested structure, with the issuing party chain leading to the root level, to a so-called *root CA*, which signs its own certificate. Such root CA certificates must be known to, and trusted by, the local system to establish all the chain of trust, from the leaf certificate presented for verification to the trusted root CA certificate.

For more information about CAs and certificate chains, refer to Certificate Authority.

Certificate Signature Hash Function

An important aspect of a TLS certificate is the cryptographic hashing function used to create the certificate signature. Certificates signed by a weak hash function can be fraudulently duplicated, resulting in a major security breach. A number of cryptographic hash functions exist in widespread usage, including the following:

- MD4
- MD5
- SHA-1
- SHA-2 (a family of algorithms)

Currently, MD4 and MD5 hashing functions are considered totally flawed and compromised. SHA-1 is currently being phased out, due to a theoretical flaw found. Major software vendors, including Mozilla and Microsoft, are working on replacing certificates signed by SHA-1, and have announced that these certificates will not be accepted starting from January, 2017. SHA-2, a family of cryptographic hash functions, is now the industry standard, including the most widely used SHA-256.

> **Important**
>
> Genesys strongly recommends that all generated certificates are signed with SHA-256.

Private and Public Keys

Another important part of the entity certificate is the cryptographic public key stored within the certificate. Asymmetric cryptography principles allow public keys to be shared universally. The public key is used to encrypt data so it can only be decrypted by using the private key of the same key pair. CA certificates contain CA public keys, which are used in the issued certificate signature verification.

To successfully decrypt the data sent by the peer, a TLS entity must use the private key generated along with the public key that is stored in the certificate. Private keys must be carefully protected, because it gives access to all cryptographic communications performed using the certificate. A compromised private key is one of the reasons a certificate can be revoked.

Certificate Exchange and Validation

Certificates are exchanged during the first stages of the TLS handshake. The purpose of this exchange is to provide the certificate to the remote party for validation.

The receiver of the certificate performs various checks before accepting the certificate as valid, such as:

- **Expiration**: Is the certificate expired?
- **Certificate Authority**: Is the certificate signed by a CA I trust? Note that this might be via a certificate chain where Intermediate Authorities signed the certificate and their certificates are signed by the root CA.
- **Revocation**: Has the certificate been revoked by being added to a Certificate Revocation List?
- **Purpose**: Is the certificate created for the purpose for which it is being used? This might be one or both of the following:
  - Is the subject of the certificate appropriate? Refer to Target Hostname Check below.
  - Is the certificate type, as provided in the `Certificate Usage` field in the certificate `Extensions`, appropriate? For example:

```
Extensions:
    Identifier: Netscape Certificate Type - 2.16.840.1.113730.1.1
      Critical: no
      Certificate Usage:
        SSL CA
        Secure Email CA
        ObjectSigning CA
```

Target Hostname Check

The client side of a TLS connection can also check that the server certificate has been issued to the same server to which it is connecting. The client side of the connection connects to the server's FQDN, and therefore might check that the target FQDN matches the FQDN listed in the server certificate. The server side of the connection is unable to perform this check, even if mutual TLS is used, because that server might not (and usually does not) have access to rDNS mechanisms required to identify the client's FQDN. The client may be located behind NATs, including server internal NATs, which do not allow the tracing of the client connection back to the originating network node. Even with a direct connection, running an rDNS check during a TLS handshake leads to additional delays in an otherwise very lengthy process.

The Target hostname check is performed against the Subject Alternative Name (SAN) in the server certificate presented for authentication. If SAN is not present, the certificate subject Common Name (CN) is used for verification. The certificate must list the server FQDN, as designed, to be accessed by client connections.

> **Warning**
> Some TLS implementations allow wildcard symbols to be used in the configured FQDN

> to match multiple (sub)domains with one certificate, but this can be very dangerous and is not recommended by Genesys. Refer to the implementation details to determine if this possible with your implementation.

For information about how Genesys enables the target hostname check, refer to Check for Certificate Host Matching.

Certificate Revocation Lists

A Certificate Revocation List (CRL) is a CA-generated list of certificates issued by the CA that are no longer to be trusted because they have been compromised in some way, or are otherwise deemed not to be trusted. Strictly speaking, a check against a valid CRL is a required step in any certificate validation. In practice, this check might be omitted if the CA is controlled by the enterprise, and certificate revocation is tracked by other means.

To set up a CRL, refer to Configuring a Certificate Revocation List.

Confidentiality

To ensure confidentiality, TLS uses a session key (derived from the public key) that is used by both parties to encrypt and decrypt messages and data. The session key is used for one session only, and then discarded, so it cannot be used in later sessions to "listen in".

For confidentiality, TLS uses symmetric encryption, which is faster than other encryption methods.

Integrity

To assure data integrity, TLS uses Message Authentication Code. The hashing algorithm in the agreed-upon cipher suite generates a checksum for the message contents and stores it in the message. If the checksum generated at the receiver's end does not match that checksum in the message, the message has been manipulated during transit, most likely as the victim of a Man-in-the-middle attack.

TLS Implementations in Genesys

TLS is a protocol with an agreed-upon standard definition. To utilize TLS in real applications, the protocol must be implemented in source code. Many different TLS implementations exist, some developing and patching newly found security issues, and some of them not. Refer to Comparison of TLS implementations to see what TLS implementations exist and how they differ.

All TLS implementations differ in the list of supported features, aspects, protocol versions and cipher lists. However, all TLS implementations still implement the same TLS, and therefore should be able to communicate with each other, given that a compatible set of features is requested at each end of the connection.

Even Genesys implementations vary. For example:

- Genesys uses different technologies (C++, Java, and so on)
- Genesys operates on different infrastructures (Windows, Unix)

Summary

Genesys products use the following TLS implementations for their proprietary protocols, depending on the component platform.

Environment	Platform	TLS Implementation Used	Keystore
Genesys native applications (with no dependency on .NET or Java)	Microsoft Windows	Microsoft SChannel (built into the operating system)	Windows Certificate Services
	*nix (Linux, AIX, Solaris)	OpenSSL (was RSA Bsafe)	File system
Genesys applications with dependency on .NET	Microsoft Windows	Microsoft SChannel (built into the operating system)	Windows Certificate Services
Genesys applications with dependency on Java	All	Java Secure Socket Extensions from Oracle JRE with configured Provider	File system; JKS; or Windows Certificate Services (on Windows only)
Selected applications (such as Genesys Voice Gateway and so on) or individual connections of a native application (such as GVP Media or Configuration LDAP)	Microsoft Windows, *nix (Linux, AIX, Solaris)	Built-in OpenSSL (review the documentation for each application)	File system (review the documentation for each application)
HTTPs (web interfaces)	Typically based on Application Server, such as J2EE, Microsoft IIS		

All Genesys TLS implementations are compatible and can communicate with each other. However, configuration details for components using different TLS implementations differ significantly.

Genesys components may also utilize open standard application protocols, such as LDAP, which, if secured, may have third-party implementations that are different (based on different versions of OpenSSL for example) than the standard implementation of Genesys TLS on a selected platform. More details can be found in sections dedicated to each specific application protocol.

Standard TLS Implementation for an Application with no .NET or Java Dependencies

Native Genesys components use an internal Genesys common library to facilitate network communication using proprietary Genesys protocols, and selected open standard protocols, like HTTP. The Genesys common library encapsulates the actual underlying TLS implementation from the component code, allowing same applications to run on different platforms while using the same API. The Genesys common library is used only by native Genesys components, not components that are written in managed code such as Java and .NET. Any exceptions to this are noted in the documentation for individual applications.

As shown in the table above, Genesys utilizes different TLS implementations to facilitate secure connections, depending on the underlying operating system.

Genesys Native Applications on Windows

When running on Windows, the Genesys common library uses Microsoft SChannel TLS implementation, technically a part of the host Windows operating system itself. All TLS operations are delegated to the operating system (SChannel) level, and all configuration is passed to the operating system level. Genesys components have very little control over TLS operations.

Because of the built-in nature of Windows' SChannel module, no specific installation of the Genesys Security Pack is required.

TLS certificates (including private keys) and CA certificates (trusted or not) are stored in Windows certificate storage. Refer to Managing Certificates using MMC on Windows for details about accessing and managing the certificate storage. Two types of certificate storage are available: user and system level. The Genesys common library first looks in user-level storage for the configured certificate, then in system-level storage.

Windows implementations do not allow any wildcard symbols to be used in the certificate Subject Alternate Name (SAN) or Common Name (CN).

In mutual TLS mode, whenever a server requests a client's certificate, it presents a list of server trusted CAs from which the client selects a certificate to present. Regardless of the client certificate configuration, Windows will lookup a certificate issued by one of the CAs provided and send it to the server.

Important

To avoid confusion and the presentation of wrong certificates, Genesys strongly recommends that you import only certificates intended for actual usage.

Available protocol versions and cipher lists may differ dependent on the version of the Windows operating system, since the SChannel module is an essential part of the operating system. Your operating system documentation provides information about the availability of particular protocol versions and ciphers. In addition, you may also want to consult the following:

- Information about the availability of protocol versions: Support for SSL/TLS protocols on Windows
- Information about availability of ciphers is available in the Cipher Suites in SChannel

Policies for CRL verification and protocol version availability are controlled on the operating system level by registry settings and system policies. A detailed description of Windows operating system security administration is outside the scope of this document, but for information about setting available TLS versions on Windows, see TLS/SSL Settings.

For information about configuring the availability of cipher lists, see How to restrict the use of certain cryptographic algorithms and protocols in Schannel.dll.

For information about how to configure the selection of ciphers with Genesys TLS, see Tuning Available Cipher Lists. To understand how protocol selection works in Genesys TLS, see Tuning Protocol Version Availablity.

Genesys Native Applications – Genesys Security Pack on UNIX

When running on Unix-like operating systems (Linux, AIX, Solaris, and so on) Genesys common library loads and uses the Genesys Security Pack on UNIX component (referred to in this document as *Security Pack*) to facilitate secure connections. Security Pack encapsulates the OpenSSL library, and provides most of the features of the OpenSSL library.

Important

Genesys Security Pack does not currently support the Mac operating system. When running on Mac OS X, Genesys Common Library loads and uses the **libgsecurity_openssl.dylib** module that is included with the SIP endpoint SDK installation package (and/or bundled with endpoint executable).

Prior to release 8.5.1, Genesys used the RSA BSAFE SSL-C implementation of secure protocols. Starting with release 8.5.1, the default implementation is replaced with the OpenSSL library. The RSA BSAFE-based implementation is still provided as an alternative in cases when a higher level of backward compatibility and interoperability with legacy applications is required.

OpenSSL was chosen as the underlying TLS implementation because it is a de-facto industry standard

implementation. OpenSSL is constantly under observation by the development and cryptoanalyst community, so all security issues found are resolved promptly. The OpenSSL version used by the Genesys Security Pack is updated whenever it is needed from a security point of view.

> **Important**
>
> If you are using a pre-8.5 release of a Genesys product and want to utilize the latest version of the Security Pack (because you need the latest protocols or security fixes), refer to individual product documentation to determine if you can use it with the version of your product. There is limited interoperability of the latest versions of the Security Pack with Genesys releases before 8.5.

For information about installing and using Genesys Security Pack, see Installing Genesys Security Pack.

Genesys Security Pack is loaded as a shared (**.so**) library whenever the application requires a secure connection for the first time in its lifecycle. Genesys Security Pack is linked to OpenSSL statically, so distribution of additional shared modules is not required.

Genesys Security Pack is designed as a drop-in replacement library. Updating to a newer version (or rolling back to a previous version) of Genesys Security Pack is trivial, and requires only a restart of the application.

Refer to the Genesys Security Pack on UNIX Release Note to determine the OpenSSL version used, and recent modifications.

Backward Compatibility of Genesys Security Pack

The new Security Pack is a drop-in replacement of an existing Security Pack. To upgrade to the OpenSSL version, replace the binary modules. You do not have to make any change to the configuration of existing deployments.

> **Important**
>
> You must restart those applications using secured connections after upgrading to the new Security Pack.

To ensure backward compatibility, the new Security Pack includes a new mode (referred to as *compatibility mode*) that restores some behavior of the old Security Pack. This mode is disabled by default.

> **Warning**
>
> - Compatibility mode should be enabled only as a last resort if the new Security Pack is

encountering compatibility errors in your environment.

- If you are in compatibility mode, Genesys strongly recommends that you take the necessary actions to avoid long-term usage of this mode.

To enable the Security Pack compatibility mode, set the environment variable GCTI_SECPACK_COMPAT_MODE to 1 before starting the application. After starting it, you cannot disable the mode during application runtime.

The following compatibility issue workarounds are enabled by compatibility mode:

- When verifying a peer certificate chain, a chain entry certificate revocation status will be ignored if the certificate is explicitly trusted as a CA in the local configuration (that is, listed in the ca certificate list).
- The peer certificate chain verification process will ignore any non-compatible **Key usage** extension value. For example, a peer certificate without authentication usage will be accepted in compatibility mode. RSA did not verify the **Key usage** extension values; OpenSSL does.

If you want to continue using the RSA BSafe implementation instead of OpenSSL, make sure you set up your environment so that shared modules from the **<Security Pack root>/legacy** folder have been loaded instead of the default modules (located in **<Security Pack root>**).

Java PSDK Implementation

The following provides information about the behaviour of TLS as implemented by the specified options in Java 8 and Java 7. Refer to Genesys Platform SDK documentation and the appropriate Oracle website (given below) for more information.

JAVA 8

No default value for **sec-protocol**
Possible values for **sec-protocol**: `SSLv23 SSLv3 TLSv1 TLSv11 TLSv12`
Default value for **tls-version**: `TLSv1.2` (the real default value is TLS)
Default value for **protocol-list**: `SSLv2Hello SSLv3 TLSv1 TLSv1.1 TLSv1.2`
Other values for **tls-version**: `SSLv3 TLSv1 TLSv1.1 TLSv1.2`
See Oracle JDK 8 for more details.

JAVA 7

No default value for **sec-protocol**
Possible values for **sec-protocol**: `SSLv23 SSLv3 TLSv1 TLSv11 TLSv12`
Default value for **tls-version**: `TLSv1` (the real default value is TLS)
Default value for **protocol-list**: `SSLv2Hello SSLv3 TLSv1 TLSv1.1 TLSv1.2`
Other values for **tls-version**: `SSLv3 TLSv1 TLSv1.1 TLSv1.2`
See Oracle JSE 7 for more details.

> **Warning**
> Java 7 supports TLS 1.1 and TLS 1.2. For client connections, neither version is enabled by default. For server connections, TLS 1.1 and TLS 1.2 are enabled by default.

.NET PSDK Implementation

Platform SDK-based applications that utilize the .NET Framework are configured similarly to what has been described for core applications running on the Windows operating system. Refer to the Release Notes of particular application to see if there are any special considerations.

What You Need

Before starting to configure your secure connections with TLS, you must have done the the following:

- Generated certificates, with associated private and public keys, and CRLs.
- Made certificates available for applications on hosts.
- Installed the Genesys Security Pack, if you are using native applications on Unix platforms.

Each of these requirements are described below.

TLS Certificates

TLS certificates must be generated and installed appropriately on any host that runs Genesys applications that utilize TLS secure connections. A certificate is generated and signed using a certification authority (CA) entity, which is able and authorized to issue certificates signed with its own name.

> **Important**
>
> Genesys strongly recommends that you use the same CA to generate all the certificates for your contact center environment.

The actual process of certificate generation in a specific environment is highly dependent on the security policies of your IT organization and tools used, and can, therefore, be different from the process described in this chapter. Genesys recommends that you consult with your network administrator before generating certificates for secure data exchange between Genesys components.

Certificates can be purchased from well-known certificate authorities, such as VeriSign. Certificates can be generated and self-signed on Linux using the OpenSSL tool, assisted by scripts distributed with the Genesys Security Pack. Windows Certificate Service can be used to generate and sign certificates. You may self-sign your certificates, or you Windows Certificate Service might already be used in your organization as a part of a chain of trusted authorities.

Generated and signed certificates must also be installed to be used by Genesys components. This procedure differs depending on the host operating system on which the certificates are installed.

> **Tip**
>
> TLS certificates can be stored and used in a number of different formats. Different TLS implementations use different certificate formats. Self-signed certificates, generated

> using OpenSSL and Genesys Security Pack, can be used across all supported Genesys implementations. For certificates obtained from other sources, you must confirm that their format is compatible with your target platforms.

Recommended Certificate Properties

When retrieving or generating a certificate, the following properties are recommended to ensure the connections using this certificate are as secure as possible:

[+] Show properties

- How is it signed by Issuer (CA)?
  - RSA2048+ Public Key (Encryption) Algorithm
  - SHA2 (SHA256+) Signature Hash Algorithm
- When does it expire?
  - It is critical to have a certificate rotated or replaced before it expires.
- Who is it for?
  - Subject and Subject Alternate Name (SAN)
  - Subject CN field (Common Name) of the DN (Distinguished Name).
  - Most often, it is the host machine’s domain.
  - SAN overrides Subject's CN field to allow for list of valid names; Genesys recommends that you set both.
- What is it for?
  - KeyUsage, ExtendedKeyUsage
  - Authentication of Server, Client, or both.
- Certificates that will be used by Genesys server applications must contain these extended attributes: `serverAuth` and `clientAuth`.
- Certificates that will be used by Genesys GUI applications must contain this extended attribute: `clientAuth`.

The following figure displays an example of a partial certificate, showing some of these recommended properties.

A sample TLS security certificate

Generating Certificates using OpenSSL and Genesys Security Pack

The OpenSSL toolkit can be used to set up a CA infrastructure and issue self-signed certificates. Use this method if certificates are to be utilized in a mixed environment, including Windows and Linux hosts, along with Java PSDK-based components.

Scripts distributed with Genesys Security Pack simplify CA creation and certificate generation. These scripts use the SHA-256 message digest algorithm by default, but allow fallback to SHA-1 if required. Genesys recommends that you do not use SHA-1.

Prerequisites

To generate certificates, you need an OpenSSL command line tool installed on the system. Obtain the most recent valid binary distribution from the OpenSSL official website.

To utilize Genesys Security Pack scripts, you need a set of standard GNU console utilities, including `bash`, `awk` and `touch`.

Generation Process

1. Create a CA directory in which CA files—scripts, configuration files, and generated certificates—will be stored.
2. Copy the **create_ca.sh** and **create_cert.sh** scripts from the installation package to the CA directory that you just created. Make sure that these scripts have executable permissions.
3. Run the **create_ca.sh** script from the **bash** shell by specifying the proper parameters (described in the table below) in the following command line:

```
create_ca.sh [-keySz KEY_SIZE] [-dgst DIGEST_ALGORITHM] [-time VALID_TIME] -CN
COMMON_NAME [-E EMAIL] [-OU ORG_UNIT] [-O ORGANIZATION] [-L LOCALITY] [-S STATE] [-C
COUNTRY]
```

The parameters are described in the following table:

[+] Show table

Parameter	Description
KEY_SIZE	(Optional) The size, in bits, of the CA private key. The default value is 2048 bits.
DIGEST_ALGORITHM	(Optional) Digest algorithm to use on certificate generation. The default value is sha256. Valid values are: sha1 (not recommended), sha224, sha256 (default), sha384, and sha512.
VALID_TIME	(Optional) The amount of time, in days, that the CA is valid. The default value is 365 days.
COMMON_NAME	(Mandatory) The name of the CA.
EMAIL	(Optional) The e-mail address of the person who is responsible for this CA.
ORG_UNIT	(Optional) The name of the organization unit that is responsible for this CA.
ORGANIZATION	(Optional) The name of the organization that is responsible for this CA.
LOCALITY	(Optional) The name of the city.
STATE	(Optional) The name of the state or region.
COUNTRY	(Optional) The two-letter abbreviation for the country.

For example:

```
create_ca.sh -CN “Basic Certification Authority” -E “youremail@yourdomain.com” -OU
“Department” -O “Genesys Telecommunication Labs” -L “Daly City” -S CA -C US
```

4. Generate certificates as required. To generate a certificate for a particular host computer, go to the CA directory in which the CA files are stored, and run the **create_cert.sh** script from the **bash** shell by specifying the parameters (see the following table) in the following command line:

```
create_cert.sh [-keySz KEY_SIZE] [-time VALID_TIME] [-dgst DIGEST_ALGORITHM] -host
HOST_NAME -CN COMMON_NAME [-IP HOST_IP] [-E EMAIL] [-OU ORG_UNIT] [-O ORGANIZATION] [-L
LOCALITY] [-S STATE] [-C COUNTRY] [-SANdomain "list"] [-SANip "list"]
```

The parameters are described in the following table:

[+] Show table

Parameter	Description
KEY_SIZE	(Optional) The size, in bits, of the host private key. The default value is 2048 bits.

Parameter	Description
VALID_TIME	(Optional) The amount of time, in days, that the certificate is valid. The default value is 100 days.
DIGEST_ALGORITHM	(Optional) Digest algorithm to use on certificate generation. The default value is sha256. Valid values are: sha1 (not recommended), sha224, sha256 (default), sha384, and sha512.
HOST_NAME	(Mandatory) The full name of the DNS host.
COMMON_NAME	(Mandatory) The name of the host.
IP HOST_IP	(Optional) The host IP.
ORG_UNIT	(Optional) The name of the organization unit.
ORGANIZATION	(Optional) The name of the organization.
LOCALITY	(Optional) The name of the city.
STATE	(Optional) The name of the state or region.
COUNTRY	(Optional) The two-letter abbreviation for the country.
SANip "list"	(Optional) Comma-separated Subject Alternative Name list. Should contain IPs.
SANdomain "list"	(Optional) Comma-separated Subject Alternative Name list. Should contain domain names.

For example:

```
create_cert.sh -host myHOST.domain1.domain2.com -CN myWorkstation
```

5. If you are installing certificates on any Java-based PSDK applications that cannot get the certificate information from Configuration Server, convert the private key file to PKCS #8 format. Use the following command:

```
convert_priv_key.sh -in INPUTFILE -out OUTFILE [-informat pfx|pkcs8|pkcs12|rsa] [-
outformat pkcs8|rsa] [-encrypt]
```

The parameters are described in the following table:

[+] Show table

Parameter	Description
INPUTFILE	Input private key filename.
OUTFILE	Output private key filename.

Parameter	Description
-informat	(Optional) Input private key format. For **.pem** private key files, use `rsa` Default is `rsa`.
-outformat	(Optional) Output private key format. For PSDK, use `pkcs8`. Default is `pkcs8`.
-encrypt	(Optional) Use password encryption for the resulting private key file. Password will be requested interactively.
ORG_UNIT	(Optional) The name of the organization unit.

Certificate Authority Files

After successful script execution, the following data structure is created:

- **ca_conf**—This directory contains the following files:
  - **ca_cert.pem**—The CA self-signed certificate file for UNIX (or any other implementation where pkcs#8 format is required)
  - **ca_cert.pfx**—The CA self-signed certificate file for Windows or Java (any other implementation where pkcs#12 format is required).

    > **Important**
    >
    > You must copy **ca_cert.pem** and **ca_cert.pfx** to each computer that will host Genesys components that might require secure data exchange, even if client certificates are not required.

  - **ca_priv_key.pem**—The CA private key.
    This file is used to sign all certificates that this CA issues. This file must be read-only, and it must be readable only by the CA administrator account.
  - **ca.db**—The internal CA database used by the OpenSSL toolkit.
  - **serial.num**—The internal CA file that contains the serial number of the next generated certificate. The serial number is a unique identifier of the certificate that the CA issues.
  - **ca.conf**—The internal CA configuration file.
- **repository**—This directory contains the files that this CA generates.

Host Certificate Files

After successful script execution, the following files are created in the repository directory:

- **<serial_#>_<host_name>_cert.pem**—The host certificate for UNIX.
- **<serial_#>.pem**—The auxiliary file for certificate generation for UNIX.
- **<serial_#>_<host_name>_priv_key.pem**—The host private key for UNIX.
- **<serial_#>_<host_name>_cert.pfx**—The PKCS (Public-Key Cryptography Standards) #12 file format,

including the private key and certificate for Windows.

where:

- **<serial_#>** is the serial number of the generated certificate. This number is unique for all certificates that this CA generates.
- **<host_name>** is the name of your host computer, which is the first part of the full DNS host name.

Certificate Revocation Lists

Revocation lists are maintained using plain text files when certificates are managed using OpenSSL or Genesys Security Pack. See Securing Connections Using TLS for details about how to configure Certificate revocation functionality for native applications that rely on Genesys Security pack. Note that for Windows native and .NET applications you must turn off certificate revocation and check if your certificates are produced as discussed in this section.

Generating Certificates Using Windows Certificate Services

This section describes how to create certificates using Windows Certificate Services. If necessary, you can also obtain a certificate from a remote machine. Use these certificates if you intend to run all of your applications on Windows. If you intend to run one or more applications that might require secure connections on UNIX, Genesys strongly recommends that you use OpenSSL to create your certificates. Make sure that certificate templates are properly configured for server and GUI applications to satisfy requirements for necessary certificate attributes, discussed above.

> **Important**
>
> The examples provided in this section assume that Windows Certificate Services have been installed and configured. For information about how to install and configure Windows Certificate Services, see the appropriate Windows documentation.

Generation Process

To generate certificates with Windows Certificate Services, do the following:

1. Generate a certificate on a computer that is running the Windows Server operating system, and that has Windows Certificate Services installed and configured. **[+] Show Steps**
.

a. Open a web browser, and enter the following URL:
http://<server-name>/certsrv
where **<server-name>** is the server that runs the Windows Server operating system, and on which Windows Certificate Services is installed and configured.

b. On the **Microsoft Certificate Services Welcome** page, click **Request a certificate**.

c. On the **Request a Certificate** page, click **Advanced certificate request**.

d. On the **Advanced Certificate Request** page, click **Create and submit a request to this CA**.

e. On the subsequent **Advanced Certificate Request** page, enter the following information:

- In the **Certificate Template** section, select an appropriate certificate template—for example, `MutualTLS2`.
- Enter the full **Name** of the DNS host as a Fully Qualified Domain Name.
- In the **Key Options** section:
  - Select **Create new key set**.
  - In the **Key Size** text box, specify the size of the key.
  - Select either **Automatic key container name** or **User specified key container name**, as appropriate.
  - Select **Mark key as exportable**.
- Click **Submit**.

After you submit the certificate request, the confirmation page appears, followed by the **Certificate Issued** page.

f. On the **Certificate Issued** page, click **Install this certificate**.

g. After you accept the system warning prompts that appear, the **Certificate Installed** page appears.

2. If you did not install the certificate in Step 1, retrieve and install it. **[+] Show Steps**

a. On the **Microsoft Certificate Services Home** page, click **View the status of a pending certificate request**.

b. Select the appropriate request from the list.
If the certificate request is approved, the **Certificate Issued** page appears.

c. Click **Install this certificate** to install the certificate.

3. Configure Microsoft Management Console (MMC). You can use MMC to manage certificates on a Windows platform. See Configuring MMC for Certificate Management.

4. Install the certificate and private key on the computer that hosts Genesys applications. If this computer is different from the one on which you generated the certificate, you must first export the certificate and its private key.

Certificate Revocation Lists

The Microsoft SChannel security provider retrieves Certificate Revocation List (CRL) information for the certificate being verified using a CRL distribution point (CDP) mechanism. If the CDP URL specified in the certificate is not reachable from the current host (or is blocked by a firewall or other network policy), the process of certificate verification might pause for a time interval specified in system settings (15 seconds by default). This might be because the CDP URL is accessible but resides on a slow network resource, or connection quality may be very low, resulting in significant delays when retrieving the CRL. This might lead to various undesired consequences. To avoid these

problems, Genesys strongly recommends that you use a local CDP in certificates and make sure all CDPs are accessible without any significant delay. You can also turn off CRL verification using Windows tools; refer to the documentation for your version of Windows.

Installing Certificates on Windows for Native Applications and Applications with a .NET Dependency

> **Important**
>
> For server applications, the certificates must be installed under the Local Computer account. For desktop applications, the certificates must be installed under the Current User account. For more information, see Managing Certificates in MMC.

To install the certificates, use the following procedure:

[+] Show steps

1. From the Windows Start menu, select **Run**, and then execute the `mmc` command to start the Microsoft Management Console (MMC).
2. On the left pane of MMC, click the **Certificates** folder. (If there is no **Certificates** folder on the left pane, see Managing Certificates in MMC.
3. Right-click the **Trusted Root Certification Authorities** folder, and select **All Tasks > Import** from the shortcut menu. This starts the Certificate Import Wizard.
4. On the first Wizard page, click **Next**.
5. On the **File to Import** page, type the full path to the **ca_cert.pfx** file or use **Browse** to navigate to the **ca_cert.pfx** that was created during CA setup. Make sure you select `All Files` for the **Files of type** option, and then click **Next**.
6. On the **Certificate Store** page, select **Place all certificates in the following store**. Make sure that the **Certificate store** text box is set to **Trusted Root Certification Authorities**. Click **Next**.
7. Click **Finish**.
8. On the left pane, click the **Certificates** folder.
9. On the left pane, right-click the **Personal** folder, and select **All Tasks > Import** from the shortcut menu. This starts the Certificate Import Wizard.
10. On the first Wizard page, click **Next**.
11. On the **File to Import** page, type the full name of the **<serial_#>_<host_name>_cert.pfx** file that was created during certificate generation. Click **Next**.
12. On the **Password** page, click **Next**. The host certificates in PKSC #12 format are generated with an empty password.
13. On the **Certificate Store** page, select **Place all certificates in the following store**. Make sure that the **Certificate store** text box is set to `Personal`. Click **Next**.

14. Click **Finish**.
15. Press **F5** to update the MMC view.
16. On the left pane, select **Certificates > Personal > Certificates**.
17. On the right pane, locate the imported certificate in the list, and double-click it.
18. In the **Certificate** dialog box, click the **Details** tab.
19. To view the certificate thumbprint, select **Thumbprint** from the list. The thumbprint, consisting of a string of hexadecimal digits, appears in the lower part of the dialog box.

Managing Certificates using MMC on Windows

You can use the Microsoft Management Console (MMC) to manage certificates on a Windows platform.

Configuring MMC for Certificate Management

To configure MMC for certificate management:

[+] Show steps

1. From the Windows **Start** menu, select **Run**, and execute the `mmc` command to start the Microsoft Management Console.
2. Select **File > Add/Remove Snap-in**.
3. In the **Add/Remove Snap-in** dialog box, click **Add**.
4. In the **Add Standalone Snap-in** dialog box, select Certificates from the list and click **Add**.
5. In the **Certificates snap-in** dialog box, select **Computer account** and click **Next**.

> **Important**
>
> To manage certificates for client applications, select **My user account**.

6. In the **Select Computer** dialog box, select **Local computer** and click **Finish**.
7. In the **Add Standalone Snap-in** dialog box, click **Close**.
8. In the **Add/Remove Snap-in** dialog box, click **OK**.
   The `Certificates` item is added under **Console Root** on the left pane.

You can save the MMC configuration in a file by selecting **File > Save As**.

Exporting Certificates

If the computer that is running Genesys applications is different from the one on which you generated the certificate, you must first export the certificate and its private key, as follows:

[+] Show steps

1. From the Windows **Start** menu, select **Run** and execute the `mmc` command to start the Microsoft Management Console.
2. Open your saved console configuration, or select **File > Add/Remove Snap-in**.
3. In the **Add/Remove Snap-in** dialog box, click **Add**.
4. In the **Add Standalone Snap-in** dialog box, select **Certificates** from the list and click **Add**.
5. In the **Certificates snap-in** dialog box, select **Computer account** and click **Next**.
6. In the **Select Computer** dialog box, select **Local computer** and click **Finish**.
7. In the **Add Standalone Snap-in** dialog box, click **Close**.
8. In the **Add/Remove Snap-in** dialog box, click **OK**.
   The **Certificates** item is added under **Console Root** on the left pane.
9. On the right pane, right-click the certificate in the list. Select **All Tasks > Export** from the shortcut menu to start the Certificate Export Wizard.
10. On the first Wizard page, click **Next**.
11. On the next page, select **Yes**, export the private key, and click **Next**.
12. On the **Export File Format** page, the only available export file format will be PKCS #12. Click **Next**.
13. On the **Password** page, type and confirm your password. Click **Next**.
14. On the **File to Export** page, specify the path and file name for your certificate. Click **Next**.
15. Click **Finish** to complete the export procedure.

Obtaining Certificates from a Remote Computer

To obtain a certificate from a remote computer:

[+] Show steps

1. From the Windows **Start** menu, select **Run**, and execute the `mmc` command to start Microsoft Management Console.
2. Select **File > Add/Remove Snap-in**.
3. In the **Add/Remove Snap-in** dialog box, click **Add**.
4. In the **Add Standalone Snap-in** dialog box, select **Certificates** from the list. Click **Add**.
5. In the **Certificates snap-in** dialog box, select **Computer account** and click **Next**.
6. In the **Select Computer** dialog box, select **Another computer** and either type the name of the remote target computer or click **Browse** to search for it. Click **Finish**.
7. In the **Add/Remove Snap-in** dialog box, click **OK**.
   A new snap-in item is added under **Console Root** in the main snap-in console window.
   You can browse for examples of all the certificates on the target computer, or you can view information about a particular certificate. Depending on the options that you select, MMC also enables you to remotely manage certificates on a target computer.
8. On the left pane, select **Certificates > Personal > Certificates**.

9. On the right pane, double-click the certificate in the list.
10. In the **Certificate** dialog box, click the **Details** tab.
11. Select **Thumbprint** from the list. The value, consisting of a string of hexadecimal digits, appears in the lower part of the dialog box.
12. Use the string of hexadecimal digits for the security configuration.

Installing Certificates on UNIX for Native Applications

Important

You must install the CA self-signed certificate file, **ca_cert.pem**, the certificate issued by this CA **<serial_#>_<host_name>_cert.pem**, and the certificate private key **<serial_#>_<host_name>_priv_key.pem** on each computer that hosts Genesys Server applications that might require secure data exchange.

1. Copy the **ca_cert.pem** file to the computer.
2. Copy the certificate and private key files to a local directory on the computer, as follows:

```
<serial_#>_<host_name>_cert.pem
 <serial_#>_<host_name>_priv_key.pem
```

3. Make sure that these files are readable by all Genesys applications that are running on this host computer.

Warning

The **<serial_#>_<host_name>_priv_key.pem** file contains critical security information. Make sure it can only be accessed by personnel authorized to work with this type of information.

Installing Certificates for Applications with Java dependency

If the Java application is using the file system to access certificates, follow the steps described for native applications using Genesys Security Pack, ensuring that **<serial_#>_<host_name>_cert.pem** and the private key file is converted to PKCS#8 format **<serial_#>_<host_name>_priv_key_NEW.pem**.

If the Java application is using a certificate store, for the instructions in Java documentation to install the certificates, ensuring that **<serial_#>_<host_name>_cert.pem** is converted to PKCS#12 (**pfx**) format . Typically you will:

1. Use the Java command `keytool –import x –file y –keystore z`.
2. If not already created, create a password for the keystore.

> **Tip**
> Genesys recommends that you use PEM files with PSDK Java for consistency.

Installing Genesys Security Pack

> **Important**
> The Security Pack on UNIX must be installed on each UNIX host computer on which Genesys native applications that use TLS are installed.

> **Tip**
> Along with libraries and scripts, a file README.version is installed as a part of Security Pack. If you have to report any security-related issues, you must also include the (cut-and-pasted) contents of this file. The information in this file contains complete Security Pack version information, which is difficult to obtain otherwise.

For information about the operating systems supported by the Security Pack on UNIX, refer to the *Genesys Supported Operating Environment Reference Guide*.

Genesys Security Pack is consistent with Federal Information Processing Standards (FIPS) starting in release 8.1.1. For information about these standards, and how to enable FIPS in Genesys software, refer to the FIPS section of this Guide.

To install Security Pack, complete the following steps:

1. Install the Security Pack on each UNIX host with which secure connections will be configured: **[+] Show steps**
   a. On the Security Pack product CD, in the **security_pack** directory, open the directory corresponding to your operating system, and locate the shell script called **install.sh**.
   b. Run this script from the command prompt by typing the following at the command line:

      ```
      sh install.sh
      ```

   c. When prompted, specify the host name of the computer on which you want to install the Security Pack.

d. Specify the full path to the directory in which you want to install the Security Pack. The installation process places the Security Pack in this directory. It also places the following scripts that are used by the OpenSSL tool in that directory:

- **create_ca.sh**—Creates the CA structure in which CA files and generated certificates are stored.
- **create_cert.sh**—Creates the certificates to use on UNIX and Windows computers.

> ### Important
>
> - Along with the necessary libraries and scripts, a file **README.<version>** is installed as part of Security Pack. Contents of this file must be reported (copied and pasted) with any security-related reported issues. This file contains complete Security Pack version information, which can be difficult to obtain otherwise.
> - For information about the installed files, see Certificate Generation and Installation.

When the installation process is finished, a message appears, indicating that the installation was successful.

2. Set the environment variable that corresponds to your operating system (see the following table), to the path to the Security Pack libraries.

Operating System	Environment Variable Name
AIX	LIBPATH
Linux	LD_LIBRARY_PATH
Solaris	LD_LIBRARY_PATH and/or LD_LIBRARY_PATH_64

> ### Warning
>
> Access permissions to the path to the Security Pack libraries, and the libraries themselves, must be set to enable Genesys applications to access them. If necessary, use the chown command to change the access permissions, as follows:
> `sudo chown <account name> -R <path to Security Pack libraries>`

Securing Connections Using TLS

This section describes securing only those connections that are defined using objects in Genesys Configuration Server, using both simple and mutual TLS. Those are typically connections between Genesys applications that utilize proprietary protocols, such as connections to Message Server, StatServer, and T-Server. Some connections have additional considerations discussed later.

Where to Set TLS Properties

Important

Configuration section names, configuration option names, and predefined option values are case-sensitive. Type them in Genesys Administrator exactly as they are documented in this Guide.

Tip

You must configure all required settings at each level (or leave that level empty). Do not configure, for example, **certificate** and **certificate-key**, or **certificate** and **sec-protocol**, on different levels. If you have to configure TLS at a particular level, add all options that have no default values and/or are otherwise needed for this TLS connection to work properly.

Server Listening Ports

You must specify a port as secure by setting its port properties. The Transport parameters of a secure port contains **tls=1**, and optionally **tls-mutual=1**. You will typically use Genesys Administrator to enable secure mode on the server port. This is a mandatory step to make the server port available for TLS, and must be set only on this (port) level. You can configure additional parameters at the Port level, or you can do that at higher levels (Application or Host levels).

For secure listening ports on the server to work properly, each must be provisioned with **certificate**, and for native applications on UNIX with **certificate-key** and **trusted-ca**. Additional options can also be specified, as required.

Tip

Not all listening ports of an application are configured using Port configuration objects.

> Some ports (such as a SIP Server SIP port) are configured differently. Refer to descriptions of individual connections later in this guide and to specific product-specific documentation to confirm how to configure server ports for each particular case.

Client Connections

When you link your application object with a server object in your configuration, and specify the secure port as the target for the connection, then and only then is this connection considered secure. Additional configuration might be needed for this application to operate properly. You will typically use Genesys Administrator to define the secure connection by establishing this link.

For client connections to operate in native applications on UNIX you must configure **trusted-ca**. If the application also supports mutual TLS, then **certificate** and, for native UNIX applications, **certificate-key**, is mandatory. Specify these and other parameters at connection level, or you can use upper levels to do so.

> ### Tip
> Not all client connections of an application are configured using objects. Refer to descriptions of individual connections (such as secure SIP Server connections to a remote trunk) later in this guide, and to product-specific documentation for more information about configuring client connections in each particular case.

Application Objects

Parameters for all TLS connections and listening ports can be specified for each application object (*Application-level*). For native UNIX applications that are clients of other secure servers, you must configure **trusted-ca**. For applications that have server ports, you must also configure **certificate** and, for native UNIX applications, **certificate-key**. Additional options might also be specified, as necessary, and will apply to all connections and ports not yet provisioned with their own parameters.

Some applications will need to be restarted for changes made on this level to take effect. In addition, connections and ports of applications that do not use the relevant configuration objects might require that you specify additional parameters. Refer to each type of connection for more details.

Host Objects

Parameters for TLS connections and listening ports for all applications running on a particular machine can be specified on the host object (*Host-level*) if no such parameters are already set on the Application or Connection/Port level. If any of the host's applications act as a server (that is, a listening port is in secure mode), then **certificate**, and, for native UNIX applications, **certificate-key** and **trusted-ca** are required. If only client applications (that is, applications that open client connections) are deployed on the host, and if mutual TLS connections are supported , **certificate** (and **certificate-key** and **trusted-ca** for native UNIX applications) are required. **trusted-ca** is mandatory for hosts on which native UNIX applications are running. Additional options can also be specified, as necessary, and apply to all connections/ports of all applications, where the options are

supported but the applications are not yet provisioned with their own parameters at lower levels.

Applications must be restarted to pick up changes made in the configuration at the host level, unless specified otherwise for each individual application. Connections and Ports of applications that do not use relevant configuration objects typically do not support configuring parameters on host level; refer to each type of connection for more details.

Using Genesys Administrator for Configuring Secure Parameters

In Genesys Administrator, configure settings in the **Network Security** section of the **Configuration** tab, noting the following:

- This is done most easily at the Host level, but there are mixed cases because components are often both clients and servers. When set at the Host level, you must specify that Host settings at to be used at the Application level.
- Settings made at the Connection/Port level are the most precise, but can be a lot of work in a large configuration.
- Because of different implementations (for example, native applications vs. PSDK Java-based), you might have to configure settings for specific cases at lower levels, even if default settings are done at a higher level.

Certificates for Windows Native Applications and Applications with .NET dependency

For Windows native applications, and applications that depend on the .NET framework, Genesys implementation of secure connections rely on the presence of certificates (an individual host certificate and a trusted certificate authority certificate) stored in Windows Certificate Storage on every host where Genesys applications are installed. With these certificates properly installed and managed (as described in TLS Certificates), the only configuration that is mandatory on the Genesys side for applications that are configured to open secure listening ports is the **certificate** option set to the Host certificate thumbprint, on the server at the Port, Application, or Host level. There is no mandatory provisioning of certificate-related options for an application to establish secure client connections when working in such environment. An application that does not use Port/Connection configuration objects might have different requirements.

Enabling a Secure Port on a Server

On Linux

1. In the **Listening Ports** field of the **Configuration** tab of the server Application object, select the port to be configured as secured, and click edit. The **Port Info** window opens.
2. On the **General** tab, choose Secured in the **Select Listening Mode** field. This automatically enters `tls=1` in the **Transport Parameters** field of the **Advanced** tab.
3. If you are setting up Mutual TLS, also add `tls-mutual=1` to the **Transport Parameters** field. All

parameters in this field must be separated by semi-colons (`;`).

4. Configure the secure port parameters at the appropriate level, as follows:
   - **Host level**:
     a. In the Host object on which the server is running, in the **Network Security** section of the **Configuration** tab, enter the absolute paths to the certificate, certificate key, and Trusted CA in the corresponding fields.
     b. In the server Application object, in the **Network Security** section of the **Configuration** tab, select *Host* in the **Certificate Source** field.
     c. Restart the server.
5. **Application level**: In the server Application object, in the **Network Security** section of the **Configuration** tab, do the following:
   a. Select *Application* in the **Certificate Source** field.
   b. Enter the absolute paths to the certificate, certificate key, and Trusted CA in the corresponding fields.
6. **Port level:** In the **Network Security** tab of the **Port Info** window, enter the absolute paths to the certificate, certificate key, and Trusted CA in the corresponding fields.

On Windows

1. Import the server Host certificate and the Trusted CA certificate into Windows Certificate Storage.
2. In the **Listening Ports** field of the **Configuration** tab of the Message Server Application object, select the port to be configured as secured, and click edit. The **Port Info** window opens.
3. On the **General** tab, choose **Secured** in the **Select Listening Mode** field. This automatically enters `tls=1` in the **Transport Parameters** field of the **Advanced** tab.
4. If you are setting up Mutual TLS, also add `tls-mutual=1` to the **Transport Parameters** field. All parameters in this field must be separated by semi-colons (`;`).
5. Configure the secure port parameters at the appropriate level, as follows:
   - **Host level:** In the Host object on which Message Server is running, in the **Network Security** section of the **Configuration** tab :
     a. Enter the thumbprint of the certificate in the **Certificate** field.
     b. In the server Application object, in the **Network Security** section of the **Configuration** tab, select *Host* in the `Certificate Source` field.
     c. Restart the server.
   - **Application level:** In the Message Server Application object, in the **Network Security** section of the **Configuration** tab:
     - Select *Application* in the `Certificate Source` field.
     - Enter the thumbprint of the certificate in the **Certificate** field.
   - At the **Port level**: On the **Network Security** tab of the **Port Info** window, enter the thumbprint of the certificate in the **Certificate** field.

Configuring Client side of Secure Connections

On Linux

1. In the **Connections** field of the **Configuration** tab of the Client Application object, select the connection to the server. The **Connections Info** window opens.
2. On the **General** tab, in the **ID** field, select the ID/number of the secured port on the server from the drop-down list.
3. Configure the parameters of the secure connection at the appropriate level, as follows:
   - **Host level**: In the Host object on which client is running, in the **Network Security** section of the **Configuration** tab, do the following:
     a. Enter the absolute path to the Trusted CA in the corresponding field.
     b. If you are configuring Mutual TLS, enter the absolute path to the certificate in the corresponding field.
     c. In the Client Application object, in the **Network Security** section of the **Configuration** tab, select `Host` in the `Certificate Source` field.
   - **Application level**: In the Server Application object, in the **Network Security** section of the **Configuration** tab, do the following:
     a. Select `Application` in the `Certificate Source` field.
     b. Enter the absolute path to the Trusted CA in the corresponding field.
     c. If you are configuring Mutual TLS, enter the absolute path to the certificate in the corresponding field.
   - **Connection level:** In the **Network Security** tab of the **Connection Info** window, do the following:
     a. Enter the absolute path to the Trusted CA in the corresponding field.
     b. If you are configuring Mutual TLS, enter the absolute path to the certificate in the corresponding field.

On Windows

1. Import the trusted CA certificate and, if using Mutual TLS, the client host certificate into Windows certificate storage.
2. In the **Connections** field of the **Configuration** tab of the Client Application object, select the connection to the server. The **Connections Info** window opens.
3. On the **General** tab, in the **ID** field, select the ID or number of the secured port on the server from the drop-down list.
4. If you are setting up Mutual TLS, configure the parameters of the secure connection at the appropriate level, as follows:
   - **Host level:** In the Host object on which the client Application is running, in the **Network Security**

section of the **Configuration** tab, do the following:

a. Enter the thumbprint of the certificate, imported in step 1, in the **Certificate** field.

b. In the client Application object, in the **Network Security** section of the **Configuration** tab, select `Host` in the `Certificate Source` field.

- **Application level:** In the client Application object, in the **Network Security** section of the **Configuration** tab, do the following:

  a. Select `Application` in the `Certificate Source` field.

  b. Enter the thumbprint of the certificate, imported in step 1, in the **Certificate** field.

- **Connection level:** On the **Network Security** tab of the **Connection Info** window, do the following :

  a. Enter the thumbprint of the certificate, imported in step 1, in the **Certificate** field.

Configuring Certificate Chains

Starting with release 8.1.3, Genesys Security Pack on UNIX supports security certificate chains, sending out the intermediate certificates along with the root certificate. The certificate container PEM format used to load certificates allows storing multiple certificates in a single PEM file.

However, note that the OpenSSL Security Pack requires the full path of local certificate chains to be specified in the configuration, starting with the root CA, even if some of the intermediate certificates are explicitly listed as trusted in the **CA** field.

PEM file Containing Multiple Certificates
A PEM file can contain multiple certificates, listed in order. Certificate info is stored in a **.pem** file, starting with `-----BEGIN CERTIFICATE-----` and ending with `-----END CERTIFICATE-----`. A PEM file can contain multiple certificates, each starting and ending with these tags.

To generate a multi-certificate PEM file from many simple PEM files, all source PEM files must be concatenated into the multi-certificate PEM file, for example:

```
cat cert_1.pem cert_2.pem cert_3.pem > cert_result.pem
```

Certificates should be listed in order from the end entity certificate, then intermediate certificates, up to the root CA certificate. Security Pack attempts to form a correct certification chain from the provided certificates by rearranging them, but it is best to provide the certificate chain already arranged.

Supplying Multiple Certificate PEM Files
The Security Pack enables configuration of multiple certificate storage PEM files that are to be loaded when forming its security credentials. PEM files must be listed in a comma-separated list. Each of the provided files can contain one or many PEM certificates. All the provided certificates will be loaded by Security Pack. The general guideline—to keep the certificate chain in order—still applies.

To enable this support, specify the multiple certificates in a comma-delimited list in the **Certificate**

field when configuring TLS. The certificates are sent in the order in which they are specified.

Configuring Certificate Revocation Lists (CRLs)

CRL on Windows

The Microsoft SChannel security provider retrieves Certificate Revocation List (CRL) information for the certificate being verified using a CRL distribution point (CDP) mechanism. If the CDP URL specified in the certificate is not reachable from the current host (or is blocked by a firewall or other network policy), the process of certificate verification might pause for a time interval specified in system settings (default is 15 seconds). This might be because the CDP URL is accessible but resides on a slow network resource, or because connection quality is very low, resulting in significant delays when retrieving the CRL. This in turn might lead to various undesired consequences. To avoid these problems, Genesys strongly recommends that you use a local CDP in certificates and make sure all CDPs are accessible without any significant delay. You can also turn off CRL verification using Windows tools; refer to the documentation for your version of Windows.

CRL with Security Pack on Unix

Set the **crl** configuration option (in the **[security]** section) at the same level (host, application, or connection) as the certificates it will contain, to allow the supporting Genesys component to verify certificates against a CRL. Specify the name of a .PEM file that contains one or more certificates defining the Certificate Revocation List. Refer to the *Framework Configuration Options Reference Manual* for a full description of this option.

Configuring Multiple Trusted CAs

Starting with release 8.0.0, Genesys Security Pack on UNIX supports multiple Trusted CA certificates for TLS connections. To enable this support, you must create a PEM file listing all of the certificates issued by the Trusted CAs, as follows.

Multiple Trusted CA PEM file

A multiple trusted CA PEM file is a file that contains information about two or more certificates, like this:

```
----BEGIN CERTIFICATE----
<encoded certificate-1 data>
----END CERTIFICATE----
----BEGIN CERTIFICATE----
<encoded certificate-2 data>
----END CERTIFICATE----
----BEGIN CERTIFICATE----
<encoded certificate-3 data>
----END CERTIFICATE----
```

To view the contents of a certificate PEM file, use the **openssl** utility, as follows:

```
openssl x509 -text -noout -in TrustedCA.pem
```

To create a multiple trusted CA certificate file, concatenate the individual certificate files, like this:

```
cat TrustedCA1.pem TrustedCA2.pem > multipleCAs.pem
```

Specify the full path to the multiple trusted CA certificate file in the **trusted-ca** field when configuring TLS. As security circumstances and requirements change, you can modify the file by adding and removing certificates, or completely replace it by specifying a single Trusted CA. Refer to the *Framework Configuration Options Reference Manual* for a full description of the **trusted-ca** option.

Check Certificate Host Matching

The **tls-target-name-check** option, set in the **[security]** section, enables a case-insensitive comparison of the TLS host name and the certificate's subject field during the authentication process. This option is transferred to a third-party library and describes whether it is necessary or not to check the names. To ensure a correct match, the client must be connecting to the server in exactly the same manner as stated in the certificate. The client must use the same FQDN, not an IP address.

Important

- Configure this option only on the client's side, at the same level where the certificate is configured.
- Security Pack 8.1 and earlier supported only a case-sensitive check of host names.

Refer to Validation (Authentication) by Receivers of Certificates for details on authentication of TLS-Server and TLS-Client identity, which includes a step to check for certificate-host matching.

If **tls-target-name-check** is set to `Host`, Genesys Security Pack uses the certificate SAN (if present, otherwise, the subject CN). Genesys Security Pack does not accept any wildcard symbols in the certificate SAN (or CN) field.

If the supporting Genesys component has a TLS-Client role for outbound connection and **tls-target-name-check=no**, then comparison of TLS-Server host name and the certificate's subject field is not made. This is used in cases when some phone devices or programs have the certificate without the host name in subject field, but have a MAC-address or other information.

By default, a comparison is not made, and the connection is allowed. Refer to the *Framework Configuration Options Reference Manual* for a full description of the **tls-target-name-check** option.

Sample Basic Configurations

This section contains examples of TLS configurations, both simple and mutual TLS.

Simple TLS on UNIX

Setting	Server Side	Client Side
Port set to:	**Listening Mode =** Secured	
Application to use:	Host TLS settings	Host TLS settings

Host set with:	`[security]`
	`trusted-ca=/etc/sec/ca/certauth.pem`
	`certificate=/etc/sec/certs/hostcert.pem`
	`certificate-key=/etc/sec/certs/privkey.pem`
		`[security]`
		`trusted-ca=/etc/sec/ca/certauth.pem`

Simple TLS on Windows

Setting	Server Side	Client Side
Port set to:	**Listening Mode =** Secured	
Application to use:	Host TLS settings	
Host set with:	`[security]` `certificate=89 A0 C1 D4 67 01 93 5D …`	

Simple TLS on Mixed Operating Systems

Setting	Server Side (*nix)	Client Side (Windows)
Port set to:	**Listening Mode =** Secured	
Application to use:	Host TLS settings	Host TLS settings

Host set with:	`[security]`
	`trusted-ca=/etc/sec/ca/certauth.pem`
	`certificate=/etc/sec/certs/hostcert.pem`
	`certificate-key=/etc/sec/certs/privkey.pem`	

Mutual TLS on UNIX

Setting	Server Side	Client Side
Port set to:	**Listening Mode =** Secured Advanced Transport Parameters for port	

	include **tls-mutual=1**	
Application to use:	Host TLS settings	Host TLS settings

Host set with:	`[security]`
	
	`certificate=/etc/sec/certs/servhostcert.pem`
	`certificate-key=/etc/sec/certs/servprivkey.pem`
	`trusted-ca=/etc/sec/ca/certauth.pem`
	`tls-mutual=1`
		`[security]`
		
		`certificate=/etc/sec/certs/clthostcert.pem`
		`certificate-key=/etc/sec/certs/cltprivkey.pem`
		`trusted-ca=/etc/sec/ca/certauth.pem`
		`tls-crl=/etc/sec/crl/crllist.pem`

Securing Core Framework Connections

This page contains information about securing connections between core Genesys servers, including Configuration Server, and Genesys clients. It also provides instructions for automatically securing a connection with Configuration Server when a client starts up.

Configuring TLS on Genesys Servers

For all server applications, configure a new or existing server port for secure connections. A port must be secure before you can configure a secure connection to that port.

Server-type applications that support Genesys TLS also support multiple server ports. This enables you to set up secure communications on only those connections for which security is considered necessary, rather than all server connections at the same time.

> **Tip**
>
> - If you intend to use the secure data exchange capabilities on connections to a specific server, Genesys recommends that you configure a new port for such secure connections, and that you leave the existing port intact for connections that do not require security .while protecting that port using network security.
> - If you want to use mutual TLS, set the **tls-mutual** configuration option to 1 .on the server side, at the same level (host, application, port) as your certificate.

Configuring TLS on Configuration Server

To configure TLS on Configuration Server, do the following:

1. In the Configuration Server Application object, configure an Auto-Detect port to enable clients to connect securely to Configuration Server.
2. If you want to use mutual TLS, set the **tls-mutual** option to 1 at the Auto-Detect port level.
3. Assign a certificate to be used by Configuration Server at the Auto-Detect port level.

Configuring TLS on Other Genesys Servers

To configure a secure port on a TLS server application, do the following:

1. In Genesys Administrator, click the **Provisioning** tab and navigate to the folder containing the server

application.

2. Select the server application and click the **Configuration** tab.
3. In the **Server Info** section, click **Add** in the **Listening Ports** table. The **Port Info** dialog box appears.
4. In the **Port Info** dialog box, on the **General** tab:
   - In the **ID** box, enter the port ID.
   - In the **Port** box, enter the number of the new port.
   - In the **Connection Protocol** box, select the connection protocol, if necessary.
   - In the **Select Listening Mode** box, select **Secured**.
   - Click **OK**.
5. Click **Save** to save the new configuration.
6. Assign a certificate to be used by this server. Genesys recommends that you assign the certificate on the host level, but you can assign it at the application level or port level if required.
7. If you want to use mutual TLS, set the **tls-mutual** option to 1 at the same level as the certificate you assigned to the server in the previous step.

Configuring TLS on Genesys Clients

After you configure your server applications so that they have secure ports, you must change the configuration of your client applications, so that they connect to these ports. Remember that you must do this only for the connections on which extra measures are necessary to protect the data that is transferred between the Genesys applications.

Configuring a Secure Bootstrap Connection to Configuration Server

Client applications must be provisioned with their secure connection parameters at application or host level when connecting to auto-upgrade port of configuration server. Configuration Server validate client configuration and send it to client application as a part of auto-upgrade procedure. Client apply receivied parameters. To configure a secure bootstrap connection to Configuration Server, follow the appropriate procedure based on the type of client application.

Server Applications

Prerequisite:

- A trusted CA certificate is configured and accessible by the client application.

Procedure:

1. Verify that the command-line parameters in the initial startup **.bat** or **.sh** file (if it exists) are set to connect to the Autodetect port.

2. In the configuration of the client server Application object, do the following:
    a. Configure a connection to the Autodetect port of Configuration Server.
    b. If mutual TLS (**tls-mutual=1**) is configured on the server side, obtain a client certificate and configure it accordingly.
3. If you need to install a new instance of the client application, use the appropriate Installation Package and provide the Application object you just configured.
4. Start the client server application using Solution Control Server and/or a startup batch file, if applicable.

User Interface Applications

For User Interface applications that utilize a login requiring host and port information, do the following:

Prerequisite:

- A trusted CA certificate is configured and accessible by the client application.

Procedure:

1. In the configuration of the client server Application object, do the following:
    a. Configure a connection to the Autodetect port of Configuration Server.
    b. If mutual TLS (**tls-mutual=1**) is configured on the server side, obtain a client certificate and configure it accordingly.
2. If you need to install a new instance of the client application, use the appropriate Installation Package and provide the Application object you just configured.
3. Start the client application by entering the Configuration Server Autodetect port in the login dialog box.

Configuring a Secure Client Connection to Other Genesys Servers

To configure a secure connection of a client application to other Genesys servers, do the following:

1. Click the **Configuration** tab of the client application.
2. Select a server to which you need to make a secure connection, and click **Edit**.
3. In the **Connection** table in the **General** section, click **Add**, and enter the properties of the secure port that you created for the server during the previous configuration steps. The read-only **Connection Type** property indicates that this connection is secure.
4. If you are configuring mutual TLS, assign the certificate, private key, and Trusted CA to this application.
5. Click **OK**.
6. Click **Save** to save the new connection configuration.

The next time this application starts, it will connect to the server over a secure connection.

Securing Local Control Agent Connections

To secure connections using TLS, you must configure TLS on the two connecting parties. This page describes how to install TLS on Local Control Agent (LCA) and on Solution Control Server (SCS) clients. It also describes how to secure the connections between LCA and SCS, and between Genesys Deployment Agent and its clients.

Configuring TLS on LCA Server

To configure TLS on the LCA server, do the following:

Prerequisites:

- Generate and install a certificate and its CA on the LCA Host computer.
- Have the certificate information available to you.

Procedure:

1. In the LCA configuration file, **lca.cfg**:
   a. If it does not already exist, add the new section **[security]**.
   b. In this section:
      - Use the **upgrade** option to designate this port as secure.
      - Specify the certificate parameters that will be used to secure the connections. The actual parameters are determined by the type of operating system you are using, as follows:
        - If you are using *nix, set the **certificate**, **certificate-key**, and **trusted-ca fields**.
        - If you are using Windows, set only the **certificate** field to the thumbprint value of the certificate.
      - If you are using multiple TLS, set the **tls-mutual** option to 1.

   For more information, see Sample Configuration Files for LCA.
2. In the annex of the host on which LCA is running, in the **[security]** section, set the **lca-upgrade** option to 1 (`true`).
3. Restart the host machine and LCA.

Sample Configuration Files for LCA

The following are sample configuration files for LCA in which the values of the `upgrade` options of the `security` section are set.

[+] Show Files

On *nix:

```
[log]
verbose=standard
standard=stdout, logfile
[security]
upgrade=1
tls-mutual=1          #only if using mutual TLS
certificate=/home/tech/sec/aix_cert.pem
certificate-key=/home/tech/sec/aix_priv_key.pem
trusted-ca=/home/tech/sec/techpubs_dco.pem
```

On Windows:

```
[log]
verbose=standard
standard=stdout, logfile
[security]
upgrade=1
tls-mutual=1          #only if using mutual TLS
certificate=f4 15 c5 d8 f3 17 b4 f9 4f d2 37 30 56 4b 07 ec b1 14 75 ee
```

Securing Connections between LCA and Solution Control Server

A secure connection between Local Control Agent (LCA) and Solution Control Server (SCS) is optional, and requires that you modify the LCA configuration file and the Host object on which LCA is running. Note that if TLS is configured between LCA and SCS on a host machine, LCA uses TLS only on the connection with SCS. Other applications running on the host are connected through TCP.

Use the **upgrade** option (in the **lca.cfg** file) and the **lca-upgrade** option (in the **[security]** section of the annex of the Host object) to configure secure data exchange using TLS on connections between LCA and SCS. These options are configured on the Host computer on which LCA and SCS are running, and where the certificate information is available to you. For more information about these two options, refer to the *Framework Configuration Options Reference Manual*.

Configuring TLS on SCS Clients

To configure TLS on SCS Clients, do the following:

Prerequisites:

- Generate and install a certificate and its CA on the SCS Host computer.
- Have the certificate information available to you.

Procedure:

In the **Options** tab of the SCS application object or the host object on which SCS is running::

1. If it does not already exist, add the new section **[security]**.
2. In this section:

- Specify the certificate parameters that will be used to secure the connections. The actual parameters are determined by the type of operating system you are using, as follows:
  - If you are using *nix, set the **certificate**, **certificate-key**, and **trusted-ca fields**.
  - If you are using Windows, set only the **certificate** field to the thumbprint value of the certificate.

3. If you are using multiple TLS, set the **tls-mutual** option to 1.

For more information, see Sample Configuration of security Section.

Sample Configuration of security Section

The following are sample configurations of the **[security]** section:

[+] Show Samples

On *nix:

```
[security]
tls-mutual=1          #only if using mutual TLS
certificate=/home/tech/sec/aix_cert.pem
certificate-key=/home/tech/sec/aix_priv_key.pem
trusted-ca=/home/tech/sec/techpubs_dco.pem
```

On Windows:

```
[security]
tls-mutual=1          #only if using mutual TLS
certificate=f4 15 c5 d8 f3 17 b4 f9 4f d2 37 30 56 4b 07 ec b1 14 75 ee
```

Configuring TLS on Genesys Deployment Agent

To configure TLS on Genesys Deployment Agent:

Prerequisites:

- Generate and install a certificate and its CA of the Genesys Deployment Agent Host computer.
- Have the certificate information available to you.

Procedure:

1. In the Genesys Deployment Agent configuration file, **gda.cfg**:
   a. If it does not already exist, add the new section **[security]**.
   b. In this section:
      - Use the **upgrade** option to designate this port as secure.
      - Specify the certificate parameters that will be used to secure the connections. The actual parameters are determined by the type of operating system you are using, as follows:

- If you are using *nix, set the **certificate**, **certificate-key**, and **trusted-ca fields**.
- If you are using Windows, set only the **certificate** field to the thumbprint value of the certificate.

- If you are using multiple TLS, set the **tls-mutual** option to 1.

For more information, see Sample Configuration Files for Genesys Deployment Agent.

2. In the annex of the host on which Genesys Deployment Agent is running, in the **[security]** section, set the **gda-tls** option to 1 (`true`).
3. Restart Genesys Deployment Agent.

Sample Configuration Files for Genesys Deployment Agent

The following are sample configuration files for Genesys Deployment Agent, in which the values of the transport options in the **[security]** section are set. The settings are the same as for any TLS setup, except that they are set in the configuration file instead of the configuration objects.

[+] Show files

On *nix:

```
[log]
verbose=standard
standard=stdout, gdalog
[security]
tls=1
tls-mutual=1          #only if using mutual TLS
certificate=/home/tech/sec/aix_cert.pem
certificate-key=/home/tech/sec/aix_priv_key.pem
trusted-ca=/home/tech/sec/techpubs_dco.pem
```

On Windows:

```
[log]
verbose=standard
standard=stdout, gdafile
[security]
tls=1
tls-mutual=1          #only if using mutual TLS
certificate=f4 15 c5 d8 f3 17 b4 f9 4f d2 37 30 56 4b 07 ec b1 14 75 ee
```

Securing Connections Between Genesys Deployment Agent and its Clients

A secure connection between Genesys Deployment Agent and its clients is optional, and requires that you modify the Genesys Deployment Agent configuration file and the Host object on which Genesys Deployment Agent is running.

Use the **tls** and **gda-tls** configuration options to configure secure data exchange using TLS on connections between Genesys Deployment Agent and its clients. Refer to the *Framework Configuration Options Reference Manual* for detailed descriptions about these configuration options.

Secure Network Logging Connections

When configuring secure connections for Centralized Logging, the Message Server designated as the Centralized Log Message Server acts as the server and opens a secure port to which its clients connect. Configuration Server and Solution Control Server act as the clients and connect to that secure port. Both Simple TLS and Mutual TLS are supported on these connections are supported.

> **Warning**
> For each component, the parameters for the secure connection can be configured at any level (Host, Application, or Port (server-side) or Connection (client side)). Be sure to configure all related security options at the same level as the certificate.

Using Genesys Administrator, use the following instructions to set up the connections.

Configuring a Secure Port on Message Server

> **Warning**
> Message Server must configure its default port with the security settings. TLS configuration of secondary listening ports on Message Server is not supported.

On Linux

1. In the **Listening Ports** field of the **Configuration** tab of the Message Server Application object, select the port to be configured as secure, and click edit. The **Port Info** window opens.
2. On the **General** tab, choose Secured in the **Select Listening Mode** field. This automatically enters `tls=1` in the **Transport Parameters** field of the **Advanced Tab**.
3. Configure the parameters of the secure port at the appropriate level, as follows:
   - **Host level**:
     a. In the Host object on which Message Server is running, in the **Network Security** section of the **Configuration** tab, enter the absolute paths to the certificate, certificate key, and Trusted CA in the corresponding fields. For example:
        - Certificate: `/root/Desktop/GENESYS_COMP/certificate/172.24.131.162.pem`
        - Certificate Key: `/root/Desktop/GENESYS_COMP/certificate/172.24.131.162.pem`
        - Trusted CA: `/root/Desktop/GENESYS_COMP/certificate/cert_auth.pem`

b. In the Message Server Application object, in the **Network Security** section of the **Configuration** tab, select `Host` in the **Certificate Source** field.

- **Application level**: In the Message Server Application object, in the **Network Security** section of the **Configuration** tab, do the following:

  a. Select `Application` in the **Certificate Source** field.

  b. Enter the absolute paths to the certificate, certificate key, and Trusted CA in the corresponding fields. For example:

  - Certificate: `/root/Desktop/GENESYS_COMP/certificate/172.24.131.162.pem`
  - Certificate Key: `/root/Desktop/GENESYS_COMP/certificate/172.24.131.162.pem`
  - Trusted CA: `/root/Desktop/GENESYS_COMP/certificate/cert_auth.pem`

- **Port level:** In the **Network Security** tab of the **Port Info** window, enter the absolute paths to the certificate, certificate key, and Trusted CA in the corresponding fields. For example:
  - Certificate: `/root/Desktop/GENESYS_COMP/certificate/172.24.131.162.pem`
  - Certificate Key: `/root/Desktop/GENESYS_COMP/certificate/172.24.131.162.pem`
  - Trusted CA: `/root/Desktop/GENESYS_COMP/certificate/cert_auth.pem`

4. If you are setting up Mutual TLS, configure it at the same level as you configured the server certificate in the previous step. Specifically:
   - If the server certificate is configured at the host level, set `tls-mutual=1` in the Annex tab of the Host object where the server application is installed.
   - If the server certificate is configured at the application level, set `tls-mutual=1` in the **Options** tab of the server application.
   - If the server certificate is configured at the port level, set `tls-mutual=1` in the **Transport Protocol Parameters** field of the **Advanced** tab of the server port. All parameters in this field must be separated by semi-colons (`;`).
5. Restart Message Server.

On Windows

1. Import the Message Server host certificate and the trusted CA certificate into Windows certificate storage.
2. In the **Listening Ports** field of the **Configuration** tab of the Message Server Application object, select the port to be configured as secured, and click edit. The **Port Info** window opens.
3. On the **General** tab of the **Port Info** window, choose `Secured` in the **Select Listening Mode** field. This automatically enters `tls=1` in the **Transport Parameters** field of the **Advanced Tab**.
4. Configure the parameters of the secure port at the appropriate level, as follows:
   - **Host level:**

     a. In the Host object on which Message Server is running, in the **Network Security** section of the **Configuration** tab, enter the thumbprint of the certificate in the **Certificate** field. The thumbprint is a string of hexadecimal characters; for example, `61 cc b8 76 3c ap 2a ff 00 13 98 6d 8e 51 7c 41 47 be f5 ee`.

b. In the Message Server Application object, in the **Network Security** section of the **Configuration** tab, select `Host` in the **Certificate Source** field.

- **Application level:** In the Message Server Application object, in the **Network Security** section of the **Configuration** tab:

  a. Select `Application` in the **Certificate Source** field.

  b. Enter the thumbprint of the certificate in the **Certificate** field. The thumbprint is a string of hexadecimal characters; for example, `61 cc b8 76 3c ap 2a ff 00 13 98 6d 8e 51 7c 41 47 be f5 ee`.

- **Port level**: On the **Network Security** tab of the **Port Info** window, enter the thumbprint of the certificate in the **Certificate** field. The thumbprint is a string of hexadecimal characters; for example, `61 cc b8 76 3c ap 2a ff 00 13 98 6d 8e 51 7c 41 47 be f5 ee`.

5. If you are setting up Mutual TLS, configure it at the same level as you configured the server certificate in the previous step. Specifically:
   - If the server certificate is configured at the host level, set `tls-mutual=1` in the Annex tab of the Host object where the server application is installed.
   - If the server certificate is configured at the application level, set `tls-mutual=1` in the **Options** tab of the server application.
   - If the server certificate is configured at the port level, set `tls-mutual=1` in the **Transport Protocol Parameters** field of the **Advanced** tab of the server port. All parameters in this field must be separated by semi-colons (`;`).
6. Restart Message Server.

Configuring TLS on Solution Control Server

On Linux

1. In the **Connections** field of the **Configuration** tab of the Solution Control Server (SCS) Application object, select the connection to Message Server. The **Connections Info** window opens.
2. On the **General** tab, in the **ID** field, select the ID of the secured port on Message Server from the drop-down list.
3. Configure the parameters of the secure connection at the appropriate level, as follows:
   - **Host level**: In the Host object on which SCS is running, in the **Network Security** section of the **Configuration** tab, do the following:

     a. Enter the absolute path to the Trusted CA in the corresponding field. For example:

     ```
     /root/Desktop/GENESYS_COMP/certificate/cert_auth.pem
     ```

     b. If you are configuring Mutual TLS, also enter the absolute path to the certificate and certificate key in the respective fields. For example:

     ```
     Certificate field: /root/Desktop/GENESYS_COMP/certificate/172.24.131.162.pem
     ```

```
Certificate Key field: /root/Desktop/GENESYS_COMP/certificate/172.24.131.162.pem
```

c. In the SCS object, in the **Network Security** section of the **Configuration** tab, select `Host` in the **Certificate Source** field.

- **Application level**: In the Message Server Application object, in the **Network Security** section of the **Configuration** tab, do the following:

  a. Select `Application` in the **Certificate Source** field.

  b. Enter the absolute path to the Trusted CA in the corresponding field. For example:

  ```
  /root/Desktop/GENESYS_COMP/certificate/cert_auth.pem
  ```

  c. If you are configuring Mutual TLS, also enter the absolute path to the certificate and certificate key in the respective fields. For example:

  ```
  Certificate field: /root/Desktop/GENESYS_COMP/certificate/172.24.131.162.pem
  Certificate Key field: /root/Desktop/GENESYS_COMP/certificate/172.24.131.162.pem
  ```

- **Connection level:** In the **Network Security** tab of the **Connection Info** window, do the following:

  a. Enter the absolute path to the Trusted CA in the corresponding field. For example:

  ```
  /root/Desktop/GENESYS_COMP/certificate/cert_auth.pem
  ```

  b. If you are configuring Mutual TLS, also enter the absolute path to the certificate and certificate key in the respective fields. For example:

  ```
  Certificate field: /root/Desktop/GENESYS_COMP/certificate/172.24.131.162.pem
  Certificate Key field: /root/Desktop/GENESYS_COMP/certificate/172.24.131.162.pem
  ```

On Windows

1. Import the trusted CA certificate and, if using Mutual TLS, the SCS host certificate into Windows certificate storage.
2. In the **Connections** field of the **Configuration** tab of the Solution Control Server (SCS) Application object, select the connection to Message Server. The **Connections Info** window opens.
3. On the **General** tab, in the **ID** field, select the ID of the secured port on Message Server from the drop-down list.
4. If you are setting up Mutual TLS, configure the parameters of the secure connection at the appropriate level, as follows:

   - **Host level:**

     a. In the Host object on which SCS is running, in the **Network Security** section of the **Configuration** tab, enter the thumbprint of the certificate, imported in step 1, in the **Certificate** field. The thumbprint is a string of hexadecimal characters; for example, `61 cc b8 76 3c ap 2a ff 00 13 98 6d 8e 51 7c 41 47 be f5 ee`.

     b. In the SCS Application object, in the **Network Security** section of the **Configuration** tab, select *Host* in the `Certificate Source` field.

   - **Application level:** In the SCS Application object, in the **Network Security** section of the **Configuration** tab, do the following:

a. Select *Application* in the `Certificate Source` field.

b. Enter the thumbprint of the certificate, imported in step 1, in the **Certificate** field. The thumbprint is a string of hexadecimal characters; for example, `61 cc b8 76 3c ap 2a ff 00 13 98 6d 8e 51 7c 41 47 be f5 ee`.

- **Connection level:** On the **Network Security** tab of the **Connection Info** window, enter the thumbprint of the certificate, imported in step 1, in the **Certificate** field. The thumbprint is a string of hexadecimal characters; for example, `61 cc b8 76 3c ap 2a ff 00 13 98 6d 8e 51 7c 41 47 be f5 ee`.

Configuring TLS on Configuration Server

> **Important**
>
> This section contains instructions for configuring TLS on Configuration Server acting as a client of the the Centralized Log. For other connections involving Configuration Server, refer to instructions for the appropriate connection elsewhere in this guide.

TLS is configured on the Configuration Server that is acting as a client of the Centralized Log Message Server in the same way as for Solution Control Server, above.

However, in a distributed Configuration Server environment, note the following:

- TLS must not be configured on the master Configuration Server until after the server has been started for the first time without network logging (that is, starting based on its configuration file).
- TLS can be configured on Configuration Server Proxy at any time.

Configuring TLS on a Centralized Log Client

TLS is installed on a Centralized Log Client in the same way as for Solution Control Server, with two exceptions:

- The client establishing a connection to the Centralized Log Message Server must configure the certificate information at the Application or Connection level.
- In the **Advanced** tab of the **Connection Info** window, do not configure **tls=1**.

Securing High Availability Connections

This section describes how to configure secure connections between primary and backup servers in a high-availability (HA) configuration.

See Supporting Components for information about components that support HA configurations. For information about setting up an HA environment for these Genesys components, see the corresponding product documentation.

Securing Connection Between Configuration Servers Configured as HA Pair

Configuration Servers can only communicate securely when both configured with their default ports in auto-upgrade mode. Follow the instructions for configuring Configuration Server (for port) and Configuring client of Configuration Server (for certificates and other parameters when Configuration server instance is playing client role). Configuration of both instances in HA pair must be identical.

Securing Connection Between Genesys Servers Configured as HA Pair

The HA synchronization connection is configured by selecting the **HA sync** check box in the **Port Info** dialog box of a specific port. This indicates that the port will be used by the former primary server to connect to the new primary server after a failover. If the **HA sync** check box is not selected, the former primary server will connect to the default port of the new primary server.

Important

If the security certificate is configured on the Connection level of the Primary application server and the Backup application server is configured, then the security certificate parameters are propagated automatically to the Backup server's Application Configuration object. However, if the Primary and Backup application servers are located on different hosts, ensure that the correct security certificate parameters are applied manually in the Backup application server's object .

Important

Genesys does not support using the ports with the port-level assigned certificates for an HA synchronization connection between redundant servers. The secure connection

should be configured on a host or application level instead.

To configure TLS on each component in the HA pair:

1. In the **Server Info** section on the **Configuration** tab of the properties of both the primary and backup servers in a redundant pair, create a new port with the same **ID**, and with **Select Listening Mode** set to Secured.

   > **Warning**
   > When multiple ports are configured for a server in a Hot Standby redundancy pair, their **ID**s and the **Select Listening Mode** settings of the primary and backup servers must match respectively.

2. In the **Port Info** dialog box of each server, click **OK** to save the new configuration. Then, in the **Configuration** tab of each, click **Save**.
3. In the **Listening Ports** table of each server, select the port that you just created, and click **Edit**.
4. In the **Port Info** dialog box, select the **HA sync** check box, and click **OK**.
5. Click **Save & Close**, **Save**, or **Save & New**, as appropriate, to save the configuration changes.

Securing Application Protocol Connections

This topic describes how to secure connections between components that connect via a specific protocol.

Securing Connections Between Distributed Solution Control Servers

To have secure connections between Solution Control Servers in Distributed mode, the connections between all of the Solution Control Servers must be secured. To accomplish this, on each SCS in the distributed configuration, secure the port to which the other distributed servers connect.

Refer to Configuring TLS on Other Genesys Servers for details about how to secure connections between Solution Control Servers.

Session Initiation Protocol (SIP)

To secure connections using SIP, refer to the Genesys SIP Server Deployment Guide.

Other Protocols

To secure connections with components using protocols in this section or otherwise described in this document, refer to the Deployment Guide for the appropriate component.

Configuring Secure Connections to Java/PSDK-Based Applications

Secure connections to Java/PSDK-based applications (such as Universal Contact Server) running on UNIX are configured in the same way as described in Other Genesys Servers, with one exception:

- If you are running Java/PSDK-based applications on the same host as C++-based applications, do not use the host certificate to secure data exchange at the application or port level. Although both types of applications use a PEM file for their private key, the internal format differs—Java/PSDK uses PKCS#8 and C++ uses RSA. Instead, use the application's certificate to enable secure data exchange on all secure ports of that application.

Advanced TLS

This topic contains additional information about TLS.

Tuning Protocol Version Availability

In release 8.5.1, as part of the transition to OpenSSL from RSA Bsafe, the behavior of the **sec-protocol** option has been modified.

> **Important**
>
> Refer to Security Pack 8.5.100.25 for information on OpenSSL version 1.1.1g and later, TLS 1.3, and SAN certificate.

The availability of a particular protocol setting in **sec-protocol** strongly depends on the actual component version.

Generally, the protocol versions currently available are as follows:

- On UNIX and Linux, TLS 1.3 is the highest available protocol with the OpenSSL Security Pack; TLS 1.1 with the RSA Security Pack.
- On Windows, refer to Microsoft documentation for the list of supported TLS versions for particular Windows deployment. Genesys recommends that you explicitly enable the desired protocol version in the Windows registry; refer to the Windows document TLS/SSL Settings for more information about enabling and disabling protocols in the Windows registry.

> **Warning**
>
> Genesys components use the Windows implementation of TLS on Windows platforms, and therefore Windows settings take precedence over **sec-protocol** settings. Genesys software is unable to use a protocol version if it is disabled on the Windows operating system level.

sec-protocol

Valid Values: SSLv23, TLSv12, TLSv13 or an empty string

Default Value: an empty string

Specifies the protocol used by the component to set up secure connections:

- SSLv23 - The highest TLS protocol version supported by both sides of communication, from TLS 1.1 and up (remains for backward compatibility, not recommended for new deployments).
- empty string - the default Security Pack settings (currently the highest TLS version supported by both sides from 1.2 upwards).
- TLSv12 - TLS version 1.2.
- TLSv13 - TLS version 1.3.

The supported protocol version modes can be categorized as one of two types:

- **strict** mode— TLSv12 and TLSv13 are the strict protocol version modes. These settings can be used to enforce a specific protocol version. The connection will not be established if the remote server does not accept the enforced protocol version.
- **compatibility** mode—SSLv23 and the default mode, are compatible with all modes from TLSv1.1 or TLSv1.2 up to and including TLSv13, and will connect with the highest mode offered by the other side of the TLS connection.

Tuning Available Cipher Lists for TLS v1.2

Normally, the set of available ciphers is provided by your InfoSec, and can be configured to the preferences of the user. The **cipher-list** configuration option allows the supporting Genesys component to select a list of cipher suites used in TLSv 1.2 and lower. This option is transferred to a third-party library and describes the set of possible cipher suites.

Cipher List Formatting Rules

> **Important**
>
> This section describes cipher list format for an application using the Genesys common library. If you are configuring a cipher list for the PSDK-based application, refer to the *Platform SDK Developer's Guide* for the proper format, and more information about cipher lists in PSDK.

For applications using the Genesys common library , the cipher list string is a list of cipher operations. Each operation consists of an optional operator character followed by a name. See OpenSSL cipher commands for more information. Cipher list strings must conform to the following formatting rules:

[+] Show rules

- The name is either a valid cipher name or a cipher alias. Valid names contain the characters a-z, A-Z, 0-9, and - (dash).
- List separator characters are used to separate the names and aliases in the list. A list separator character must be a colon.
- Multi-part names are joined with +.

- The character ! appearing immediately after a separator indicates a kill operation. The cipher following the character becomes unavailable.
- The character + appearing immediately after a separator indicates an order operation. This moves the active cipher to the current position in the list of ciphers.
- The character - appearing immediately after a separator indicates a delete operation. The cipher following the character becomes inactive. The cipher remains available for further operations.
- A non-operator character appearing immediately after a separator indicates an add operation. If the cipher following the character is not currently active, the cipher is added as an active cipher to the end of the list of available ciphers.

All operations occur in the order in which they appear in the list. If the cipher corresponding to a name (or part of a name, for multi-part names) is not available in the library, it is ignored during loading. In this situation, no error message is logged.

Aliases

Ciphers also have aliases. The following table details the primary cipher aliases.

[+] Show table

Alias	Description
kRSA, kDHr, kDHd, kEDH	Key exchange types
aRSA, aDSS, aNULL, aDH	Authentication
DES, 3DES, RC4, RC2, eNULL	Ciphers
MD5, SHA1	Message digests

Groups of commonly-used ciphers also have aliases. This enables multiple aliases to be specified easily. The following table details the cipher group aliases.

[+] Show table

Alias	Description
SSLv2	All SSLv2 ciphers
SSLv3	All SSLv3 ciphers
EXP	All export ciphers
LOW	All low strength ciphers (no export ciphers, normally single DES)
MEDIUM	128-bit encryption
HIGH	Triple DES with key lengths larger than 128 bits, and some cipher suites with 128-bit keys

Aliases can also be joined in a colon-separated list to specify the ciphers to add, move, or delete.

Example

The following is an example of a cipher string:

```
!ADH:RC4+RSA:HIGH:MEDIUM:LOW:EXP:+SSLv2:+EXP
```

This cipher string is interpreted in the following sequence:

1. Do not consider any ciphers that do not authenticate.
2. Use ciphers that use RC4 and RSA.
3. Include the HIGH, MEDIUM, and LOW security ciphers.
4. Add all export ciphers.
5. Pull all SSLv2 and export ciphers to the end of the list.

Tuning Available Cipher Lists for TLS v1.3

ciphersuites

Valid Values: The colon-separated list of TLSv1.3 ciphersuite names, as defined in RFC 8446, in preference order. The list may include one or more of the following:

- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256
- TLS_AES_128_GCM_SHA256
- TLS_AES_128_CCM_8_SHA256
- TLS_AES_128_CCM_SHA256

Default Value: empty string, which is equivalent to `TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256`

Specifies the defined list of ciphersuites to be used for TLSv1.3, if that TLS version is supported by both side of the connection (and negotiated during handshake). This option supplements `cipher-list` option (which is still applicable for TLSv1.2 and lower).

Applications should use the SSL_CTX_set_ciphersuites() or SSL_set_ciphersuites() functions to configure TLSv1.3 ciphersuites.

> **Important**
>
> The functions SSL_CTX_get_ciphers() and SSL_get_ciphers() return the full list of ciphersuites that have been configured for both TLSv1.2 and lower and TLSv1.3.

Troubleshooting Genesys TLS

Follow the suggestions in this section if your Genesys TLS configuration does not seem to work correctly.

Secure Connection Cannot be Established

When a secure connection between a client and server cannot be established, review the following suggestions:

- Make sure the Genesys components support the Genesys TLS functionality. See the corresponding product documentation.
- Make sure that Genesys TLS is supported on your operating system. See step 2 of Installing the Security Pack.
- Make sure that the CA self-signed certificate file and at least one certificate issued by this CA are installed on the host computers where a client and server applications run.
- For UNIX, make sure that the Genesys Security Pack on UNIX is installed on each UNIX host computer on which Genesys components are installed.
- For UNIX, make sure the environment variables that correspond to your operating systems are properly set (see the table in step 2 of Installing the Security Pack).
- For UNIX, make sure the environment variables that correspond to your operating systems are also properly set for the LCA environment (see the table in step 2 of Installing the Security Pack).
- For Windows, check if the certificates are installed under the Local Computer account for server applications and under the Current User account for client GUI applications.
- Make sure that configured certificates including CA certificates are not expired.
- If DB Server starts from the configuration file and cannot open a secure port, make sure that the transport option is configured correctly and there are no spaces before or after the delimiter characters ; and =.
- Genesys recommends that only one instance of CA is used for your entire call center environment.
- Certificates are generated for a particular host with the full host name specified. When the certificate is installed on the host where applications run, make sure that the host name complies with these two requirements:
  - The Subject CN/SAN field of the host name contains the fully qualified domain name (FQDN) of this host.
  - The host name must match the name that is resolved from other computers.

Supporting Components

This section lists the Genesys components that currently support TLS and on what connections. For detailed information about TLS support by Genesys components, see the corresponding product documentation.

> **Important**
> This list indicates that secure data exchange using TLS is supported on the given connections; it does not specify the type of TLS supported. Refer to product- or component-specific documentation to determine if Mutual TLS and/or Simple TLS is supported.

CCPulse+

Secure data exchange is supported on the following CCPulse+ and Framework connections:

- Between CCPulse+ and Stat Server
- Between CCPulse+ and DB Server
- Between CCPulse+ and Configuration Server

Configuration Layer Components

Secure data exchange is supported on all Configuration Layer connections:

- Between Configuration Server and Configuration Manager
- Between Configuration Server and Configuration Server Proxy
- Between Configuration Server and DB Server
- Between primary and backup Configuration Servers
- Between Configuration Server and External Authentication LDAP Directory (LDAPS)

eServices Components

Secure data exchange is implemented on those connections involving eServices components, as indicated in the following table.

[+] Show table

From	To	
Component	**Port (Secure Listening Port)**	
Chat Server 8.1.0 and later	Interaction Server 8.1.0 and later	`default` (or alternate name)

From	To	
	Universal Contact Server 8.1.0 and later	default (or alternate name)
	Configuration Server or Configuration Server Proxy	default (or alternate name)
	Message Server	default (or alternate name)
Interaction Server 8.1.0 and later	Universal Contact Server	default (or alternate name)
	DB Server	default (or alternate name)
	Stat Server	default (or alternate name)
	Chat Server 8.1.0 and later	ESP (required name)
	Configuration Server or Configuration Server Proxy	default (or alternate name)
	Message Server	default (or alternate name)
	E-mail Server 8.1.0 and later	default (or alternate name)
	Interaction Server 8.1.0 and later	default (or alternate name)
E-mail Server 8.1.0 and later	Configuration Server or Configuration Server Proxy	default (or alternate name)
E-mail Server 8.1.2 and later	Universal Contact Server 8.1.1 and later	default (or alternate name)
	Interaction Server 8.1.0 and later	default (or alternate name)
	Configuration Server or Configuration Server Proxy (writable)	default (or alternate name)
	Message Server	default (or alternate name)
Universal Contact Server Proxy 8.1.0 and later	Configuration Server or Configuration Server Proxy	default (or alternate name)
	Universal Contact Server 8.1.0 and later	default (or alternate name)
SMS Server 8.1.0 and later	Configuration Server or Configuration Server Proxy	default (or alternate name)
	Solution Control Server	default (or alternate name)
	Message Server	default (or alternate name)
	Interaction Server 8.1.0 and later	default (or alternate name)
Social Messaging Server 8.1.0 and later	Configuration Server	default (or alternate name)

From	To	
	or Configuration Server Proxy	
	Message Server	default (or alternate name)
	Interaction Server 8.1.0 and later	default (or alternate name)
Classification Server 8.1.0 and later	Configuration Server or Configuration Server Proxy	default (or alternate name)
Web API Server Java 8.1.0 and later	Configuration Server or Configuration Server Proxy	default (or alternate name)
	Solution Control Server	default (or alternate name)
	Message Server	default (or alternate name)
	Interaction Server 8.1.0 and later	default (or alternate name)
	E-mail Server 8.1.0 and later	default (or alternate name)
	Chat Server 8.1.0 and later	default (or alternate name)
	Stat Server 8.1.0 and later	default (or alternate name)
	Universal Contact Server 8.1.0 and later	default (or alternate name)
Web API Server .NET	Configuration Server or Configuration Server Proxy	default (or alternate name)
	Solution Control Server	default (or alternate name)
	Message Server	default (or alternate name)
	Interaction Server 8.1.0 and later	default (or alternate name)
	E-mail Server 8.1.0 and later	default (or alternate name)
	Chat Server 8.1.0 and later	default (or alternate name)
	Stat Server 8.1.0 and later	default (or alternate name)
	Universal Contact Server 8.1.0 and later	default (or alternate name)

In addition to the general procedures discussed in this Guide:

- Additional steps are required to configure TLS for Universal Contact Server and E-mail Server, both of which are Java-based servers. Refer to the *eServices Deployment Guide* for additional information.
- If TLS is configured on Universal Contact Server (UCS), E-mail Server, or Social Messaging Server, either as a server on its ESP port or as a client of Configuration Server, Interaction Server, Chat Server, UCS, or Message Server, follow these steps to enable it as a Windows Service:
  1. Select the Windows service related to UCS, E-mail Server, or Social Messaging Server.
  2. Select the Log 0n tab.

3. Select `Log on as this account` and provide the username and password of a local host user.

Genesys Composer

Secure data exchange is supported between Genesys Composer and Configuration Server/ Configuration Server Proxy, and on both TCP and SIP connections to GVP Debugger.

Genesys Co-browse Server

Secure data exchange is supported on the following connections:

- Between Co-browse Server and Configuration Server/Configuration Server Proxy
- Between Co-browse Server and Message Server
- Between Co-browse Server and External Cassandra
- HTTPS to communicate with proxied resources target servers.

Genesys Info Mart

Secure data exchange is supported between Genesys Info Mart and:

- Configuration Server/Configuration Server Proxy
- Message Server
- Interaction Concentrator database and Info Mart databases (via SSL)

Genesys Knowledge Center

Secure data exchange is supported between Genesys Knowledge Center and:

- Configuration Server/Configuration Server Proxy
- Message Server
- Solution Control Server

Genesys Softphone

Secure data exchange is supported between Genesys Softphone and the other components of the SIP Infrastructure.

Genesys Voice Platform

Secure data exchange is supported on the following connections within Genesys Voice Platform (GVP) and between GVP and Framework:

- Between GVP components and Configuration Server/Configuration Server Proxy
- Between GVP components and SIP Server
- Between GVP Reporting Server and GVP Media Control Platform/Call Control Platform/Resource Manager/ MRCP Proxy

- SIP interface on GVP Resource Manager/Media Control Platform/Call Control Platform/CTI Connector
- MRCP Platform on GVP Media Control Platform/MRCP Proxy
- HTTP interface on GVP Media Control Platform/Call Control Platform
- HTTP interface on GVP Supplementary Service Gateway

> **Important**
> GVP does not use standard TLS configuration in all cases. It uses a different format for its internal connections (for example, `sip.transport.0=transport0 tls:any”<SIP Port>`) that is described in the GVP Deployment Guide and GVP User’s Guide.

Gplus Adapter for Siebel CRM

Secure data exchange is supported on all internal connections of the Gplus Adapter for Siebel CRM, and between the adapter and:

- Configuration Server/Configuration Server Proxy
- Interaction Server
- Siebel

intelligent Workload Distribution

Secure data exchange is supported on Workload Distribution (iWD) connections to all other Genesys Servers. In addition, Business Context Management Service (BCMS) supports TLS on its connection with Interaction Server.

Interaction Concentrator (ICON)

Secure data exchange is supported between the ICON Server and all other Genesys Servers.

Interaction Layer Components

Secure data exchange is supported on the following Interaction Layer Components:

- From the web browser to the Genesys Administrator/Genesys Administrator Extension server (HTTPs/ SSL)
- From the Genesys Administrator Extension server to:
  - Configuration Layer components—Configuration Server, Solution Control Server, Genesys Deployment Agent
  - Interaction Layer components—Genesys Administrator Extension Database, Genesys Administrator API
  - Database Management Systems—Oracle, MS SQL
  - License Reporting Manager (LRM) Database

Interaction Routing Designer (IRD)

Secure data exchange is supported between IRD and Message Server.

Interaction Workspace Components

See Workspace Desktop Edition.

IVR Server and IVR Drivers Components

Secure data exchange is supported on the following IVR Drivers, IVR Server, and Framework connections:

- Between IVR Driver for WVR for AIX, IVR Driver for MPS and Configuration Server/Configuration Server Proxy
- Between IVR Drivers WVR for AIX, IVR Driver for MPS and IVR Server(s)
- Between IVR Server and Configuration Server/Configuration Server Proxy and/or T-Servers
- Between IVR SDK (C-library version only)

License Resource Manager (LRM)

Secure data exchange is supported between License Resource Manager and:

- All other Genesys Servers
- All supported databases

Load Distribution Server

Load Distribution Server supports secure data exchange on all connections.

Management Layer

Secure data exchange is supported on the following Management Layer connections:

- Between Message Server and DB Server
- Between Message Server and its clients
- Between Message Server and Solution Control Servers
- Between Solution Control Server (SCS) and Solution Control Interface (SCI)
- Between SCS and Configuration Server/Configuration Server Proxy
- Between SCI and Configuration Server/Configuration Server Proxy
- Between SCI and DB Server
- Between Local Control Agent (LCA) and SCS
- Between Genesys Deployment Agent and its clients
- Between primary and backup Solution Control Servers

Media Layer Components

Secure data exchange is supported on the following Media Layer connections:

- Between T-Servers
- Between Network T-Servers
- Between T-Server and Network T-Server
- Between T-Server/Network T-Server and Configuration Server/Configuration Server Proxy
- Between primary and backup T-Servers in hot standby mode
- Between T-Server and custom client applications that have been created with the new T-Library

SIP Server supports secure data exchange on all connections listed above, plus on all SIP traffic.

Orchestration Server

Secure data exchange is supported on the following connections between Orchestration Server and:

- Configuration Server/Configuration Server Proxy
- DB Server
- Message Server
- T-Server
- SIP Server
- IVR Server
- Interaction Server
- Federation Server
- Stat Server
- Universal Routing Server
- Intracluster connections

Outbound Contact Components

Secure data exchange is supported on the following Outbound Contact and Framework connections:

- Between Outbound Contact Server and CPD Server/CPD Proxy Server
- Between Outbound Contact Server and Configuration Server/Configuration Server Proxy
- Between Outbound Contact Server and T-Server
- Between Outbound Contact Server and DB Server
- Between Outbound Contact Server and Stat Server
- Between CPD Server and CPD Proxy Server
- Between CPD Server and T-Server

- Between CPD Server/CPD Proxy Server and Configuration Server/Configuration Server Proxy

Platform SDK

Platform SDK supports TLS for Genesys components that support this feature. For details about how TLS can be used in custom-built applications, refer to the appropriate API Reference for your development platform.

Pulse

Secure data exchange is supported between Pulse and all other Genesys Servers.

Services Layer Components

Secure data exchange is supported on the following Services Layer connections:

- Between Stat Server and Configuration Server/Configuration Server Proxy
- Between Stat Server and T-Server/SIP Server
- Between Stat Server and DB Server
- Between Stat Server and Interaction Server
- Between Stat Server and Message Server
- Between Stat Server II and Configuration Server/Configuration Server Proxy

In addition, secure data exchange is supported between Stat Server and all client connections that support this feature.

Universal Contact Components

Refer to eServices Components to determine on what connections involving Universal Contact Server/ Universal Contact Server Proxy support secure data exchange using TLS.

Universal Routing Components

Secure data exchange is supported between all Universal Routing components and those Framework components that support this feature.

Starting with Security Pack on Unix 8.1.2, a secure HTTP (HTTPS) connection for Universal Routing Server can be configured without a client certificate.

Workforce Management Components

Secure data exchange is supported on the following connections within Workforce Management (WFM) and between WFM and Framework:

- Between WFM Data Aggregator and Configuration Server/Configuration Server Proxy, Message Server, and Stat Server
- Between WFM Data Aggregator and WFM Server

- Between WFM Daemon and Configuration Server/Configuration Server Proxy and Message Server
- Between WFM Daemon and WFM Server
- Between WFM Server and Configuration Server/Configuration Server Proxy and Message Server
- Between WFM Server and WFM Builder and WFM Server (acting as a server application)
- Between WFM Builder and Configuration Server/Configuration Server Proxy and Message Server
- Between WFM Builder and WFM Server
- Between WFM Web and WFM Server, WFM Data Aggregator, and WFM Builder
- Between WFM Configuration Utility and WFM Server
- All internal connections, and all connections to Configuration Server/Configuration Server Proxy, and Message Server.

Workspace Desktop Edition (formerly known as Interaction Workspace) Components

Secure data exchange is supported on the following Workspace Desktop Edition connections:

- Between Workspace Desktop Edition and Stat Server
- Between Workspace Desktop Edition and T-Server
- Between Workspace Desktop Edition and Configuration Server
- Between Workspace Desktop Edition and Universal Contact Server
- Between Workspace Desktop Edition and Interaction Server
- Between Workspace Desktop Edition and Chat Server Server
- Between Workspace Desktop Edition SIP Endpoint and SIP Server

Workspace Desktop Edition can connect to any Genesys application configured for TLS, and whose Host is assigned a certificate.

TLS Feature Support Matrix

Genesys is continually updating TLS implementations to keep up with latest revisions and best practice recommendations while enabling configurability to maintain a high degree of backward compatibility and allow customers to tune the protocol to their own security preferences. The below table outlines for specific interconnections between Genesys products, compatibility with some key considerations around the TLS protocol.

How to read this table:

- Product (acting as client): This indicates for a given connection the product which is connecting to another Genesys product (the client).
- Product connections (acting as server): This indicates to which product is being connected (the server).

Thus, each line defines a unique connection between two Genesys products.

The remaining columns indicate current support levels for attributes of this connection as indicated below:

- TLS 1.2 Support Release #: This column indicates the minimum version of the server-side component necessary to support version 1.2 of the TLS protocol.
- sec-protocol option support: TLS relies on a handshake (mutual agreement) between client and server to select protocol version to use. This column indicates whether this product can be configured, for this connection, using option sec-protocol to control which protocol versions may be offered in handshaking process. See Advanced TLS for more details.
- Mutual TLS Support: This column indicates whether in addition to server offering certificate to the client in the connection, the client may also offer certificate to the server (mutual certificate exchange). See Securing Connections using TLS for an example of configuring a connection for mutual certificate exchange.
- Host configuration to Message Server: Typically, TLS settings can be configured explicitly for each connection, or for convenience at application or host level. However, in earlier implementations connections to Message Server would not leverage TLS settings unless configured at the explicit connection. This column indicates whether when product connects to Message Server whether Host level configuration can be used.
- FIPS 140-2: This column indicates whether there is optional configuration that leverages a FIPS 140-2 validated cryptographic module for this product's side of connection. See Federal Information Processing Standards for more details.
- Compatible with SHA-2 certificates: This column indicates if server certificate can be SHA-2 signed. SHA-2 is preferred over earlier signing algorithms such as MD5 or SHA1.
- Refer to Security Pack 8.5.100.25 for information on OpenSSL version 1.1.1g, TLS 1.3, and SAN certificate.

Product (acting as client)	Product Connections (acting as server)	TLS 1.2 Support Release #	sec-protocol option support	Mutual TLS Support	Host configuration to Message Server	FIPS 140-2	Compatible with SHA-2 certificates	Comments
T-Server for Skype for Business 9.0.000.06	Configuration Server	8.5.101.18	Yes				Yes	
	Message Server	8.5.100.25	Yes				Yes	
	Stat Server	8.5.112.05	Yes				Yes	
	SIP Server	8.1.102.73	Yes				Yes	
	Orchestration Server	8.1.400.86	Yes				Yes	
	Universal Routing Server	8.1.400.52	Yes				Yes	
	SkFB_Connector	9.0.000.06	Yes				Yes	
	SkFB_TServer_backup	9.0.000.06	Yes				Yes	
Connector for Skype for Business	Configuration Server	8.5.101.18	Yes				Yes	
	Message Server	8.5.100.25	Yes				Yes	
	Microsoft Skype for Business 2015		Yes				Yes	Connection to Microsoft Skype for Business 2015 is entirely controlled by Microsoft libraries and has not been tested in house.
Intelligent	Universal	8.5.100.19			NA			

Automation 9.0	Contact Server							
	Chat Server	8.5.107.11			NA			
	Interaction Server	8.5.109.01			NA			
	Configuration Server	8.5.100.22			NA			
Intelligent Automation 3.3.0	Universal Contact Server	8.5.300.05			NA			
Product (acting as client)	**Product Connections (acting as server)**	**TLS 1.2 Support Release #**	**sec-protocol option support**	**Mutual TLS Support**	**Host configuration to Message Server**	**FIPS 140-2**	**Compatible with SHA-2 certificates**	**Comments**
Management Framework 8.5+ – all components	Configuration Server	8.5.100.22	NA	NA	NA	NA	NA	
	Message Server	8.5.100.13	Yes	Yes	Yes	Yes	Yes	
	Solution Control Server	8.5.100.17	NA	NA	NA	NA	NA	
	Local Control Agent	8.5.100.20	NA	NA	NA	NA	NA	
	Configuration Server Proxy	8.5.100.22	NA	NA	NA	NA	NA	
	DB Server	8.1.300.06	NA	NA	NA	NA	NA	
Universal Contact Server 8.5.100.19+	Configuration Server	8.5.100.22	Yes	Yes	NA		Yes	TLSv1.2 support comes from Java. Use **-D jdk.tls.client.protocols** and **jdk.tls.disabledAlgorithms** options to enable.
	Message Server	8.5.100.13	Yes	Yes	Yes		Yes	
	Chat Server	8.5.107.11	Yes	Yes	NA		Yes	
	Interaction Server	8.5.109.01	Yes	Yes	NA		Yes	

	Email Server	8.5.104.06	Yes	Yes	NA		Yes	Starting with 8.5.3, PEM private key format is supported.
	Local Control Agent	8.5.100.20	Yes	Yes	NA		Yes	
	Social Media Server	8.5.400.03	Yes	Yes	NA		Yes	
Email Server 8.5.104.08+	Configuration Server	8.5.100.22	Yes	Yes	NA	Yes		
	Message Server	8.5.100.13	Yes	Yes	Yes	Yes		
	Interaction Server	8.5.109.01	Yes	Yes	NA	Yes		
	Universal Contact Server	8.5.100.19	Yes	Yes	NA	Yes		
Social Media Server 8.5.400.03+	Configuration Server	8.5.100.22	Yes	Yes	NA		Yes	
	Message Server	8.5.100.13	Yes	Yes	Yes		Yes	
	Universal Contact Server	8.5.100.19	Yes	Yes	NA		Yes	
	Interaction Server	8.5.109.01	Yes	Yes	NA		Yes	
Universal Contact Server Proxy 8.5.100.04+	Universal Contact Server	8.5.100.19	Yes	Yes	NA	Yes	Yes	
	Configuration Server	8.5.100.22	Yes	Yes	NA	Yes	Yes	
	Message Server	8.5.100.13	Yes	Yes	Yes	Yes	Yes	
Product (acting as client)	**Product Connections (acting as server)**	**TLS 1.2 Support Release #**	**sec-protocol option support**	**Mutual TLS Support**	**Host configuration to Message Server**	**FIPS 140-2**	**Compatible with SHA-2 certificates**	**Comments**

TServer for Avaya Communication Manager 8.1.010.30+	Configuration Server	8.5.100.22	Yes	Yes	NA	Yes	Yes	
	Message Server	8.5.100.13	Yes	Yes	Yes	Yes	Yes	
	ISCC	8.1.010.30	Yes	Yes	NA	Yes	Yes	
Outbound Contact Server 8.1.508.06+	Message Server	8.5.100.13	Yes	Yes	Yes	Yes	Yes	
	Configuration Server	8.5.100.22	Yes	Yes	NA	Yes	Yes	
	SIP	8.1.102.58	Yes	Yes	NA	Yes	Yes	
	TServer for Avaya	8.1.010.30	Yes	Yes	NA	Yes	Yes	
	Interaction Server	8.5.109.01	Yes	Yes	NA	Yes	Yes	
	DB Server	8.1.300.06	Yes	Yes	NA	Yes	Yes	
	Real Time Metrics Engine (Stats Server)	8.5.102.00	Yes	Yes	NA	Yes	Yes	
Orchestration Server 8.1.400.82+	Message Server	8.5.100.13	Yes	Yes	Yes	Yes	Yes	
	Configuration Server	8.5.100.22	Yes	Yes	NA	Yes	Yes	
	Universal Routing Server	8.1.400.22	Yes	Yes	NA	Yes	Yes	
	Real Time Metrics Engine (stats Server)	8.5.107.00	Yes	Yes	NA	Yes	Yes	
	SIP	8.1.102.58	Yes	Yes	NA	Yes	Yes	
	Interaction Server	8.5.109.01	Yes	Yes	NA	Yes	Yes	
Interaction	Configuration	8.5.100.22	Yes	Yes	NA	Yes	Yes	

Concentrator 8.1.514.09+	Server							
	Message Server	8.5.100.13	Yes	Yes	Yes	Yes	Yes	
	DB Server	8.1.301.03	Yes	Yes	NA	Yes	Yes	
	SIP	8.1.102.58	Yes	Yes	NA	Yes	Yes	
	Outbound Contact Server	8.1.508.00	Yes	Yes	NA	Yes	Yes	
	Interaction Server	8.5.109.01	Yes	Yes	NA	Yes	Yes	
Classification Server 8.5.300.01+	Configuration Server	8.5.100.22	Yes	Yes	NA		Yes	
	Configuration Server Proxy	8.5.100.22	Yes	Yes	NA			
	Message Server	8.5.100.13	Yes	Yes	Yes		Yes	
	Universal Contact Server	8.5.100.19	Yes	Yes	NA		Yes	
Interaction Server 8.5.110.01+	Configuration Server	8.5.100.22	Yes	Yes	NA	Yes	Yes	
	Configuration Server Proxy	8.5.100.22	Yes	Yes	NA	Yes	Yes	
	Message Server	8.5.100.13	Yes	Yes	Yes	Yes	Yes	
	Universal Contact Server	8.5.100.19	Yes	Yes	NA	Yes	Yes	
	DB Server	8.1.301.03	Yes	Yes	NA	Yes	Yes	
	Chat Server	8.5.107.11	Yes	Yes	NA	Yes	Yes	
	Social Media Server	8.5.400.03	Yes	Yes	NA	Yes	Yes	
	Classification	8.5.300.01	Yes	Yes	NA	Yes	Yes	

	Server							
	Email Server	8.5.104.06	Yes	Yes	NA	Yes	Yes	
	(Server Port)	From supporting clients	Yes	Yes	NA	Yes	Yes	
Chat Server 8.5.109.05+	Configuration Server	8.5.100.22	Yes	Yes	NA	Yes	Yes	
	Message Server	8.5.100.13	Yes	Yes	Yes	Yes	Yes	
	Interaction Server	8.5.110.01	Yes	Yes	NA	Yes	Yes	
	Universal Contact Server	8.5.200.19	Yes	Yes	NA	Yes	Yes	
	Cassandra Database	2.28	Yes	Yes	NA		Yes	
Digital Messaging Server (with WeChat Driver) 9.000.03+	Configuration Server	8.5.100.22	Yes	NA	Yes	Yes	Yes	
	Solution Control Serer	8.5.100.17	Yes	NA	Yes	Yes	Yes	
	Message Server	8.5.100.13	Yes	Yes	Yes	Yes	Yes	
	Universal Contact Server	8.5.200.19	Yes	NA	Yes	Yes	Yes	
	Interaction Server	8.5.110.01	Yes	NA	Yes	Yes	Yes	
	Chat Server	8.5.109.05	Yes	NA	Yes	Yes	Yes	
	(Server Port)	From supporting clients	Yes	NA	Yes	Yes	Yes	Digital Messaging Server supports only one port "default" with

								Listening Mode = secured.
Interaction Server Proxy 8.5.110.01+	Configuration Server	8.5.100.22	Yes	NA	Yes	Yes	Yes	
	Message Server	8.5.100.13	Yes	Yes	Yes	Yes	Yes	
	Interaction Server	8.5.110.01	Yes	NA	Yes	Yes	Yes	
GWS (Web Services and Applications) 8.5.201.85+	Configuration Server	8.5.101.08			NA	Yes	Yes	• Cannot connect to a Configuration Server 'auto-detect' port. Must connect to a 'secured' port. • TLS not supported for connection to Message Server. • GWS CA Trusted certificate must be configured in **application.yaml** file only:
	Interaction Server	8.5.107.11			NA	Yes	Yes	
	Universal Contact Server	8.5.200.10			NA	Yes	Yes	
	Chat Server	8.5.109.06			NA	Yes	Yes	
	SIP	8.1.102.58			NA	Yes	Yes	

								GWS does not read configuration from Configuration Management Environment.
								• For TLSv1.2 use command line option **-Djdk.tls.client.protocol TLSv1.2** or into JAVA_OPTIONS of `/etc/default/gws` for CentOS6 or `/usr/lib/systemd/system/gws.service` for CentOS7.
								Example configuration:
								serverSettings:
								`caCertificate:/usr/local/genesys/cacert/ca_cert.pem`
								onPremiseSettings:
								`cmeHost: fmk`
								`cmePort: 2021`

								`tlsEnabled: true`
Product (acting as client)	**Product Connections (acting as server)**	**TLS 1.2 Support Release #**	**sec-protocol option support**	**Mutual TLS Support**	**Host configuration to Message Server**	**FIPS 140-2**	**Compatible with SHA-2 certificates**	**Comments**
Mobile Engagement 8.5.107.19+	Configuration Server	8.5.100.22			NA			
	Message Server	8.5.100.13			Yes			
	Real Time Metrics Engine (Stats Server)	8.5.102.22			NA			
	Cassandra Database				NA			TLS is supported between Cassandra nodes and on JMX port of Cassandra. TLS is not supported from GMS to Cassandra DB.
	Chat Server	8. 5.105.05			NA			TLS between GSG/GMS and Chat Server in trust server mode (encryption only, no certificate checks). For Chat version 1,

								add the following option in chat section: **chat_ssl_trust_all**=`true`
	Universal Contact Server	8.5.200.10			NA			TLS between GSG/GMS and Chat Server in trust server mode (encryption only, no certificate checks).
	Email Server	8.5.104.06			NA			You can set up an HTTPS connection (even in the **GMS Connection** tab). **Note**: GMS uses HTTPClientFactory, and a TLS option can be set (section **gms**, option **http.ssl_trust_all**, value=`false`, `true`).
	Orchestration Server	8.1.400.53			NA			You can set up an HTTPS connection (even in the **GMS Connection** tab).

								Note: GMS uses HTTPClientFactory, and a TLS option can be set (section **gms**, option **http.ssl_trust_all**, value=`false`, `true`).
	Solution Control Server	8.5.100.17			NA			
	Universal Routing Server	8.1.400.22			NA			You can set up an HTTPS connection (even in the **GMS Connection** tab). **Note**: GMS uses HTTPClientFactory, and a TLS option can be set (section **gms**, option **http.ssl_trust_all**, value=`false`, `true`).
Co-Browse 8.5.101+	Configuration Server	8.5.100.22	Yes		NA		Yes	
	Message Server	8.5.100.13	Yes		Yes		Yes	
	Cassandra Database		Yes		NA		Yes	
Product (acting as client)	**Product Connections (acting as server)**	**TLS 1.2 Support Release #**	**sec-protocol option support**	**Mutual TLS Support**	**Host configuration to Message Server**	**FIPS 140-2**	**Compatible with SHA-2 certificates**	**Comments**

Workforce Management Server 8.5.207.09+		To other supporting servers				Yes	Yes	
Workforce Management Builder 8.5.207.05+		To other supporting servers					Yes	
Workforce Management Daemon 8.5.207.01+	Configuration Server	8.5.101.16	Yes		NA		Yes	
	Message Server	8.5.100.13	Yes		Yes			
Workforce Management Web 8.1.301.02+	Configuration Server	8.5.101.16	Yes		NA		Yes	
	Message Server	8.5.100.13	Yes		Yes			
Workforce Management Aggregator 8.5.203.00+		To other supporting servers					Yes	
Workforce Management DB Server 8.1.301.02+		To other supporting servers					Yes	
Genesys Administrator Extensions 8.5.290.09+	(Server Port)	From supporting clients	Yes		NA		Yes	Add **setIncludeProtocols**=TLS1. in gax.properties.
	Configuration Server	8.5.101.16	Yes	Yes	NA		Yes	For TLSv1.2 with Java 7, set **-Djdk.tls.client.protocols**=TLSv1.2 (not required for

								Java 8).
	Solution Control Server	8.5.100.26	Yes	Yes	NA		Yes	For TLSv1.2 with Java 7, set **-Djdk.tls.client.protocols**=TLSv1.2 (not required for Java 8).
	MSSQL DB	MSSQL:SQLServer2014, SQLServer2016, SQLServer2012	Yes		NA		Yes	For TLSv1.2 with Java 7, set **-Djdk.tls.client.protocols**=TLSv1.2 (not required for Java 8).
Product (acting as client)	**Product Connections (acting as server)**	**TLS 1.2 Support Release #**	**sec-protocol option support**	**Mutual TLS Support**	**Host configuration to Message Server**	**FIPS 140-2**	**Compatible with SHA-2 certificates**	**Comments**
Workspace Desktop Edition 8.5.105.12+	Universal Contact Server	8.5.300.09	Yes	Yes 8.5.148.04+	NA		Yes	Through PSDK.NET 9:ESP Protocol TLS 1.2: 8.5.115.17+
	Configuration Server	8.5.100.22	Yes	Yes 8.5.148.04+	NA		Yes	Through PSDK.NET Config 9 TLS 1.2: 8.5.115.17+
	Interaction Server	8.5.110.10	Yes	Yes 8.5.148.04+	NA		Yes	Through PSDK.NET OpenMedia 9

								TLS 1.2: 8.5.115.17+
	SIP Server	8.1.102.58	Yes	Yes 8.5.148.04+	NA		Yes	Through PSDK.NET Voice 9 TLS 1.2: 8.5.115.17+
	Chat Server	8.5.107.11	Yes	Yes 8.5.148.04+	NA		Yes	Through PSDK.NET 9: Basic Chat Protocol TLS 1.2: 8.5.115.17+
	Real Time Metrics Engine (Stats Server)	8.5.102.22	Yes	Yes 8.5.148.04+	NA		Yes	Through PSDK.NET 9 TLS 1.2: 8.5.115.17+
Voice Platform Resource Manager 8.5.175.95+	SIP Server	8.1.102.58	Yes	Yes	NA		Yes	
	Media Control Platform	8.5.176.05	Yes	Yes	NA		Yes	
	CTI Connector	9.0.010.07	Yes	Yes	NA	Yes	Yes	
	Reporting Server	8.5.181.77			NA			
	RM Internode				NA			
	Configuration Server	8.5.100.22			NA		Yes	
	Message Server	8.5.100.13						

Voice Platform Media Control Platform 8.5.176.05+	Resource Manager	8.5.175.95	Yes	Yes	NA		Yes	
	Reporting Server	8.5.181.77			NA			
	Configuration Server	8.5.100.22	Yes	Yes	NA		Yes	
	Message Server	8.5.100.13						
	HTTPs (client)	8.5.176.05	Yes	Yes	NA		Yes	
	ASR/TTS (MRCP v2 Nuance)	8.5.176.05	Yes	Yes	NA		Yes	
	ASR/TTS (MRCP v1)				NA			
	Nuance/MRCPP				NA			
Voice Platform MRCP Proxy 8.5.184.42+	Reporting Server	8.5.181.77		Yes	NA			
	Configuration Server	8.5.100.22	Yes	Yes	NA		Yes	
	Message Server	8.5.100.13						
	MRCP ASR/TTS				NA			Media Control Platform can be connected directly to MRCP resource.
	MRCP Client				NA			
Voice Platform UCMConnector (T-Server Cisco UCM to Media	Resource Manager	8.5.175.95	Yes					
	T-Server							

Server Connector) 8.5.184.06+	Configuration Server	8.5.100.22	Yes				Yes	
	Message Server	8.5.100.13						
Voice Platform Policy Server 8.5.010.10+	Configuration Server						Yes	
	HTTPs						Yes	
	Genesys Administrator						Yes	
	Message Server						Yes	
Voice Platform CTIConnector 9.0.010.07+	IVR Server	8.5.000.09	Yes	Yes		Yes	Yes	
	Cisco ICM							
	Configuration Server	8.1.100.06	Yes	Yes		Yes	Yes	
	Resource Manager	8.5.181.38	Yes	Yes		Yes	Yes	
	Message Server	8.5.100.16	Yes	Yes	Yes	Yes	Yes	
Voice Platform Reporting Server 9.0.010.62+	Configuration Server		NA					TLSv1.2 configured on Java.
	Database			Yes	NA			Mutual TLS with Oracle RAC 12
	HTTPS			Yes	NA			TLSv1.2 configured on Java.
	RC (Active MQ)			Yes	NA			
	Message Server			Yes	Yes			TLSv1.2 configured on

								Java.
SIP Feature Server 8.1.201.91+	SIP Server				NA			
	Genesys Administrator Extensions				NA			
	Media Control Platform	Supporting version			NA			
	Cassandra DB	Supporting version		Yes	NA			
	Configuration Server				NA			
	Phone				NA			
SIP Server 8.1.102.25+	Configuration Server	8.5.100.22	Yes	Yes	Yes	Yes	Yes	
SIP Proxy 8.1.100.57+	Message Server	8.5.100.13	Yes	Yes	Yes	Yes	Yes	
iWD Manager 9.0.004.07+	Configuration Server	8.1.300.24	Yes	Yes	NA		Yes	
	Message Server	8.5.100.03	Yes	Yes	Yes		Yes	
	iWD History Node	9.0.004.07	Yes	Yes	NA		Yes	
	Interaction Server	8.5.105.04	Yes	Yes	NA		Yes	
	Universal Contact Server	8.5.300.09	Yes	Yes	NA		Yes	
iWD History Node 9.0.004.07+	Configuration Server	8.1.300.24	Yes	Yes	NA		Yes	
	JMSQ		Yes	Yes	NA		Yes	
iWD Runtime Node	Configuration Server	8.1.300.24	Yes	Yes	NA		Yes	

9.0.004.07+	iWD History Node	9.0.004.07	Yes	Yes	NA		Yes	
iWD Web 9.0.004.01+	Configuration Server	8.1.300.24	Yes	Yes	NA		Yes	
	Message Server	8.5.100.03	Yes	Yes	Yes		Yes	
	Interaction Server	8.5.105.04	Yes	Yes	NA		Yes	
	iWD Web Capture Point	9.0.003.07	Yes	Yes	NA		Yes	
Browser	iWD Web	9.0.004.01	Yes	Yes	NA		Yes	
	iWD Manager	9.0.004.07	Yes	Yes	NA		Yes	
iWD GAX Plugin 9.0.012.07+	iWD Runtime Node	9.0.004.07	Yes	Yes	NA		Yes	
Product (acting as client)	**Product Connections (acting as server)**	**TLS 1.2 Support Release #**	**sec-protocol option support**	**Mutual TLS Support**	**Host configuration to Message Server**	**FIPS 140-2**	**Compatible with SHA-2 certificates**	**Comments**
Load Distribution Server 8.1.005.02+	Configuration Server	8.5.100.25	Yess	Yes	NA	Yes	Yes	
	Message Server	8.5.100.11	Yes	Yes	Yes	Yes	Yes	
	SIP	8.1.101.79	Yes	Yes	NA	Yes	Yes	
	Load Distribution Server	8.1.005.02	Yes	Yes	NA	Yes	Yes	
Universal Routing Server 8.1.400.28+	Load Distribution Server	8.1.005.02	Yes	Yes	Yes	Yes	Yes	
	Interaction Server	8.5.110.01	Yes		NA		Yes	

Platform SDK for Java 8.5.102.02+	Any supporting server	Any supporting server	Yes	Yes		Yes	Yes	FIPS 140-2 requires OpenJDK 8u212 b04+
Platform SDK for .NET 8.5.102.03+	Any supporting Server	Any supporting Server	Yes	Yes		Yes	Yes	
TServer for Avaya TSAPI v.8.1.010.12+	Configuration Server	8.5.100.18		Yes	NA			
	Message Server	8.5.100.20	Yes	Yes	Yes		Yes	
	Real Time Metrics Engine (Stats Server)	8.5.100.22	Yes	Yes	NA		Yes	
	Universal Routing Server	8.1.400.52	Yes	Yes	NA		Yes	
	TServer for Avaya TSAPI	8.1.010.12	Yes	Yes	NA		Yes	
TServer for Cisco UCM 8.1.202.34+	TServer for Cisco UCM	8.1.202.34	Yes	Yes	NA		Yes	
	Configuration Server	8.5.100.25	Yes	Yes	NA		Yes	
	Message Server	8.5.100.11	Yes	Yes	Yes		Yes	
	Real Time Metrics Engine (Stats Server)	8.5.104.22	Yes	Yes	NA		Yes	
	Universal Routing Server	8.1.400.28	Yes	Yes	NA		Yes	
Genesys InfoMart 8.5.011.11+	Configuration Server	8.5.100.22	Yes	Yes	NA	Yes	Yes	
	Message	8.5.100.13	Yes	Yes	Yes	Yes	Yes	

	Server							
Genesys Web Engagement 8.5.000.42+	Configuration Server	8.5.100.02	Yes	Yes	NA		Yes	
	Message Server	8.5.000.17	Yes	Yes			Yes	
	Interaction Server	8.5.105.00	Yes	Yes	NA		Yes	
	Interaction Server Proxy	8.5.109.13	Yes	Yes	NA		Yes	
	Real Time Metrics Engine (Stats Server)	8.5.101.05	Yes	Yes	NA		Yes	
	External Cassandra	2.2.3	Yes	Yes	NA		Yes	
Product (acting as client)	**Product Connections (acting as server)**	**TLS 1.2 Support Release #**	**sec-protocol option support**	**Mutual TLS Support**	**Host configuration to Message Server**	**FIPS 140-2**	**Compatible with SHA-2 certificates**	**Comments**
Pulse 9.0.001.00+	Configuration Server	8.5.101.20	Yes	Yes	NA		Yes	
	Message Server	8.5.100.11	Yes	Yes	Yes		Yes	
	MSSQL DB	SQL Server 2012 on Windows 2012 R2 and SQL Server 2017 on Linux		Yes	NA		Yes	
	PostgreSQL DB	PostgreSQL Server 10 on Windows 2012 R2 and PostgreSQL		Yes	NA		Yes	

		Server 10 on Linux						
	Oracle DB	Oracle 12c RAC on Linux			NA		Yes	
Pulse Collector 9.0.001.01+	Configuration Server	8.5.101.20	Yes	Yes	NA		Yes	
	Message Server	8.5.100.11	Yes	Yes			Yes	
	Real Time Metrics Engine (Stats Server)	8.5.104.22	Yes	Yes	NA		Yes	
	DB Server	8.1.302.02	Yes	Yes	NA		Yes	
	MSSQL DB	SQL Server 2012 on Windows 2012 R2 and SQL Server 2017 on Linux		Yes	NA		Yes	
	PostgreSQL DB	PostgreSQL Server 10 on Windows 2012 R2 and PostgreSQL Server 10 on Linux		Yes	NA		Yes	
	Oracle DB	Oracle 12c RAC on Linux			NA		Yes	
Real Time Metrics Engine (Stats Server) 8.5.110.14+	Configuration Server	8.5.100.22	Yes		NA		Yes	
	Message Server	8.5.100.13	Yes	Yes	Yes		Yes	
	SIP-Server	8.1.102.58	Yes	Yes	NA		Yes	
	Interaction	8.5.201.05	Yes	Yes	NA		Yes	

	Server							
	DB Server	8.1.301.03	Yes	Yes	NA		Yes	
Universal Routing Server 8.1.400.56+	Configuration Server	8.5.100.22			NA			
	Message Server	8.5.100.13	Yes	Yes	Yes	Yes	Yes	
	SIP Server	8.1.102.50	Yes	Yes	NA	Yes	Yes	
	Orchestration Server	8.1.400.82	Yes	Yes	NA	Yes	Yes	
	Real Time Metrics Engine (Stats Server)	8.5.102.00	Yes	Yes	NA	Yes	Yes	
	IIS Web Server	IIS7			NA	Yes		

TLS SNI Extension Support

Introduction

Starting with Genesys Security Pack on UNIX 8.5.100.23, it's possible to specify TLS extension `server_name` by setting the **tls-target-name** option. Server Name Indication (SNI) is an extension to the Transport Layer Security (TLS) computer networking protocol by which a client indicates which host name it is attempting to connect to at the start of the handshaking process. For related RFC, see here.

This feature requires the **tls-target-name** option to work correctly. For information on the **tls-target-name** option, refer to tls-target-name.

Client-side and server-side support

On the client side:

- Both Windows and UNIX Security Pack implementations send the `server_name` extension.

> **Important**
>
> Both implementations send the `server_name` extension all the time. However, if you do not set the **tls-target-name** value, then wrong server name may be sent. This issue will be fixed in future iterations.

On the server side:

- Neither Windows nor UNIX Security Pack support this feature.

The **tls-target-name** setting causes the `server_name` extension to be sent to the server and causes the client to check this value against the subject/CN and/or SAN in the returned certificate from the server, even if connection was made using IP address instead of hostname. This check happens only if the **tls-target-name-check** option's value is set to `host`.

Federal Information Processing Standards (FIPS)

Federal Information Processing Standards, also known as FIPS, are a set of standards created by the United States federal government for use in computer systems of non-military government agencies and their contractors. They are concerned primarily with interoperability of different systems, portability of data and software, and computer security.

A FIPS standard is developed only when there are no voluntary standards in existence to address federal requirements. In some cases, the standards are modified and updated restatements of technical standards already in use, such as those of the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO).

Generally speaking, the Genesys implementation of TLS is considered to be consistent with FIPS 140-2, based on FIPS capabilities of the underlying libraries.

Supporting Components

The following Genesys components support data security using FIPS:

- Management Framework (except on Apple OS)
- Genesys Security Pack on UNIX
- Workforce Management
- Network T-Servers
- Media T-Servers
- Performance Manager Advisors CCA-ME
- eServices (partial)
- Composer (except on connections to/from the Web Request Block)
- Genesys Rules System
- Outbound Contact
- intelligent Workload Distribution (iWD)
- Interaction Concentrator
- Genesys Info Mart
- Workspace Desktop Edition
- Orchestration Server
- Platform SDK

Genesys Voice Platform

Genesys Voice Platform (GVP) components support data security using FIPS, but some GVP components will require an additional step to enable it. These components use the security library directly and require the additional configuration option **FIPS Mode Enabled** to control their usage. Refer to the *Genesys Voice Platform User's Guide* for more information.

Enabling FIPS in your Environment

Enabling FIPS depends on the operating system that is running in your environment, as follows:

Windows

> **Important**
> FIPS is disabled by default

To set up a FIPS-compliant set of ciphers to be used on Windows, configure the operating system as described in Windows documentation at: http://support.microsoft.com/kb/811833

Then, to enable or disable FIPS, set the following registry variable to `1` (enable) or `0` (disable), as appropriate:

- On Windows 2012 and Windows 8:
  HKLM\System\CurrentControlSet\Control\Lsa\FIPSAlgorithmPolicy
- On Windows 2008, Windows Vista, and Windows 7:
  HKLM\System\CurrentControlSet\Control\Lsa\FIPSAlgorithmPolicy\Enabled

UNIX or Linux

Starting in release 8.1.1, the Genesys Security Pack contains both the original non-FIPS and FIPS-consistent shared libraries. To specify which library to use (FIPS or non-FIPS), set the given environment variables (and related variables) to the location of the library (**<install directory>**) to be used, as follows:

- To use the FIPS library, do one of the following, as appropriate:
  - On AIX platforms, set both the LD_LIBRARY_PATH and LIB_PATH environment variables to either:
    - **<install directory>/fips140_lib32** (for 32-bit libraries)

      or

    - **<install directory>/fips140_lib64** (for 64-bit libraries)
  - On Solaris 64-bit platforms, set both the LD_LIBRARY_PATH and LD_LIBRARY_PATH_64 environment variables to **<install directory>/fips140_lib64**.
  - On Linux platforms that use the Genesys Security Pack version 8.5.100.30 or later, use the following procedure. For Linux with Genesys Security Pack versions prior to 8.5.100.30, use the regular configuration(s) specified in the later section of this page.
    - Run the `fipsinstall.sh` script provided in the **fips140_lib64** directory. This runs the FIPS module self-test and generates proper OpenSSL configuration files which are mandatory for using the FIPS module.
    - Set both the LD_LIBRARY_PATH and OPENSSL_MODULES environment variables to **<install directory>/fips140_lib64** and OPENSSL_CONF variable to **<install directory>/fips140_lib64/openssl.cnf**.

    Note: The master OpenSSL configuration file (**openssl.cnf**) configured in **OPENSSL_CONF** is not included in the installation package but it is generated dynamically by the **fipsinstall.sh** script along with the **fipsmodule.cnf** configuration for FIPS.
  - On all other platforms (including Linux with Genesys Security Pack version prior to 8.5.100.30), set only the LD_LIBRARY_PATH environment variable to **<install directory>/fips140_lib32** (for 32-bit libraries) or **<install directory>/fips140_lib64** (for 64-bit libraries).
- To use the non-FIPS library, set the LD_LIBRARY_PATH environment variable to **<install directory>**.

Platform SDK for .NET

To enable FIPS in an application built using Platform SDK for .NET, use the same procedure as you do for configuring the common library for IIRC.

Platform SDK for Java

To enable FIPS in a Genesys Java environment, you must set up the Java Runtime Environment (JRE) to be compliant with FIPS, as described in the Platform SDK Java documentation.

To configure a FIPS-enabled service-provider, refer to Platform SDK FIPS documentation.

Secure HTTP (HTTPS)

In addition to Transmission Control Protocol (TCP) support for TLS, most Genesys connections using Hypertext Transfer Protocol (HTTP) also support Communications Integrity through the use of HTTP Secure (HTTPS). HTTPS applies SSL or TLS to HTTP connections. In most cases, HTTP is used for communications between web servers and web browsers, and therefore applications are set up to be compatible with the HTTPS setup at the web server. In some cases, HTTP is also used for connections which do not include a web browser, such as web services like Representational State Transfer (REST). See product documentation for details.

Supporting Components

The following components, or elements thereof (as indicated), support the use of HTTPS:

- Universal Routing (connecting to external Web Services)
- Outbound Contact (HTTP Connections for Pre-Validation)
- eServices Web API Server
- Genesys Co-browse (if customer site does not already support)
- Genesys Knowledge Center
- Context Services (REST API)
- Workforce Management (Web and connections between internal components)
- SIP Feature Server
- Voice Platform
- Interaction SDK
- Mobile Services API
- Genesys Agent Desktop
- Genesys Pulse
- Workspace Desktop Edition (ClickOnce)
- Agent Scripting (Plug-in for Workspace Desktop Edition)
- Performance Management Advisors (to Web Server)
- Genesys Administrator
- Genesys Administrator Extension
- Pulse (Plug-in for Genesys Administrator Extension)
- License Reporting Manager (Plug-in for Genesys Administrator Extension)
- Composer (Web Request Block, Context Services)
- Gplus Adapter for Siebel CRM

- Gplus Adapter for SAP CRM
- Intelligent Workload Designer (Web User Interface)
- Genesys Rules System
- Web Engagement
- WebRTC
- Genesys Web Services
- Speech & Text Analytics
- Genesys Video Gateway 9.0
- Genesys Predictive Routing

Secure Real-Time Transport Protocol (SRTP)

Secure Real-Time Transport Protocol (SRTP) is supported by Genesys SIP Solutions. Genesys recommends that SIP Connections to negotiate SRTP connections be protected by TLS. SRTP negotiation is done using SDES methodology in SDP attachments to SIP messaging. See product documentation for specific details.

Supporting Components

- Genesys Video Gateway 9.0 and later (DTLS-SRTP on WebRTC (Web) interface)
- SIP Endpoint SDK, including Workspace Desktop Edition SIP Endpoint
- Voice Platform Media Control Platform

Web Application Security

Genesys software provides web application security that meets or exceeds industry-wide security standards and recommendations defined by governing bodies and security-related organizations.

Genesys provides protection from the following weaknesses:

- Open Web Application Security Project (OWASP)
- RESTful Web Services

Some Genesys products include bundled web servers. Web Servers should be hardened to comply with your corporate standards. Considerations may include (but are not limited to): activating SSL, hiding management consoles, removing default error pages, and configuring session cookie attributes.

Open Web Application Security Project

Open Web Application Security Project (OWASP) is a world-wide organization that drives the evolution of safe and secure software, and the visibility and awareness of the need for it. It does not provide security solutions. Instead, it identifies and brings security issues to the attention of the software industry encouraging the industry to addressing these issues in their software.

OWASP is perhaps best known for its Top Ten Application Security Risks, commonly referred to as the OWASP Top Ten. This is a list of what OWASP considers to be the ten most important web application security weaknesses, and provides information to help address and mitigate these weaknesses. The weaknesses identified by the OWASP Top Ten have and will change over time, as software and the digital infrastructure becomes more complex and open. For more information about OWASP, the OWASP Top Ten, and what companies and organizations are using OWASP Top Ten, refer to the OWASP website.

This section identifies what and how Genesys addresses the OWASP Top Ten Weaknesses.

Top 10 2010-A3—Broken Authentication and Session Management

Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, session tokens, or exploit other implementation flaws to assume other users' identities.

To mitigate the risk of improper access to session-related data stored in cookies, Genesys uses the HTTPOnly and Secure flags when dealing with cookies related to web application sessions. The HTTPOnly flag prevents access to the cookie from non-HTTP protocols; the Secure flag prevents access outside of an SSL session.

Supporting Components

The following components address OWASP Top 10 2010-A3:

- Genesys Administrator
- Genesys Administrator Extension

Top 10 2007-A6—Information Leakage and Improper Error Handling

Error messages generated by failed authentication attempts used to display specific information about why the attempt failed. This could be used by an unauthorized user to gain access to the Genesys system. For example, one error used to indicate that either the login username or the

password was incorrect. A malicious user could use this information to discover credentials, and gain access to data.

Now, the information returned by failed authentication requests, while still clear about the nature of the error, is less specific about its cause. In the above example, the message still indicates that it is an authentication error, but combines the possible causes into being the username and/or the password. A malicious user would then have to try all possible combinations of all possible usernames and passwords, a task that is much greater than trying just a password or username.

Supporting Components

The following Genesys components address OWASP Top 10 2007-A6:

- Management Framework
- Genesys Administrator

RESTful Web Services

Representational State Transfer (REST) is a software architecture style exemplified most notably by the World Wide Web. It enforces proper interactions between internal components of a product, without imposing on the users of the product as a whole.

A RESTful web service is a web service that meets the constraints imposed by REST. Four HTTP verbs are normally used to implement a RESTful web service: GET, PUT, POST, and DELETE. Of these, GET is the safest method, being similar to a READ operation. PUT and DELETE are the most harmful methods, capable of overwriting or removing data.

Components Using RESTful Web Services

The following Genesys components use RESTful Web Services:

- Genesys Mobile Services
- Genesys Voice Platform
- Orchestration Routing Server
- Context API
- Genesys Predictive Routing

Genesys Software and RESTful Web Services

To minimize the possible detrimental impact of exposing data to the RESTful methods, especially PUT and DELETE, follow the implementation described in the following message:

> ### Warning
> Any products that provide a RESTful interface (GSG, GVP, ORS, Context API), must be located on a web server that is not used for any other purpose. This web server must be protected by appropriate user authentication and access controls. These APIs rely on exposing Web Server functions (PUT and DELETE) that you might not want exposed with other applications.

Data Privacy

> **Important**
>
> The purpose of this document is to help organizations understand how Genesys Services can be utilized to help them comply with certain regulatory requirements, including EU General Data Protection Regulation (GDPR). Some of the Genesys Services features described herein may or may not be available based upon an organization's specific environment and Genesys Services acquired.
>
> The information in this document may not be construed or used as legal advice about the content, interpretation or application of any law, regulation or regulatory guideline. Customers and prospective customers must seek their own legal counsel to understand the applicability of any law or regulation on their processing of personal data, including through the use of Genesys' products or services.

Existing Data Privacy Regulations

Genesys understands the need for enterprises using our software to comply with all local and global laws. It is possible for a customer to use the Genesys suite of products in a manner that complies with GDPR (European Union General Data Protection Regulation), LGPD (Brazilian General Data Protection Act), APP (Australian Privacy Principles), CCPA (California Consumer Privacy Act) by following the guidelines listed in General Data Protection Regulation. However, the Genesys products are merely tools to be used by the customer and Genesys recommends that the customer take steps to ensure compliance with the applicable regulations.

General Data Protection Regulation (GDPR)

This page provides general information about Genesys support for customer compliance with the General Data Protection Regulation (GDPR).

What is GDPR?

GDPR is a regulation in EU law passed by the European Union in 2016, setting new rules for how companies manage and share personal data. It addresses the export of personal data outside the EU. The GDPR is applicable for enterprises across globe that store EU citizens data.

The regulation applies if the data controller, an organisation that collects data from EU residents, or processor, an organisation that processes data on behalf of a data controller like cloud service providers or the data subject (person) is based in the EU. The regulation also applies to organisations based outside the EU if they collect or process personal data of individuals located inside the EU.

Important

The purpose of this document is to help organizations understand how Genesys Services can be utilized to help them comply with certain regulatory requirements, including EU General Data Protection Regulation. Some of the Genesys Services features described herein may or may not be available based upon an organization's specific environment and Genesys Services acquired.

The information in this document may not be construed or used as legal advice about the content, interpretation or application of any law, regulation or regulatory guideline. Customers and prospective customers must seek their own legal counsel to understand the applicability of any law or regulation on their processing of personal data, including through the use of Genesys' products or services.

What data comes under the scope of GDPR?

According to the European Commission, "personal data is any information relating to an individual, whether it relates to his or her private, professional or public life. It can be anything from a name, a home address, a photo, an email address, bank details, posts on social networking websites, medical information, or a computer's IP address." This data is called personally identifiable information (PII).

How does Genesys support compliance with the rights defined by GDPR?

Genesys holds EU citizens' data for the purposes of executing processing on behalf of customers. While Genesys customers are the data controllers for GDPR purposes, Genesys has a responsibility to support customer compliance with GDPR requests. The following table describes Genesys support for GDPR rights.

Right	Genesys Support
Right of Consent	Requirements to meet Right of Consent apply outside the Genesys platform. In general, Genesys does not collect data unless it has been determined to be necessary to meet the use cases of customers, who are the data controllers from the point of view of GDPR compliance. Although Genesys might collect aggregate or pseudo-anonymized information for purposes such as statistical and best-practices analysis, Genesys does not utilize customer data for purposes that require consent from consumers. However, be aware that some information you collect for business purposes might incidentally be captured in the Genesys platform (for example, in a transcript record).
Right of Access and Portability	Genesys provides processes to export PII if the data is held for more than 30 days, so that customers can comply with Right of Access requests from consumers.
Right of Erasure (Forget Me)	Genesys provides processes to delete, redact, or pseudo-anonymize PII if the data is held for more than 30 days, so that customers can comply with Right of Erasure requests from consumers.
Breach Notification	Genesys maintains a Product Security Incident Response Team (PSIRT) to collaborate with customers in data breach scenarios.
Privacy by Design	As described on other pages in the *Genesys Security Deployment Guide* (this document), security measures that protect customer data are part of standard Genesys design requirements.

Genesys Engage cloud Support for GDPR

This page provides information that is specific to the Genesys Engage cloud implementation of support for General Data Protection Regulation (GDPR). For general information about Genesys support for GDPR, see General Data Protection Regulation (GDPR).

> **Important**
>
> In general, Genesys support for GDPR compliance is based on default configuration settings and typical application usage. Other underlying components within Genesys Engage cloud do not store sensitive information beyond 30 days.

Genesys has a standard JSON format for both Right of Erasure (Forget Me) and Right of Access and Portability (Export).

Process for Genesys Engage cloud

- Open a support case using the Customer Care portal.
- Select the Case Sub Type of Data Privacy.
- Provide the input JSON file. This JSON file will contain information on what action is required to be taken and will also include the consumer-identifying input for GDPR requests (forget me and/or export).
- Customer Care can assist in creating the JSON files that will form the input for the Export or Delete request. Customers can also create these JSON files. See JSON Format for a sample format.
- Genesys will then process the JSON files to fulfill the request.
- After the file is processed, Customer Care will post the execution log or exported records to the SFTP site in the Case (Under Transfer Files).
- The customer can validate the execution log and based on the response messages, a follow-up action may be required. If the execution was not successful, a corrected request may be resubmitted.

Notes for Outbound

- The *shortcodes* array in the request is used only to execute searches for the mobile channel (SMS/MMS). You can skip it or provide an empty array.
- If shared shortcodes are specified in the request, they will not be included in the exported result.
- The *accountid* is created during the provisioning process and is required for Outbound. PEC Care can obtain this value for each tenant if it is not already known.

Notes for Portico Aggregation or Mobile Messaging Manager (MMM)

- The *shortcodes* array in each JSON request is mandatory for these products since they are exclusively used for mobile messaging (SMS/MMS).
- If shared shortcodes are specified in the request, they will not be included in the exported result.
- The *accountid* parameter can be ignored.

JSON File Format

Genesys has a standard JSON format for both Right of Erasure (Forget Me) and Right of Access and Portability (Export).

Input JSON File Naming Conventions

Customers create a plain-text JSON file using the following pattern: **<request_type>-<date:yyyyMMdd>_<uniqueID_or_timestamp>.json** where

- **<request_type>** is either *forget* or *export*.
- **<date:yyyyMMdd>** allows for limited pickup of files placed on the directory structure by date. Automated processing will occur on files matching the current date only.
- **<uniqueID_or_timestamp>** to ensure filenames are unique within a day. Files with the exact same name should not be posted to the pickup location.

Example file name: **forget-20180315_120000.json** OR **export-20180315_120000.json**

Submit File Content

The following sample can be used as a starting template. It models a request for 2 different GDPR compliance requests. This request format can be pasted into a tool like https://jsoneditoronline.org/ to visualize and edit within the structure. It is beyond the scope of these directions to cover a description of the JSON structure or suggest any tools that can be used to help create or edit JSON.

- requests - An array of one or more request objects. At least one is required.
- requestcase - (Optional and not processed) Used to match requests with results. This can be used to store a ticket case number or internal identifier.
- shortcodes - An array of shortcodes used for SMS activity. If SMS channel is not used, submit an empty array.
- accountid - A string containing the account number to search within the Engage platform. This can be an enterprise account number which will search all accounts in an enterprise.
- type - "FORGET" or "EXPORT" are the only supported values. All requests in the payload should have the same type and should match the file name prefix.
- contacts - An array of device type and value pairings. Specify as many or as few as specified by the end user. Allowable device types within the contacts array: phone, email, ipaddr:
  - phone - must be in ITU E.123 international format: like +1 781 555 1212

- email - must follow standard email conventions (include 1 "@" and no less than 1 dot in domain portion, no illegal characters in username)
- ipaddr - likely to be supplied in IPv4 format (4 sets of numbers with 3 dots between)

```
{
   "requests": [{
           "requestcase": "97456596893834",
           "shortcodes": ["11111","22222"],
           "accountid": "30003748347",
           "type": "FORGET",
           "contacts": [{
                   "phone": "+1 781 555 1212"
               },
               {
                   "phone": "+1 617 555 1212"
               },
               {
                   "email": "test@test.com"
               },
               {
                   "email": "genesys@genesys.com"
               },
               {
                   "ipaddr": "10.10.10.10"
               },
               {
                   "ipaddr": "11.11.11.11"
               }
           ]
       },
       {
           "requestcase": "6457657657",
           "shortcodes": ["11111","22222"],
           "accountid": "30003748347",
           "type": "FORGET",
           "contacts": [{
                   "phone": "+1 781 555 1313"
               },
               {
                   "phone": "617 555 1313"
               },
               {
                   "email": "test2@test.com"
               },
               {
                   "email": "genesys2@genesys.com"
               },
               {
                   "ipaddr": "10.10.10.11"
               },
               {
                   "ipaddr": "11.11.11.12"
               }
           ]
       }
   ]
}
```

Submit File Location

Properly named files shall be posted to a directory named **GDPR_Submit** on the supplied SFTP

account. Only files placed in this specifically named directory will be processed.

Delete/Export Result Format and Retrieval

Result File Retrieval Location

A properly named and submitted file will generate a result file for DELETE or EXPORT requests. All result files will be found in a directory on the same SFTP account named **GDPR_Result**.

Result File Naming

Result files will have the exact same name as submitted files except a suffix will be appended: *-execution-log*

Example result file name format: *forget-20180315_120000-execution-log.json* OR *export-20180315_120000-execution-log.json*

Result File Content: Execution Log

Result files contain the original request array. The result array is appended and follows the same format as the original request with the inclusion of a new property named "response" within each "contacts" object. The string value for "response" will always either be "SUCCESS" or "ERROR" followed by some additional explanation for the error. It is important to note that any errors found apply only to processing for that contact device. This means that a request against other contacts will not be stopped because of the failure for one. The expected action simply will not have been completed for the error `device: Not deleted` in the case of "forget" requests and is not provided in any result content for "export" requests. Any "ERROR" devices will need to be corrected and resubmitted.

Both "SUCCESS" and "ERROR" responses include support for an additional reason message. All "ERROR" responses will include this additional reason. "SUCCESS" responses will include a "not found" message if the device did not return results.

```
{
    "requests": [{
            "requestcase": "97456596893834",
            "shortcodes": ["11111","22222"],
            "accountid": "30003748347",
            "type": "FORGET",
            "contacts": [{
                    "phone": "+1 781 555 1212"
                },
                {
                    "phone": "+1 617 555 1212"
                },
                {
                    "email": "test@test.com"
                },
                {
                    "email": "genesys@genesys.com"
                },
                {
                    "ipaddr": "10.10.10.10"
                },
                {
                    "ipaddr": "11.11.11.11"
```

```
                }
            ]
        },
        {
            "requestcase": "6457657657",
            "shortcodes": ["11111","22222"],
            "accountid": "30003748347",
            "type": "FORGET",
            "contacts": [{
                    "phone": "+1 781 555 1313"
                },
                {
                    "phone": "617 555 1313"
                },
                {
                    "email": "test2@test.com"
                },
                {
                    "email": "genesys2@genesys.com"
                },
                {
                    "ipaddr": "10.10.10.11"
                },
                {
                    "ipaddr": "11.11.11.12"
                }
            ]
        }
    ],
    "result": [{
            "requestcase": "97456596893834",
            "shortcodes": ["11111","22222"],
            "accountid": "30003748347",
            "type": "FORGET",
            "contacts": [{
                    "phone": "+1 781 555 1212",
                    "response": "SUCCESS"
                },
                {
                    "phone": "+1 617 555 1212",
                    "response": "SUCCESS"
                },
                {
                    "email": "test@test.com",
                    "response": "SUCCESS"
                },
                {
                    "email": "genesys@genesys.com",
                    "response": "SUCCESS"
                },
                {
                    "ipaddr": "10.10.10.10",
                    "response": "SUCCESS"
                },
                {
                    "ipaddr": "11.11.11.11",
                    "response": "SUCCESS"
                }
            ]
        },
        {
            "requestcase": "6457657657",
            "shortcodes": ["11111","22222"],
```

```
            "accountid": "30003748347",
            "type": "FORGET",
            "contacts": [{
                    "phone": "+1 781 555 1313",
                    "response": "SUCCESS"
                },
                {
                    "phone": "617 555 1313",
                    "response": "ERROR: incorrect device format"
                },
                {
                    "email": "test2@test.com",
                    "response": "SUCCESS"
                },
                {
                    "email": "genesys2@genesys.com",
                    "response": "SUCCESS"
                },
                {
                    "ipaddr": "10.10.10.11",
                    "response": "SUCCESS"
                },
                {
                    "ipaddr": "11.11.11.12",
                    "response": "SUCCESS"
                }
            ]
        }
    ]
}
```


Result File Content: Export Results

Export results will have the exact same name as submitted files except a "-archive" suffix will be appended and the results will be bundled in a .zip archive. Example name: export-20180315_120000-archive.zip

Any exported data that is derived from a list (active contact records or historical contact attempt records) will be exported as plaintext, comma-delimited CSV.

Export result archives WILL NOT include call recordings (though they are in scope for FORGET requests). To extract call recordings, customers should run one or both of the Script Recording or Client Recording reports via the Account Manager's "Reports" tab.

Customer Expectations & Assumptions

For "forget" or "export" requests, customers are expected to:

- Limit the set of "request initiators" to a small, qualified set of resources.
- Ensure a consumer's requested devices are accurately provided. By transposing a digit in a phone device, a "forget" request for an unintended consumer could be initiated OR possibly worse, "export" results for the wrong party could be produced which, if not reviewed before distribution, are a breach of privacy. Any processed "forget" requests will result in irreversible anonymisation for their devices, if found.

- Ensure expected execution-log files are received. Depending on number of requests in the posted payload, more than 1 execution-log may result. If no results are found, correct posted file names.
- Check execution-log files for error messages for any provided device and repost any corrected files as needed to completely satisfy their consumer's request.
  - For "forget" requests, translate execution-log results into a consumable confirmation to their consumers, if needed.
  - For "export" requests, review, filter, or transform archive results to ensure that only results for the intended consumer(s) are provided.
- Refrain from posting their consumer's GDPR requests or results to forums or portals (whether Genesys-provided or not) as doing so perpetuates PII on other platforms.
- Remove a consumer's data from future files posted to Genesys platforms. GDPR requests will result in a one-time removal as of that point in time and do NOT result in additions to device or clientID suppression/DNC. If suppression is additionally desired as a "safety check," customers already have UI or API methods to take that action.

Important

A "forget" request will result in an entity in scope (historical contact attempt record, list record, etc.) to be no longer searchable by device. This is because found devices are turned into unrecognizable placeholder equivalents. For this reason, the only way to confirm "forget" behavior after processing a request is to submit an "export" request or do a contact trace/search by ClientID to confirm that fields with PII have been redacted.

Genesys Engage On-Premises Support for GDPR

> **Important**
>
> For information on how Genesys Cloud compliance with GDPR, see Genesys Cloud and GDPR compliance page and for Genesys Engage cloud, see Genesys Engage cloud Support for GDPR.

The following list summarizes Genesys support for Right of Access (export) and Right of Erasure (forget)—across Genesys solutions and products. Products that potentially process but do not store PII are not included:

- Feature Server Support for GDPR
- Genesys Intelligent Automation Support for GDPR
- Genesys Interaction Recording and Analytics Support for GDPR
- Genesys Rules System Support for GDPR
- Genesys Voice Platform Support for GDPR
- Mobile Engagement Support for GDPR
- Predictive Routing Support for GDPR
- Web Services and Applications Support for GDPR
- Workspace Desktop Edition Support for GDPR
- Genesys CX Insights Support for GDPR
- Genesys Info Mart Support for GDPR
- Universal Contact Server Support for GDPR
- intelligent Workload Distribution Support for GDPR
- Outbound Contact Support for GDPR

Universal Contact Server Support for GDPR

This page describes product-specific aspects of Universal Contact Server support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

Dependencies

Universal Contact Server (UCS) has no dependency on other products for the management of contacts and interactions. However the following products depend on UCS for contact and interaction data:

- Interaction Server has interaction data related to UCS contacts.
- Genesys Mobile Services (GMS) uses Context Service profiles to look up GMS callback requests.
- Genesys Info Mart uses interaction data gathered by Interaction Concentrator (ICON) during routing.

Before deleting data in UCS, ensure that the Genesys Info Mart ETL cycle is complete and any pending GMS and Interaction Server activities are completed.

Exporting Data

The data export format is JSON. Functionality is provided by a custom Java command-line application. The application will export contact and interaction data in JSON format. Attachments will be inlined in interactions using base64 format. The script to export contacts is named `exportUcsInteractions.bat` (`.sh`).

Command-line parameters

- The script `exportUcsInteractions.bat` (`.sh`) must be edited to provide the UCS `HOST` and `PORT`.
- A contact ID or comma-separated list of contact IDs surrounded with quotation marks (mandatory if no contact identification request is provided).
- A contact identification request in format JSON surrounded with quotation marks (mandatory if no

contact ID provided). In the JSON string the quotation marks must be escaped.

- `-count` (optional)—if used, provides the number of contacts and interactions that will be exported without doing the export (simulation mode).

Command-line examples

- `exportUcsInteractions.sh 0001SaD2PVY4312K -count`
- `exportUcsInteractions.sh 0001SaD2PVY4312K`
- `exportUcsInteractions.sh "0001SaD2PVY4312K,0001SaD2PVY4315J" -count`
- `exportUcsInteractions.sh "0001SaD2PVY4312K,0001SaD2PVY4315J"`
- `exportUcsInteractions.sh "{\"TenantId\":101,\"EmailAddress\":\"jane.doe@company.com\"}" -count`
- `exportUcsInteractions.sh "{\"TenantId\":101,\"EmailAddress\":\"jane.doe@company.com\"}"`

The result files are generated in the folder where the export script is running. The result will have the following structure where `0001SaD2PVY4312K` is the contactId:

```
contact-0001SaD2PVY4312K.json
interactions-for-contact-0001SaD2PVY4312K.json
```

Implementing Forget Me

This functionality is provided for all existing UCS versions by the PSDK contact RequestDelete (contact) API. Deleting a contact using this API will delete all contact entities as well as all interaction entities regardless of their states (in progress or not).

Deleting individual contacts—On Premise

To delete an individual contact, use the following procedure:

Step #1—Verify the contact and the request

1. The customer requests deletion using media such as email, chat or voice. This incoming interaction is routed to an agent who can either capture the information on the caller or perform the deletion while connected.
2. The Agent makes sure the request is valid according to the contact center's policy and that the deletion request is genuine.
3. The Agent may confirm to the customer that deletion is about to take place and warn that no further communication will be made. The customer must understand that if he/she communicates with the Call Center again their data will be recreated.
4. If the Agent is not currently on an interaction with a customer requesting to be forgotten, or the Contact History is not currently showing, they will need to find the customer in the Contact History using this procedure:
   - Finding and viewing an interaction in the contact database

> **Important**
> Be aware that there may be multiple contacts with the same name and/or multiple contact records for the same person.

5. The Agent disconnects from chat, voice or makes sure that a confirmation email has been sent.

Step #2—Close any in-progress interactions

1. Having located the appropriate contact, the Agent then reviews their contact history for in-progress interactions. See Step #8 of Finding and viewing an interaction in the contact database.
2. If interactions are in progress, such interactions must be stopped. Agents must remove all interactions from workbins.

Step #3—Clean up the Context Services database
If Conversation Manager is installed, any Service/State/Task records in the Context Services database must be deleted. To do this use the Delete Customer Profile API.

Step #4—Delete the contact in UCS:
If the Agent is using Workspace Desktop and has permission to manage contacts, then they delete the contact using one of the methods described at the links below:

- Deleting a contact (WDE User Guide)
- Deleting contacts (WDE Help)
- Contact actions (WWE Help)

Deleting individual contacts—Cloud

To delete an individual contact, use the following procedure:

Step #1—Verify the contact and the request

1. The customer requests deletion using media such as email, chat or voice. This incoming interaction is routed to an agent who can either capture the information on the caller or perform the deletion while connected.
2. The Agent makes sure the request is valid according to the contact center's policy and that the deletion request is genuine.
3. The Agent may confirm to the customer that deletion is about to take place and warn that no further communication will be made. The customer must understand that if he/she communicates with the Call Center again their data will be recreated.
4. If the Agent is not currently on an interaction with a customer requesting to be forgotten, or the Contact History is not currently showing, they will need to find the customer in the Contact History using one of the procedures described here:
   - Where are my contacts?
   - Quick Search

> **Important**
> Be aware that there may be multiple contacts with the same name and/or multiple contact records for the same person.

5. The Agent disconnects from chat, voice or makes sure that a confirmation email has been sent.

Step #2—Close any in-progress interactions

1. Having located the appropriate contact, the Agent then reviews their contact history for in-progress interactions—see Contact and interaction history
2. If the Agent finds any interaction listed as "In Progress" they must wait until it is routed and handled before proceeding to delete the contact.

Step #3—Delete the contact
If the Agent is using Workspace Desktop and has permission to manage contacts, then they delete the contact using the procedures described here:

- What actions can I take with a contact?

Batch deletion of contacts

Batch deletion named `deleteContact.bat` (`.sh`) deletes contacts with their interactions.

Command-line parameters

- The script `deleteContact.bat` (`.sh`) must be edited to provide the UCS `HOST` and `PORT`.
- A contact ID or comma separated list of contact IDs surrounded with quotation marks (mandatory if no contact identification request is provided).
- A contact identification request in format JSON surrounded with quotation marks (mandatory if no contact ID provided). In the JSON string the quotation marks must be escaped.
- -confirm: mandatory to perform the contact deletion. If not provided the script only reports the contacts found and their attributes.

Command-line examples

- `exportUcsInteractions.sh 0001SaD2PVY4312K`
- `exportUcsInteractions.sh 0001SaD2PVY4312K -confirm`
- `exportUcsInteractions.sh "0001SaD2PVY4312K,0001SaD2PVY4315J"`
- `exportUcsInteractions.sh "0001SaD2PVY4312K,0001SaD2PVY4315J" -confirm`
- `exportUcsInteractions.sh "{\"TenantId\":101,\"EmailAddress\":\"jane.doe@company.com\"}"`
- `exportUcsInteractions.sh "{\"TenantId\":101,\"EmailAddress\":\"jane.doe@company.com\"}"`

```
    -confirm
```

Downloadable script

> **Important**
> The downloadable scripts referenced below support both UCS 8.5 and UCS 9.1.

The script (download using this link) is provided for exporting interactions linked to a contact.

To use it:

1. Unzip the file to a target directory and cd to this directory.
2. Edit the script file to match your requirements (such as UCS host, UCS port, contact IDs, and so on).
3. Launch the script.

Windows example

```
deleteContact.bat
Mon May 21 08:34:16 CEST 2018: Starting interaction export
Mon May 21 08:34:16 CEST 2018: Export path is \tmp
Mon May 21 08:34:16 CEST 2018: Directory exists: C:\tmp
Mon May 21 08:34:16 CEST 2018: Connecting to UCS at 192.168.1.2:8889
Mon May 21 08:34:17 CEST 2018: Retrieve contact '0002Ha7QB2MX000K'
Mon May 21 08:34:17 CEST 2018: Found contact Id '0002Ha7QB2MX000K'
Mon May 21 08:34:17 CEST 2018: Saving contact info to C:\tmp\contact-0002Ha7QB2MX000K.json
Mon May 21 08:34:17 CEST 2018: Saving interactions to C:\tmp\interactions-for-
contact-0002Ha7QB2MX000K.json
Mon May 21 08:34:17 CEST 2018: Found interactions for contact '0002Ha7QB2MX000K'
Mon May 21 08:34:31 CEST 2018: Successfully exported all interactions for contactId
0002Ha7QB2MX000K from DataSource main
Mon May 21 08:34:31 CEST 2018: Saving interactions to C:\tmp\interactions-for-
contact-0002Ha7QB2MX000K.json
Mon May 21 08:34:31 CEST 2018: No interactions found for contact Id'0002Ha7QB2MX000K' in
source archive
Mon May 21 08:34:31 CEST 2018: Reason: 06:34:30.964 Server: ucs85-49d888d5a732 Msg: No data
source with role 'archive' found
Mon May 21 08:34:31 CEST 2018: Done
```

Linux example

```
$ ./ExportUcsInteractions.sh
Mon May 21 08:31:15 CEST 2018: Starting interaction export
Mon May 21 08:31:15 CEST 2018: Export path is C:\Users\username\AppData\Local\Temp
Mon May 21 08:31:16 CEST 2018: Directory exists: C:\Users\username\AppData\Local\Temp
Mon May 21 08:31:16 CEST 2018: Connecting to UCS at 192.168.1.2:8889
```

```
Mon May 21 08:31:17 CEST 2018: Retrieve contact '0002Ha7QB2MX000K'
Mon May 21 08:31:17 CEST 2018: Found contact Id '0002Ha7QB2MX000K'
Mon May 21 08:31:17 CEST 2018: Saving contact info to C:\Users\username\AppData\Local\Temp\
contact-0002Ha7QB2MX000K.json
Mon May 21 08:31:17 CEST 2018: Saving interactions to C:\Users\username\AppData\Local\Temp\
interactions-for-contact-0002Ha7QB2MX000K.json
Mon May 21 08:31:17 CEST 2018: Found interactions for contact '0002Ha7QB2MX000K'
Mon May 21 08:31:30 CEST 2018: Successfully exported all interactions for contactId
0002Ha7QB2MX000K from DataSource main
Mon May 21 08:31:30 CEST 2018: Saving interactions to C:\Users\username\AppData\Local\Temp\
interactions-for-contact-0002Ha7QB2MX000K.json
Mon May 21 08:31:30 CEST 2018: No interactions found for contact Id'0002Ha7QB2MX000K' in
source archive
Mon May 21 08:31:30 CEST 2018: Reason: 06:31:30.793 Server: ucs85-49d888d5a732 Msg: No data
source with role 'archive' found
Mon May 21 08:31:30 CEST 2018: Done
```

Outbound Contact Support for GDPR

This page describes product-specific aspects of Outbound Contact support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

Logging

Sensitive data masking in logs

Behavior of the OCS component in relation to handling PII data in the logs should be configured with a set of options defined in sections **[log-filter]** and **[log-filter-data]**.

The **[log-filter]** section defines the default treatment of filtering data in log output. It defines the treatment of all KV pairs in the User Data, Extensions, and Reasons attributes of the log, and also defines the behavior of selected call handling (such as T-Servers) and reporting applications when processing call related data.

The **[log-filter-data]** section defines the treatment of specific KV pairs in the User Data, Extensions, and Reasons attributes of the log. It overrides the general settings in the **[log-filter]** section. This section defines how the set of keys in User Data, Extensions, and Reason should be handled when they are printed out into the log files.

Refer to Common Configuration Options section for full details on sensitive data masking in logs.

Log rotation

Logging should be configured with the “expire” option that defines if the log files will expire, and if so, the maximum number of days before log files are deleted.

Sensitive data masking in log files and log retention implements the Privacy by Design GDPR requirement.

Refer to Common Configuration Options section for full details on log rotation.

Handling PII data

The PII data with which OCS operates consists of phone numbers, company names, and any user data that can be dynamically formed based on the business process. A company that implements the structure of the user data based on its business process should care about handling this user data. OCS stores this PII data in databases in the form of calling lists; the PII data could also appear in different log files of the OCS component itself.

The following table summarizes OCS sources which could contain PII data:

Source	Form of storage	PII data
Calling List(s)	Table in the relational database	Potentially any type of PII data, stored in user-defined fields
Application Logs	Flat file(s)	Potentially any type of PII data
Audit Trail Logs	Flat file(s)	Phone numbers
Do Not Call List(s)	Table in the relational database	Phone numbers, Customer IDs
GSW Request Log(s)	Table in the relational database	Phone numbers
Record History Log(s)	Flat files(s)	Potentially any type of PII data, stored in user-defined fields

Databases

OCS uses databases for two types of entities--Calling and Do Not Calling Lists, and GSW Request Logs. Database administrators should follow general rules for maintaining GDPR-compliant databases. The general approach is as follows.

- Design data location – operating systems, primary and backup nodes.
- Design data access – limiting personal data access to as few as possible persons and roles.
- Design data storage – different storage systems provide number of mechanisms allows to store sensitive data securely. It could be full or part data encryption, secure protocols, and so on.

All these items implement Privacy By Design GDP requirement for Calling List.

If there are databases that are not encrypted and contains PII data, these databases should be reviewed for sensitive data and corresponding records should be either modified or removed.

OCS database:

1. OCS Calling Lists – these contain phone numbers and user-defined fields which could contain PII
2. Do Not Call lists
3. GSW Request Logs

Handling Requests

Find / Export PII data

- Find/Export all records in the Calling List where phone number equals given:

```
SELECT * FROM <cl_table_name> WHERE contact_info LIKE '<phone number>'
```

- Find/Export all records in Calling List where user_field contains given identifier

```
SELECT * FROM <cl_table_name> WHERE <user_field> LIKE '%<identifier>% '
```

- Find/Export data in Do Not Call Lists where phone number equals given.

```
SELECT * FROM <dnc_table_name> WHERE phone LIKE '<phone number>'
```

- Find/Export data in Do Not Call Lists where Customer ID equals given.

```
SELECT * FROM <dnc_table_name> WHERE customer_id LIKE '<identifier> '
```

- Find/Export data in GSW Request Log where phone number equals given.

```
SELECT * FROM <rl_table_name> WHERE phone LIKE '<phone number>'
```

Edit PII data For archived Calling Lists, it can be done using SQL queries.

For example, the following SQL statement will mask phone numbers in the GSW Request Log where phone number matches a given number.

```
UPDATE <rl_table_name> SET phone = '***' WHERE phone LIKE '<phone number>'
```

A similar SQL statement could be used for masking PII data in the Calling List:

```
UPDATE <cl_table_name> SET contact_info = '***', <user_field> = '***' WHERE contact_info LIKE
'<phone number>'
```

Note, it is not recommended to update records that are retrieved or may be retrieved by OCS. To avoid updating such records, add the following clause to the WHERE part of the SQL statement above:

```
record_status NOT IN (2)
```

SQL Statement for Do Not Call List:

```
UPDATE <dnc_table_name> SET phone = '***', customer_id = '***' WHERE phone LIKE '<phone
number>'
```

Note, updates in the Do Not Call List table will not update data in the OCS memory immediately. This happen only when the Do Not Call List table is re-read by OCS (refer to https://docs.genesys.com/Documentation/OU/8.1.5/Dep/CallingLists#Rereading_of_the_Do-Not-Call_List).

Delete PII dataSQL query example for individual entries deletion based on phone number or unique ID stored in the user-defined field:

```
DELETE FROM <cl_table_name> WHERE contact_info LIKE '<phone number>'

DELETE FROM <cl_table_name> WHERE <user_field> LIKE '%<identifier>% '
```

OCS Log files

All log files should be checked to determine if they contain any PII data. If any is found, it should be

either masked or removed from the files.

Log files:

- Main OCS and OCS HTTP proxy logs
- OCS Audit Trail logs
- Record History logs

Handling Log Files

Find / Export data in log files

To find and/or export PII data in a log file, simple console utilities like grep can be used.

For example, using the grep utility with regexp request for a log file will find all strings that contains a given string, such as `SocialSecurityNumber: 123456789` that is not masked, such as `SocialSecurityNumber: grep -n -e "SocialSecurityNumber: \([0-9]\) " OCSLogFile.log`

The found data could be easily exported using redirecting of the output into the file instead of the standard output:

grep -n -e "SocialSecurityNumber: \([0-9]\) " OCSLogFile.log > ExportedData.txt

It implements Right of Access and Portability GDP requirement for log files.

Edit data in log files

To mask the PII data in the log files, find and edit procedures should be implemented for the log files. It can be done with some already-existing tools, some special tools designed for this purpose, or some general tools like SED in Linux based systems.

For example, using the SED utility with regexp request will update the log file in place and all strings like `SocialSecurityNumber: 123456789` will be changed to strings like `SocialSecurityNumber: ***`.

```
sed -i -e 's/SocialSecurityNumber: \([0-9]*\)/SocialSecurityNumber: ***/g'
OCSLogFile.log
```

Delete data from log files

Deleting of the PII data from the log files also should be implemented using find and edit procedures. It can also be done with special or general tools.

For example using the SED utility with regex request will completely remove the string where the pattern `SocialSecurityNumber: 123456789` has been found.

```
sed -i -e 's/^.*SocialSecurityNumber: \([0-9]*\).*$//g' tst.txt
```

The following example looks in all log files for the pattern `SocialSecurityNumber: 123456789` and deletes these files.

```
find ./ -iname "*.log" -exec grep -e "SocialSecurityNumber: \([0-9]\)" '{}' \;
-delete
```

Genesys Administrator Support for GDPR

This page describes product-specific aspects of Genesys Administrator support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

Deleting Agents and Users

The following pages describe the process to delete agent and user information:

- Deleting Agents
- Bulk Import/Export of Agent Data
- Users (Persons)

> **Warning**
> "Usage of Genesys Administrator by an employee requires processing of the employee’s Personal Data for functioning of the Genesys Administrator(user’s name, work phone number, and work email). Without storing this Personal Data associated with an employee, the Genesys Administrator could stop functioning. Thus, for current employees, the processing of their Personal Data is necessary for the purposes of the legitimate interests pursued by the [CUSTOMER]. Further, [CUSTOMER] may be required to keep employee call records in order to meet other regulatory requirements. Based on the lawfulness of this processing and the design of the Genesys Administrator, we do not recommend erasing Personal Data associated with an ongoing user."

Genesys CX Insights Support for GDPR

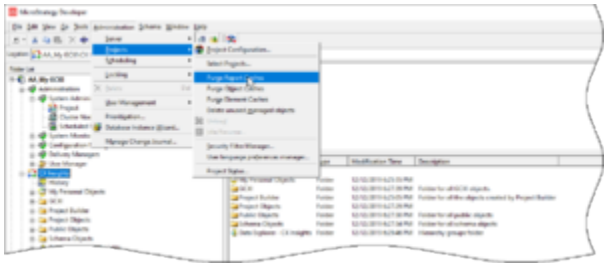
Genesys provides the following information to help you ensure that you handle Personal Identifiable Information (PII) in accordance with General Data Protection Regulation (GDPR) standards.

Overview of Genesys CX Insights support for GDPR compliance

Genesys Customer Experience Insights (Genesys CX Insights or simply GCXI) can store Personal Identifiable Information (PII) in logs, history files, and in reports (in scenarios where customers include PII data in reports). Genesys recommends that you do not capture PII in reports, however, if you do so, it is your responsibility to remove any such report data within 21 days or less, if required by General Data Protection Regulation (GDPR) standards. Note that work email addresses are often used in reports, but are not considered PII.

Deleting PII from reports

If your reports contain PII, perform the following steps to remove old report results:



Purge Report Caches

1. Delete any saved reports that contain PII.
2. Purge the GCXI report cache:
    1. Open MicroStrategy Developer, log in, and select your project.
    2. Select the **CX Insights** project.
    3. Click **Administration** > **Projects** > **Purge Report Caches**.

Clearing PII from logs and history

To remove PII from logs and history files, complete the following steps:

1. Clean up logs — Periodically delete the following logs, which can include login names:

Microstrategy logs:

- `/mnt/log/mstr/DistributionService_DeliveryDetails.log`
- `/mnt/log/mstr/AuthenticationServer_Trace.log`
- `/mnt/log/mstr/Kernel_UserTrace.log`
- `/mnt/log/mstrWeb/MSTRLog<date>.log`
- `/mnt/log/tomcat/localhost_access_log.<date>.txt`

GCXI utils logs:

- `/mnt/log/gcxi/com.genesys.jdbc.driver.log`

2. Configure the maximum size and age of the history list, at the project level. For detailed steps to manage the history list, see the MicroStrategy website for information about:
   - Controlling the maximum size of the History List
   - Controlling the lifetime of History List messages (in days)
   - Scheduling History List message deletion

Other steps

Genesys does not support or recommend using cubes. If you do use them, be aware that, depending on the data uploaded, cubes can contain PII data which you might need to delete in order to comply with GDPR requirements.

Predictive Routing Support for GDPR

This page describes product-specific aspects of Predictive Routing support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

Genesys Predictive Routing (GPR) provides you the ability to do the following actions, in compliance with GDPR requirements:

- Export Personally Identifiable Information (PII)
- Remove PII ("Forget me")

> **Important**
> GDPR compliance is fully managed by the client. GPR provides the necessary endpoints, but compliance requires you to perform the steps required to locate and remove PII.

- For additional details, refer to the *Predictive Routing API Reference* (access requires a password; contact your Genesys representative for assistance).

Locating and Handling PII via the GPR API

Use the following procedure to read and delete PII:

1. Specify the unique field (Field) that identifies the person making a GDPR request (Person).
2. Find the data structures where the Person's data might be located:
   - Agent Profile schema
   - Customer Profile schema
   - Datasets (there might be multiple datasets)
   - Predictors (there might be multiple predictors)
   - (Optional) Accounts and user-management data might also contain PII. However, note that changes

to these types of data might interfere with GPR operations, because accounts and users are required for administration.

3. Compose an API request directed to each data structure containing PII for the Person.

> **Important**
>
> Refer to Table of API Commands Used to Handle PII in GPR (below) and choose the correct command syntax for each data structure.

4. (Optional) Before removing PII, execute a data export request with the same filter to ensure you are about to remove the right data.
5. Execute the request. In case of removal, the request removes the entire row (document) that matches the filter from the selected data structure.

Example

This example demonstrates how to find and remove information about the customer with the email address `johndoe@example.com` from the data stored in a predictor.

To start, inspect the data structure to find the relevant field:

```
curl \
    --request GET \
    "https://localhost/api/v2.0/predictors/{id}?token={token}"
```

The result indicates that the email address is stored in a field called `customer_email`.

Next, export the PII associated with the email address from the specified predictor:

```
curl \
    --request GET \
    "https://localhost/api/v2.0/
predictors/{id}/data?token={token}&filter=%28ctx.customer_email%3Djohndoe%40example.com%29"
```

> **Important**
>
> - Customer fields have the prefix `ctx` and agent fields have the prefix `act`. These prefixes exist only in the predictor data structure.
> - As shown In the example above, the value of the filter parameter in the query of a GET request (in this example, `ctx.customer_email=johndoe@example.com`) must be URL (percent) encoded.

To remove the PII from this predictor, use the following command:

```
curl \
    --request POST \
    --header "Content-Type: application/json" \
```

```
--data '{"data_filter": "(ctx.customer_email=johndoe@example.com)"}' \
"https://localhost/api/v2.0/purge/predictors/{id}?token={token}"
```

Important

- PII removal is a permanent (hard) delete operation. You cannot restore deleted data.
- PII removal is an asynchronous job. Check the job status to ensure that the job has removed the PII successfully.
- PII removal deletes the entire row (document) matching the filter. If you filter the data only by the email field, other fields are removed as well.

Table of API Commands Used to Handle PII in GPR

Data	Read	Delete
Agent profiles	GET /agents?ID={id}	POST /purge/agents + body {"data_filter": "({field}={value})"}
Customer profiles	GET /customers?ID={id}	POST /purge/customers + body {"data_filter": "({field}={value})"}
PII in datasets	GET /datasets/{id}/data?filter=({field}={value})	POST /purge/datasets/{id} + body {"data_filter": "({field}={value})"}
PII in predictors	GET /predictors/{id}/data?filter=({field}={value})	POST /purge/predictors/{id} + body {"data_filter": "({field}={value})"}

Genesys Info Mart Support for GDPR

Overview of Genesys Info Mart support for GDPR compliance

Starting with the initial 8.5.010 release, Genesys Info Mart processes input JSON files that customers provide to comply with Right of Access ("export") or Right of Erasure ("forget") requests from their customers (*consumers*). Starting with release 8.5.010.16, Genesys Info Mart also processes GDPR requests that customers provide from their employees, who are contact center agents and supervisors.

Genesys Info Mart uses the JSON files customers place in configured, tenant-specific locations (see Input file location). The JSON files identify the consumers or employees who have made GDPR requests to either "export" or "forget" their personally identifiable information (PII). The Info Mart database potentially stores requesting consumers' and employees' PII.

The daily Info Mart database maintenance job, Job_MaintainGIM, processes any "export" or "forget" JSON files that have been added or modified since the last processed JSON file. To execute the requests, the maintenance job uses a SQL script, **gdpr.sql** or **gdpr_partitioned.sql**, located in the **sql_scripts** folder in the installation directory.

- For "export" requests, the applicable PII is reported in a database table, CTL_GDPR_HISTORY.
- For "forget" requests, the data is redacted in Info Mart tables (primarily fact tables), and the PII that was searched for (in other words, the data prior to redaction) is reported in the CTL_GDPR_HISTORY table.

The PII is stored in the CTL_GDPR_HISTORY table for a configurable amount of time (maximum 30 days). You can query the CTL_GDPR_HISTORY table to obtain:

- PII data to satisfy an "export" request (see Example: SQL for export)
- A detailed audit trail of all the fields that were interrogated to satisfy the GDPR requests (see Example: SQL for audit)

To query the CTL_GDPR_HISTORY table, you must have read-only privileges for the Info Mart schema.

> **Important**
>
> No special configuration is required on the Genesys Info Mart side to enable support for GDPR. Equally, no special actions are required as long as the maintenance job runs regularly. Genesys recommends that you do not disable the run-maintain option, which enables the maintenance job to run on an automatic schedule.

Data retention policies

A number of configuration options control how long data is retained in the Info Mart database. The following table summarizes the applicability of GDPR to different types of Info Mart data. For more information about the categories of data in the Info Mart database, see the "Genesys Info Mart Database Schema" page in the *Genesys Info Mart Physical Data Model* for your RDBMS.

Important

Genesys Info Mart support for GDPR compliance is based on default configuration settings and typical application usage. For example, given the recommended values of the **days-to-keep-*** options, fact data in GIDB tables and logs is short-lived (ephemeral) data that is automatically deleted within a short period of time, and Genesys Info Mart does not redact this data outside of the automated deletion.

Particularly in multimedia deployments with long-living interactions, Genesys recommends that you reduce the values of certain options that affect purge behavior. See Recommendations for purge-related options for multimedia for more information.

Configuration option	Controls what type of data?	Contains PII?	Included in GDPR processing?
Never deleted—not configurable	Configuration object data	Yes, for employees	Yes (starting with release 8.5.010.16)
gim-etl section			
days-to-keep-gim-facts	Fact data in the dimensional model	Yes	Yes. **Note:** Dimension tables in the dimensional model are never purged. Dimensions store low-cardinality data that does not contain PII.
days-to-keep-active-facts	Active multimedia interaction data in the dimensional model, GIDB, and Staging tables	Yes	Active interaction data in dimensional model fact tables is processed. Active interaction data in GIDB and Staging tables is short-lived and is not processed.
days-to-keep-gidb-facts	Fact data in GIDB	Yes	No. See Important note, above.
days-to-keep-gdpr-history (introduced to support GDPR)	The CTL_GDPR_HISTORY table	Yes	Not considered for GDPR reprocessing, because the data is short-lived (maximum 30 days).
days-to-keep-cfg-facts	Deleted configuration fact data	No consumer- or employee-related PII	No

Configuration option	Controls what type of data?	Contains PII?	Included in GDPR processing?
days-to-keep-discards-and-job-history	Discard tables, and audit and history tables	No	No
gim-export section			
days-to-keep-output-files	Files generated by Job_ExportGIM to provide Info Mart data to customers who use Genesys Info Mart's Data Export feature	Yes	No. See Important note, above. **Note:** Customers are responsible for implementing adequate processes to ensure that any PII in their imported data is handled in accordance with GDPR requirements, including using suitable retention periods or redacting data to comply with "forget" requests.
log4j section			
max-backup-index	Log files	Yes	No. See Important note, above.

Recommendations for purge-related options for multimedia

Particularly in multimedia deployments with long-living interactions, ensure that the **days-to-keep-*** options that control retention policies for short-lived data that is not included in GDPR processing provide a sufficient buffer for unexpected delays—for example, if a database purge was not executed because the maintenance job was interrupted. In deployments where the Info Mart database is partitioned, you must also factor in partition sizes, since a partition is not purged until all the data it contains is eligible to be purged.

Genesys recommends that you base option settings on the following calculations:

- **days-to-keep-active-facts** = 30 – <buffer> – "partitioning-interval-size-gidb-mm"/24/3600
- **days-to-keep-gidb-facts** = 30 – <buffer> – max("partitioning-interval-size-gidb", “partitioning-interval-size-gidb-mm”, “partitioning-interval-size-gidb-ocs”)/24/3600

Example

If the partition size for multimedia interactions in GIDB has been increased to the nondefault value of one week (`partitioning-interval-size-gidb-mm=604800`), the partition size for Outbound Contact–related data in GIDB has been increased to the nondefault value of ten days (`partitioning-interval-size-gidb-ocs=864000`), and you want to allow a buffer of three days for unexpected delays, set:

- `days-to-keep-active-facts = 30 – 3 – 604800/24/3600 = 20`
- `days-to-keep-gidb-facts = 30 – 3 – 864000/24/3600 = 17`

Input file location

You specify the location for each tenant's JSON files in the **[gdpr].gdpr-directory** option on the Annex tab of the Tenant configuration object. Genesys has no special requirements for the location of the directory. The **gdpr-directory** option must simply specify a valid path that both Genesys and you can get to.

All Genesys products use the same tenant-specific directory for the input and, if the product provides them (Genesys Info Mart does not), output files for that tenant. You are responsible for maintaining this directory.

JSON input files

There are separate input files for Right of Erasure and Right of Access requests:

- forget-<DDMMYYYY>-<any optional content>.json
- export-<DDMMYYYY>-<any optional content>.json

The date part of the file name (<DDMMYYYY>) indicates the date the file was created, to maintain file uniqueness. Using timestamps in the file system, Genesys Info Mart processes any files added or modified for that tenant since the last time Genesys Info Mart processed GDPR requests.

It is your responsibility to ensure that the request does not conflict with other regulatory or legal obligations.

File specification

The JSON specification for the forget and export files for GDPR requests is identical.

- "caseid" — (Optional) Holds customer case numbers, for possible use by Customer Care to supplement customer self-service.
- "consumers" — (Required for consumer requests) Holds an array of individual "consumer" elements, so that GDPR requests from multiple consumers can be processed at the same time.
  - "consumer" — (Required) An individual consumer for whom a GDPR request is being submitted. Each consumer may be identified by one or more of the following attributes, specified in an array:
    - "phone" — Phone number, without separators
    - "email" — Email address
    - "fbid" — Facebook ID
    - "twid" — Twitter handle
    - "wcid" — WeChat ID
    - "name" — Given name
    - "ipaddr" — IP address

- "employees" — (Required for employee requests) Holds an array of individual "employee" elements, so that GDPR requests from multiple employees can be processed at the same time.
  - "employee" — (Required) An individual employee for whom a GDPR request is being submitted. Each employee may be identified by one or more of the following attributes, specified in an array:
    - "username" — (Required) Username of the person object in Configuration Server and/or Outbound Engagement configuration
    - "employeeid" — Employee ID of the person object in Configuration Server
    - "name" — Given name
- "gim-attached-data" — (Optional) Used by Genesys Info Mart to target custom user data attached to interactions and custom Outbound Contact Server (OCS) fields used in Outbound Contact campaigns. Custom user data and custom fields contain data for which customers configured customized storage in the Info Mart database.
  - "kvlist" — Holds an array of the custom user data KVPs and custom OCS fields that might contain PII.

Example

```
{
   "caseid":"123456789",
   "consumers":[
      {"consumer":
         [
            {"name":"John Doe"},
            {"name": "John Q. Doe"},
            {"phone":"555551212"}
         ]
      },
      {"consumer":
         [
            {"name":"Dan Akroyd"},
            {"phone":"555556161"},
            {"phone":"555556162"},
            {"email":"danny@hollywood.com"},
            {"email":"funnyguy@comedy.org"},
            {"fbid":"Dan Akroyd"}
         ]
      }],
    "gim-attached-data":
         {"kvlist": ["AcctNum", "SSN"]
         },
   "employees":[
      {"employee":
         [
            {"username":"SueSmith"},
            {"name": "Sue Smith"},
            {"employeeid":"RR11243"}
         ]
      }]
 }
```

Right of Access ("export") requests

Genesys Info Mart uses the GDPR input files named **export-<DDMMYYYY>-<any optional**

content>.json as the input for "export" processing. See Input files for details about the JSON file requirements. The PII that Genesys Info Mart will report is specified in the input JSON files in:

- The phone and email attributes that identify the requesting consumer
- The username attribute that identifies the requesting employee
- Custom user data KVPs and custom Outbound Contact Server (OCS) fields customers might specify in the "gim-attached-data" element

While custom KVPs and fields are included in the GDPR output, Genesys Info Mart searches only on the phone or email address in order to find fact table records associated with the requesting consumers. Similarly for employee requests, while employee ID, name, and other attributes in the configuration record (for example, email address) are included in the GDPR output, Genesys Info Mart searches only on the username in order to find configuration object and fact table records associated with the requesting employees. For details about the specific tables and fields that are searched, see the description of the CTL_GDPR_HISTORY table in the *Genesys Info Mart Physical Data Model* for your RDBMS.

In the initial implementation, Genesys Info Mart does not provide an output JSON file. Instead, the PII data is reported in the CTL_GDPR_HISTORY table.

Example: SQL query for "export"

The following is an example of SQL you can use to retrieve PII for export. The example returns distinct occurrences of consumer PII data for phone number 5551212. The same query specifying `CONSUMER_ID = 'jsmith'` will return distinct occurrences of employee PII data for the agent with username *jsmith*.

```
SELECT TENANT_KEY,
       FORGET,
       CONSUMER_ID,
       TABLE_NAME,
       COLUMN_NAME,
       KEY_VALUE
FROM CTL_GDPR_HISTORY
WHERE TENANT_KEY = <tenant>
AND FORGET = 0
AND CONSUMER_ID = '5551212'
AND KEY_VALUE IS NOT NULL
AND CREATED_TS BETWEEN <TS_1> AND <TS_2>
GROUP BY TENANT_KEY, FORGET, CONSUMER_ID, TABLE_NAME, COLUMN_NAME, KEY_VALUE
ORDER BY TENANT_KEY, FORGET, CONSUMER_ID, TABLE_NAME, COLUMN_NAME, KEY_VALUE
```

Right of Erasure ("forget") requests

Genesys Info Mart uses the GDPR input files named **forget-<DDMMYYYY>-<any optional content>.json** as the input for "forget" processing. See Input files for details about the JSON file requirements. Genesys Info Mart processing of "forget" requests parallels "export" requests, except that any PII that is found is redacted in the applicable Info Mart database tables, with the PII that was searched for (in other words, the data prior to redaction) recorded in the CTL_GDPR_HISTORY table.

As is the case for "export" requests, the PII that Genesys Info Mart will redact is specified in the input

JSON files in:

- The phone and email attributes that identify the requesting consumer
- The username attribute that identifies the requesting employee
- Custom user data KVPs and custom Outbound Contact Server (OCS) fields customers might specify in the "gim-attached-data" element

While custom KVPs and fields are redacted, Genesys Info Mart searches only on the phone or email address in order to find fact table records associated with the requesting consumers. Similarly for employee requests, while employee ID, name, and other attributes in the configuration record (for example, email address) are redacted, Genesys Info Mart searches only on the username in order to find configuration object and fact table records associated with the requesting employees. For details about the specific tables and fields that are searched, see the description of the CTL_GDPR_HISTORY table in the *Genesys Info Mart Physical Data Model* for your RDBMS.

> **Warning**
> Usage of Genesys products requires processing of employees' Personal Data (user's name, work phone number, and work email) for proper functioning of the Genesys solution. Without storing this Personal Data associated with an employee, Genesys Info Mart could stop performing its function. Thus, for current employees, the processing of their Personal Data is necessary for the purposes of the legitimate interests pursued by the customer. Further, the customer may be required to keep employee interaction records in order to meet other regulatory requirements. Based on the lawfulness of this processing and the design of Genesys products, Genesys does not recommend erasing Personal Data associated with an ongoing user.

Audit and tracking

The Genesys Info Mart log tracks the progress of GDPR processing.

Sample Genesys Info Mart log (click to enlarge)

In addition, there are two log events that report the status of GDPR request processing, by tenant: STANDARD-level log messages 55-20172 (GIM_ETL_GDPR_ERROR) and 55-20173 (GIM_ETL_GDPR_SUCCESS). (In Genesys Info Mart releases earlier than 8.5.011.18, GIM_ETL_GDPR_SUCCESS was a TRACE-level message, 55-31406.) Genesys recommends setting an alarm on message 55-20172, to alert you to problems that are preventing completion of GDPR processing. You can use log message 55-20173 (or 55-31406 in pre-8.5.011.18 releases) to cancel the alarm.

The CTL_GDPR_HISTORY table as audit report

Genesys Info Mart does not provide an execution report in the form of an output file. Instead, the CTL_GDPR_HISTORY table provides details about the PII data associated with “export” or “forget” requests. The table also provides a detailed audit trail of all the fields that were interrogated to satisfy a particular GDPR request. NULL values indicate that the field was evaluated for a particular instance of PII and was found to be empty.

Consider the following example:

- An input JSON file for a specific tenant indicates that the consumer with phone number 5551212 wishes to see PII data related to that phone number. The input JSON file also specifies custom user data keys, which, based on business practices in the customer's environment, might be associated with PII data.
- To satisfy this request, Genesys Info Mart interrogates interactions to or from 5551212, as well as related facts, such as user data related to those interactions. CTL_GDPR_HISTORY would have a row for each table/column searched.
- Say there were no interactions to or from 5551212. In this case, the row for each table searched for this phone number would have a NULL FACT_ID and a NULL KEY_VALUE.
- Say there were interactions to or from 5551212. However, in an associated record, a field populated by one of the keys specified in the JSON file is empty. In this case, the row for that table and column would have a NULL KEY_VALUE.

Example: SQL query for audit

The following is an example of SQL you can use to see the PII data for phone number 5551212, along with a full audit trail of the search for this PII, showing both the presence and the absence of this instance of PII data in tables and columns included in the search.

```
SELECT * FROM CTL_GDPR_HISTORY
WHERE TENANT_KEY = <tenant>
AND CONSUMER_ID = '5551212'
AND CREATED_TS BETWEEN <TS_1> AND <TS_2>
ORDER BY TABLE_NAME, COLUMN_NAME, FACT_ID
```

Genesys Interaction Recording and Analytics Support for GDPR

This page describes product-specific aspects of Genesys Interaction Recording and Analytics support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

Searching for Interactions

To export and delete specific interactions you first have to perform a search for the interactions.

For details about how to search for interactions, see the **Explore > Create a New Search** section in the **8.5.5 SpeechMiner UI User Manual**.

The best way to search for specific interactions is to use the **Metadata** filter, since the Metadata filter enables you to filter the search results for selected metadata and metadata values. The search results will only include interactions for which the selected types of metadata have defined values and when you specify values that match the specific conditions.

The best way to search for specific customer interactions is to use the metadata field that identifies customers (for example, telephone number, ID and so on) and to enter the value that identifies the specific customer.

> **Important**
> The types of available metadata vary from system to system.

Exporting Data

To export specific interactions you can use the **Export** batch action to export one or more interactions.

For details, see the **Explore > Search Results Grid > Batch Actions > Export Interactions** section in the **8.5.5 SpeechMiner UI User Manual**.

> **Important**
> Interactions are exported with their voice recording files. Interactions are not exported with their screen recording files.

Delete Interactions

To delete specific interactions you can use the **Delete** batch action to delete one or more interactions.

For details, see the **Explore > Search Results Grid > Batch Actions > Delete an Interaction** section in the **8.5.5 SpeechMiner UI User Manual**.

> **Important**
> When an interaction is deleted, its voice recording and screen recording files are also deleted only if your system includes Interaction Recording Web Services (RWS) version 8.5.202.18 or later. If you have not upgraded to RWS version 8.5.202.18 or later, the screen recording files will not be deleted.

.

Genesys Mobile Engagement Support for GDPR

This page describes product-specific aspects of Genesys Mobile Engagement support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

Forget Me Scenario

See Forget Me for more information.

If you need to forget a customer and his or her related information, you can use the Delete Callback API to delete one or more Callbacks by passing service IDs or Customer Numbers. See Delete Callback API for details.

> **Important**
> You will be able to delete a Callback only if it is in `SCHEDULED` or `COMPLETED` status. If the Callback is queued or in progress, first cancel it, then delete it.

Export Content Scenario

See Bulk Cancel and Export of Callback Records for more information.

Genesys Voice Platform Support for GDPR

This page describes product-specific aspects of Genesys Voice Platform support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

Data Retention Policies

GVP has configurable retention policies that allow expiration of data. GVP allows aggregating data for items like peak and call volume reporting. The aggregated data is anonymous. Detailed call detail records include DNIS and ANI data. The Voice Application Reporter (VAR) data could potentially have personal data, and would have to be deleted when requested. The log data files would have sensitive information (possibly masked), but requires the data to be rotated/expired frequently to meet the needs of GDPR.

Configuration Settings

Media Server

Media Server is capable of storing data and sending alarms which can potentially contain sensitive information, but by default, the data will typically be automatically cleansed (by the log rollover process) within 30 days.

The location of these files can be configured in the GVP Media Control Platform Configuration [default paths are shown below]:

- recordutterance.path = $InstallationRoot$/utterance/
- recording.basepath = $InstallationRoot$/record/
- record.basepath = $InstallationRoot$/record
- cpd.record.basepath = $InstallationRoot$/record/
- record.basepath = $InstallationRoot$
- record.irrecoverablerecordpostdir = $InstallationRoot$/cache/record/failed
- recordcachedir = $InstallationRoot$/cache/record

- directory = $InstallationRoot$/callrec/

Log files and temporary files can be saved. The location of these files can be configured in the GVP Media Control Platform Configuration [default paths are shown below]:

- logdir = $InstallationRoot$/logs/
- tmpdir = $InstallationRoot$/tmp/
- directories.save_tempfiles = $InstallationRoot$/tmp/

Also, additional sinks are available where alarms and potentially sensitive information can be captured. See **Table 6** and **Appendix H** of the Genesys Voice Platform User's Guide for more information. The location of these files can be configured in the GVP Media Control Platform Configuration [default paths are shown below]:

- ems.log_sinks = MFSINK|DATAC|TRAPSINK
- metricsconfig.DATAC = *
- dc.default.metricsfilter = 0-16,18,25,35,36,41,52-55,74,128,136-141
- ems.metricsconfig.MFSINK = 0-16,18-41,43,52-56,72-74,76-81,127-129,130,132-141,146-152

GVP Resource Manager

Resource Manager is capable of storing data and sending alarms and potentially sensitive information, but by default, the data will typically be automatically cleansed (by the log rollover process) within 30 days.

Customers are advised to understand the GVP logging (for all components) and understand the sinks (destinations) for information which the platform can potentially capture. See **Table 6** and **Appendix H** of the Genesys Voice Platform User's Guide for more information.

GVP Reporting Server

The Reporting Server is capable of storing/sending alarms and potentially sensitive information, but by default, these components process but do not store consumer PII. Customers are advised to understand the GVP logging (for all components) and understand the sinks (destinations) for information which the platform can potentially capture. See Table 6 and Appendix H of the Genesys Voice Platform User's Guide for more information.

By default, Reporting Server is designed to collect statistics and other user information. Retention period of this information is configurable, with most data stored for less than 30 days. Customers should work with their application designers to understand what information is captured as part of the application, and, whether or not the data could be considered sensitive.

Data Retention Specific Settings

- rs.db.retention.operations.30min.default = 7
- rs.db.retention.operations.5min.default = 1

- rs.db.retention.operations.counts.default = 7
- rs.db.retention.operations.daily.default = 90
- rs.db.retention.operations.hourly.default = 7
- rs.db.retention.operations.monthly.default = 1095
- rs.db.retention.operations.weekly.default = 364
- rs.db.retention.cdr.default = 30
- rs.db.retention.events.default = 7
- rs.db.retention.var.30min.default = 7
- rs.db.retention.var.5min.default = 1
- rs.db.retention.var.daily.default = 90
- rs.db.retention.var.hourly.default = 7
- rs.db.retention.var.monthly.default = 1095
- rs.db.retention.var.weekly.default = 364

Identifying Sensitive Information for Processing

The following example demonstrates how to find this information in the Reporting Server database – for the example where 'Session_ID' is considered sensitive:

- `select * from dbo.CUSTOM_VARS where session_ID = '018401A9-100052D6';`
- `select * from dbo.VAR_CDRS where session_ID = '018401A9-100052D6';`
- `select * from dbo.EVENT_LOGS where session_ID = '018401A9-100052D6';`
- `select * from dbo.MCP_CDR where session_ID = '018401A9-100052D6';`
- `select * from dbo.MCP_CDR_EXT where session_ID = '018401A9-100052D6';`

An example of a SQL query which might be used to understand if specific information is sensitive:

[+] View Example Query

```
USE [ems-rs]
DECLARE @SearchStr nvarchar(100) = '018401A9-100052D6'
DECLARE @Results TABLE (ColumnName nvarchar(370), ColumnValue nvarchar(3630))

SET NOCOUNT ON

DECLARE @TableName nvarchar(256), @ColumnName nvarchar(128), @SearchStr2 nvarchar(110)
SET  @TableName = ''
SET @SearchStr2 = QUOTENAME('%' + @SearchStr + '%','''')

WHILE @TableName IS NOT NULL

BEGIN
    SET @ColumnName = ''
    SET @TableName =
    (
        SELECT MIN(QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME))
```

```
        FROM     INFORMATION_SCHEMA.TABLES
        WHERE          TABLE_TYPE = 'BASE TABLE'
            AND    QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME) > @TableName
            AND    OBJECTPROPERTY(
                    OBJECT_ID(
                        QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME)
                         ), 'IsMSShipped'
                           ) = 0
    )

    WHILE (@TableName IS NOT NULL) AND (@ColumnName IS NOT NULL)

    BEGIN
        SET @ColumnName =
        (
            SELECT MIN(QUOTENAME(COLUMN_NAME))
            FROM     INFORMATION_SCHEMA.COLUMNS
            WHERE          TABLE_SCHEMA    = PARSENAME(@TableName, 2)
                AND    TABLE_NAME    = PARSENAME(@TableName, 1)
                AND    DATA_TYPE IN ('char', 'varchar', 'nchar', 'nvarchar', 'int', 'decimal')
                AND    QUOTENAME(COLUMN_NAME) > @ColumnName
        )

        IF @ColumnName IS NOT NULL

        BEGIN
            INSERT INTO @Results
            EXEC
            (
                'SELECT ''' + @TableName + '.' + @ColumnName + ''', LEFT(' + @ColumnName + ',
3630)
                FROM ' + @TableName + ' (NOLOCK) ' +
                ' WHERE ' + @ColumnName + ' LIKE ' + @SearchStr2
            )
        END
    END
END

SELECT ColumnName, ColumnValue FROM @Results
```

Web Services & Applications Support for GDPR

This page describes product-specific aspects of Web Services & Applications support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

Deleting PII

The **gdpr_forget_me.py** script provides an option to delete personal data from the Personal Favorites where the agent fills the information about the user.

The script can be run with the following parameters:

```
gdpr_forget_me.py [-h] [--gws GWS] --cass_keyspace CASS_KEYSPACE --cass_hostname
CASS_HOSTNAME --cass_port CASS_PORT --search SEARCH [--group_name GROUP_NAME] [--
cleanup CLEANUP] [--gws_username GWS_USERNAME] [--gws_user_password
GWS_USER_PASSWORD] [--gws_additional_headers GWS_ADDITIONAL_HEADERS]
```

Prerequisites

The following software packages are required to run this script:

- Python 2.7
- PyCassa 1.9.1
- Requests 2.18.4

Parameters

Parameter Name	Mandatory	Description
--cass_keyspace CASS_KEYSPACE	Yes	The Cassandra keyspace name.
--cass_hostname CASS_HOSTNAME	Yes	The Cassandra host name.

Parameter Name	Mandatory	Description
--cass_port CASS_PORT	Yes	The Cassandra thrift port.
--search SEARCH	Yes	The text (email or phone number) to be searched for.
-h, --help	No	Show this help message.
--group_name GROUP_NAME	No	The user group name.For example, interaction-workspace-personal-favorites.
--cleanup CLEANUP	No	When set to *true*, the script will send a request to GWS to delete the settings that match. The default value is *false*.
--gws_username GWS_USERNAME	No	The GWS user name with admin privileges. This user is used to delete matching data via GWS API.
--gws_user_password GWS_USER_PASSWORD	No	The password of the GWS user specified in *gws_username*.
--gws_additional_headers GWS_ADDITIONAL_HEADERS	No	The additional headers that will be added to the GWS request. CSRF headers must be added here. Headers are presented as JSON. For example, `{"header_name": "header_value"}`.

Example

The script will search personal data for deleting by email agent1@pizza.com in all settings groups, but will not delete, because --cleanup argument isn't specified.

- `python gdpr_forget_me.py --cass_keyspace sipfs --cass_hostname localhost --cass_port 9160 --search agent1@pizza.com`

The script will search email agent1@pizza.com in all settings groups and delete via GWS API.

- `python gdpr_forget_me.py --cass_keyspace sipfs --cass_hostname localhost --cass_port 9160 --search agent1@pizza.com --gws http://127.0.0.1:8090 --gws_username admin@pizza.com --gws_user_password password --cleanup true`

The script will search email agent1@pizza.com in all settings groups and delete via GWS API. The password of gws_username will be encrypted.

- python gdpr_forget_me.py --cass_keyspace sipfs --cass_hostname localhost --cass_port 9160 --search agent1@pizza.com --gws http://127.0.0.1:8090 --cleanup true

The script will search agent1@pizza.com in settings group with name 'interaction-workspace-personal-favorites' and delete via GWS API. 2 additional headers will be added to requests (X-CSRF-HEADER with value X-CSRF-TOKEN and X-CSRF-TOKEN with value 1434429f-81a8-459a-9d6d-792d17644471).

- `python gdpr_forget_me.py --cass_keyspace sipfs --cass_hostname localhost --cass_port 9160 --search agent1@pizza.com --gws https://gws-api-host:8099 --gws_username admin@pizza.com --gws_user_password password --cleanup true --group_name interaction-workspace-personal-favorites --gws_additional_headers "{\"X-CSRF-HEADER\": \"X-CSRF-TOKEN\", \"X-CSRF-TOKEN\": \"1434429f-81a8-459a-9d6d-792d17644471\"}"`

Workspace Web Edition Support for GDPR

Workspace Web Edition is an agent facing User Interface that enables the handling of interactions such as voice calls, chats, emails , and social media between a contact center and its contacts (customers).

Workspace and Customer Data

Workspace handles customer data through connections to the Genesys back end, including Genesys Universal Contact Server, Salesforce CRM, Key-Value pairs captured by an IVR, web forms, and agent/contact interaction. These data are not owned or stored by Workspace and so will be dealt with through GDPR Forget Me requirements for those services.

However, Workspace might store customer personal information in Recent and Personal Favorites. Workspace can be configured to store a number of most recent contacts to enable agents to quickly resume interacting with someone they have recently worked with. Workspace can also be configured to enable agents to specify a contact as a Favorite. This feature enables the agent to quickly locate that contact's information to facilitate initiating a new interaction. Personal Favorite contacts store contact information until an agent unfavorites the contact.

Removing Recent contacts and Personal Favorite contacts information is handled through the GDPR Forget Me requirement for GWS Server.

Warning

Workspace also supports Corporate Favorites. Administrators can specify a list of targets for all agents or for groups of agents. Normally Corporate Favorites are internal business targets, not customers or external contacts, but it is possible that an administrator might configure an external contact, including their phone number and email address, as a Corporate Favorite. Since this information is owned by the administrator of the Genesys customer, it will not be covered by the GDPR Forget Me tool.

SIP Feature Server Support for GDPR

This page describes product-specific aspects of SIP Feature Server support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

SIP Feature Server stores the voicemail data in recording or metadata format. Feature Server will fetch the ANI from the input file and process the voicemail delete process once per day scheduled by an automatic trigger.

In the Feature Server application, the following option can be configured for each vendor under **gdpr** section:

- **gdpr-directory**

Forget Me

From Feature Server release 8.1.202.10, the Forget Me feature is supported. For more information, see Forget Me.

Export Me

From Feature Server release 8.1.202.16, the Export Me feature is supported. For more information, see Export Me.

Scheduling Tasks

Scheduled maintenance tasks are executed from the master Feature Server. You can set Forget Me and Export Me using the **ScheduledTasks** application options.

Scheduling the forget-me Task

The following options will enable automatic script execution of Forget Me:

- **forget-me.active** = `true` (Activate/Deactivate)
- **forget-me.cmd** = **forgetMe.py** --**dbhost** <`host`> --**dbport** <`port`> (Command line)
- **forget-me.schedule** = `0 51 14 ? * *` (Schedule)

Scheduling the export-me Task

The following options will enable automatic script execution of Export Me:

- **export-me.active** = `true` (Activate/Deactivate)
- **export-me.cmd** = `exportMe.py --dbhost <host> --dbport <port>` (Command line: The other parameters are added from SIP Feature Server)
- **export-me.schedule** = `0 0 4 ? * *` (Schedule)

Genesys Intelligent Automation Support for GDPR

This page describes product-specific aspects of Genesys Intelligent Automation support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

Genesys Intelligent Automation, formerly known as Genesys App Automation Platform (GAAP) or SpeechStorm, is a solution that enables organizations to rapidly deploy phone self-service functionality to their customers, including intelligent call steering, for a more efficient and personalized customer experience.

Genesys Intelligent Automation provides non-technical users with a high level of control over the management and configuration of the system using a web-based interface. The system Dashboard allows business users to see at a glance how their applications are performing, as well as proactively highlighting areas for improvement, therefore avoiding potential usability issues.

Customers provide consumer-identifying input for GDPR requests (forget me and/or export) as JSON files. Genesys has a standard JSON format for both Right To Access (export data) and Forget Me (delete or redact data). Intelligent Automation will read these files and perform the appropriate tasks for each file. IA will write an execution result file to indicate if the task was successful or not. If unsuccessful, the customer should leave the appropriate JSON file in the directory so that IA can try and process again the next night. IA will only use the **phone** attribute and the **intauto-fields** attribute to find data.

Data Retention Policies

The **DBOvernightJobs.NumDaysHistoricalDataToKeep** setting controls the number of days that data is retained. When set to a positive value X, the data is retained for X days. When the value is set to -1 or 0, the data is retained permanently.

Configuration Settings

Digital customers (Web IVR, chatbots, etc.) should instead set their environment up to not store PII initially, the IA team will add digital functionality to the GDPR task in a future release. To not store PII,

the customer should:

- Turn the **Confidential Mode On** preference to *true* for any question which asks for PII.
- Turn **Remember Me** functionality off.
- Set the **CTI.FieldsToStoreInReporting.ExcludePattern** server setting to include any attached data which contains PII. The setting should be in a pipe separated string e.g., (AccountNumber|CustomerName|CustomerPhoneNumber)

JSON input files

There are separate input files for Right of Erasure and Right of Access requests:

- forget-<DDMMYYYY>-<any optional content>.json
- export-<DDMMYYYY>-<any optional content>.json

The date part of the file name (<DDMMYYYY>) is expected to indicate the date the file was created, to maintain file uniqueness. Genesys products do not use this information to trigger or manage request processing. Using timestamps in the file system, products process any files added or modified for that tenant since the last time the product processed GDPR requests.

Exporting and Forgetting Data

1. Copy the **fish-reporting-9.X.XXX.jar** file and fish-reporting.properties on to the server.

- **GDPR.ScheduledTask.ImportJsonFilesDirectory** - The location of the export and forget JSON requests. Use "/" instead of "\" delimiters. If this location is remote, it should be made available as a shared network drive on the server in question.
- **GDPR.ScheduledTask.TimeToRun.Hours** - The hour value of the time the service should run (24 HR time)
- **GDPR.ScheduledTask.TimeToRun.Minutes** - The minutes value of the time the service should run
- **GDPR.ScheduledTask.ExecutionResults.Directory** - The location to where the execution results should be written. Use "/" instead of "\" delimiters. If this location is remote, it should be made available as a shared network drive on the server in question.
- **GDPR.ScheduledTask.ExportJsonFilesDirectory** - The location to where the exported data should be written. Use "/" instead of "\" delimiters. If this location is remote, it should be made available as a shared network drive on the server in question.
- **GDPR.CompanyID** - The ID of the company you want to run these tests against

2. Take a copy of the database.properties from one of your IA/GAAP/SpeechStorm servers and paste into the same directory as your fish-reporting JAR file.

3. Open a command line and run the following commands:

```
cd <path to fish-reporting JAR e.g. cd C:\GDPR\>
```

```
 java -jar fish-reporting-9.X.XXX.jar (where X is the version number of your fish-reporting
JAR)
```

The GDPR task will then run at the scheduled time each day.

Notes

- Java 8 is required.
- The tasks should be run at a different time from the hive off.
- The server on which JSON files are located should be made visible as a shared drive on the fish-reporting server
- The tasks will be run as an executable JAR file.
- In order for this task to operate correctly, the hive off process must be operational as GDPR tasks are only run against historical data (i.e., data from the previous day and before).

Examples

JSON Input Example

```
{
  "consumers": [
    {
      "consumer": [
        {
          "phone": "02890571321"
        },
        {
          "phone": "077519654543"
        }
      ]
    }
  ],
  "intauto-fields": [
    "AccountNumber",
    "CustomerName"
  ]
}
```

JSON Output Example

```
{
  "consumers": [
    {
      "consumer": [
        {
          "phone": "02890571321"
        },
        {
          "cti_fields": "AccountNumber:123456"
        },
        {
          "question_answer": "result_detail:,nbest_meaning_1:goat,nbest_rawanswer_1:I'd like
```

```
a goat,nbest_slots_1:oof:banana|
foo:ananab,nbest_meaning_2:null,nbest_rawanswer_2:null,nbest_slots_2:null,nbest_meaning_3:null,nbest_rawanswer_
        },
        {
          "business_task_details": "Created a shiny moon man"
        }
      ]
    }
  ],
  "intauto-fields": []
}
```

Properties file Example

```
GDPR.ImportJsonFilesDirectory=Z:\json\dir\
GDPR.ScheduledTask.TimeToRun.Hours=12
GDPR.ScheduledTask.TimeToRun.Minutes=07
GDPR.ScheduledTask.ExecutionResultsDir=Z:\results\dir\
GDPR.ExportJsonFilesDirectory=Z:\results\json\dir\
GDPR.CompanyID=2
```

Genesys Pulse Support for GDPR

This page describes product-specific aspects of Genesys Pulse support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

Genesys Pulse only stores **Usernames**. A unique username is a good privacy protection approach. If a user uses email address as a username, to support GDPR, this email address and all dashboards/ widgets that were created by this user must be removed. This is done manually through the Widget Management.

Forget Me Procedure

1. Log into Genesys Pulse as a user with the `Pulse Manage Users` privilege.
2. Click on the `Gear` icon at the top right-hand corner.
3. Select the `Widget Management` menu item.
4. In the opened tab, select checkboxes for users that must be removed.
5. Click on the `Delete` button above the table.
6. In the opened dialog, select all checkboxes.
7. Press the `Delete` button.

Genesys Pulse Advisors Support for GDPR

This page describes product-specific aspects of Genesys Pulse Advisors support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

The Genesys Pulse Advisors (formerly Genesys Performance Management Advisors) dashboards display real-time metrics. Business managers, operations managers, and supervisors use the metrics data to quickly identify developing service and performance issues. Pulse Advisors business objects are created and related to Access Groups or Persons using a Genesys configuration interface, such as Genesys Administrator. These objects are then synchronized with the Advisors database, and an administrator can then configure the remaining information for each object, along with the necessary relationships, using the Advisors administration module. The Advisors business objects are not represented as standard objects in the Genesys configuration interface. The business attribute values contain just the ID and name of the object. You can enter a description for a business attribute in your Genesys configuration interface, but Advisors does not import it into the Advisors database, or use the description in any other way. No Personally Identifiable Information (PII) is stored in Advisors automatically.

Using Advisors Support Email Addresses

Advisors modules store the email addresses that you enter on the installation wizards and use those addresses to notify Support staff about operating issues. The email addresses are stored in the relevant Advisors properties file. A user's email address persists in the properties file even after a user's Person object is removed from Configuration Server. An email address that contains an employee's full name might be considered to be PII, although the email addresses in the Advisors properties files do not link to any additional user-identifying information. It is the customer's responsibility to correctly enter and manage Advisors Support email addresses, including the removal of any email address from the properties files if required for GDPR compliance (Right of Erasure). The need to update properties file(s) to remove email addresses can be avoided if you always use an email alias for Support staff, rather than user-identifying email addresses. For example, use advisors.support@yourcompany.com instead of john.doe@yourcompany.com.

The following table identifies the property in each .properties file with which an email address is associated, when configured:

Module	Location	Property
Platform	conf/	`adminUser`

Module	Location	Property
	NavigationService.properties	
Contact Center Advisor XML Generator	conf/XMLGen.properties	`connectionFailureRetryParams.supportEmai`
Frontline Advisor	conf/FrontlineAdvisor.properties	`__failure_notification_toAddress`

Email addresses that you enter on the **Notification Lists** page in the Advisors administration module are stored in the **DCC_NOTIFICATION_LIST_USERS** table in the Platform database. Genesys recommends that you configure these email addresses as non-personal email addresses. The Pulse Advisors product collects and stores email addresses only for the purposes of sending notifications. It is the customer's responsibility to correctly enter and manage user information on the **Notification Lists** page in the Advisors administration module, including the removal of email addresses, if required, for GDPR compliance (Right of Erasure). To update or remove an already-configured email in the notification lists, use the **Notification Lists** page to edit or remove a specific and potentially user-identifying email address. For more information about the **Notification Lists** page in the Advisors administration module, see Notification Lists in the *Contact Center Advisor and Workforce Advisor Administrator User's Guide*.

Workspace Desktop Edition Support for GDPR

This page describes product-specific aspects of Workspace Desktop Edition support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

Workspace Desktop Edition is an agent facing User Interface that enables the handling of interactions such as voice calls, chats, emails, SMS, and social media between a contact center and its contacts (customers).

Workspace and Customer Data

Workspace handles customer data through connections to the Genesys back end, including Genesys Universal Contact Server, Salesforce CRM, Key-Value pairs captured by an IVR, web forms, and agent/contact interaction. These data are not owned or stored by Workspace and so will be dealt with through GDPR Forget Me requirements for those services.

However, Workspace might store customer personal information in **Recent** and **Personal Favorites**. Workspace can be configured to store a number of most recent contacts to enable agents to quickly resume interacting with someone they have recently worked with. Workspace can also be configured to enable agents to specify a contact as a **Favorite**. This feature enables the agent to quickly locate that contact's information to facilitate initiating a new interaction.

Recent contacts is considered to be *held less than 30 days* and so does not require explicit processing for GDPR compliance.

Personal Favorite contacts store contact information until an agent unfavorites the contact. This might exceed the limitations specified by the GDPR Forget Me requirements, and therefore is handled by the Workspace GDPR Forget Me tool. Administrators determine where agent Personal Favorites are stored. They might be stored in the Agent Annex in the Genesys Back End, on a shared network directory, or in the Windows Profile of the agent.

Workspace also supports **Corporate Favorites**. Administrators can specify a list of targets for all agents or for groups of agents. Normally Corporate Favorites are internal business targets, not customers or external contacts, but it is possible that an administrator might configure an external contact, including their phone number and email address, as a Corporate Favorite. Since this information is owned by the administrator of the Genesys customer, it will not be covered by the

GDPR Forget Me tool.

GDPR and Personal Favorites

Workspace implements a new command line tool, the Workspace GDPR Forget Me tool, to enable administrators to remove any **Personal Favorite** record that contains a match of any of the phone numbers or email addresses specified in the JSON input file. The file specifies the list of phone numbers and email addresses to *erase*.

> ### Warning
> If one or several instances of Workspace are running while you run the Workspace GDPR Forget Me tool, then any agent favorites that are removed by the script might be written back when a Workspace instance that deals with this favorite is stopped. Genesys recommends that you run the Workspace GDPR Forget Me tool in off hours (if possible) and you make the list of favorites to delete incremental so that subsequent runs of the tool can delete occurrences that would have been restored in the meantime.

The Forget Me tool is applied to Personal Favorites in the following storage scenarios:

- Workspace User Settings are stored in the **Person** annex in Genesys Configuration Server. This is the case when the following configuration options are specified as follows:
  - options.record-option-locally-only = `false`
  - options.record-location is blank
- Workspace User Settings are stored in a shared network directory with a path where uniqueness is based on User attributes. This is the case when the following configuration options are specified as follows:
  - options.record-option-locally-only = `false`
  - options.record-location contains `$Agent.UserName$`, `$Agent.FirstName$`, `$Agent.LastName$`, or `$Agent.EmployeeId$`

> ### Warning
> If you store your personal favorites in a storage scenario that is not covered by the Forget Me tool, you must migrate to one of the supported storage models.

The Forget Me tool does *not* apply to Personal Favorites in the following storage scenarios:

- Workspace User Settings are stored in the Windows User Profile.
  - This is the case when the following configuration option is specified as follows:

    - options.record-option-locally-only = `true`
  - Use these migration steps to the supported scenario:
    1. Modify your Workspace configuration to store the Agent profile in a central location: options.record-option-locally-only = `false`.
    2. Design a clean-up script and execute it using Windows Network Administration tools against all user profiles of the workstations where agents are logging in. This script must also delete the following directory: `"%AppData%\Genesys Telecommunication\InteractionWorkspace\UserData\"`
- Workspace User Settings are stored in a shared network directory with a path that is *not* based on User attributes.
  - This is the case when the following configuration options are specified as follows:
    - options.record-option-locally-only = `false`
    - options.record-location specified with a value where uniqueness is not guaranteed by `$Agent.UserName$` or `$Agent.EmployeeId$`
  - Use these migration steps to the supported scenario:
    1. Change the value of the options.record-location option to specify the new path based on Agent unique attributes.
    2. The next time the agent logs in, the agent's profile information is merged to the new location.

Implementing the Workspace GDPR Forget Me tool

The **Workspace GDPR Forget Me tool** is a command line executable (`wde_gdpr_compliance.exe`) that uses a JSON file as input to clean out user data from a network directory or the agent annex.

To use the tool, create your JSON file (see below) then go to the Workspace Desktop Edition CD (8.5.125.01) and launch the executable. The path to the tool is here:

`WorkspaceDesktopEditionGDPRTools/Windows/`

JSON input file structure

The JSON input file must contain a list of target contacts/customers that are to be removed. The supported Request Type is 'forget'.

The following common consumer identifications types are considered attributes of a 'consumer':

- "phone":<value>: A valid phone number. Phone numbers can have separators or '+' in front to indicate an international number and can be a subset of contact phone number to be deleted. For example: (555)-123-456 or 123-456 are valid to delete +1 (555)123.456. The script automatically ignores all separators.
- "email":<value> A valid email address with or without Personal information ex: "john@company.com" or "John Doe <john@company.com>".

The structure of the JSON list of consumers should be as follows:

```
{"consumers":[{"consumer":{{"phone":<value>}{ "email":<value>} ...}]}
```

Here is an example of a JSON input file:

```
{
   "request":"forget",
   "consumers":[
      {"consumer":
         [
            {"name":"Jane Deer"},
            {"name": "Jane Q. Deer"},
            {"phone":"555551212"}
         ]
      },
      {"consumer":
         [
            {"name":"Dan Man"},
            {"phone":"555551212"},
            {"phone":"555551234"},
            {"email":"danny@mail.com"},
            {"email":"DM@mail.org"},
            {"fbid":"Dan Man"}
         ]
      }],
]}
 }
```

> **Important**
>
> - Only Consumers with an email address or phone number are considered in the executable.
> - If at least one 'email' or 'phone' parameter is matched, the consumer information is deleted.

Command Line arguments

Use the following command line arguments when running the tool:

- -host: Host of Configuration Server (mandatory).
- -port: Port Configuration Server (mandatory).
- -user: Username of a Configuration Server administrator who has full access on the Person objects representing agents (mandatory).
- -password: Password of the Configuration Server administrator specified in -user argument.
- -jsonfile: The name of the JSON input file containing a list of customers with personal information to be erased (mandatory).
- -rfolder : Network directory path containing the agent profiles to be cleaned up. The directory path must have the same value as the '`options.record-location`' option. If this argument is not specified, the tool is cleaning up the profile stored in the Annex of Person object in Configuration Server.
- -logfolder: The folder name for the output log file.

The following is a command line example to cleanup agents profiles that are stored in a shared network directory:

```
wde_gdpr_compliance.exe -host <hostname> -port <portnumber> -user <username> -password
<password> -jsonfile <jsonFilePath> -rfolder <networkDirectoryPath> -logfolder
<logDirectoryPath>
```

The following is a command line example to cleanup profiles stored in agent annexes:

```
wde_gdpr_compliance.exe -host <hostname> -port <portnumber> -user <username> -password
<password> -jsonfile <jsonFilePath> -logfolder <logDirectoryPath>
```

Phone number and Email addresses matching rules

Any separators or + in front of phone numbers are removed before phone number matching, then the specified phone number is searched.

For example, if you search for a contact phone number such as "567890", the following phone numbers will be matched and the contact information will be deleted:

- "1234567890"
- "+1 (234).567.890"
- "567.890"

When searching for email addresses, the tool only looks for the address part. For example, "test@email.com" and "Name <test@email.com>" are equal.

Log file output

The following information is output to the log after the executable is run:

```
wde_gdpr_compliance-<DDMMYYYY_HHMMSS>-execution-log (execution results, success/fail with
error information)
```

intelligent Workload Distribution Support for GDPR

This page describes product-specific aspects of intelligent Workload Distribution support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

intelligent Workload Distribution is a Business Optimization product that gathers work items or "Tasks" from third party systems such as CRM, and distributes them efficiently to the back office agents. For this, it analyzes key information ("attributes") attached to each Task, such as customer segment, due date, request type, priority and so on. With these attributes, iWD classifies the Tasks and assigns a priority to them, then pushes them at the right time to the right Agent.

iWD Core/Extended Attributes (including "Customer ID") must never contain any Personal Information (any information that may identify a natural living person). Should conveying Personal Information be necessary, Customer must convey it only via the Task(s) Custom attribute(s) and only if the Task's lifetime is less than thirty days. Customer must not include any Custom attribute(s) which contain Personal Information in the list of attributes used for historical reports.

Customer acknowledges that these steps are necessary to ensure that Personal Information retention is kept to a minimum. Failure to follow these instructions will result in the storage of Personal Information for an indefinite period of time, and Customer acknowledges that it is responsible for avoiding such retention.

Genesys Rules System Support for GDPR

This page describes product-specific aspects of Genesys Rules System support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

Genesys Rules System (GRS) is a Business Solution in which business logic is executed to make decisions related to Customer Service interactions. The client is typically a customer-facing IVR, website, or mobile application. Pertinent data collected by the client is passed into GRS for evaluation, and a decision is returned based on the configured business rules. The data that is collected and passed in is based on Genesys' customers' specific business requirements and will vary from solution to solution. While this data may contain PII, GRS itself does not store any PII as the rule execution is stateless. Data is passed in, evaluated, and a response returned but no data is stored or persisted in any form.

GRS and intelligent Workload Distribution

GRS is tightly integrated into Genesys intelligent Workload Distribution (iWD). Please refer to the GDPR statements for iWD.

Billing Data Server Support for GDPR

This page describes product-specific aspects of Billing Data Server support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

Billing Data Server stores the contact center’s activity records that are:

- Relevant for the customer invoices generation.
- Required to exert Genesys’ legal rights.
- Required to service the contract for our customers.

Depending on the contract, and thus on the set of the billable metrics that are collected to form the bills, the detailed information may include:

- Phone numbers of the end customers (consumers).
- Information about the customer's employees configured in the tenant's Genesys configuration environment:
  - Employee IDs
  - Usernames
  - First and last names

Antivirus Guidelines for Genesys Products

Genesys does not test with third-party software. It is possible, in rare cases, that the antivirus (AV) software could cause disruption in processing the applications. But the impact would depend on the settings that are configured for the antivirus software and individual Genesys products used in an environment. Technical Support will accept all Genesys related problems faced during the use of antivirus software. In most cases, the subsequent solution (recommended configuration changes, enabling/disabling particular scanners) should be addressed by the antivirus vendor.

For high performance media and signaling servers, Genesys recommends disabling any Real Time AV protections which may create delays that negatively impact media QoS and/or delivery. For digital media channels, it is critical to have AV enabled in the agent workstations. For email, AV protections should be applied in your email server.

For more information on the antivirus guidelines for each product, refer to the corresponding product documentation below:

- Genesys Mobile Services (GMS)
- Genesys Engage Chat
- Genesys Info Mart
- Interaction Concentrator (ICON)
- Load Distribution Server
- Management Framework
- Stat Server (RTME)

Document Change History

This section lists changes to the document that were made in support of the new and updated material listed in New in Release 8.5.

New/Updated Features

- [05/25/18] The Data Privacy section added to provide general information about Genesys support for customer compliance with the General Data Protection Regulation (GDPR).
- The TLS section has been completely revised and updated, and contains instructions for securing connections between and to/from Framework components.
- A brief description of the HTTP Secure (HTTPS) and Secure Real-Time Transport (SRTP) protocols has been added.

Other Changes

- The section formerly called Data Confidentiality and Integrity has been split into two sections Authentication and Authorization and Protection of Data at Rest.
- The section formerly known as Communications Integrity has been renamed Protection of Data in Transit.
- The description of the **inactivity-timeout** option has been moved to the *Framework Configuration Options Reference Manual*.