Genesys Security Deployment Guide

User Passwords

12/15/2025

User Passwords

Contents

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, MyТфьу	**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	password!, my-name,	**password-req-**

Character Type or Case	Description	Examples	Configuration Option
	contain at least one punctuation character from the set: !\"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~).	ьн-тфьу	**punctuation**

Password Expiration

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

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

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

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

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

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

Resetting Passwords

Password reset can occur in one of two ways:

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

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

Password Reset by User

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

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

Password Reset Forced by System Administrator

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

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

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

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

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

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

Re-using Passwords

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

Account Lockout After Failed Connection Attempts

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

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

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

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

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

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

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

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

made in quick succession.

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

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

When an account is locked out, the following occurs:

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

The administrator can unlock an account manually by:

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

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

Account Expiry after Inactivity

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

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

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

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

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

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

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

Password Encryption

Passwords are encrypted automatically within the system, as follows:

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

Important

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

```
confserv -p confserv ‘$Montana’
```

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

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

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

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

Hiding Passwords in Log Files

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

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

Restrictions on User Connections

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

Number of Concurrent Connections

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

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

> Important

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

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

Control over Linked Connections

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

Control over HA pairs

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