Genesys Security Deployment Guide

System-Level Guides 8.5.x

9/20/2016

Table of Contents

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
Last Login

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

Transport Layer Security (TLS)
Federal Information Processing Standards (FIPS)
Secure HTTP (HTTPS)
Secure Real-Time Transport Protocol (SRTP)
Lightweight Directory Access Protocol Secure (LDAPS)

Web Application Security

Open Web Application Security Project
RESTful Web Services

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 Professional Services 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 Professional Services 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 European Data Protection Directive (EDPD), 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.

New in This Release

This Guide has been updated with the following new Genesys security features and functions available 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, ф, 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, MyTфьу	**password-req-mixed-case**
Numeric	A password must contain at least one numeric character (0–9).	password123, myname8, кгыышф7	**password-req-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 does 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 attempt to connect to it 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 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 three 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.

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

When an account is locked out, its status changes to Locked; Configuration Server generates log event 21-22140; and the date and time of lockout, and the instance of the Configuration Server from which it is locked, is recorded in the read-only **last-locked-at** option in the options of the user's Person object, for reference purposes.

This basic configuration applies to all user accounts in the Tenant. Individual accounts can be configured to override the lockout rule by setting **account-override-lockout** to `true`. This option can also be used to cancel an existing lockout on an account.

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
- Interaction Routing Designer
- Outbound Contact Manager

- CCPulse+

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*.

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.

inactivity-timeout
Default Value: `0`
Valid Values: Any non-negative integer
Changes Take Effect: Immediately

Specifies the amount of time (in minutes) that a user who is logged in to a GUI Application can be inactive before application screens are minimized and the user forced to be re-authenticated. The default value `0` (zero) means that the feature is disabled.

This option is configured in the options of the GUI Application object.

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 proc
   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 applicati
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
  Configuration page of the installation wizard.
- You have created, and have the URLs of, the security banner and any custom error pag

Start of procedure

1. On the **Security Banner Configuration** page of the installation wizard for the applicatio
   a. In the **Select Security Banner behavior and configuration** section, select whe
   banner that you are about to define to be used by all applications that support the
   by applications of this type.
   b. Click **Next**.
2. On the **Security Banner Parameters** page, specify the parameters for the security banr
   a. Select how the security banner is displayed to the user the next time an applicati
      - **Until each user chooses to turn it off**—The security banner includes ar
      again. check box that, by default, is not selected. If the user selects this c
      security banner will not be displayed again to that user, regardless of the
      starting. Each Windows user account must explicitly select this option and
      security banner for all applications.
      - **Until each user chooses to turn it off once for each application type**-
      an **I Accept. Do not show this again.** check box that, by default, is not s
      option and clicks **Accept**, the security banner will not be displayed again
      application of this type. However, each time the user starts another type o
      security banner is active, the security banner will be displayed. Each Win
      explicitly select this option and click **Accept** to disable the security banne
      - **Every time the application starts**—The security banner does not includ
      this again. check box. The security banner is displayed to every user eve
      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 cann
   - **Proceed to login without banner**—The user can log in to the applicatio
   - **Exit, no login dialog box is displayed**—The user is not permitted to log

   > **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 win
   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
   - 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 dis
   the entire document is not available for display within this time, an intermediate m
   of use ... Please wait ..., is displayed until the security banner itself can be displ
e. Specify the height and width, in pixels, of the security banner window, intermedia
   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 c
document at the specified URL. At no time does the window exceed the work are
retains its size between logins, and once displayed, can be resized using standa

> **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 b banner, specify the URL of the document and click **Add**. When you have added all the U 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 U
   - Otherwise, specify the URL of an error document—either the default error page c 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 fo

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
Change permission) to the **HKEY_LOCAL_MACHINE** registry key—normally the system admin
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. Th
  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 redefin
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 ta

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 16
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
default key if the subkey does not exist.

String entries can be entered as STRING or EXPANDABLE_STRING registry values. If they are
EXPANDABLE_STRING, environment variable strings enclosed in percent signs (%) are replac
the environment variables (located in **%HOMEDRIVE%%HOMEPATH%\default.htm**). Integer e
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

- 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_

  where **<seq_number>** is the sequence number for multiple URLs. Multiple URLs are trie
  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:/
HKEY_LOCAL_MACHINE\SOFTWARE\GCTI\Unilogin\Banner\URLs\2\URL=http:
HKEY_LOCAL_MACHINE\SOFTWARE\GCTI\Unilogin\Banner\URLs\3\URL=%Syst
HKEY_LOCAL_MACHINE\SOFTWARE\GCTI\Unilogin\Banner\URLs\3\ErrorPage
```

  specify the following behavior:

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

Security Banner Registry Entries

This section describes the registry options used to specify and customize the appearanc
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 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 t
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 i
the user selects this check box, they will not see the security banner at subsequent atter
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 no
window, and the security banner is displayed every time anyone tries to access the appli

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
the error page is displayed, the window displays the **Exit** button in place of the **Accept** a
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 th
page window. If neither of these values is specified (the default), the window is sized to f
document specified by the URL. At no time does the window exceed the work area of the
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 notificat
control. To download and render the document, the security banner dialog uses compon
WebBrowser control. In some cases, for security reasons, the WebBrowser control does
means to detect navigation cancellation. This timeout is used to detect and properly proc

The absence of progress or status notifications from the WebBrowser control for a perioc
considered a failure to retrieve the document. If this timeout expires, the attempt to retrie
the current URL is aborted, and the dialog attempts to retrieve the next URL from the UR
last URL in the list, the System error **0x80004004: Operation aborted** error message is

If this option is set to zero (**0**), progress and status notifications are not used to detect do

> 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
by the URL. If the timeout expires before the content is displayed, an intermediate windo
use... Please wait...) is displayed. During this time, the user can close the window by cli
only be accepted when the content is fully displayed.

If the document cannot be retrieved, the behavior of the window depends on the value o

- If **AckMandatory=0**, the window closes automatically (if it is open), and the login
  can then log in to the application.
- If **AckMandatory=1**, the error page included with the software is displayed, show
  errors, refer to the HTTP specification. For system errors, refer to Microsoft techr
  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 r
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 addres
- 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 dis
the IE window
Changes Take Effect: After the application restarts

Required; specifies the URL of the document displayed in the security banner window. If 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
- Universal Routing Server
- Workspace Desktop Edition (formerly known as Interaction Workspace)
- intelligent Workload Distribution
- Interaction Routing Designer
- 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.

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 *Framework Configuration Options Reference Manual*.

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**

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'
```

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 Context Service
- Genesys Voice Platform
- Genesys Interactive Insights
- Performance Management Advisors
- Genesys Info Mart
- Interaction Concentrator

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. The PKCS #7 encryption algorithm is used for both encryption and decryption.

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
- SIP Server
- Load Distribution Server
- Stat Server
- Outbound Contact Server
- Interaction Concentrator (ICON)
- Federated
- Universal Contact Server
- Universal Routing Server
- Orchestration Server
- eServices (partial)
- Enterprise SDK
- Interaction SDK
- Platform SDK
- Real Time Metrics Engine

SIP Server

SIP Server supports this feature, except for data that appears in a SIP header.

Stat Server

Stat Server supports this feature, but in a non-standard way. For more information, refer to Stat Server-specific documentation.

Platform SDK

Platform SDK supports this feature, but 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

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

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'
```

```
          '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'
          '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.

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.

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.

Introduction to Genesys Transport Layer Security

Genesys supports the optional use of the Transport Layer Security (TLS) protocol to secure data exchange between its components. TLS is an industry-standard protocol for secure communications on the Internet, and it is the successor of Secure Sockets Layer (SSL) 3.0.

Security Benefits

TLS provides strong authentication, message privacy, and integrity capabilities. TLS secures data transmission by using a variety of encryption options. TLS authenticates servers to prove the identities of the 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 considered to be compliant with Federal Information Processing Standards (FIPS).

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.

[+] Show supporting components

> **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.

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)
	Universal Contact Server 8.1.0 and later	`default` (or alternate name)

From	To	
Interaction Server 8.1.0 and later	Configuration Server or Configuration Server Proxy	`default` (or alternate name)
	Message Server	`default` (or alternate name)
	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)
E-mail Server 8.1.0 and later	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 8.1.2 and later	Configuration Server or Configuration Server Proxy	`default` (or alternate name)
	Universal Contact Server 8.1.1 and later	`default` (or alternate name)
	Interaction Server 8.1.0 and later	`default` (or alternate name)

From	To	
Universal Contact Server Proxy 8.1.0 and later	Configuration Server or Configuration Server Proxy (writable)	`default` (or alternate name)
	Message Server	`default` (or alternate name)
	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 or Configuration Server Proxy	`default` (or alternate name)
	Message Server	`default` (or alternate name)
	Interaction Server 8.1.0 and later	`default` (or alternate name)

From	To	
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)

From	To	
	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 8.1 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 On` 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 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 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 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
- 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.

> **Important**
>
> Connections internal to WFM use OpenSSL, not the standard IIRC. Refer to WFM documentation for specific instructions about setting up secure connections using OpenSSL.

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 as described in Assigning a Certificate to a Host.

Connection to Configuration Server in TLS relies on the auto-detect mode implemented by Configuration Server as described in Configuring Secure Connections to Configuration Server.

Feature Description

TLS secures connections through the exchange of authentication digital certificates during a handshake process which negotiates ciphers and key lengths used to encrypt exchanged data.

TLS can be configured in two ways, as described in the following sections:

- Simple TLS
- Mutual TLS

See Supporting Components for the list of components and connections that support TLS.

Simple TLS

In simple TLS, only the Server has a security certificate. It sends this certificate to the Client, which checks the certificate against its own Certificate Authority (CA). In effect, this authenticates the identity of only the Server.

Basic steps of this authentication are as follows:

1. TLS Client connects as anonymous.
2. TLS Server sends to TLS Client its certificate, containing a certificate chain that begins with the server's public key certificate and ends with the CA's root certificate. See Certificate Generation and Installation.
3. TLS Client checks the CA certificate in its trusted CA list.
4. TLS Client compares the TLS Server host name and the certificate's subject field, which must be identical (**tls-target-name-check=**`host`). See Check for Certificate-Host Matching.
5. TLS Client is satisfied that the server certificate is not expired and has not been revoked. See Certificate Revocation Lists.

Mutual TLS

In mutual TLS, both the Server and the Client have security certificates. They exchange their certificates, then each checks the other's certificate against its own CA. This authenticates the identities of both the Server and the Client.

Basic steps of this authentications are as follows:

1. TLS Server and TLS Client exchange their certificates and check the root CA in the list of trusted CAs. See Certificate Generation and Installation.
2. TLS Client compares the TLS Server host name and the certificate's subject field, which must be identical (**tls-target-name-check=**`host`). See Check for Certificate-Host Matching.
3. TLS Client is satisfied that the server certificate is not expired and has not been revoked. See Certificate Revocation Lists.
4. TLS Server is satisfied that the client certificate is not expired and has not been revoked. See Certificate Revocation Lists.

You can upgrade to mutual TLS by setting the **tls-mutual** option in the **[security]** section to `1`, as follows:

tls-mutual
Default Value: `0`
Valid Values: `0`, `1`
Changes Take Effect: Immediately

Specifies if mutual TLS is used for secure data transfer. If set to `1`, TLS certificates must be configured on both the server and client applications. If set to `0` (the default), client certificates are not required, and either simple TLS or data encryption (if **client-auth=**`0`) is used.

Evolution of Genesys TLS

Prior to 8.1.3, secure data exchange was accomplished by encrypting the data, using the TLS server certificate.

Starting in 8.1.3, simple TLS is the default method of secure data exchange. On the Windows platform, Configuration Server enables automatic authentication of a server's security certificate by the Windows TLS client socket. However, this might cause the failure of existing TLS connections for which server certificates were not configured or CAs were not configured on the clients. Genesys recommends that to prevent authentication errors on those existing TLS connections, make sure that server certificates are used and/or CAs are configured on the client applications. Alternatively, you can set the **client-auth** option to zero (`0`) to disable the default behavior and restore pre-8.1.3 behavior. This option can be set at the connection, application, or host level, with the same order of precedence.

Environment Prerequisites

The instructions in this document assume that you are adding Genesys TLS to existing connections of your Genesys configuration—that is, that your applications have already been installed, properly configured, and associated with hosts and with each other. See product-specific deployment guides for instructions about, and deployment instructions for, these components.

Supported Platforms

> **Important**
>
> Genesys TLS is not supported on all operating systems that Genesys products support. For UNIX-based operating systems, see Setting the Environment Variables for more information.

Refer to the *Genesys Supported Operating Environment Reference Guide* for a list of operating systems and database systems supported in Genesys releases 7.6 and later.

Supported Versions of TLS

Genesys TLS supports the following versions of TLS:

- TLS 1.1
- TLS 1.0
- SSL v3
- SSL v2

However, the version of TLS that is actually supported depends on the involved components and the software they are running. Refer to product documentation for TLS version information.

Specifying the TLS Protocol

Starting with Security Pack 8.5, an application can specify the lowest compatible protocol used by Security Pack on UNIX to send and accept secure connection requests on one or more of its connections, thereby limiting the use of obsolete protocols. To enable this, use the following option:

sec-protocol
Default Value: `SSLv23`
Valid Values: `SSLv23`, `SSLv3`, `TLSv1`, `TLSv11`
Changes Take Effect: Immediately

Specifies the protocol used by the component to set up secure connections.

This option is configured on one of three levels:

- Host-level (the application host): In the **[security]** section of the annex of the Host object.
- Application-level: In the **[security]** section of the options of the Application object.
- Port-level (connection-level): As a transport parameter of the application's connection.

> **Important**
> If the component reads its configuration information solely from its configuration file, such as LCA or Genesys Deployment Agent, set this option in the **[security]** section of the appropriate configuration file (such as **lca.cfg** for LCA or **gda.cfg** for Genesys Deployment Agent).

On a single component, this option must be configured at the same level where the certificate is configured. Across a network, if this option is configured at multiple levels (connection, application, host), the value set at the lowest level takes precedence. That is:

- The value set at the connection level takes precedence over the value set at the application and host levels.
- The value set at the application level takes precedence over the value set at the host level.

Feature Configuration

All Genesys components are configured in Genesys Administrator. To enable secure data exchange between the components, you must configure additional parameters in the Host objects, and in the Application objects that represent these components.

To use Genesys TLS functionality, you must complete the following steps:

1. For UNIX, install the Security Pack on each host computer where Genesys components run. See Security Pack on UNIX.

2. Set up a Certificate Authority (CA) on all server and client hosts that will be using TLS. See Certificate Generation and Installation.
3. Create and install security certificates on UNIX and/or Windows platforms, as follows:
    - For simple TLS, install the certificates on only those hosts where the Server applications are running.
    - For mutual TLS, install the certificates as follows:
        - On those hosts on which the Server applications are running.
        - On other hosts that are not running Server applications but are running Client applications.

    See Certificate Generation and Installation.

4. Complete application-specific and/or host-specific configuration procedures in Genesys Administrator. See Genesys TLS Configuration.

You can create and manage certificates and the corresponding private keys by using the OpenSSL toolkit and Windows Certification Services.

Security Pack Installation

The Genesys Security Pack on UNIX provides the components—such as shared libraries and an example of a Certification Authority (CA)—that are used to generate certificates and to deploy them on UNIX computers on which Genesys components are installed.

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

Supported Operating Systems

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

Security Pack on UNIX also supports 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.

Install the Security Pack

To use Genesys TLS on UNIX platforms, 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**
      > 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. In the environment variable that corresponds to your operating system (see the following table), specify 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>
```

Certificate Generation and Installation

This chapter provides an overview of the process of certificate generation using the open source OpenSSL tool and Windows Certificate Services, and how to manage those certificates on a Windows platform using Microsoft Management Console (MMC).

Keep in mind that 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.

Important

- Although you can use OpenSSL to generate certificates on both UNIX and Windows, Windows Certificate Services is available only on the Windows Server operating system. Nevertheless, the certificates generated by both methods can be used for secure data exchange between applications that run on both Windows and UNIX operating systems.
- Genesys recommends that you use OpenSSL if you intend to run any applications that might require secure connections on UNIX. If you intend to run all

your applications on Windows, Windows Certificate Services is recommended.
- When configuring simple TLS, certificates are optional for Genesys 8.x client applications.
- The security certificates used in Genesys TLS must be valid and compatible with (acceptable to) OpenSSL.

OpenSSL Certificates

Use OpenSSL certificates if you intend to run any applications that might require secure connections on UNIX. However, if you intend to run all of your applications on Windows, Genesys strongly recommends that you use Windows Certificate Services to generate certificates.

Supported Certificate and Key File Formats

- X.509
- PKCS#8
- DER (`.cer`)
- PEM (`.pem`, `.cer`)
- PKCS#7
- PKCS#12

Java/PSDK-based Applications

If you are going to be installing certificates for Java/PSDK-based applications on UNIX, such as Universal Contact Server (UCS), you will have to convert the private-key files generated by OpenSSL to a format compatible with those applications. The conversion must be done after they are generated but before they are installed, as given in the procedure below (see step 3).

Pre-requisites

The scripts that are used to generate certificates require the OpenSSL toolkit, which you can obtain from the OpenSSL Project website.

You can also obtain build binaries of OpenSSL tools for the Windows operating system from here.

Generation and Installation

To create and install certificates using OpenSSL, complete the following steps:

1. Set up a Certification Authority (CA). **[+] Show steps**

> **Important**
> Genesys recommends that you use only one CA instance for your entire call center environment.

a. Create a CA directory in which CA files—scripts, configuration files, and generated certificates—will be stored.
b. 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.
c. Run the **create_ca.sh** script from the **bash** shell by specifying the proper parameters (see the following table) in the following command line:

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

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

Parameter	Description
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.

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.

    > **Important**
    > You must copy this file 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.

2. Generate certificates as required. **[+] Show steps**

> **Important**
>
> Genesys recommends that you use the same CA to generate all certificates for a particular environment.

To generate a certificate for a particular host computer:

a. Go to the CA directory in which the CA files are stored.
b. 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] -
host HOST_NAME -CN COMMON_NAME [-OU ORG_UNIT] [-O
ORGANIZATION] [-L LOCALITY] [-S STATE] [-C COUNTRY]
```

For example:

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

Parameter	Description
KEY_SIZE	(Optional) The size, in bits, of the host private key. The default value is `2048` bits.
VALID_TIME	(Optional) The amount of time, in days, that the certificate is valid. The default value is `100` days.
HOST_NAME	(Mandatory) The full name of the DNS host.
COMMON_NAME	(Mandatory) The name of the host.
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.

Parameter	Description
COUNTRY	(Optional) The two-letter abbreviation for the country.

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, 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.

3. If you are installing certificates on any Java-based PSDK applications, such as Universal Contact Server, convert the private key file to PKCS #8 format. Use the following command:

```
openssl pkcs8 -topk8 -nocrypt -in
<serial_#>_<host_name>_priv_key.pem -out
<serial_#>_<host_name>_priv_key_NEW.pem
```

The converted file **<serial_#>_<host_name>_priv_key_NEW.pem** will be compatible with Java-based PSDK applications.

4. Install the certificates. **[+] Show steps**

Important

- If you are using mutual TLS, you must install the CA self-signed certificate file, **ca_cert.pem**, and at least one certificate issued by this CA on each computer that hosts Genesys applications that might require secure data exchange.
- If you are using simple TLS, you need to install only the CA self-signed certificate file, **ca_cert.pem**, on each computer that hosts Genesys applications that might require secure data exchange. You do not need to install certificates on those hosts that are not running any of the server applications.

On UNIX

a. Copy the **ca_cert.pem** file to the computer.
b. Copy the certificate and private key files to a local directory on the computer, as follows:
    - For Java-based PSDK applications: **<serial_#>_<host_name>_cert.pem** and **<serial_#>_<host_name>_priv_key_NEW.pem**
    - For other applications: **<serial_#>_<host_name>_cert.pem** and **<serial_#>_<host_name>_priv_key.pem**
c. 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.

When you configure an application to support secure data exchange on UNIX:

- The full path to the **ca_cert.pem** file is copied to the **Trusted CA** text box of the **Certificate properties**.
- The full path to the **<serial_#>_<host_name>_cert.pem** file is copied to the **Certificate** text box of the **Certificate properties**.

- The full path to the **<serial_#>_<host_name>_priv_key.pem** or **<serial_#>_<host_name>_priv_key_NEW.pem** file is copied to the **Certificate Key** text box of the **Certificate properties**.

For more information, see Genesys TLS Configuration.

On Windows

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

a. From the Windows Start menu, select **Run**, and then execute the `mmc` command to start the Microsoft Management Console (MMC).
b. 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.
c. Right-click the **Trusted Root Certification Authorities** folder, and select **All Tasks > Import** from the shortcut menu. This starts the Certificate Import Wizard.
d. On the first Wizard page, click **Next**.
e. On the **File to Import** page, type the full name of the **ca_cert.pem** file that was created during the CA setup, and then click **Next**.
f. 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**.
g. Click **Finish**.
h. On the left pane, click the **Certificates** folder.
i. On the left pane, right-click the **Personal** folder, and select **All Tasks > Import** from the shortcut menu. This starts the Certificate Import Wizard.
j. On the first Wizard page, click **Next**.
k. 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**.
l. On the **Password** page, click **Next**. The host certificates in PKSC #12 format are generated with an empty password.

m. 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**.
n. Click **Finish**.
o. Press **F5** to update the MMC view.
p. On the left pane, select **Certificates > Personal > Certificates**.
q. On the right pane, locate the imported certificate in the list, and double-click it.
r. In the **Certificate** dialog box, click the **Details** tab.
s. 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.

Generating Certificates with 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.

Genesys TLS functionality requires that security certificates comply with the following requirements:

- Certificates that will be used by Genesys server applications must contain these extended attributes: `serverAuth`, `clientAuth`, and `emailProtection`.
- Certificates that will be used by Genesys GUI applications must contain these extended attributes: `clientAuth` and `emailProtection`.

Make sure that certificate templates are properly configured for server and GUI applications to satisfy these requirements.

> **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.

Generating Certificates

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.

Managing Certificates with MMC

You can use the Microsoft Management Console (MMC) to manage certificates on a Windows platform. For more information, see *Microsoft Management Console Help*.

Configuring MMC for Certificate Management

To configure MMC for certificate management:

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:

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:

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.

Configuration of Secure Connections

After you generate the certificates and install them on the host computers, you must configure Genesys applications to use them on the connections that need to be secure. (By default, connections between Genesys applications are not secure.)

Important

- The instructions in this chapter 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 Genesys components of previous releases, 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 your particular product or component.

- An application can be both a server and a client application. In this case, they are configured both as a TLS server and a TLS client application. In this case, the same security certificate is used as both a server certificate and a client certificate.

Standard Configuration

You configure all secure connections in the same way, regardless of the types of participating client and server applications. The only exceptions are:

- 8.x releases of Genesys client applications that run on Windows and support TLS do not require client security certificates.
- Connections with Java-based applications—See Configuring Secure Connections to Java/PSDK-Based Applications.
- Configuration Server connections—See Configuring Secure Connections to Configuration Server.
- Local Control Agent connections—See Configuring Secure Connections Between LCA and Solution Control Server.
- Genesys Deployment Agent connections—See Configuring Secure Connections Between Genesys Deployment Agent and its Clients.
- High Availability synchronization connections-See Configuring High Availability Synchronization Connections.

Certificate Chains

Starting with release 8.1.3, Genesys Security Pack on UNIX supports security certificate chains, sending out the intermediate certificates with the root certificate. 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.

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, create a PEM file listing all of the certificates issued by the Trusted CAs. Then specify the full path to this 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.

Configuration Steps

To configure secure connections, perform the following steps:

1. 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. **[+] Show steps**

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.

> Important
>
> 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 unsecured port intact for connections that do not require security.

Secure Port on TLS Server Application

To configure a secure port on a TLS server applications, do the following:

a. In Genesys Administrator, click the **Provisioning** tab and navigate to the folder containing the server application.
b. Select the server application and click the **Configuration** tab.
c. In the **Server Info** section, click **Add** in the **Listening Ports** table. The **Port Info** dialog box appears.
d. 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**.
e. Click **Save**, **Save & Close**, or **Save & New** to save the new configuration.

Auto-Detect Port on Configuration Server

To configure an Auto-Detect port in the Configuration Server application, so that clients can connect securely to Configuration Server, do the following:

a. In Genesys Administrator, click the **Provisioning** tab and navigate to the Configuration Server on which you want to configure a secure port.
b. Select **Configuration Server** and click the **Configuration** tab.
c. In the **Server Info** section, click **Add** in the **Listening Ports** table. The **Port Info** dialog box appears.
d. 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**.
e. In the **Port Info**> dialog box, on the **Advanced** tab:
    - In the **Transport Parameters** box, replace `tls=1` with `upgrade=1`.
    - Click **OK**.
f. Click **Save**, **Save & Close**, or **Save & New** to save the new configuration.

If security parameters have been configured for the application, during the connection through the Auto-Detect port, Configuration Server checks the validity of security settings. Depending on the results, the client is connected in secure mode or Configuration Server rejects the client connection.

2. (Optional) Enable mutual TLS on the server applications. In the options of each server application, set the **tls-mutual** configuration option to `1`.

3. Assign a certificate to be used by the server applications.

Important

- If you are configuring simple TLS, certificates are optional for Genesys 8.x client applications.
- Genesys recommends that, unless you have compelling reasons to have any of your applications and/or ports protected by their own

individual certificate, you keep the certificate assignment at either the host or application level, as follows:
    - If you are installing certificates on a Java-based PSDK application, such as Universal Contact Server, assign the certificates at the application level. Then use these certificates to provide secure data exchange for all ports configured on that application (see step 5).
    - Otherwise, assign the certificates at the host level. Then use these certificates to provide secure data exchange for all applications residing on your hosts (see step 5).

[+] Show steps

After you create new server ports for secure connections, you must configure certificates in one of the following ways:

- Assign a certificate to a host, and then use this certificate to secure data exchange via any secure port of any server application that is located on this host. Use the procedure in the Assign to a Host tab, below.
- Assign a certificate to a server application, and then use this certificate to secure data exchange via any secure port of this application. A certificate that is assigned to an application takes precedence over the certificate that is assigned to a host. Use the procedure in the Assign to an Application tab, below.
- Assign a certificate directly to a specific port of a server application to secure data exchange via this port. A certificate that is assigned to a port takes precedence over certificates that are assigned to hosts and applications. Use the procedure in the Assign to a Port tab, below.

In Genesys Administrator, security-related properties are contained in the **Network Security** section of the Configuration tab for tenant, host, and server-type application objects.

Important

Before configuring secure connections, make sure that certificates are installed on the host computers on which specific Genesys components run, and that the certificate information is available to you.

Assign to a Host

To use a host's certificate to secure data exchange via any ports of any server applications (including client applications of server type) on that host, you must first assign a certificate to the host, and then complete the server application configuration.

a. In Genesys Administrator, select the host that accommodates the server applications whose connections you need to secure. To find out the name of the host that accommodates an application, look it up in the properties of that application.
b. Click on the host's **Configuration** tab.
c. In the **Network Security** section:
    i. In the **Certificate** text box, specify the full path to the **<serial_#>_<host_name>_cert.pem** file.

        Important

        If this client uses a certificate chain, specify a comma-delimited list of certificates with their paths.

    ii. (Optional) In the **Description** text box, enter a description for this certificate.
    iii. In the **Certificate Key** text box, specify the full path to the **<serial_#>_<host_name>_priv_key.pem file**.
    iv. In the **Trusted CA** text box, specify the full path to the **ca_cert.pem** file.

        Important

If this client uses multiple Trusted CAs, specify the name of the **PEM** file that contains the list of certificates issued by those Trusted CAs.

For information about the installed files, see step 3 of Installing Certificates.

The certificate information now appears in the appropriate fields of the **Network Security** section of the host's **Configuration** tab.

d. Click **Save**, **Save & Close**, or **Save & New**, as appropriate, to save the new configuration.

When you configure the Host object in Genesys Administrator, the certificate information that you specify in the **Network Security** section of the Host's **Configuration** tab is also displayed in the Host's annex (**Options tab > View = Advanced View (Annex)**) in the **[security]** section. The configuration options in the **[security]** section have the same meaning as those in the **Network Security** section of the **Configuration** tab—namely, the parameters of the certificate assigned to this Host.

Assign to an Application

If you intend to use an application's certificate to secure data exchange via any ports of a specific server application, you must first assign a certificate to this application, and then complete the server application configuration.

a. In Genesys Administrator, click the **Provisioning** tab and navigate to the folder containing the application to which you want to assign a certificate.
b. Select the application and click the **Configuration** tab.
c. In the **Certificate Source** box of the **Network Security** section, specify whether the application is going to use a certificate installed locally (select **Application**) or the certificate installed on the host (select **Host**).
d. If you selected **Application** in step c, specify the certificate parameters as follows:
    i. In the **Certificate** text box, specify the full path to the **<serial_#>_<host_name>_cert.pem** file.

Important

If this client uses a certificate chain, specify a comma-delimited list of certificates with their paths.

ii. (Optional) In the **Description** text box, enter a description for this certificate.
iii. In the **Certificate Key** text box, specify the full path to the **<serial_#>_<host_name>_priv_key.pem** file.
iv. In the **Trusted CA** text box, specify the full path to the **ca_cert.pem** file.

Important

If this client uses multiple Trusted CAs, specify the name of the `PEM` file that contains the list of certificates issued by those Trusted CAs.

For information about the installed files, see step 3 of Installing Certificates.

The certificate information now appears in the appropriate fields of the **Network Security** section of the host's **Configuration** tab.

e. Click **Save**, **Save & Close**, or **Save & New**, as appropriate, to save the new configuration.

When you configure the Application object in Genesys Administrator, the certificate information that you specify in the **Network Security** section of the host's **Configuration** tab is also displayed in the **[security]** section of the host's annex (**Options tab > View = Advanced View (Annex)**). The configuration options in the **[security]** section have the same meaning as those in the **Network Security** section of the **Configuration** tab—namely, the parameters of the certificate assigned to this Application.

Assign to a Port

a. In Genesys Administrator, click the **Provisioning** tab and navigate to the folder containing the server application for which you want to assign a certificate on its port.
b. Select the application and click the **Configuration** tab.
c. Open the **Server Info** section.

d. In the **Listening Ports** table, select the port whose connections you need to secure, and click **Edit**. The **Port Info** dialog box opens.
e. On the **General** tab, set **Select Listening Mode** to **Secured**. In the **Port Properties** dialog box, click the **Certificate** tab and click **Certificate properties**.
f. On the Network Security**Bold text** tab, specify the security parameters, as follows:
    i. In the **Certificate** text box, specify the full path to the **<serial_#>_<host_name>_cert.pem** file.

> Important
>
> If this client uses a certificate chain, specify a comma-delimited list of certificates with their paths.

    ii. (Optional) In the **Description** text box, enter a brief description of the certificate.
    iii. In the **Certificate Key** text box, specify the full path to the **<serial_#>_<host_name>_priv_key.pem** file.
    iv. In the Trusted CA text box, specify the full path to the **ca_cert.pem** file.

> Important
>
> If this client uses multiple Trusted CAs, specify the name of the **PEM** file contains the list of certificates issued by those Trusted CAs.

g. In the **Port Info** dialog box, click **OK**.

> Important
>
> The **Advanced** tab of the **Port Info** dialog box presents certificate parameters in a different form. This tab is reserved for future use.

h. In the **Port Properties** dialog box, click **OK**.
i. In the Application Properties dialog box, click **OK** to save the new configuration.

4. If necessary, remove (unassign) a certificate from a host, application, or port. **[+] Show steps**

From a Host

a. In Genesys Administrator, select the host from which to remove the certificate and open the **Configuration** tab.
b. Delete all certificate parameters in the **Network Security** section.
c. Click **Save & Close**, **Save**, or **Save & New**, as appropriate.

From an Application

> Important
> When you switch from Application certificate assignment to Host assignment, the Application certificate parameters are deleted.

a. In Genesys Administrator, select the application from which to remove the certificate and open the **Configuration** tab.
b. In the **Network Security** section, select **Host** in the **Certificate Source** dropdown list. This will delete all certificate parameters.
c. Click **Save & Close**, **Save**, or **Save & New**, as appropriate to save your changes.

From a Port

a. In Genesys Administrator, select the server application from which you want to remove a certificate from on its port and open the **Configuration** tab.
b. In the **Listening Ports** table in the **Server Info** section, select the port from which you want to delete the certificate and click **Edit**.
c. In the **Port Info** dialog box, click the **Network Security** tab, delete all certificate parameters, and click **OK**.
d. Click **Save & Close**, **Save**, or **Save & New**, as appropriate to save your changes.

5. (Optional, but recommended) Configure each server application to use the host certificate (for non-Java based applications) or application certificate (for Java-based applications), as applicable. If you are configuring mutual TLS, you must also configure each participating client application to use that host certificate. **[+] Show steps**

Use Host Certificate

After you assign a certificate to a host, you can use it to secure data exchange through any secure port of any server application that resides on that host.

a. In Genesys Administrator, navigate to the application that you want to configure to use the host certificate.
b. Select the application, and open the **Network Security** section of the **Configuration** tab.
c. In the **Select Source** field, select **Host** from the drop-down list.
d. Click **Save & Close**, **Save**, or **Save & New**, as appropriate to save your changes.

Use Application Certificate

After you assign a certificate to an application, you can use it to secure data exchange through any secure port of any server application that resides on that host.

a. In Genesys Administrator, navigate to the application that you want to configure to use the application certificate.
b. Select the application, and open the **Network Security** section of the **Configuration** tab.
c. In the **Listening Ports** table, select the port whose connections you need to secure, and click **Edit**. The **Port Info** dialog box opens.
d. Enter the certificate information as required.
e. Click **Save & Close**, **Save**, or **Save & New** as appropriate to save your changes.

6. Configure secure connections from the client applications. **[+] Show steps**

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.

The same configuration procedure is used for client applications of server type and user-interface type.

Important

When configuring simple TLS, certificates are optional for Genesys 8.x client applications.

a. Click the **Configuration** tab of the client application.
b. Select a server to which you need to make a secure connection, and click **Edit**.
c. 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.
d. If you are configuring mutual TLS, assign the host certificate to this application. Use the procedure in step 5, above.
e. Click **OK**.
f. Click **Save & Close**, **Save**, or **Save & New**, as appropriate, to save the new connection configuration.

The next time this application starts, it will connect to the server over a secure connection.

Configuring Secure Connections to Java/PSDK-Based Applications

Secure connections to Java/PSDK-based applications (such as Universal Contact Server) are configured in the same way as described in Configuration Steps, with the following 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.

Configuring Secure Connections to Message Server

Secure connections to Message Server are configured in the same way as described in Configuration Steps, with the following exceptions:

- Message Server must configure its default port with the security settings if TLS is to be enabled. TLS configuration on secondary listening ports is not supported.

- The client establishing a connection to Message Server must configure the certificate information at the Application or connection level.

Message Server supports TLS communication between it and the Solution Control Servers in a distributed configuration. In this situation, Message Server acts as the server and its port is configured as secured. The Solution Control Servers then connect to this port.

Configuring Secure Connections to Configuration Server

To configure a secure connection of a client application to Configuration Server, complete the following procedures.

New Server Clients

1. Specify the Auto-Detect port number of Configuration Server during application installation, by using the Installation Wizard. The Installation Wizard will propagate these parameters to the following locations:
    - To the **Command–Line** text box, in the **Server Info** section of the server's **Configuration** tab.
    - To the server application's **startServer.bat** file (for Windows) or **run.sh** file (for UNIX).
    - To the ImagePath in the Application folder in the Registry Editor.
2. Modify a client application configuration by adding a connection to the Configuration Server Application object to the client, as described in step 6 of Configuring Secure Client Connections, but select the Auto-Detect port in step c.

New User-Interface Clients

Start the applications and enter the Auto-Detect port number of Configuration Server in the **Log In** dialog box that appears.

Existing Clients

1. Verify or create the Auto-Detect port of Configuration Server in the Configuration Server Application object.
2. Modify a client application configuration by adding a connection to a Configuration Server Application object to the client, as described in step 6 of Configuring Secure Client Connections, but select the Auto-Detect port in step c.
3. Depending on the method that you use for starting client applications, for existing client applications of server type, change the port information to correspond to the port ID of the Auto-Detect port that you specified for Configuration Server, as follows:

- In the **Command Line Arguments** text box in the **Server Info** section of the server's **Configuration** tab.
- In the server application's **startServer.bat** file (for Windows) or **run.sh** file (for UNIX).
- In the **ImagePath** in the Application folder in the Registry Editor.

Configuring Secure 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.

> **Important**
>
> 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** and **lca-upgrade** configuration options 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*.

Before you configure the secure connection, you must:

- Install a certificate on the Host computer on which LCA is running.
- Have the certificate information available to you.

[+] Show steps

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 TLS you are using, as follows:

- If you are using mutual TLS, set the **certificate**, **certificate-key**, and **trusted-ca fields**.
- If you are using simple TLS, set only the **certificate** and **certificate-key** fields.
-

For more information, see the Sample Configuration Files for LCA.

2. In the annex of the host machine on which LCA is running, set the option **lca-upgrade** 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**

Mutual TLS:

```
[log]
verbose=standard
standard=stdout, logfile
[security]
upgrade=1
tls-mutual=1
certificate=/home/tech/sec/aix_cert.pem
certificate-key=/home/tech/sec/aix_priv_key.pem
trusted-ca=/home/tech/sec/techpubs_dco.pem
```

Simple TLS:

```
<tt>[log]
verbose=standard
standard=stdout, logfile
[security]
upgrade=1
certificate=/home/tech/sec/aix_cert.pem
certificate-key=/home/tech/sec/aix_priv_key.pem
```

Configuring Secure 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.

Before you start to configure the secure connections, you must ensure that:

- Certificates are installed on the Host computers on which Genesys Deployment Agent is running.
- The certificate information is available to you.

[+] Show steps

> **Important**
>
> Refer to the *Framework Configuration Options Reference Manual* for detailed descriptions of the options modified in this 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 **tls** 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 TLS you are using, as follows:
            - For Mutual TLS, set the **certificate**, **certificate-key**, and **trusted-ca** fields.
            - For Simple TLS, set only the **certificate** and **certificate-key** fields.

    For more information, see the Sample Configuration Files for Genesys Deployment Agent.
2. In the annex of the host machine on which Genesys Deployment Agent is running, set the option **gda-tls** to `1`.
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 any TLS setup, except that they are set in the configuration file instead of the configuration objects. **[+] Show files**

Mutual TLS:

```
[log]
verbose = standard
standard = stdout, gdalog
[web]
rootdir=./gdaroot
[security]
tls=1
tls-mutual=1
certificate=/home/tech/sec/aix_cert.pem
certificate-key=/home/tech/sec/aix_priv_key.pem
trusted-ca=/home/tech/sec/techpubs_dco.pem
```

Simple TLS:

```
[log]
verbose = standard
standard = stdout, gdalog
[web]
rootdir=./gdaroot
[security]
tls=1
certificate=/home/tech/sec/aix_cert.pem
certificate-key=/home/tech/sec/aix_priv_key.pem
```

Configuring Secure HA Synchronization connections

This section describes how to configure secure connections between primary and backup servers in a high-availability (HA) configuration.

> Important

See “Supporting Components” on page 136 for information about components that support secure connections in HA configurations. For information about setting up a HA environment for these Genesys components, see the corresponding product documentation.

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

Genesys does not recommend 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.

[+] Show steps

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.

Certificate Revocation Lists

TLS uses digital certificates to identify the parties in a conversation and then to negotiate an encryption algorithm to use. If the certificates are revoked or expired, the connection will fail to identify the parties and TLS will not set up the encrypted channel.

A Certificate Revocation List (CRL) is a time-stamped list identifying revoked certificates. This list is signed by a CA or CRL issuer and is made freely available in a public repository. Each revoked certificate is identified in a CRL by its certificate serial number. When a system uses a certificate for verifying a remote user's digital signature, for example, that system not only checks the certificate signature and validity but also acquires a suitably-recent CRL and checks that the certificate serial number is not on that CRL. The meaning of “suitably-recent” may vary with local policy, but it usually means the most recently-issued CRL. A new CRL is issued on a regular periodic basis (such as hourly, daily, or weekly). An entry is added to the CRL as part of the next update following notification of revocation. An entry must not be removed from the CRL until it appears on one regularly-scheduled CRL issued beyond the revoked certificate's validity period.

Configuration

Use the configuration option **tls-crl** (in the **[security]** section) to allow the supporting Genesys component to verify certificates against a CRL, by specifying a filename, in PEM format, 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.

> Important
>
> Configuration of a CRL in SIP Server slightly differs from other Genesys components. Refer to the *Framework SIP Server Deployment Guide* for more information.

Cipher Lists

The **cipher-list** configuration option allows the supporting Genesys component to select a list of cipher suites used in TLS. This option is transferred to a third-party library and describes the set of possible cipher suites.

Cipher Formatting Rules

> **Important**
>
> Cipher list format for an application using the PSDK library is different from that for an application using the Genesys common library. If you are configuring a cipher list for the PSDK-based appliction, 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. Cipher list strings must conform to the following formatting 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 `-`.
- 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.

Cipher Aliases

Ciphers also have aliases. The following table details the primary cipher aliases.

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.

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

Aliases can be joined in a colon-separated list to specify the ciphers to add, move, or delete.

Ciphers Example

The following string 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.

Configuration

Use the **cipher-list** option to define the list of ciphers. Refer to the *Framework Configuration Options Reference Manual* for a detailed description of the option.

If you are configuring a cipher list for an application using the Genesys common library, refer to the following:

- Cipher Formatting Rules for valid formats of a cipher list.
- Ciphers Example for an example of a valid cipher list.

If you are configuring a cipher list for an application using the PSDK library, refer to the *Platform SDK Developer's Guide*.

> **Warning**
>
> If you are going to use cipher lists on a host running both PSDK library-based applications and Genesys common library-based applications, do not configure **cipher-list** at the host-level. Configure the option at the application level or lower.

Check for Certificate-Host Matching

The **tls-target-name-check** option 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.

> **Important**
>
> Security Pack 8.1 and earlier supported only a case-sensitive check of host names.

Refer to Introduction to Genesys Transport Layer Security for details on authentication of TLS-Server and TLS-Client identity, which includes a step to check for certificate-host matching.

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 this option.

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 Install 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 of environment variables).
- For UNIX, make sure the environment variables that correspond to your operating systems are also properly set for the LCA environment (see the table of environment variables).
- 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** 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.

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 FIP Standard is developed only when there is 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, 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
- Interaction Workspace
- 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 systems 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 RSA-compatible and the FIPS-consistent shared libraries. To specify which library to use (FIPS or RSH), 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 HP-UX 32-bit platforms, set both the LD_LIBRARY_PATH and SHLIB_PATH environment variables to **<install directory>/fips140_lib32**.
      - On Solaris 64-bit platforms, set both the LD_LIBRARY_PATH and LD_LIBRARY_PATH_64 environment variables to **<install directory>/fips140_lib64**.
      - On all other platforms, 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 RHS library, set the LD_LIBRARY_PATH environment variable to **<install directory>**.

PSDK.NET

To enable FIPS in a PSDK.NET, use the same procedure as you do for configuring the common library for IIRC.

PSDK.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 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 (to be released in 2015)
- 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
- 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

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

- 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 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.

Document Change History

This section lists changes to the document that were made in addition to the new and updated material listed in New in Release 8.5.

8.5.001.00

This version of the Guide has been updated with the following:

- When configuring and using TLS:
  - Lists of supported TLS versions and OpenSSL certificate and key file formats.
  - Instructions for configuring TLS connections on Java/PSDK applications. See certificate conversion and installation, configuration of secure connections, and cipher lists.
  - Information about support of certificate chains and multiple trusted CAs.
- A brief description of the HTTP Secure (HTTPS) and Secure Real-Time Transport (SRTP) protocols.
- Overall, the structure of this Guide has been changed slightly, as follows:
  - 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.