Framework 8.5

Stat Server

Deployment Guide

About Genesys

Genesys is the world's leading provider of customer service and contact center software - with more than 4,000 customers in 80 countries. Drawing on its more than 20 years of customer service innovation and experience, Genesys is uniquely positioned to help companies bring their people, insights and customer channels together to effectively drive today's customer conversation. Genesys software directs more than 100 million interactions every day, maximizing the value of customer engagement and differentiating the experience by driving personalization and multi-channel customer service - and extending customer service across the enterprise to optimize processes and the performance of customer-facing employees. Go to www.genesys.com for more information.

Each product has its own documentation for online viewing at the Genesys Documentation website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Customer Care from Genesys

If you have purchased support directly from Genesys, please contact **Genesys Customer Care**. Before contacting Customer Care, please refer to the ***Genesys Care Program Guide*** for complete contact information and procedures.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the ***Genesys Licensing Guide.***

Released by

Genesys Telecommunications Laboratories, Inc. www.genesys.com

Document Version: 85fr_dep_statserver_10-2014_v8.5.001.00

Table of Contents

Supplements

Index

Preface

Welcome to the *Framework 8.5 Stat Server Deployment Guide.* This document introduces you to the configuration, installation, and start procedures that are relevant to Stat Server. This guide is valid only for the 8.5.*x* releases of Stat Server.

Note: For releases of this document that have been created for other releases of this product, please visit the Genesys Documentation website, or request the Documentation Library DVD, which you can order by e-mail from Genesys Order Management at `orderman@genesys.com`.

This preface provides an overview of this guide, identifies the primary audience, introduces document conventions, and lists related reference information:

About Stat Server

Stat Server is part of the Services Layer of the Genesys Framework. This key component is used by other Genesys solutions and Solution Reporting to track the real-time states of interaction management resources and to calculate basic measurements about the performance of contact center events and activities.

Intended Audience

This guide, primarily intended for network, IT, and contact center administrators, assumes that you have a basic understanding of:

- Computer-telephony integration (CTI) concepts, processes, terminology, and applications.

- Network design and operation.
- Your own network configurations.

You should also be familiar with Genesys Framework and Genesys solutions architecture and functions.

Making Comments on This Document

If you especially like or dislike anything about this document, feel free to e-mail your comments to Techpubs.webadmin@genesys.com.

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document. Please limit your comments to the scope of this document only and to the way in which the information is presented. Contact your Genesys Account Representative or Genesys Customer Care if you have suggestions about the product itself.

When you send us comments, you grant Genesys a nonexclusive right to use or distribute your comments in any way that it believes appropriate, without incurring any obligation to you.

Contacting Genesys Customer Care

If you have purchased support directly from Genesys, please contact Genesys Customer Care.

Before contacting Customer Care, please refer to the *Genesys Care Program Guide* for complete contact information and procedures.

Document Change History

This is the first release of the Framework 8.5 Stat Server Deployment Guide. In the future, this section will list topics that are new or have changed significantly since the first release of this document.

Genesys

Chapter

1 Overview of Stat Server

Stat Server is the component responsible for converting single interactions into statistical data that is useful for interaction processing and contact center reporting. This data is then used by various Genesys solutions and applications to determine the availability of resources and to generate statistical indicators of contact center performance.

This chapter contains the following sections:

New in This Release

This section describes the new or changed functionality that was introduced in release 8.5.0.

- Stat Server supports multiple Interaction Servers that handle the same `Tenant`.
- Stat Server supports the `EventHint` panic signal from Interaction Server. Refer to the *eServices Reference Manual* for more information about this feature.
- Stat Server supports Interaction Server Proxy.
- Stat Server supports direct database connection. See page 63 for details.
- Stat Server features more robust DND model implementation for voice.
- Stat Server processes network messages (client and server) in a separate thread.
- Stat Server supports higher number of concurrent client connections (on most platforms).
- Stat Server no longer supports Resource Capacity Wizard and Stat Server Wizard. All deployment wizards migrate to Genesys Administrator Extension (GAX).

Other Changes Documented Elsewhere

The *Supported Operating Environment Reference Guide* lists the operating systems (OSs) on which Stat Server 8.5 can operate. Also, the *Genesys Migration Guide* describes the high-level changes that were implemented with each release.

Stat Server Overview

The Genesys Statistics Server (Stat Server) is part of the Genesys Management Framework Services Layer shown in Figure 1.

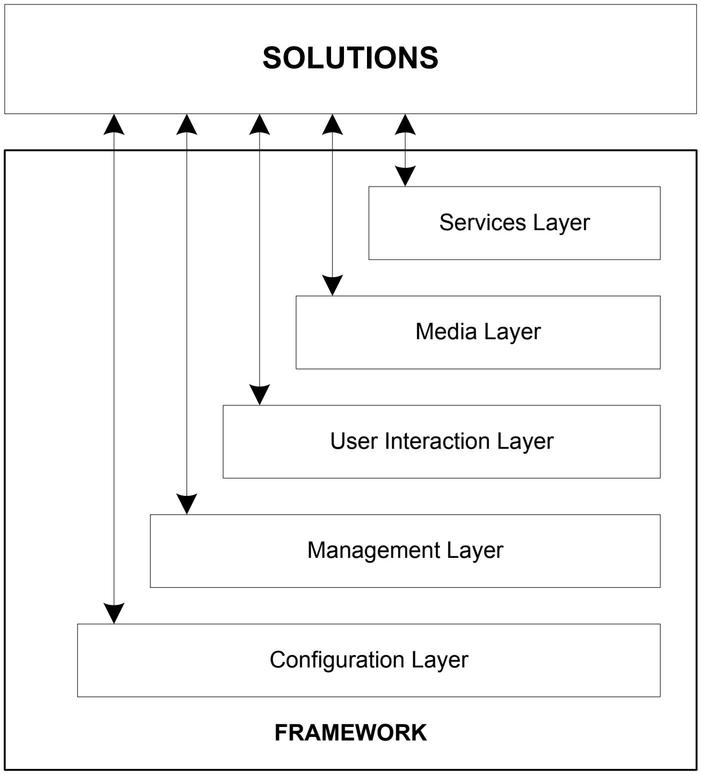


Figure 1: Management Framework Architecture

Refer to the *Management Framework Deployment Guide* for more information about the Services Layer and Stat Server's role within the Genesys Management Framework.

As a client of T-Server, Session Initiation Protocol (SIP) Server (a type of T-Server), and Interaction Server, Stat Server processes raw information received from these applications. As a client of Configuration Server, Stat Server retrieves information about the following contact center objects:

- Regular DNs
- Agents
- Places
- Queues
- Group of agents, places, and queues
- Routing Points
- Staging areas
- Tenants
- Workbins
- Switches
- Campaigns
- Calling lists

Refer to "Stat Server Object Types" in the *Framework Stat Server User's Guide* for information about which of these object types is supported for different modes of Stat Server operation.

In short, Stat Server reconstructs the behavior of contact center objects in order to provide its clients with more elaborate and statistically useful reporting data.

To receive statistical information, Stat Server clients must specify the kind of data they need, following the Genesys Statistical Model described in the *Reporting Technical Reference 8.0* series. This specification consists of a request for statistics retrieval, via stat types, from the Stat Server application programming interface (API)—the Statistics Library, or Stat Lib for short. (This API is not documented.) For instance, the Universal Routing Server requests information for the purpose of monitoring virtual queues and determining agents' availability to process additional simultaneous interactions. And, the Genesys Outbound Contact Solution requests statistical information from Stat Server about the performance of its outbound campaigns and calling lists.

This *Framework Stat Server Deployment Guide* describes the configuration and installation of the Stat Server application (and supporting topics) whereas the *Framework Stat Server User's Guide* describes the configuration of statistics—their stat type definition. The *Reporting Technical Reference 8.0* series describes the application of Stat Server statistical types employed by the Solution Reporting applications—CCPulse+ and CC Analyzer. You can also reference other Genesys solution user guides for information about how the various solutions rely on Stat Server to provide statistical information.

Genesys

Chapter

2 Configuring a Stat Server Application

You must configure a Stat Server `Application` object before you install it. To configure Stat Server, Configuration Server must be running. In the following sections, this chapter describes how to configure a Stat Server `Application` object:

Notes: To use secure Transport Layer Security (TLS) connections between Stat Server and its clients, or between Stat Server and Configuration Server, you must configure such connections manually following the procedures described in the *Genesys Security Deployment Guide.*

Configuration

You can configure a Stat Server 8.5 `Application` object manually within Genesys Administrator/GAX. Refer to the *Genesys Administrator/Genesys Administrator Extension Help file* for more information. You use the Stat Server application template to accomplish this. This template is located in the `templates` directory of the Real-Time Metrics Engine CD and is named Stat_Server_850.apd. You should import this file into your configuration environment before configuring Stat Server application.

Configuring a Stat Server Application Using Genesys Administrator/GAX

In the templates directory of the *Real-Time Metrics Engine* CD—Stat_Server_850.xmlfile—contains the metadata that defines the default and valid values for most of the Stat Server configuration options that are available to you in the 8.5.*x* release. (The complete listing of configuration options is located in this document and accompanying release notes.) To use this metadata, you must import it into the Stat Server application template. As you configure a Stat Server application within Genesys Administrator, Genesys Administrator validates the values that you specify for configuration options against this metadata.

Refer to the *Genesys Administrator/Genesys Administrator Extension Help file* for instructions on how to import and use metadata and for instructions on how to configure applications.

Notes: If you specify more than one database access point, Stat Server will use only the first one.

For Stat Server operating in cluster mode, you must select auto-restart.

The Stat Server applications that function within a Stat Server cluster ignore the stat type definitions that are defined at the application level.

Configuring Stat Server as Part of a SIP Cluster

Chapter 4 will have you adding Stat Server applications to a Stat Server solution for the purposes of servicing a SIP cluster. The Stat Server solution is the core object that controls when its components are started and what common statistical options all Stat Server applications share. This collection of Stat Server applications forms a *Stat Server cluster* where the nodes of the Stat Server cluster are paired in some capacity with nodes in the SIP cluster.

You can add other nonnodal applications to the Stat Server solution (including other Stat Server applications that are not paired to any SIP cluster node at all). Refer to “Configuring the Components of a Stat Server Solution” on pages 24 for additional information. Contact your Genesys representative for information about SIP Cluster architecture.

Adding SIP Server Applications to Stat Server Configuration

On the Connections tab of a Stat Server application, add all of the SIP Server applications that each node is to monitor.

- If a particular Stat Server node must connect to both the Interaction Proxy and T-Controller interfaces of SIP Server, then leave the `Connection Protocol` field blank. With Interaction Proxy and T-Controller listening ports properly provisioned within the SIP Server application, Stat Server will retrieve that information and open connections to both Interaction Proxy and T-Controller ports.
- If a particular Stat Server node must connect to the Interaction Proxy interface only of SIP Server, configure the connection by selecting `IPport` in the `Port ID` field and by typing `IProxy` in the `Connection Protocol` field.
- If a particular Stat Server node must connect to the T-Controller interface only of SIP Server, configure the connection by selecting `TCport` in the `Port ID` field and by typing `TController` in the `Connection Protocol` field.

How Stat Server Pairs Its Nodes to SIP Server Nodes

A Stat Server node connects to the SIP Server applications in the `Connections` tab according to the following rules:

TController Connection Protocol. If the server connection is configured with `TController` as the connection protocol and the SIP Server has a port ID of `TCport,` then Stat Server connects to this SIP Server as to a T-Controller.

IProxy Connection Protocol. If the server connection is configured with `IProxy` as the connection protocol and the SIP Server has a port ID of `IPport,` then Stat Server connects to this SIP Server as an Interaction Proxy.

No Specified Connection Protocol. If no connection protocol is specified, then Stat Server connects to SIP Server both as a T-Controller and an Interaction Proxy if the corresponding ports are specified within SIP Server configuration.

Configuring Secure Connections in an HA Environment

Stat Server connects to server applications by reading the properties of its own `Application` object in Configuration Server and connecting to each of the servers that are listed under the `Connections` tab in Genesys Administrator/ GAX. Each connected application, in turn, has its own properties and advanced transport parameters—for example, TLS mode, client-side port definition (CSPD), IP address, security certificate signatures, backup servers—that define how connections are to be made.

Stat Server supports a redundant server configuration in warm standby mode only. (Refer to “Redundancy Types” in the *Genesys Security Deployment Guide* for a discussion about this and other modes of operation.) In warm standby mode, Stat Server tries to connect to the backup application only after repeated attempts to connect to the primary application have failed. *The* *backup application does not have to be listed among Stat Server connections in order for these attempts to take place.* In this scenario, however, Stat Server uses the connection parameters that are specified within the properties of the primary application to connect to the backup application. This propagation of parameters to the backup application is the default behavior.

But, what if you want Stat Server to use instead those parameters that are specified within the application properties of the backup application—a different CSPD, for instance? When both TLS and HA are enabled and you require individual security settings for each connection, you must add the backup application explicitly to Stat Server's connections list in order for Stat Server to read its properties.

This configuration will cause Stat Server to generate the error ; however, selecting `Yes` to continue the assignment will yield the desired effect without Stat Server inadvertently launching the backup server when it is not needed.

To set client-side ports for each connection, refer to the "Client-Side Port Definition" chapter of the *Genesys Security Deployment Guide* and to all of the chapters within the "Communications Integrity–Transport Layer Security" part of this document for further information.

Chapter

3 The HTTP Interface

In order for the SIP Cluster Solution to receive information about the state of a regular DN or to receive information about performance data about each Stat Server node within a Stat Server cluster, you can configure one or more Stat Server applications within a Stat Server cluster to use an HTTP interface. In the following sections, this chapter explains how to configure Stat Server to provide this information:

This chapter pertains only to Stat Server that operates in cluster mode. Specifying the HTTP protocol for Stat Server that operates in regular mode is not currently supported.

Using the HTTP Interface for Feature Server

One manner in which Feature Server (a SIP Cluster server) sends call forwarding and DND requests from a particular DN to T-Controller is triggered through the activity that is transmitted through an HTTP interface. (Other manners are described in the *SIP Cluster Solution Guide*.) Stat Server operating in cluster mode can provide DN state information via this interface when it is configured to do so. When requested through the HTTP interface, Feature Server sends the `TCallSetForward`, `TCallCancelForward`, `TSetDNDOn`, or `TSetDNDOff` request, as appropriate, to the appropriate T-Controller that is in charge of the DN within the SIP Cluster.

Configuring an HTTP Listening Port

Within Genesys Administrator, you configure an HTTP listening port at the `Server Info` tab of an application's properties. Refer to the *Genesys Administrator Help* for more information.This listening port is specific to a particular Stat Server—you do not configure an HTTP listening port at the Stat Server solution level for all components to share. However, you can designate as few as one HTTP listening port within one Stat Server node to provide DN state information for all nodes within a Stat Server cluster.

The default protocol that Stat Server uses when you specify no connection protocol is an internal proprietary simple protocol.

Internal Performance Counters

Through the HTTP interface, Stat Server also supplies performance measurements to T-Controller for the events that Stat Server receives and sends. To provide server performance information, you must configure an HTTP listening port for every Stat Server node that must supply this information. Such configuration is required because the performance measurements that Stat Server provides will differ based on the object that initiated the request. Table 1 lists the performance measurements that Stat Server provides.

Table 1: Stat Server Performance Counters

Request / Input	Output
Configuration Server events	Rate of Configuration Server events per second calculated for the last checkpoint interval. (The default checkpoint interval is internally set at 30 seconds.) Rate of Configuration Server events per second calculated for the entire time since startup Maximum rate of Configuration Server events per second calculated for the checkpoint interval
Configuration Server delayed events	Number of delayed Configuration Server events during the last checkpoint interval
T-Server events	Rate of T-Server events per second calculated for the last checkpoint interval Rate of T-Server events per second calculated for the entire time since startup Maximum rate of T-Server events per second calculated for the checkpoint interval

Table 1: Stat Server Performance Counters (Continued)

Request / Input	Output
T-Server delayed events	Number of delayed T-Server events during the last checkpoint interval
Requests from Stat Server clients	Rate of events per second calculated for the last checkpoint interval Rate of events per second calculated for the entire time since startup Maximum rate of events per second calculated for the checkpoint interval
Events sent to Stat Server clients	Rate of events per second calculated for the last checkpoint interval Rate of events per second calculated for the entire time since startup Maximum rate of events per second calculated for the checkpoint interval
Samples published to shared memory	Number of samples during the latest checkpoint interval
Subscriber notifications from shared memory	Number of subscriber notifications during the latest checkpoint interval

You can also get response within an html browser by issuing the following string in the URL:

```
http://<StatServer HTTP listener host name>:<listener port>/genesys/
statserver/<path to specific resource>
```

Genesys

Chapter

4 Configuring a Stat Server Cluster Solution

You create a Stat Server solution in order to provide low-level reporting for a cluster of Session Initiation Protocol (SIP) servers. A Stat Server solution is implemented as group of identical Stat Server applications—a *Stat Server cluster*—running on the same host and utilizing shared memory-based publish/subscribe layer. Reporting for T-Server that is not a SIP Server and Interaction Server applications are not controlled by Stat Server solutions.

Unlike Genesys applications, you do not install Genesys solutions; rather, you configure and install the components that a solution controls, and then you configure the solution.

In the following sections, this chapter describes how to configure a Stat Server solution manually:

The Stat Server Cluster

You configure a Stat Server cluster as `Solution` object within Configuration Server of type `Multimedia`. This solution defines the connectivity parameters to all Stat Server instances, or *nodes,* within the cluster. Other applications can belong to the cluster as well (including other Stat Server applications that do not service the cluster); however, this chapter focuses on setup of Stat Server nodes for a Stat Server cluster only. The Stat Server `Solution` object also defines and stores all of the statistical parameters that service the cluster.

Each Stat Server instance within the cluster might connect to:

- The T-Controller interface of one or more SIP Server instances within a SIP Cluster and/or
- One or more Interaction Proxy interfaces of one or more SIP Server instances within a SIP Cluster

However, in order to monitor the entire SIP cluster, a Stat Server solution must connect to:

- All T-Controller interfaces of the SIP cluster—you can configure all such connections within as few as one Stat Server instance—and
- Every Interaction Proxy interface—using as many Stat Server instances that are required to handle call volume.

Figure 2 on page 25 illustrates one possible Stat Server solution configuration.

Any statistic for any object could be requested from any of the Stat Server instances within the cluster; therefore, each Stat Server instance shares a connection to all other Stat Server instances to proxy statistical requests and events between a particular aggregation instance and client.

You configure each Stat Server instance as an `Application` object of `Stat Server` type following the instructions and precautions described in Chapter 2, “Configuring a Stat Server Application,” beginning on page 13.

Upon startup, each Stat Server instance retrieves the name of its solution from the value specified with the `-cluster` command-line parameter (described on page 26). Using this information, each Stat Server instance then reads the configuration specified within the Stat Server `Solution` object, and establishes connections with all other Stat Server instances configured within the solution.

Stat Server Cluster Solution Prerequisites

- Stat Server Cluster solution instances must be configured and running in cluster mode, which is supported only on Microsoft Windows 2008, 64-bit.
- All solution Stat Server components must be of the same Stat Server release.
- You must install all nodes of the Stat Server cluster on the same host.
- A host can host no more than one Stat Server cluster that designates the same Stat Server `Solution`. Other Stat Server clusters that run on same host must designate different Stat Server solutions.

Creating a Stat Server Solution

Before you can complete solution configuration, you must first configure the components that will be added to the solution as described in Chapter 2, “Configuring a Stat Server Application,” beginning on page 13. You can use

Genesys Administrator or Genesys Administrator Extension (GAX) to create a Stat Server solution.

Using Genesys Administrator

1. At the Provisioning menu in Genesys Administrator, select Environment and then Solutions.
2. Click New to create a new Solution object.

   The Solutions screen opens and displays the three tabs (Configuration, Options, and Permissions) for you to configure the properties of your new solution.

   On the Configuration tab are three frames: General, Components, Component Definitions.
3. In each of the three frames on the Configuration tab, define the configuration properties of your solution, as described in Table 2, and click Save.
4. On the Options tab, add and configure the following:
   a. The options that are listed in Table 3, "Configuration Options for the cluster Section," on page 26.
   b. All stat type options that will be used by nodes of the Stat Server solution.

No changes are required on the Permissions tab. Refer to *Genesys Administrator Help* for information on how to use this tool.

Table 2: Stat Server Solution Properties

On This Tab/Frame	Provide the Following Information
General	• In the Name field, type a unique name for your Stat Server solution. • In the Assigned to Tenant field, choose the one tenant that the Stat Server solution should monitor. The tenant selection of each Stat Server instance must match this value. This field does not appear in single-tenant environments. • In the Solution Type field, select Multimedia. Once the solution is saved, you cannot change this value. • In the Solution Control Server field, select the appropriate server. • In the Version field, select the version. This field becomes uneditable once the solution is saved.
Component Definitions	Add all of the application types that the solution will house, and adjust the startup priority of each as needed.

Table 2: Stat Server Solution Properties (Continued)

On This Tab/Frame	Provide the Following Information
Components	Add all Stat Server instances that will service the solution. Each instance represents a node within the cluster. Refer to “Configuring the Components of a Stat Server Solution” on page 24 for more information. **Note:** Stat Server is supported on both 32- and 64-bit platforms. Stat Server operating in cluster mode, however, is supported on 64-bit platforms only. Starting a Stat Server cluster requires that all component nodes invoke one or more Stat Server executables from the **same Stat Server memory model.**
Options	Specify the appropriate stat-type sections, options, and values to define the statistics that are common to all Stat Server nodes. Refer to the *Framework Stat Server User’s Guide* for a description of configuration options that pertain to statistics.
Security	This tab appears after you save, close, and reopen the solution’s properties. It is not used for Stat Server solutions.

Configuring the Components of a Stat Server Solution

Chapter 11 provides hardware and software recommendations addressing the number of Stat Server instances required for optimal performance.

To properly monitor a SIP cluster:

- All Stat Server nodes within the Stat Server cluster solution must be running.
- The Stat Server cluster solution must include one and only one connection to every T-Controller interface of the SIP cluster One Stat Server instance can connect to only one T-Controller interface.
- The Stat Server cluster solution must include connections to all Interaction Proxy interfaces. More than one Stat Server instance can connect to the same Interaction Proxy interface.

For example, let us assume that a particular SIP Cluster has four nodes (`SIPServerA` through `SIPServerD`) and the Stat Server solution that services this SIP Cluster contains eight Stat Server applications (`StatServer1` through `StatServer8`). In order to maintain an even distribution of events amongst the eight Stat Server instances, each could be configured to connect to SIP Cluster components as illustrated in Figure 2; namely:

- StatServer1 includes connections to T-Controller$_A$ and IProxy$_A$ of SIPServerA
- StatServer2 includes connections to T-Controller$_B$ and IProxy$_B$ of SIPServerB
- StatServer3 includes connections to T-Controller$_C$ and IProxy$_C$ of SIPServerC.
- StatServer4 includes connections to T-Controller$_D$ and IProxy$_D$ of SIPServerD.
- StatServer5 includes connection to IProxy$_A$.
- StatServer6 includes connection to IProxy$_B$.
- StatServer7 includes connection to IProxy$_C$.
- StatServer8 includes connection to IProxy$_D$.

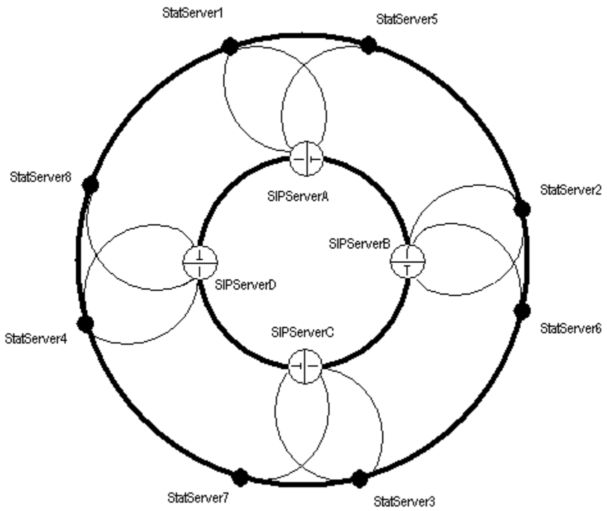


Figure 2: Sample Configuration Pairing Stat Server Solution to a SIP Cluster

In this configuration, call-related TEvents from each Interaction Proxy interface will be balanced among two Stat Server applications.

Configuring Options for a Stat Server Solution

This section describes the options that you can use on the Options tab to configure a Stat Server solution. Refer to Chapter 2 on page 13 to learn how to configure the individual Stat Server components of a Stat Server solution and to the "Statistics Configuration Options" chapter in the *Framework 8.5 Stat Server User's Guide* to learn about the options that you can use to configure statistics for your Stat Server solution.

Importing Statistical Types into a Solution

You can use the statistical parameters that are defined within the Genesys-provided Stat Server application templates within a Stat Server solution as well as within a Stat Server application.

However, this template includes configuration sections that do not pertain at all to a Stat Server solution. Moreover, the template does not include the one section that is mandatory for all Stat Server solutions; namely, the [cluster] section, described below. You can either modify your configuration file as needed before importing it or modify the solution's configuration following import.

Mandatory Options

You must configure the host option within the [cluster] section of a Stat Server solution in order to start the solution.

cluster Section

A Stat Server solution must contain the [cluster] section in order provide reporting for a SIP cluster. Table 3 describes this section's options

Table 3: Configuration Options for the cluster Section

Option	Description
host	Specifies the host name for the Stat Server solution. Stat Server uses this value as an integrity check only for Stat Server instances upon their start-up. If the host name of a Stat Server instance does not match that specified within this section, that Stat Server instance will not start. Specifying a value for this option is mandatory. Default Value: Not applicable Valid Values: host name Changes Take Effect: When Stat Server is restarted
reset-delay	Specifies the delay, in seconds, during which time Stat Server tries to synchronize data for a given interval from all nodes of the Stat Server cluster. This delay is applicable to reset-based statistics. The busier your network, the higher the value you should set for this option. Default Value: 2 Valid Values: 1–30 Changes Take Effect: When Stat Server is restarted

Table 3: Configuration Options for the cluster Section (Continued)

Option	Description
take-event-attached-data-changed-from-iproxy	Controls whether Stat Server will mask EventAttachedDataChanged TEvents through its T-Controller connection of SIP Server when SIP Server transmits attached data about regular DNs to Stat Server. If set to no, Stat Server receives and processes EventAttachedDataChanged TEvents through its connections with both T-Controller and Interaction Proxy; the traffic between SIP cluster and Stat Server cluster in this scenario is higher than the resulting traffic when this option is set to yes. If set to yes, Stat Server stops processing such TEvents through its T-Controller connection with SIP Server but continues processing them through its Interaction Proxy connection using Stat Server's best effort. Nodes connected to the Interaction Proxy interface of SIP Server propagate EventAttachedDataChanged TEvents through shared memory. • If EventAttachedDataChanged arrives too early (such as when there is lower latency on Stat Server's connection with Interaction Proxy than on its connection with T-Controller), Stat Server buffers EventAttached DataChanged TEvents. • If EventAttachedDataChanged arrives too late (such as when there is lower latency on Stat Server's connection with T-Controller than on its connection with Interaction Proxy), Stat Server discards EventAttached DataChanged (thereby causing some loss of data). • If a call is in progress during the startup of Stat Server node that is connected to T-Controller, changes to user data (propagated through shared memory) become visible only after other call-related events become visible and are received by that node. Data loss results if the node receives no other call-related events in this scenario. • If Stat Server's Interaction Proxy connection is lost, no changes to user data changes will be visible on the associated agent DNs until such time that the connection is reestablished. Default Value: no Valid Values: yes, no Changes Take Effect: When Stat Server is restarted **Note:** Setting this value to no can adversely impact performance.

Statistical Parameter Sections

All stat-type, time-profile, filter, and time-range sections of a Stat Server cluster must be defined at the solution level. When Stat Server operates in cluster mode (SS_c), Stat Server ignores any of these sections that might be defined at the Stat Server application level. Log-level options, must be defined at the application level. Conversely, SS_c ignores this section when it is defined on the Options tab of the Stat Server Solution object.

You define statistical parameter sections on the `Options` tab of the Stat Server solution in the same manner that you define these sections on the `Options` tab of a Stat Server application. Refer to the *Stat Server User's Guide* for this information.

Configuring High Availability for Stat Server Solutions

High availability (HA) for Stat Server solutions is not achieved in the same manner as high availability is achieved for Stat Server applications (described on page 15). However, you can simulate high availability by configuring an identical Stat Server solution that operates on a different host. Understand that in this release, this second solution is completely independent of the first.

Both the SIP Servers from which Stat Server receives TEvents and the clients that Stat Server services must be able to connect to either host. All instances of both clusters (primary and backup) must be running (with auto-restart enabled). And, all Stat Server instances must configure the `accept-clients-in-backup-mode` option (described in the next chapter) to `yes`. In this configuration, each Stat Server instance within the solution must specify its backup as one application from the second solution, but this is a solution for backup of solution components—not the solution itself.

Chapter

5 Fine-Tuning the Configuration of a Stat Server Application

This chapter describes the options that you can use on the Options tab to configure a Stat Server application. Refer to Chapter 6, “Other Factors Affecting Stat Server,” beginning on page 65, for descriptions of options in other Genesys applications that affect Stat Server behavior and Chapter 7, “Common Log Options,” beginning on page 71, for descriptions of log configuration options that are common to most Genesys server applications. Finally, to learn about the options that you can use to configure statistics for your Stat Server application, refer to the “Statistics Configuration Options” chapter in the *Framework 8.5 Stat Server User's Guide.*

The information in this chapter is divided among the following topics:

All of these options are defined on the Options tab of the Stat Server Application object.

Note: The configuration options that relate to Stat Server logging are described separately in Chapter 7.

Mandatory Options

You are not required to configure any options to start a Stat Server application.

statserver Section

You must name this section `statserver.` Table 4 lists some options available for configuring Stat Server `Application` objects. Use the options listed in Table 5 on page 49 to configure a Stat Server application to write data to a database. Use the options listed in Table 9 on page 63 to configure a direct connection to a database. And use the options listed in Tables 7 and 8, beginning on page 56, to enable Java functionality. You are directed to read your operating-system and/or Java Runtime Environment (JRE) vendor documentation to learn of any peculiarities regarding JVM installation or the setup of JRE. Table 6 lists the valid time-format codes you can use with the `time-format` configuration option (which is described in Table 4).

The notation in the upper right hand corner of each description in the tables indicates additional information:

- **C,** indicates that the option applies to Stat Server operation in cluster mode.
- **R,** indicates that the option applies to Stat Server operation in regular mode.

The absence of the **C** or **R** notation denotes the converse; namely, the option does not apply to Stat Server operation in cluster mode or regular mode, respectively. Note that nonapplicability does not stop Stat Server from reading the option, if it has been configured. Nonapplicable under these circumstances means that either Stat Server behavior cannot be altered by setting the option or that Stat Server will ignore the option and its value. Note also that if you specify an unsupported option in configuration, Stat Server will log the outcome and continue operating as if the option were never specified.

For those configuration options that indicate valid values of `true` and `false,` any of the following additional values are also valid:

- `t` and `f`
- `yes` and `no`
- `y` and `n`
- `1` and `0`
- `on` and `off`

These alternates might not be indicated within Tables 4–7. Also, the default values listed in the tables refer to those in the Stat Server application template , which is provided on the *Real-Time Metrics Engine* CD or that are inherent to Stat Server (if the options are not listed in the template). Lastly, where the name of a configuration option changed between releases, Stat Server continues to support the former name.

Table 4: Configuration Options for the statserver Section

Option	Description
accept-clients-in-backup-mode	C,R Specifies whether Stat Server accepts client connections when operating in backup mode. With this option set to `yes`, Stat Server notifies the clients about its redundancy mode after a client's registration and after a change in mode. Moreover, when its redundancy mode is changed to backup, Stat Server does not close the communication port and accepts clients' connections and requests. Default Value: `no` Valid Values: `yes`, `no` Changes Take Effect: When Stat Server is restarted
allow-asm-outbound-on-established	R Controls whether Stat Server is allowed to generate `ASM_Engaged` and `ASM_Outbound` actions upon `EventEstablished`. For all other cases, the option control is not applicable. When this option is set to `true`, Stat Server will try to recognize interaction-flow scenarios where the `GSW_RECORD_HANDLE` key is present in the `UserData` of `EventEstablished` TEvent and the `ANI` or `OtherDN` attribute points to a DN of type `Call Processing Port`. If these conditions are met, Stat Server starts these actions as follows: • Stat Server starts and ends the `ASM_Engaged` actions instantaneously. • Stat Server starts the `ASM_Outbound` action. Default Value: `true` Valid Values: `true`, `false` Change Take Effect: When Stat Server is restarted.
allow-vq-orig-dns-from-environment	R Specifies whether Stat Server will generate retrospective actions, reflecting regular DNs, to virtual queue objects that belong to the `Environment` tenant. If this option is set to `yes` and the `Environment` tenant is listed among those assigned to Stat Server (in a multi-tenant environment), Stat Server will generate such actions when these virtual queues are also assigned as origination DNs to `GroupAgents` and `GroupPlaces` objects belonging to a particular tenant (that is, to a tenant that is not the `Environment` tenant) as a result of call activity on that tenant. For this functionality to work properly, you must also set the `vq-treat-unknown-third-party-dn-as-agent-dn` option to `true` and the `vq-ignore-third-party-dn` option to `false`.

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
allow-vq-orig-dns-from-environment (continued)	Default Value: `yes` Valid Values: `yes, no` Changes Take Effect: When Stat Server is restarted **Note:** Stat Server operating in cluster mode behaves as if this option were forever set to `no`. No value for this option—whether specified or not—can change this inherent behavior.
auto-backup-interval	C,R Sets the time, in minutes, for checking persistent statistics and storing them in the file specified by the `backup-file-name` option. A value of `0` disables automated backups. This option was previously named `AutoBackupInterval`. Default Value: `15` Valid Values: Integers `0` through `2147483647` ($2^{31}-1$) Changes Take Effect: When Stat Server is restarted
backup-file-name	C,R Specifies the name of the backup file that stores persistent statistics for synchronization. Stat Server memorizes all parameters for statistics in demand, initiating their collection immediately after restart. If a particular statistic has not been requested for a long time period (three days, by default, as specified in the `old-stats-remove-interval` option), the statistics are removed from both the cache and the backup file. For operation in cluster mode, it is especially important that you specify a unique backup file name for each Stat Server node within a cluster. Upon startup, Stat Server reads the value of this option—and that of every other Stat Server node within the cluster—logs an error if Stat Server encounters a file collision, and exits. **Note:** Stat Server ignores backup files when: • They were generated by a different version of Stat Server. • You reconfigure the Stat Server solution. This option was previously named `BackupFileName`. Default Value: `ssbackup.000` Valid Values: Any valid path (optional) and file name Changes Take Effect: When Stat Server is restarted

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
capacity-treat-acw-as-interaction	C,R Determines whether Stat Server treats ACW activity as interactions while the associated DN is in after-call work (ACW) status. The routability of additional, simultaneous interactions to a device is dependent on the number of interactions that currently are occurring at that device. Setting this option to `yes` instructs Stat Server to treat any ACW activity as an interaction *for the purpose of determining capacity*—synonymous to any other type of voice interaction, such as handling customer-initiated (inbound) calls, internal calls among agents, and so forth. *For the purpose of reporting current activity,* this treatment does not increment the count of `CurrentNumber` or `TotalNumber` statistics. The presence of ACW on a device also affects the routability of interactions of other media types, as defined in the capacity model for your environment. For information about defining capacity rules, refer to the *Genesys Resource Capacity Planning Guide*. If this option is set to `no`, Stat Server does not consider ACW-related activities that occur at a device in its calculation of the `current_number` component of the capacity vector. In fact, Stat Server may allow additional, simultaneous interactions to be routed to that device per the capacity rules defined in your environment. Default Value: `no` Valid Value: `yes, no` Changes Take Effect: When Stat Server is restarted
check-stuck-calls	R When you set the value of this option to `yes`, Stat Server checks DNs of the `Extension`, `ACD Position`, `VTO` (IVR), `ACD Queue`, `Service Number`, and `Routing Point` types for calls with no activity during the last 10 minutes. When detecting such calls, Stat Server queries T-Server on the current DN status. If T-Server indicates that the call has been cleared from the DN in question, Stat Server deletes the call from memory. The checkup interval is 600 seconds (10 minutes). Stat Server does not check Internet DNs or DNs of `Virtual Routing Point` type. This option was previously named `CheckStuckCalls`. **Note:** Calls can be stuck in T-Server and/or Stat Server. The `check-stuck-calls` configuration option enables Stat Server to clear calls that it determines to be stuck in Stat Server. Default Value: `no` Valid Values: `yes, no` Changes Take Effect: Immediately upon notification

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
check-vq-stuck-calls-frequency	R Works in conjunction with the frequency of EventReserved_2 heartbeats that accompany live calls from URS, specifying the frequency, in seconds, with which Stat Server checks virtual queues for stuck calls. Stat Server suspends checks for stuck calls if no calls are queued at any virtual queue that Stat Server monitors. Stuck calls result if a URS connection breaks when a live calls is completed. In this situation, Stat Server does not receive the EventReleased TEvent to indicate the end of the call, and Stat Server views the interaction as continuing. Setting this option to a high value for large environments alleviates CPU load and helps to avoid situations where Stat Server inadvertently clears live calls due to network latency. In such situations, you should also consider resetting the call_kpl_time URS option, which measures the frequency of EventReserved_2 heartbeats, to a higher value. Small environments can set both options to relatively lower values. Also, you should be aware that setting call_kpl_time to 0 (zero) means that Stat Server will receive no EventReserved_2 events from URS. In this case, Stat Server considers all calls currently residing at the virtual queue as stuck and eliminates them from processing following the period of time specified by the check-vq-stuck-calls-frequency option. Refer to the *Universal Routing Reference Manual* for additional information about the call_kpl_time configuration option. Prior to release 8.0, this functionality was not configurable. Stat Server's checks for stuck calls was internally hard-coded at 60 seconds. Beginning with the 8.0 release, the default for this option, whether explicitly set or not, is 600 seconds. Default Value: 600 Valid Values: 30 to 2147483 (just over 24 days; 2147483=2^{31}/1,000) Changes Take Effect: When Stat Server is restarted
debug-level	C,R A comma-separated list of debug categories that are visible in the Stat Server log. This option is enabled only if you have set the verbose common log option to all. In graphical environments, log output often takes more than half of a server's execution time. To maintain performance, use only the debug levels that you need and run Stat Server in the background. Also, minimize the Stat Server window or redirect log output to a different device, such as a file. Be very careful, however, when directing log output to a file and consider the available free disk space, directory and file permissions, and possible conflicts with different software trying to use the log file at the same time.

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
debug-level (continued)	For each debug category, you can also set the level of debug logging by specifying a numerical value from [0–9] (with 9 being the most verbose) and appending the number to each category. For example: `Init, Status:6, Cluster:8` Debug level 0 is synonymous to no logging at all for the specified debug category. Debug levels 1–4 provide less logging information than was provided in prior releases but more than debug level 0. Debug level 5 provides exactly the same logging information that was provided in prior releases. This level is the default level if none is otherwise specified. Debug levels 6–7 provide more detailed output than level 5. Debug levels 8–9 provide the most extensive log output requiring further internal processing which, in turn, further degrades Stat Server performance. Default Value: Init Valid Values: all — Synonymous with Init, Server, Client, Status, Action, SQL, Mngmnt, Java, Reset for Stat Server in regular mode. Synonymous with Init, Server, Client, Cluster, Status, Action, Mngmnt, Reset for Stat Server in cluster mode. The debug level that you designate for this category supersedes any debug level that you designate for other categories. Action — Logs changes to the internal Stat Server object model and provides a significant source of troubleshooting data, which includes entries following every TEvent. Client — Logs all Stat Server communication with its clients, such as the opening of statistics and all statistical values sent to the client. This value generates a large amount of data, and should be sparingly used for troubleshooting reproducible problems with statistics. Cluster — Logs activity related to Stat Server operating in cluster mode. **Note:** This value does not apply to Stat Server operating in regular mode.

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description	
debug-level (continued)	`Init`	Used for capturing data related to Configuration Server that affects Stat Server, including dynamic Configuration Server changes made as Stat Server starts—such as the addition, deletion, and/or change of objects or their properties having an affect on Stat Server. This value is useful for tracking initial configuration and dynamic changes and is much more compact than the information provided in the Configuration Server log. Genesys recommends that you always include this value in this option.
	`Java`	Displays information related to Java extension functionality. Use this value only for statistics in the Outbound Contact 7.2.0^{+} or MCR 7.0.1^{+}. **Notes:** • MCR has been renamed eServices beginning with release 8.0. • This value does not apply to Stat Server operating in cluster mode.
	`Mngmnt`	Displays profiling information, including the number of currently connected clients, statistics being computed at the moment, and statistics to be reported to clients.
	`Reset`	Enables the log messages Stat Server sends to clients while sending statistics requested with a reset-based notification mode.
	`Server`	Logs T-Server events pertaining to Stat Server. Genesys recommends that you not include this value if you maintain logs for the related T-Server(s).
	`Shmem`	Logs shared memory communication among Stat Server cluster nodes. Applicable for Stat Server in cluster mode only.
	`SPT`	Logs events related to Stat Server startup. This value is provided to maintain backward compatibility and may be eliminated in future releases.
	`SQL`	Displays the SQL statements issued if you have configured a database for Stat Server. **Note:** This value does not apply to Stat Server operating in cluster mode.
	`Status`	Logs events related to the current state of objects and is useful for troubleshooting Stat Server–Router problems.

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
debug-level (continued)	Changes Take Effect: Immediately upon notification The SIP Server product provides a troubleshooting tool that parses the log output of several Genesys servers including Stat Server. Refer to the *SipSpan2 User's Guide* for information on how to use this tool. This option was previously named `DebugLevel`.
DefaultAgentSPT, DefaultDNSPT,	C,R These options create a precedence list of actions, which Stat Server uses to assign status to DNs, agents, places, or routing points, when there is more than one action occurring at each point. The `DefaultGroupSPT` and `DefaultGroupRPSPT` options available in the initial 7.0 release and prior releases are no longer required. For information on the operating mechanism of Status Priority tables, refer to the "Object Statuses" chapter in the *Framework 8.5 Stat Server User's Guide.* **Warning!** Do not change these options without consulting a Genesys technical representative. Default Value: `...` (an ellipsis) Valid Value: A list of actions separated by a comma or an ellipsis. If you specify a list, it overrides the list hard-coded in the Stat Server Status Priority tables. Changes Take Effect: When Stat Server is restarted
do-backup-in-background	C,R Specifies whether Stat Server spawns a separate thread to store statistic definitions in its backup file. If this option is set to `yes` (the default), Stat Server spawns a separate thread. If it is set to `no,` Stat Server writes to its backup file using the main thread. Default Value: `yes` Valid Values: `yes, no` Changes Take Effect: When Stat Server is restarted

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
filters-allow-wildcards-in-values	C,R Specifies whether Stat Server accepts the wild-card characters `*` and `?` in the `<value>` argument of `PairExist` functions in filters. If this option is set to `yes`, Stat Server interprets these characters as wildcards. If i is set to `no`, Stat Server interprets these as literal characters. Prior to release 7.5, Stat Server interpreted a `<value>` argument of `"*"` as *any string* and `"*"`, embedded within a string, as a literal character. For example, Stat Server interprets the `PairExist(KY1, "Mr.*")` function in one of two ways depending on the value of the `filters-allow-wildcards-in-values` option: • As a function whose filter returns any statistic where the values for `KY1` begin with `Mr.`, if the value of this option is set to `yes`. • As a function whose filter returns only those statistics where the value for `KY1` is equivalent to the four characters `Mr.*` if the value of this option is set to `no`. Stat Server interprets the `PairExist(KY2, "*")` filter as one where `KY2` is equal to any number of characters regardless of the value of this option. Default Value: `no` Valid Values: `yes, no` Changes Take Effect: When Stat Server is restarted
generate-stat-validity-events	C,R Controls whether Stat Server sends `EventStatInvalid` and `EventStatValid` events to Stat Server clients: Default Value: `yes` Valid Values: `yes, no` Change Takes Effect: When Stat Server is restarted. Setting this option to `no` can reduce load on Stat Server and its clients in scenarios in which, due to disconnects with T-Server, Stat Server generates large volumes of `statvalid` and `statinvalid` messages (in addition to statistic and registration messages) that otherwise could affect Stat Server operation negatively. This feature is intended to be used in Stat Server applications that service Data Sourcer 7.6–only clients. Setting this option to `no` is not recommended in Stat Server applications that service other types of clients—especially Universal Routing Server clients that rely on `statvalid/statinvalid` events for making routing decisions.
generate-transfer-taken-on-ringing	C,R Controls when Stat Server generates the `CallTransferTaken` action for a transferred call—either while it is ringing at a regular DN or after it has been answered on that DN.

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
generate-transfer-taken-on-ringing (continued)	Default Value: `yes` Valid Values: `yes, no` Changes Take Effect: When Stat Server is restarted. If this option is set to `yes,` Stat Server generates `CallTransferTaken,` as in previous releases: • If the transfer completes on a regular DN while ringing or on some mediation DN before distribution to a regular DN. • After the call has been answered, if the transfer completes following receipt of `EventEstablished` on a regular DN. If this option is set to `no,` Stat Server does not generate the `CallTransferTaken` action while the transferred call is ringing. Instead, Stat Server generates this action after the transferred call has been answered (that is, upon receipt of `EventEstablished`). **Note:** Stat Server does not generate the `CallTransferTaken` action for direct single-step transfers made to agents or routing points that are located at different sites whether or not Stat Server monitors such agents.
ignore-disabled-objects-in-group-statistics	C,R Specifies whether Stat Server takes into account the `Person` and `Place` objects that are disabled in the Configuration Layer when calculating statistics for corresponding groups of objects. Setting this option to `yes` excludes agents and places in the calculation of group status for `CurrentState` statistics as long as the relevant `Person` and `Place` configuration objects are disabled in the Configuration Layer. This option also affects any number-related group statistics in the same manner. Genesys recommends setting this option to `yes` for a Stat Server application serving Universal Routing Server, and `no` (the default) for a Stat Server application serving CCPulse+ and/or CC Analyzer. Default Value: `no` Valid Values: `yes, no` Changes Take Effect: When Stat Server is restarted
ignore-disabled-objects-in-queue-statistics	C,R In the 8.1.0$^-$ releases, this option specifies whether Stat Server takes into account disabled `Person` and `Place` objects when calculating certain queue statistics. When this option is set to `yes,` Stat Server abstains from updating queue statistics having any of the `ActionLogin, AgentReady,` and `AgentActive` actions in the main mask while `Person` and `Place` objects continue to be disabled. When set to `no,` Stat Server considers all `Person` and `Place` objects—disabled or enabled—and all masks in computations of queue statistics.

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
ignore-disabled-objects-in-queue-statistics (continued)	Default Value: `no` Valid Values: `yes, no` Changes Take Effect: When Stat Server is restarted **Note:** The aforementioned statistics are such that actions are propagated from a place to a queue only when an agent is logged on to the place and either of the following: • Both the agent and place are enabled, in which case the value of this configuration option is irrelevant. • The agent and/or place is disabled and the value of this option is set to `no`. Starting with 8.1.2 release, only the agent should be enabled to be accounted for in queue statistics that use `ActionLogin, AgentReady,` and `AgentActive` actions.
ignore-off-hook-on-position	R Specifies whether to ignore `On-Hook/Off-Hook` events on `Position` DNs. If this option is set to `yes`, `On-Hook/Off-Hook` events are ignored on `Position` DNs. Use this option if your version of T-Server does not properly propagate `On-Hook` or `Off-Hook` TEvents. This option was previously named `IgnoreOffHookOnPosition`. Default Value: `no` Valid Values: `yes, no` Changes Take Effect: When Stat Server is restarted
load-balance-aht	R Specifies the initial value, in seconds, for handling time. Stat Server uses this figure in the operand of the formula for calculating load-balancing of all mediation DNs that Stat Server monitors. Refer to the `LoadBalance` statistical category in the *Stat Server User's Guide* for more information about this formula. You can also configure this value locally within the application options of mediation DN objects. A value that is specified at the mediation DN level supersedes the values that are specified within the Stat Server `Application` object for that mediation DN. Refer to page 68 of this document for information about configuring this option within mediation DN objects. Prior to release 8.0, this value was hard-coded at `90` seconds. Default Value: `90` Valid Values: Positive integers less than `4294967296` (2^{32}) Changes Take Effect: When Stat Server is restarted (when defined within the Stat Server `Application` object).

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
management-port	C,R Specifies the TCP/IP port that Stat Server reserves for connections that its SNMP Option Management Client establishes. **Note:** For Stat Server operating in cluster mode, specify a different management port for each node in the cluster. **Warning!** You must specify a value for this option if you are using an SNMP connection. Do not change the value for this option while Stat Server is running. Default Value: `3031` Valid Values: Any available TCP port (Integers `1` through `65535`) Changes Take Effect: When Stat Server is restarted
max-client-connections	C,R Specifies the maximum number of clients that can be connected to Stat Server at any given time. The default value, `-1,` or `0` (zero) indicates that an unlimited number of clients can be connected to Stat Server. Default Value: `-1` Valid Values: `-1, 0,` or any positive integer less than `2147483648` (2^{31}) Changes Take Effect: When Stat Server is restarted
nec-position-extension-linked	R Specifies whether Stat Server applies a special model when processing after-call work (ACW) notifications from NEC T-Server. This model, normally used with Meridian T-Server, consists of Position and Extension DNs linked together in Stat Server logic when they belong to the same phone. Refer to the *Framework 8.5 Stat Server User's Guide* for a description of the `AfterCallWork` action and models for its generation. **Note:** For switch types, such as the Nortel Meridian, in which a place is configured with both Position and Extension DNs and in which an agent is required to log in to the Position DN, this option must be set to `yes` in order for `EstimWaitTime` and `LoadBalance` statistics to return expected values. The option name is case-insensitive. Default Value: `no` Valid Values: `yes, no` Changes Take Effect: When Stat Server is restarted

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
old-stats-remove-interval	C,R Sets the amount of time, in minutes, that unused statistics should continue to calculate on Stat Server. A value of `0` causes Stat Server to close a statistic as soon as the application requesting it closes its request or disconnects. This option was previously named `OldStatsRemoveInterval`.The default template does not include this option. The internal, hard-coded default value is `4320` (three days). Default Value: `4320` Valid Values: Integers `0` through `2147483647` ($2^{31}-1$) Changes Take Effect: When Stat Server is restarted
position-extension-linked	R Specifies how Stat Server interprets the status of a place and an agent when the place contains a position and an extension that belong to the same switch. Default Value: `yes` Valid Values: `yes, no` Changes Take Effect: When Stat Server is restarted By default (`yes`), the status of a DN of the `Extension` type affects the place status under these conditions: • An agent is logged in at the DN of the `Position` type that belongs to the same place. • An agent may or might not be logged in at the DN of the `Extension` type. With the option set to `no,` the status of the DN of the `Extension` type affects the place status under these conditions: • An agent might or might not be logged in at the DN of the `Position` type that belongs to the place. • An agent *must* be logged in at the DN of the `Extension` type.
queue-disable-dcid-for-missed-calls	R Controls whether Stat Server ignores the `CONNID` attribute of the `ACWMissed` and `CallMissed` actions on mediation DNs: Default Value: `no` Valid Values: `yes, no` Changes Take Effect: When Stat Server is restarted. In environments that contain a large number of origination DNs, setting this option to `yes` has the beneficial side effect of improving Stat Server performance as well as the unfortunate effect of Stat Server no longer being able to distinguish interactions by connection ID when `Formula` is set to `DCID` for statistics that have the `ACWMissed` and/or `CallMissed` actions specified in the main mask.

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
queue-use-pseudo-actions	R Restricts Stat Server's use of the following mediation DN actions to unfiltered statistics defined using only the `CurrentNumber` or `CurrentRelativeNumberPercentage` statistical categories: • `DNLogin` • `DNActive` • `DNReady` • `AgentLogin` • `AgentActive` • `AgentReady` If this option is set to `true,` Stat Server enables this restriction and minimizes the possible overhead that could results in environments that contain a large number of origination DNs assigned to `GroupPlaces` or `GroupAgents` objects. If it is set to `false,` all statistical categories (but not filters) will be applicable to the listed actions. Default Value: `true` Valid Values: `true, false` Changes take effect: When Stat Server is restarted
reconnect-timeout	R Indicates the time interval, in seconds, between Stat Server attempts to reconnect to a T-Server or the database (DB Server if `[db-direct-connection]/enable` is set to `no`) if either is disconnected or not running. This option was previously named `reconnect_timeout` (spelled with an underscore). Default Value: `10` Valid Values: Positive integers less than `4294967296` (2^{32}); Stat Server sets any negative or `0` values that you might configure to `10`. Changes Take Effect: When Stat Server is restarted
reg-delay	R Causes Stat Server to wait the specified number of seconds before registering DNs that have been added to Configuration Server. This option was previously named `reg_delay` (spelled with an underscore). Default Value: `3` Valid Value: `0` (zero) or positive integers less than `4294967296` (2^{32}) Changes Take Effect: When Stat Server is restarted

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
reg-dns-chunk-delay	R Specifies the interval, in seconds, between two subsequent DN registration requests. Stat Server waits for the specified interval before sending a request to T-Server to register a subsequent set of DNs, thus allowing T-Server to process the previous request. In a large configuration environment, use this option in conjunction with reg-dns-chunk-volume to optimize DN registration at Stat Server startup. Default Value: 10 Valid Values: Positive integers less than 4294967296 (2^{32}) Changes Take Effect: When Stat Server is restarted
reg-dns-chunk-volume	R Specifies the number of DNs that Stat Server submits in a single registration request to T-Server. Instead of trying to register for all configured DNs at once, Stat Server divides the DN registration among several requests, each for the specified number of DNs. In a large configuration environment, use this option in conjunction with reg-dns-chunk-delay to optimize DN registration at Stat Server startup. Default Value: 1000 Valid Values: Any positive integer less than 2147483648 (2^{31}) Changes Take Effect: When Stat Server is restarted
rp-handle-queueing-events	C,R Controls Stat Server's recognition of the CallState attribute of EventQueued and EventRouteRequest TEvents that occur at routing points. Default Value: no Valid Values: yes, no Changes Take Effect: When Stat Server is restarted. If this option is set to yes, Stat Server analyzes the CallState attribute value on EventQueued and EventRouteRequest TEvents that might occur at routing points. If it is set to no, Stat Server ignores EventQueued TEvents that occur at routing points and considers only EventRouteRequest TEvents when it analyzes the CallState attribute. Setting this option to yes enables Stat Server to count the correct number of transfers that are taken for single-step transferred calls that pass through routing points in a single-site environment, such as SIP Server.

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
send-timeout	C,R Specifies the interval, in seconds, that Stat Server keeps client requests in the output queue. When this timeout expires for a given client request, Stat Server disconnects this client as being "too slow." Consider increasing this option's value in an environment with a slow network or where client disconnects are frequent. Default Value: `300` Valid Values: `60-3600` (1 hour) Changes Take Effect: When Stat Server is restarted
show-attached-data	R Beginning with release 8.1.1, this option is obsolete. Refer instead to the description of the `default-filter-type` log option on page 90. For Stat Server 8.1.0 and prior releases, if this option is set to `yes`, Stat Server outputs call-extracted `UserData` to the Stat Server log. If it is set to `no`, Stat Server stops outputting attached data to its log—regardless of the log-level setting (`trace`, `debug`, and so forth). T-Server propagates attached data (`UserData`) by way of TEvents; this data is used for internal computations. To output `UserData` to the log, in the `Filters` section of the Stat Server application object, add a `PairExist("key","value")` filter where `key` is the name of the `UserData` key; `value` may denote a specific value or `"*"`. Setting this option does not affect Stat Server's processing of `UserData`. For memory, performance, and security reasons, however, Stat Server strips away any attached data that is not directly used for internal computations. Refer to the `UserData` property in the "Call Properties" table of the *Stat Server User's Guide* for more information. Default Value: `no` Valid Values: `yes`, `no` Changes Take Effect: Immediately upon notification
show-queued-interactions	C,R Controls the appearance of the list of queued interactions for mediation DNs in the Stat Server log. Default Value: `yes` Valid Values: `yes`, `no` Changes Take Effect: Immediately upon notification When this option is set to `yes`, the Stat Server log will contain log entries for every interaction within each mediation DN. When it is set to `no`, Stat Server displays only the number of interactions in the mediation DN. **Note:** This option does not pertain to interaction queues that are controlled by Interaction Server.

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
subscribe-for-all-ixn-server-events	R Limits the types of events that Stat Server receives from Interaction Server and, as a result, improves performance for environments that regularly handle a high volume of interactions. Default Value: `no` Valid Values: `no, yes` Changes Take Effect: When Stat Server is restarted If this option is set to `no,` Stat Server subscribes from Interaction Server for place-related events only. Note that because Stat Server will not receive other types of events, this setting might cause Stat Server to miscalculate other than place-related statistics. A change in userdata that is detected by Interaction Server, for instance, will not be known to Stat Server because Interaction Server will not transmit `event_properties_changed` events when this option is set to `no`. In order to receive all Reporting protocol events, this option must be set to `yes` at all times.
suppress-agent-status-updates-for-ixn-server	R Enables suppression of `EventCurrentAgentStatus` notifications by Stat Server in environments that deploy multiple Stat Server applications. Disabling this statistic request from select clients avoids situations in which Stat Server clients receive multiple and identical notifications about current status for the same agent. Default Value: `no` Valid Values: `no, yes` Changes Take Effect: Immediately upon notification
suppress-user-data	R You configure this option in the `[statserver]` section on `Options` tab of switch and/or DN objects. Refer to “To Suppress the Transmission of Attached Data” on page 66 for the description and permissible values of this option.
vag-statistics-active-agents-only	C,R Limits the membership of virtual agent groups to only those active agents satisfying a particular script condition. (An active agent is `Person` object that has been enabled in Configuration Server.) Default Value: `no` Valid Values: `yes, no` Changes Take Effect: When Stat Server is restarted

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
vq-ignore-third-party-dn	R Controls whether Stat Server relies on the ThirdPartyDN attribute of EventDiverted TEvents to determine the DN to which a call was diverted from a given virtual queue. Default Value: true Valid Values: true, false Changes Take Effect: Upon DN re-registration **Note:** Stat Server operating in cluster mode behaves as if this option were forever set to true. No value for this option—whether specified or not—can change this inherent behavior.
vq-treat-unknown-third-party-dn-as-agent-dn	C,R Indicates whether Stat Server generates the CallAnswered action for virtual queue objects in the following scenario: 1. Stat Server receives an EventDiverted TEvent for the virtual queue. 2. The ThirdPartyDN attribute value of this TEvent contains the ID of an unknown DN—one that is monitored by a switch other than that to which the virtual queue belongs. 3. The call is subsequently routed to an agent. If this option is set to true, Stat Server generates the CallAnswered action under the preceding circumstances. If it is set to false, Stat Server does not generate this action under the same circumstances. If the ThirdPartyDN attribute value is null or contains an ID that coincides with that of the answering DN, Stat Server generates the CallAnswered action on virtual queue objects, regardless of this option's setting. **Note:** The vq-ignore-third-party-dn option must be set to false in order for Stat Server to consider the value of this option. Default Value: yes Valid Values: yes, no Changes Take Effect: When Stat Server is restarted
vq-use-alt-enter-time	C,R Controls whether Stat Server uses an alternative enter time when it calculates the durations of some actions in some scenarios that involve virtual queues.

Table 4: Configuration Options for the statserver Section (Continued)

Option	Description
vq-use-alt-enter-time (continued)	Specifically, prior to release 7.6.100.43, when Stat Server received `EventPartyChanged` with `CallState=ok` on a virtual queue and the connection ID differed from the previous connection ID (`Connid!=Previous Connid`), Stat Server considered each receipt of `EventPartyChanged` to constitute a new call and, therefore, reset the durations of the following retrospective actions: • `CallAnswered` • `CallDistributed` • `CallAbandoned` • `CallAbandonedFromRinging` • `CallRingingPartyChanged` • `CallForwarded` • `CallCleared` Beginning with release 7.6.100.43, Stat Server supports this scenario by *not* updating the enter time—this becomes the “alternate enter time”—that is associated with the previously listed actions when either the `vq-use-alt-enter-time` option or `use-alt-enter-time` local DN-level option (described on page 68) is set to `yes`. The scenario, where `Connid!=PreviousConnid`, is common in some SIP deployments in which Stat Server receives multiple `EventPartyChanged` TEvents for a call that remains in a virtual queue waiting for its target to become available. The permissible values for this option are the following: Default Value: `no` Valid Values: `yes`, `no` Changes Take Effect: When Stat Server is restarted.
xx-disconnect-clients-on-ixn-server-disconnect	R Controls whether Stat Server disconnects all clients—including voice clients—upon receiving notification of disconnection from Interaction Server: Default Value: `no` Valid Values: `yes`, `no` Changes Take Effect: When Stat Server is restarted. In large environments, setting this option to `yes` enables Stat Server to handle Interaction Server disconnections more efficiently by ceasing to open new statistics from Stat Server clients, a time-consuming operation in very large environments. It is assumed that you will perform the reconnection after the Interaction Server disconnect has been resolved.

Table 5: Configuration Options for Operating Stat Server with a Database

Option	Description
binding-threshold	R Specifies the number of records in a binding block—that is, the number of records to be sent to the DBMS simultaneously. This option is enabled only if you have set the value of the `enable-binding` configuration option to `yes`. The default template does not include this option. Default Value: `10` Valid Values: Any positive integer less than `2147483648` (2^{31}) Changes Take Effect: When Stat Server is restarted
enable-binding	R Specifies whether to enable binding functionality. By default, Stat Server uses a regular method of sending requests. If you set the value of this option to `yes,` Stat Server uses binding for sending requests. This option works in conjunction with the `binding-threshold` configuration option and is supported for Oracle, Microsoft SQL, and DB2 relational database management systems. The default template does not include this option. This option was previously named `OracleBinding`. Default Value: `no` Valid Values: `yes, no` Changes Take Effect: Immediately upon notification
identity-in-login-table	R Turning this option on enables Stat Server to operate with an Oracle Real Application Clusters (RAC). This option requires a database access point connection to an Oracle RAC database. The user must also initialize their LOGIN table with the oracle/login_oracle.sql script that comes with the Stat Server 8.5 installation. Default Value: `off` Valid Values: `on, off` Changes Take Effect: When Stat Server is restarted

Table 5: Configuration Options for Operating Stat Server with a Database (Continued)

Option	Description
ixn-id-in-status-table	R Specifies whether Stat Server will populate the IxnID field for records written to the STATUS table. If you set this option to off, or if you do not configure this option, the IxnID field will be null. This field provides functionality, comparable to connection IDs for calls, for Multimedia interactions that rely predominantly on the number generated by Interaction Server for identification in the interaction flow. **Note:** If you set this option to on, consider also setting the multimedia-activity-in-status-table configuration option to yes, so that Stat Server will record information about the status of multimedia interactions in the other fields of the STATUS table. Default Value: off Valid Values: on, off Changes Take Effect: Immediately upon notification Refer to A, “Physical Data Models for Stat Server Tables” on page 119 for a complete description of the STATUS table. **Warning**! To avoid data loss, do not change the setting of this option in runtime if you have also set enable-binding to yes.
local-time-in-status-table	R Specifies whether to populate the StartLocalTime and EndLocalTime fields in the STATUS table. If you set the value of this option to off, or if you do not specify a value, the StartLocalTime and EndLocalTime fields will contain no data. For Solution Reporting applications, set this option to off; such reports do not use the local time fields, and setting this option to on could affect performance. When setting this option to on, also set the time-format option to the desired format. Refer to page 124 for a complete description of the STATUS table. This option was previously named LocalTimeInStatusTable. Default Value: off Valid Values: on, off Changes Take Effect: Immediately upon notification **Warning!** To avoid data loss, do not change the value of this option in runtime if you have also set enable-binding to yes.

Table 5: Configuration Options for Operating Stat Server with a Database (Continued)

Option	Description
login-table	R Specifies whether Stat Server writes records about login and logout TEvents directly to the LOGIN table in the Stat Server database. Refer to page 121 for a complete description of this table. This option was previously named LoginTable. Default Value: off Valid Values: on, off Changes Take Effect: Immediately upon notification
max-unsent-sql-statements	R Specifies the maximum number of SQL statements that Stat Server is allowed to maintain in memory. As soon as Stat Server's connection to the RDBMS is broken, Stat Server starts storing SQL statements in memory. These statements will be issued against the Stat Server database once the connection is restored. If the number of SQL statements in memory exceeds the value that is specified by this option, data loss might result. To avoid data loss, Stat Server must remain connected to the database for the entire period of the records submission to the RDBMS. Your addp timeout for connection from Stat Server to DB Server should be set as large as possible to prevent disconnection by addp. Refer to the *Framework Deployment Guide* for information about setting addp. If the number of SQL statements in memory ever exceeds this option's value, data loss of the entire memory pool will result and the accumulation of SQL statements will begin anew. Please be aware that setting this option's value too high might cause your system to run out of memory. Configure this option in conformance with the amount of RAM installed on the machine where Stat Server operates. If you specify any value that is less than the default (100000), Stat Server resets it to 100000. Default Value: 100000 (SQL statements) Valid Values: Integers greater than or equal to 100000 and less than 2147483648 (2^{31}). Changes Take Effect: When Stat Server is restarted

Table 5: Configuration Options for Operating Stat Server with a Database (Continued)

Option	Description
multimedia-activity-in-status-table	R Specifies whether multimedia-related actions are counted while computing status values that are written to the STATUS table. (For a complete classification of actions, refer to the *Stat Server User's Guide.*) If this option is set to no, Stat Server ignores multimedia-related actions in its computation of place and agent status. Stat Server also reads the value of the multimedia configuration option in the TServer section of the monitored DN (whose type is Extension) to determine whether the corresponding DN is a multimedia DN, capable of processing interactions of different media types, such as those DNs that are controlled by a SIP-compliant T-Server. Refer to page 66 for more information. **Note:** If you set this option to yes, you might also consider setting the ixn-id-in-status-table configuration option to yes so that Stat Server populates the IxnID field for multimedia interactions. Default Value: yes Valid Values: no, yes Changes Take Effect: When Stat Server is restarted
qinfo-table	R Specifies whether Stat Server writes records about queue statuses directly to the QINFO table. Refer to page 122 for a complete description of this table. This option was previously named QInfoTable. Default Value: off Valid Values: on, off Changes Take Effect: Immediately upon notification
status-table	R Specifies whether Stat Server writes records about agent statuses directly to the STATUS table. Refer to page 124 for a complete description of this table. This option was previously named StatusTable. Default Value: off Valid Values: on, off Changes Take Effect: Immediately upon notification

Table 5: Configuration Options for Operating Stat Server with a Database (Continued)

Option	Description
status-table-update-end-time-at-end-only	R Setting this option to `yes` enables Stat Server to set the `EndTime` and `EndLocalTime` fields of the `STATUS` table to `0` (zero) during updates, provided that the corresponding status has not yet ended. A zero value implies `0` for integer fields and `""` (empty string) for character fields. As soon as the statuses complete, Stat Server updates those fields with the time when the statuses ended. Default Value: `no` Valid Values: `yes, no` Changes Take Effect: When Stat Server is restarted
time-format	R Specifies the time format of data stored in the `StartLocalTime` and `EndLocalTime` fields in the `STATUS` table. You must set the `local-time-in-status-table` option (see page 50) to `yes` to use the `time-format` option. The format string consists of one or more codes preceded by a percent sign (%). Character strings that do not begin with % are copied unchanged to `strDest`. This option was previously named `TimeFormat`. Default Value: `%m/%d/%Y %H:%M:%S` Valid Values: See Table 6 on page 55 for a complete listing and description of valid time formats. Changes Take Effect: When Stat Server is restarted **Example** Suppose you are using the default time format `%m/%d/%Y %H:%M:%S.` If the start time for a particular state is Tuesday, January 1, 1999, at 3 PM and 10 seconds, character data stored in the `STARTLOCALTIME` field in the `STATUS` table is stored as `01/01/1999 15:00:10`. Changing the format codes for the date in the `time-format` option to `%Y/%m/%d` means the date is stored in the international date format as `1999/01/01`. Spaces can also be used. For example, `%Y %m %d` would be stored as `1999 01 01`.

Table 5: Configuration Options for Operating Stat Server with a Database (Continued)

Option	Description
use-server-id	R This option prevents constraint-violation errors from occurring in a database when more than one Stat Server application attempts to write to the same database. If only one Stat Server application writes to the same database table or you have set the value of the `status-table` option (see page 52) to `no`, you do not have to specify a value for this option. The default template does not include this option. To set this option, enter any number from `0` to `63`. Use a different value for each Stat Server application that writes to the same database table. Each Stat Server application uses its assigned value to generate internally stored IDs. **Note:** Configure this option only for those Stat Server applications writing to the same database and monitoring different switches. Do not configure Stat Server applications to write to the same database if they monitor the same switches. This option was previously named `UseServerID`. Default Value: No default value Valid Value: Any integer from `0` (zero) to `63` Changes Take Effect: Immediately upon notification
voice-reasons-table	R Specifies whether Stat Server stores the reasons for agents to change or continue `Ready` and `NotReady` states and `AfterCallWork` work mode. If this option is set to `yes`, Stat Server writes the reasons records directly to the `VOICE_REASONS` table. Refer to page 126 for a description of this table. Default Value: `no` Valid Values: `yes`, `no` Changes Take Effect: Immediately upon notification
warn-unsent-sql-statements	R Defines the threshold upon which Stat Server begins logging warning messages about the number of unsent SQL statements. To avoid data loss, Stat Server must remain connected to the database for the entire period of the records submission to the RDBMS. If you use DB Server for connection to RDBMS your addp timeout for connection from Stat Server to DB Server should be set as large as possible to prevent disconnection by addp. Default Value: `5000` (SQL statements) Valid Values: Any positive value, both less than `2147483648` (2^{31}) and less than the value that is specified by the `max-unsent-sql-statements` configuration option. Changes Take Effect: When Stat Server is restarted

Table 6: Valid Time-Format Codes

Format Code	Description
%a	Abbreviated weekday name
%A	Full weekday name
%b	Abbreviated month name
%B	Full month name
%c	Date and time representation appropriate for locale
%d	Day of month as a two-digit number (01–31)
%H	Hour in 24-hour format (00–23)
%I	Hour in 12-hour format (01–12)
%j	Day of year as a three-digit number (001–366)
%m	Month as a two-digit number (01–12)
%M	Minute as a two-digit number (00–59)
%p	Current locale's AM/PM indicator for 12-hour clock
%S	Second as a two-digit number (00–59)
%U	Week of year as a two-digit number, with Sunday as the first day of week (00–51)
%w	Weekday as a one-digit number (0–6; Sunday is 0)
%W	Week of year as a two-digit number, with Monday as first day of week (00–51)
%x	Date representation for current locale
%X	Time representation for current locale
%y	Year without century, as a two-digit number (00–99)
%Y	Year with century, as a four-digit number (1970–*x*)
%z, %Z	Time-zone name or abbreviation; no characters, if time zone is unknown
%%	Percent sign
%#c	Long date and time representation, appropriate for current locale—for example, `Wednesday, March 14, 2001, 12:41:29`

Table 6: Valid Time-Format Codes (Continued)

Format Code	Description
%#x	Long date representation, appropriate to current locale—for example, `Wednesday, March 14, 2001`
#	The pound sign (#) can precede any formatting code. This changes the meaning of the format code as shown in entries with the pound sign in this table. **Notes:** • The pound sign is ignored in these format codes: `%#a, %#A, %#b, %#B, %#p, %#X, %#z, %#Z, %#%` • The pound sign in these format codes removes any leading zeroes: `%#d, %#H, %#I, %#j, %#m, %#M, %#S, %#U, %#w, %#W, %#y, %#Y`

Table 7: Java-Related Options in the [statserver] Section

Option	Description
debug-level	R Adding `Java` to the value of this option enables Stat Server to log messages that are related to Java extension functionality. For the complete description of this option, see page 35.
enable-java	R When you set the value of this option to `true,` Stat Server tries to load JVM at startup.The `jvm-path` configuration option described on page 59 defines the location of JVM. If you set this value to `false` at Stat Server startup, but later set it to `true,` Stat Server attempts to load JVM at runtime. **Note:** Stat Server ignores the change in setting from `true` to `false.` To unload JVM, you must stop Stat Server. Default Value: `false` Valid Values: `true, false` Changes take effect: When Stat Server is restarted

Java Sections

Note: For this Stat Server release, Java functionality is reserved for use in conjunction with Genesys-provided reports for Outbound Contact and eServices (formerly known as Multimedia). You cannot use Java extensions while Stat Server operates in cluster mode.

Upon startup, Stat Server reads the enable-java configuration option to determine whether SSJE (Stat Server Java Extension) functionality is enabled. If the value of this option is true, Stat Server processes the information specified in the following Stat Server sections:

- [java-config]
- [jvm-options]
- [java-extensions]

using the following high-level procedure:

1. Stat Server verifies that the [java-config] section exists.
2. Stat Server verifies that the jvm-path option within that section has been specified.
3. Stat Server verifies that the [jvm-options] section exists.
4. If all three are true, Stat Server loads Java Virtual Machine (JVM) from the path specified by jvm-path using any options that you might have specified within the [jvm-options] section.

   For Stat Server to be able to load JVM, a platform-appropriate environment variable has to be set on the host:

   - LD_LIBRARY_PATH for Linux/Solaris.
   - LIBPATH for AIX.
   - PATH for Microsoft Windows.

   In general, the parent folder of the jvm.dll or libjvm.so (specified in the jvmpath option) should be included within the value of the environment variable.

   For example:

   - On Linux or Solaris, if the location of the file is /usr/java/jdk1.7.0_60/jre/lib/amd64/server/libjvm.so, then LD_LIBRARY_PATH should contain /usr/java/jdk1.7.0_60/jre/lib/amd64.
   - On AIX, if the location of the file is /usr/java/sdk7/jre/lib/ppc64/j9vm/libjvm.so, then LIBPATH should contain /usr/java/sdk7/jre/lib/ppc64.
   - On Windows, if the location of the file is C:\Java\jre7\bin\server\jvm.dll, then PATH should contain C:\Java\jre7\bin.
5. Stat Server loads Java classes from the Genesys Platform SDK (kv65_adapter.jar and kvlists.jar) and from the Stat Server Java SDK (statserver.jar and statserver_impl.jar).
6. Stat Server loads the Java libraries indicated by the java-libraries-dir configuration option of the [java-config] section.

7. If Stat Server successfully loads the Java host environment, Stat Server next tries to load Java Extensions (specified by the `java-extensions-dir` configuration option of the `[java-config]` section) from archives specified in the `[java-extensions]` section.
8. Stat Server takes the initial parameters for each `<extension.jar>` extension from the section where `java-extension-jar=<extension.jar>`, and uses them for this extension execution.

Each Java configuration section is further described in Table 8. For those configuration options for which you specify `true/false` values, any of the following additional values are also valid:

- `yes` and `no`
- `y` and `n`
- `1` and `0`
- `on` and `off`

These alternates might not be indicated in the table.

Table 8: Java Sections and Pertinent Configuration Options

Option	Description
java-config Section	
java-extensions-dir	R The value of this option must contain the path to the directory where all Java Extensions are stored. Default Value: `./java/ext` Valid Value: Any valid, fully specified directory path Changes Take Effect: Upon Stat Server restart, or upon setting the `enable-java` configuration option to `true`
java-libraries-dir	R The value of this option must contain the path to the directory where all Java libraries are stored. Default Value: `./java/lib` Valid Value: Any valid, fully specified directory path Changes Take Effect: Upon Stat Server restart, or upon setting the `enable-java` configuration option to `true`

Table 8: Java Sections and Pertinent Configuration Options (Continued)

Option	Description
java-extension-loading-timeout	R Specifies the length of time, in seconds, that Stat Server allocates for loading Java Extensions. If an Extension does not load within this timeout, Stat Server sends a message to its log indicating this. Stat Server makes no further attempts to load the Extension during runtime. Default Value: `20` Valid Value: Any positive integer less than `2147483648` (2^{31}) Changes Take Effect: Upon Stat Server restart, or upon setting the `enable-java` configuration option to `true` Only under rare circumstances should you change this option, such as if your particular Java Extension is very large or if its execution is very time consuming.
jvm-path	R The value of this option must contain the path to JVM: • `jvm.dll` on Windows • `libjava.so`, `libjvm.so`, `libjvm.a`, or `libjvm.sl` on UNIX Default Value: No default value Valid Value: Any valid, fully specified path (including file name) to the particular file Changes Take Effect: Upon Stat Server restart, or upon setting the `enable-java` configuration option to `true`

Table 8: Java Sections and Pertinent Configuration Options (Continued)

Option	Description
jvm-option Section R The configuration options you specify for this section correspond to the Java executable (`java.exe` on Windows, `java` on UNIX), and command-line options specific to your branch and version of JVM. Refer to your JVM documentation to find out its applicable configuration options. For Solaris platforms, set the stack space to at least 4,096K. For example, for HotSpot JVM, configuring the following would accomplish this: `-XX:ThreadStackSize=4096` Note that Genesys neither recommends nor endorses any particular JVM. Configuration options follow the `Name/Value` format used in other Stat Server sections, where `Name` is the name of the Java command-line option. If you specify a value for a named configuration option in this section, Stat Server converts the two to `Name=Value` before passing the option to JVM. If you do not specify a value, Stat Server passes the name only.	

Example 1	Assume that `foo` is a valid option requiring a value for your Java application. To specify a value of `some string,` create the following configuration option within the `[jvm-options]` section of your Stat Server application.
	`Name = -Dfoo`
	`Value = "some string"`
	Note: Include quotes in the value's definition, if JVM requires them on the command line.

Example 2	This example demonstrates how to configure an option—the Java HotSpot Client VM—that does not require a value.
	`Name = -Client`
	Note: You must include the hyphen if JVM requires it.
	`Value = <null>`

Table 8: Java Sections and Pertinent Configuration Options (Continued)

Option	Description
java-extensions Section Default behavior creates neither this section nor any of its configuration options. You must manually add the section to the Stat Server `Application` object and provide an arbitrary section name. Use this section to pass initialization parameters to the Java Extension.	
<filename>.jar	R The name of this Java configuration option is the relative path of the Java Extension jar archive with respect to the SSJE installation directory described with `[java-config]/java-extensions-dir`. The resulting combined path should point inside the SSJE installation directory (note that on UNIX systems, all symbolic links are resolved). Otherwise, Stat Server logs a security violation message and does not load the corresponding SSJE. Furthermore, if Stat Server cannot match the resulting path to any existing Java Extension configured to be loaded, Stat Server ignores the content of this entire section. The corresponding value is either `false` (indicating that Stat Server is not to consider this particular Java Extension) or `true` (indicating that it is). The path is relative to that specified by the `java-extensions-dir` configuration option described on page 58—for example, `ext1.jar` or `subdir3/ext3.jar`. If you initially do not set this option when Stat Server first starts, but later set it, Stat Server attempts to dynamically load the extension at runtime. Refer to How to Configure a Particular Java Extension below for additional information. Default Value: No default value Valid Values: `false, true` Changes Take Effect: During Java Extension initialization phase
<Name>	R <Value> You can specify additional configuration options following the `Name/Value` format used in other Stat Server sections, where `Name` is name of the parameter to be passed to SSJE and `Value` is the parameter's value. If you do specify a value for a parameter in this section, Stat Server converts the `Name/Value` pair to `Name=Value` before passing it to SSJE. If you do not specify a value, Stat Server passes only the name. Default Value: No default value Changes Take Effect: During Java Extension initialization phase

How to Configure a Particular Java Extension

When Stat Server loads SSJE, Stat Server passes a set of parameters during the initialization phase. To specify those parameters in Stat Server, follow these steps:

1. Create a new configuration section, with an arbitrary name, on the Stat Server `Options` tab in Configuration Server.
2. Within this section, create the `java-extension-jar` option and, as its value, specify the relative path of the corresponding SSJE jar archive with respect to the SSJE installation directory; for example, `MySSJE.jar`.
3. Add any other options to this section. Stat Server passes the corresponding `name:value` pairs to SSJE during the initialization phase.

How to Configure a Particular Java Extension Stat Type

Some Stat Server clients (such as CC Analyzer) require an explicit statistical type (stat type) configuration in Configuration Server. Java stat types are configured slightly differently than regular stat types. To configure a particular stat type defined in a Java Extension:

1. Create a new section, with an arbitrary name, on the Stat Server `Options` tab in Configuration Server.
2. Within the newly created section, create these new mandatory options:
    - `Category`
    - `Objects`
    - `JavaSubCategory`

    The first two are standard for all stat types. (Refer to the “Statistical Type Sections” section in the *Framework 8.5 Stat Server User’s Guide* for a description of these and other options.)

    The value of the third option must have the format *`extension-jar-path:stat-type-name,`* where:

    - *`extension-jar-path`* is the relative path of the Java Extension jar archive with respect to the SSJE installation directory described by `[jvm-options]/java-extensions-dir`.
    - *`stat-type-name`* is the name of the stat type residing in SSJE.
3. Add any other options to the newly created section. Stat Server will pass the corresponding `name:value` pairs to SSJE whenever the statistics associated with this corresponding stat type are requested.

db-direct-connection Section

Note: This section is only applicable to Stat Server operating in regular mode.

Stat Server 8.5 is capable of working with the DBMS through:

- A direct database connection, without using DB Server. Make sure to install the corresponding DBMS client software on the Stat Server host for direct database connection.
- An indirect database connection, using DB Server (backward compatibility mode)

By default, Stat Server connects through a configured DB Server. To enable the direct database connection, create the `[db-direct-connection]` section.

Note: This section is not applicable to Stat Server operating in cluster mode.

Table 9: Options in the [db-direct-connection] Section

Option	Description
debug	This option controls advanced debugging information such as function calls. Default Value: `0` Valid Values: `0-5` Changes Take Effect: When Stat Server is restarted.
enable	This option enables direct database connection to the DBMS. Default Value: `off` Valid Values: `on, off` Changes Take Effect: When Stat Server is restarted.
verbose	This option controls the SQL statement and message sending debug information. Default Value: `0` Valid Values: `0-4` Changes Take Effect: When Stat Server is restarted.

See also options listed in Table 5 on page 49 to configure a Stat Server application to write data to a database.

Starting with the 8.5 release Stat Server installation package includes the following additional executables:

On Unix

- dbclient_db2_32
- dbclient_db2_64
- dbclient_oracle_32
- dbclient_oracle_64

On Windows

- dbclient_db2.exe
- dbclient_oracle.exe
- dbclient_msql.exe

These executables are located in the same directory as your Stat Server executable.

The appropriate executable is used to establish the connection to a particular database, using DB Info provided in the connected Database Access Point.

See Framework Database Connectivity Reference Guide for more information.

Chapter

6 Other Factors Affecting Stat Server

Stat Server receives events from the Genesys applications that are configured in Stat Server’s application connections and processes them within the confines of Stat Server’s configuration. In addition, Stat Server directly reads general information about the switch underlying these applications. Stat Server uses this information, in part, to determine which action(s) to generate and report to its clients. Though Stat Server does not read the values of the configuration options of such applications, Stat Server does consider certain attributes about these applications (such as their type and version) in its handling of events that originate from these applications.

Note: For the purpose of this chapter, DNs and switches are not considered to be applications. However, Stat Server does read the configuration options of these objects to provide certain functionality.

This chapter describes factors other than Stat Server’s own configuration that have an impact on Stat Server output. It contains the following sections:

For information about manipulating Stat Server behavior via the configuration of the Stat Server `Application` object, refer to the previous chapter, “Fine-Tuning the Configuration of a Stat Server Application” on page 29.

Stat Server Reads Switch and DN Attributes

To Determine Capacity and Impact Routing of Interactions to Multimedia DNs

In support of reporting for multimedia DNs, whenever Stat Server in regular mode of operation (SS_r) detects a multimedia DN, Stat Server now reads the DN's attributes, and those of its switch, to determine whether the DN is capable of handling multiple, simultaneous interactions of differing media types. Stat Server in cluster mode does not support reporting of multimedia DNs.

SS_r looks for the following:

- A DN switch type of either of the following:
  - VoIPSMCPSwitch (Voice over IP SMCP Switch in Genesys Administrator.
  - SIPSwitch (SIP Switch in Genesys Administrator)
- A DN type of CFGExtension (Extension in Genesys Administrator)
- A value of yes in the [TServer]/multimedia configuration option for the DN. (This option is defined on the Options tab of the DN object in Genesys Administrator.)
- Version 7.6.*x* or greater of T-Server, if the switch type is SIPSwitch.

Stat Server uses the switch's media attributes as the default for all Extension DNs that belong to it.

If these criteria are met, Stat Server supports routing of interactions with chat or voice media types to multimedia DNs. (For more information on this subject, refer to "Capacity Planning for Multimedia DNs" in the *Genesys Resource Capacity Planning Guide*.) Prior to release 7.6, Stat Server supported routing of voice interactions only to such DN types.

To Suppress the Transmission of Attached Data

For the switches and DNs that Stat Server monitors, Stat Server checks the [statserver] section of the Options tab for the value of the suppress-user-data configuration option. The value of this option determines whether Stat Server should transmit call-extracted attached data to Stat Server clients for the particular DN on which the option was set or for all DNs registered on a switch. Setting this option is useful for reducing network traffic in environments where many Stat Server applications are connected to a single T-Server, for example, and where each Stat Server application in such a scenario serves a different business purpose.

A value of no (the default value) indicates that Stat Server will continue to receive attached data (and transmit attached data to its clients). If the option value is set to yes, however, T-Server will not send any EventAttachedData

TEvents or `AttributeUserData` attributes of any other TEvent to Stat Server; and, as a result, Stat Server will not transmit userdata, for the associated DN or switch object, to its clients.

If this option is defined for a particular DN, its value overrides any value that may be specified at the switch. Dynamic changes to this option take effect upon DN re-registration.

Note: The selective suppression of attached data is possible only with T-Server release 7.6 and later.

For Processing Stuck Calls and ACW Notifications

In addition, Stat Server regularly references a switch's type and a DN's type to perform many other operations, such as checking for stuck calls or processing ACW notifications.

Stat Server Reads Resource Attributes

To Determine Which Objects Are Enabled

To calculate group- and queue-related statistics, Stat Server considers whether member `Person` objects and, for Stat Server operating in regular mode, `Place` objects have been enabled in Configuration Server, depending on the values of the `ignore-disabled-objects-in-group-statistics` and `ignore-disabled-objects-in-queue-statistics` Stat Server configuration options (described on page 39). This property of contact center resources is but one attribute that Stat Server directly reads about configuration objects.

To Determine if Origination DNs Are Configured

Stat Server, also reads the properties of `GroupAgents` objects and, for Stat Server operating in regular mode, `GroupPlaces` objects as well, to determine if origination DNs have been configured therein (in the `Options` tab of the object's properties in Genesys Administrator/GAX). If configured, Stat Server reflects the events occurring at these origination DNs for agent group and place group statistics computations—Stat Server also generates retrospective, interaction-related actions reflecting regular DNs onto these origination DNs.

Stat Server Reads Virtual Agent Group Definitions

To Determine Group Membership

For agent group objects, Stat Server also reads the `script` configuration option (located in the `virtual` section of the `Options` tab) to determine the objects to which actions apply. Refer to the "Virtual Agent Groups" chapter of the *Framework 8.5 Stat Server User's Guide* for more information about how to define this object.

Stat Server Reads Mediation DN Attributes

To Determine Average Handling Time

When it is calculating statistics for URS so that it can balance call loads over several mediation DNs, Stat Server reviews each mediation DN's setting of average handling time, which is configured through use of the `load-balance-aht` option in the `[statserver]` section on the `Options` tab of mediation DN objects. Values specified at the mediation DN–level supersede the global value, which is controlled and set within the Stat Server `Application` object, and the same range of values apply. See the description of the `load-balance-aht` Stat Server configuration option on page 40.

Dynamic changes to this option, at the mediation DN level, take effect immediately upon notification of mediation DN re-registration.

To Calculate Action Durations

Stat Server reads the `use-alt-enter-time` configuration option in the `[statserver]` section on the `Options` tab of virtual queue DN objects to determine whether Stat Server should use an alternative enter time when calculating the durations of some actions in some scenarios that involve virtual queues. The value of this dynamic option overrides the value of the `vq-use-alt-enter-time` global option that is set in the Stat Server application. Refer to the description of the global option on page 47 for further details.

To Calculate Estimated Wait Time

As part of Stat Server's calculation of the estimated wait time in which agent capacities are greater than one, Stat Server reads the `media-type` configuration option in the `[statserver]` section of the `Options` tab of virtual queue objects to determine the media type of interactions that the virtual queue has been configured to handle. This dynamic option is set only for a virtual queue object; there exists no global option that defines the media type for all virtual queues. Its permissible values are those that have been preconfigured within Configuration Server—in the `Business Attributes/MediaType` folder in Genesys Administrator. Only one media type should be configured for any given virtual queue and this value cannot be voice.

Refer to the `ExpectedWaitTime2` statistical category in the *Framework Stat Server User's Guide* to learn how Stat Server calculates estimated wait time for multimedia interactions. The `ExpectedWaitTime2` statistical category is not supported for Stat Server operating in cluster mode.

Stat Server Reads SIP Server

While running in cluster mode, Stat Server reads information about agent logins and DNs from SIP Server. Refer to SIP Cluster documentation for further information.

Chapter

7 Common Log Options

This chapter describes log configuration options that are common to all Genesys server applications and applicable to any Framework server component. As presented in this chapter, default values of these options reflect, first and foremost, those default values that are set by the Stat Server application template. Where the Stat Server template omits an option, this chapter, otherwise, provides the inherent default value. The default values listed in this chapter might differ from what is documented *Framework Configuration Options Reference Manual.* Refer to that *Reference Manual* and to the *Genesys Security Deployment Guide* for additional information.

This chapter includes the following sections:

For your convenience, the SIP Server product provides a troubleshooting tool that parses the log output of several Genesys servers including Stat Server. Refer to the *SipSpan2 User's Guide* for information on how to use this tool.

Mandatory Options

You do not have to configure any common log options to start Stat Server applications.

log Section

You must name this section log. Table 10 lists the log configuration options available to you. Note that to use these options, you must actively set them, manually on the Options tab of the Stat Server Application object within Genesys Administrator. The default Stat Server application template includes only the verbose option. These options are generic options that apply to all Genesys server applications. They do not, however, apply to the Stat Server solution.

Table 10: Log Options

Option	Description
verbose	Determines whether a log output is created. If it is, this option specifies the minimum level of log events generated. The log events levels, starting with the highest-priority level, are standard, interaction, trace, and debug. Refer to “Log Output Options” on page 81 for more information. Default Value: all Valid Values: all — All log events (that is, log events of standard, trace, interaction, and debug levels) are generated if you set the debug-level option in the statserver section to all. debug — The same as all. trace — Log events of trace and higher levels (that is, log events of standard and interaction levels) are generated, while log events of the debug level are not. interaction — Log events of the interaction and higher levels (that is, log events of standard level) are generated, while log events of the trace and debug levels are not generated. standard — Log events of the standard level are generated, while log events of the interaction, trace, and debug levels are not generated. none — Produces no output. Changes Take Effect: Immediately Refer to the *Framework Management Layer User’s Guide* for more information on the standard, trace, interaction, and debug log levels.

Table 10: Log Options (Continued)

Option	Description
buffering	Turns operating system file buffering on or off. This option applies only to `stderr` and `stdout` output (see page 81). Setting this option to `true` increases output performance. **Note:** When you enable buffering, messages might appear at the console with delay. Default Value: `true` Valid Values: `true` Enables buffering `false` Disables buffering. Changes Take Effect: Immediately
segment	Specifies whether there is a segmentation limit for a log file. If there is, this option sets the mode of measurement along with the maximum size. If the current log segment exceeds the size set by this option, the current file is closed and a new file is created. This option is ignored if log output is not configured to be sent to a log. Default Value: `100` MB Valid Values: `false` No segmentation allowed. `<number>` KB or `<number>` Sets the maximum segment size in kilobytes. The minimum value is `100 KB`. `<number> MB` Sets the maximum segment size in megabytes. The maximum value is `2047 MB`. `<number> hr` Sets the number of hours for the segment to stay open. The minimum number is `1 hr`. Changes Take Effect: Immediately

Table 10: Log Options (Continued)

Option	Description
keep-startup-file	Specifies whether a startup segment of the log, containing the initial configuration, is to be kept. If it is, you can set this option to `true` or to a specific file size. A `true` setting means that the size of the initial segment will be equal to the size of the regular log segment defined by the `segment` option (defined on page 73). The value of this option is ignored if you set the `segment` option to `false.` Default Value: `false` Valid Values: `false` No startup segment of the log is kept. `true` A startup segment of the log is kept. The size of the segment equals the value of the `segment` option. `<number> KB` Sets the maximum size, in kilobytes, for a startup segment of the log. `<number> MB` Sets the maximum size, in megabytes, for a startup segment of the log. Changes Take Effect: After restart

expire	Determines whether log files expire. If they do, this option sets the measurement for determining when they expire, along with the maximum number of files (segments) or days before the files are removed. Stat Server ignores this option if you configure log output to be sent to other than a log file.
	Default Value: `10`
	Valid Values:
	`false` No expiration; all generated segments are stored.
	`<number> file` or `<number>` Sets maximum number of log files to store. Specify a number from `1-1000.`
	`<number> day` Sets the maximum number of days before log files are deleted. Specify a number from `1-100.`
	Changes Take Effect: Immediately
	Note: If an option's value is set incorrectly to out of the range of valid values, it is automatically reset to `10`.

Table 10: Log Options (Continued)

Option	Description
messagefile	Specifies the file name for Stat Server log events. The name must be valid for the operating system on which Stat Server is running. The option value can also contain the absolute path to the `statserver.lms` file. Otherwise, Stat Server looks for the file in its working directory. Default Value: `statserver.lms` Valid Values: `<string>.lms` Changes Take Effect: After Stat Server restarts if Stat Server locates `statserv.lms` at startup or immediately if Stat Server cannot locate this file at startup **Warning!** An application that does not find its *.lms file at startup cannot generate application-specific log events and send them to Message Server.
message_format	Specifies the format of log record headers that an application uses when it writes to its log file. Using compressed log record headers improves the application performance and reduces the log's file size. Default Value: `short` Valid Values: `short` An application uses compressed headers when writing log records to its log file. `full` An application uses complete headers when writing log records to its log file.

message_format (continued)	Changes Take Effect: Immediately
	With the value set to `short`:
	• A header of the log file or the log file segment contains information about the application (such as the application name, application type, host type, and time zone) while single log records within the file or segment omit this information.
	• A log message priority is abbreviated to `Std`, `Int`, `Trc`, or `Dbg`, for `standard`, `interaction`, `trace`, or `debug` messages respectively.
	• `Message ID` does not contain the prefix GCTI or the application type ID.
	A log record in the `short` format looks like this:
	`2007-05-07T18:15:33.952 Std 05060 Application started`
	A log record in the `full` format looks like this:
	`2007-05-07T18:11:38.196 Standard localhost cfg_dbserver GCTI-00-05060 Application started`
	Note: Whether the `full` or `short` format is used, time is printed as specified by the `time_format` option.

Table 10: Log Options (Continued)

Option	Description
time_convert	Specifies the system by which an application calculates the log record time when generating a log file. The time is converted from the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970). Default Value: `Local` Valid Values: `local` Time of log-record generation expressed as a local time, based on the time zone and any seasonal adjustments. Time zone information of the Stat Server host computer is used. `utc` Time of log-record generation expressed as Coordinated Universal Time (UTC). Changes Take Effect: Immediately
time_format	Specifies how to represent, in a log file, the time when an application generates log records. Default Value: `time` Valid Values: `time` Time string is formatted according to the HH:MM:SS.sss (hours, minutes, seconds, and milliseconds) format. `locale` Time string is formatted according to the system's locale. `ISO8601` Date in the time string is formatted according to ISO 8601 format. Fractional seconds are given in milliseconds. Changes Take Effect: Immediately A log record's time field in ISO 8601 format looks like this: `2007-07-24T04:58:10.123`
print-attributes	Specifies whether the application will attach extended attributes, if any exist, to a log event that the application sends to log output. Typically, log events of the `Interaction` log level and `Audit`-related log events contain extended attributes. Setting this option to `true` enables audit capabilities, but negatively affects performance. Genesys recommends enabling this option only when testing new interaction scenarios. Default Value: `false` Valid Values: `true, false` Changes Take Effect: Immediately Refer to the *Genesys Combined Log Events Help* for information about extended attributes.

Table 10: Log Options (Continued)

Option	Description
check-point	Specifies how often, in hours, an application generates a `check-point` log event to divide the log into sections of equal time. By default, the application generates this log event every hour. Setting the option to `0` prevents generation of `check-point` events. Default Value: `1` Valid Values: `0-24` Changes Take Effect: Immediately
memory	Specifies the name of the file to which the application regularly prints a snapshot of the memory output (see page 81). The new snapshot overwrites previously written data. If the application terminates abnormally, this file contains the latest log messages. Memory output is not recommended for processors with a CPU frequency lower than 600 MHz. **Note:** If the file specified as the `memory` file is located on a network drive, an application does not create a snapshot file (with the extension `*.memory.log`). Default Value: No default value Valid Values: `<string>` (memory file name) Changes Take Effect: Immediately
memory-storage-size	Specifies the buffer size for log output to the memory. Refer also to “Log Output Options” on page 81 for more information. Default Value: `2 MB` Valid Values: `<number> KB` or `<number>` — Size of the memory output, in kilobytes. The minimum value is `128 KB`. `<number> MB` — Size of the memory output, in megabytes. The maximum value is `64 MB`. Changes Take Effect: When memory output is created

Table 10: Log Options (Continued)

Option	Description
spool	Specifies the folder, including full path to it, in which an application creates temporary files related to network log output. If you change this value while the application is running, the change does not affect the currently opened network output. Default Value: The Stat Server working directory Valid Values: `<path>` The full path of the folder Changes Take Effect: Immediately
compatible-output-priority	Specifies whether the application uses 6.*x* output logic. Default Value: `false` Valid Values: `true` The log of the level specified by one of the log output options described on page 82 is sent to the specified output. `false` The log of the level specified by one of the log output options described on page 82 and higher levels is sent to the specified output.

compatible-output-priority (continued)	Changes Take Effect: Immediately
	For example, you configure the following options in the `log` section for a 6.*x* application and for a 8.*x* application:
	`[log]`
	`verbose=all`
	`debug=file1`
	`standard=file2`
	Stat Server 6.*x* log file content is as follows:
	• `file1` contains `debug` messages only.
	• `file2` contains `standard` messages only.
	Stat Server 8.*x* log file content is as follows:
	• `file1` contains `debug`, `trace`, `interaction`, and `standard` messages.
	• `file2` contains `standard` messages only.
	Warning! Genesys does not recommend changing the default value of the `compatible-output-priority` option unless you have specific reasons to use the 6.*x* log output logic—that is, to mimic the output priority as implemented in releases 6.*x*. Setting this option to `true` affects log consistency.

The configuration options listed in Table 11 enable you to generate debug logs containing information about specific Stat Server operations. You designate these options in the `log` section of the Stat Server application.

Warning! Genesys advises you to use these options only when requested by Genesys Customer Care.

Table 11: Debug Log Options

Option	Description
x-conn-debug-open	Generates debug log records about "open connection" operations. Default Value: `0` Valid Values: `0` Log records are not generated. 1 Log records are generated. Changes Take Effect: After restart
x-conn-debug-select	Generates debug log records about "socket select" operations. Default Value: `0` Valid Values: `0` Log records are not generated. 1 Log records are generated. Changes Take Effect: After restart
x-conn-debug-timers	Generates debug log records about the timer creation and deletion operations. Default Value: `0` Valid Values: `0` Log records are not generated. 1 Log records are generated. Changes Take Effect: After restart
x-conn-debug-write	Generates debug log records about "write" operations to the socket within the common library. Default Value: `0` Valid Values: `0` Log records are not generated. 1 Log records are generated. Changes Take Effect: After restart

Table 11: Debug Log Options (Continued)

Option	Description
x-conn-debug-security	Generates debug log records about security-related operations, such as Transport Layer Security (TLS) and security certificates. **Note:** This option has no effect on Stat Server 7.6 and earlier releases, which do not support TLS operations. Default Value: `0` Valid Values: `0` Log records are not generated. `1` Log records are generated. Changes Take Effect: After restart
x-conn-debug-api	Generates debug log records about connection library function calls. Default Value: `0` Valid Values: `0` Log records are not generated. `1` Log records are generated. Changes Take Effect: After restart
x-conn-debug-dns	Generates debug log records about DNS operations. Default Value: `0` Valid Values: `0` Log records are not generated. `1` Log records are generated. Changes Take Effect: After restart
x-conn-debug-all	Generates debug log records about open connection, socket select, timer creation and deletion, write, security-related, DNS operation, and connection library function calls. This option is the same as enabling or disabling all of the previous `x-conn-debug-<optype>` options. Default Value: `0` Valid Values: `0` Log records are not generated. `1` Log records are generated. Changes Take Effect: After restart

Log Output Options

To configure log outputs, set log level options (all, alarm, standard, interaction, trace, memory, and/or debug) to the desired types of log output (stdout, stderr, network, memory, and/or [filename] for log file output).

You can use:

- One log level option to specify different log outputs.
- One log output type for different log levels.
- Several log output types simultaneously for logging the events of the same or different log levels.

You must separate the log output types by a comma when you are configuring more than one output for the same log level. See "Examples" on page 86.

Note: The log output options are activated according to the setting of the verbose configuration option.

Warnings!

- If you direct log output to a file on the network drive, an application does not create a snapshot log file (with the extension *.snapshot.log) in case it terminates abnormally.
- Directing log output to the console (by using the stdout or stderr settings) can affect application performance. Avoid using these log output settings in a production environment.

Table 12: Log Output Options

Option	Description
all	Specifies the outputs to which an application sends all log events. You must separate log output types with commas when you configure more than one output type. Default Value: No default value Valid Values: `stdout` Log events are sent to the standard output (`stdout`). `stderr` Log events are sent to the standard error output (`stderr`). `network` Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores log events in the Log Database. Setting the `all` log level option to `network` enables an application to send log events of `standard, interaction,` and `trace` levels to Message Server. Log events of `debug` level are neither sent to Message Server nor stored in the Log Database. `memory` Log events are sent to the memory output on the local disk. This output is the safest in terms of Stat Server performance. `[filename]` Log events are stored in a file with the specified name. If you do not specify a path, the log file is created in the working directory. Changes Take Effect: Immediately For example, `all = stdout, logfile` **Note:** To ease the troubleshooting process, consider using unique prefixes for log files that different Stat Server applications generate.

Table 12: Log Output Options (Continued)

Option	Description
alarm	Specifies the outputs to which an application sends log events of `Alarm` level. You must separate log output types with commas when you configure more than one output type. For example, `alarm = stdout, logfile` Default Value: No default value Valid Values (log output types): `stdout` Log events are sent to the standard output (`stdout`). `stderr` Log events are sent to the standard error output (`stderr`). `network` Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores log events in the Log Database. `memory` Log events are sent to the memory output on the local disk. This output is the safest in terms of the application performance. `[filename]` Log events are stored in a file with the specified name. If you do not specify a path, the log file is created in the working directory. Changes Take Effect: Immediately
standard	Specifies the outputs to which an application sends log events of the `Standard` level. You must separate log output types with commas when you configure more than one output type. For example, `standard = stderr, network` Default Value: No default value Valid Values (log output types): `stdout` Log events are sent to the standard output (`stdout`). `stderr` Log events are sent to the standard error output (`stderr`). `network` Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. `memory` Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. `[filename]` Log events are stored in a file with the specified name. If you do not specify a path, the log file is created in the working directory. Changes Take Effect: Immediately

Table 12: Log Output Options (Continued)

Option	Description
interaction	Specifies the outputs to which an application sends log events of the `Interaction` and higher levels (that is, log events of `Standard` level). You must separate log outputs with commas when you configure more than one output type. For example, `interaction = stderr, network` Default Value: No default value Valid Values (log output types): `stdout` Log events are sent to the standard output (`stdout`). `stderr` Log events are sent to the standard error output (`stderr`). `network` Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. `memory` Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. `[filename]` Log events are stored in a file with the specified name. If you do not specify a path, the log file is created in the working directory. Changes Take Effect: Immediately
trace	Specifies the outputs to which an application sends log events of `Trace` and higher levels (that is, log events of `Standard` and `Interaction` levels). You must separate log outputs with commas when you configure more than one output type. For example, `trace = stderr, network` Default Value: No default value Valid Values (log output types): `stdout` Log events are sent to the standard output (`stdout`). `stderr` Log events are sent to the standard error output (`stderr`). `network` Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. `memory` Log events are sent to the memory output on the local disk. This output is the safest in terms of the application performance. `[filename]` Log events are stored in a file with the specified name. If you do not specify a path, the log file is created in the working directory. Changes Take Effect: Immediately

Table 12: Log Output Options (Continued)

Option	Description
debug	Specifies the outputs to which an application sends log events of debug and higher levels (that is, log events of standard, trace, interaction, and debug levels). You must separate log output types with commas when you configure more than one output type. For example, debug = stderr, /usr/local/genesys/logfile Default Value: No default value Valid Values (log output types): stdout — Log events are sent to the standard output (stdout). stderr — Log events are sent to the standard error output (stderr). memory — Log events are sent to the memory output on the local disk. This output is the safest in terms of the application performance. [filename] — Log events are stored in a file with the specified name. If you do not specify a path, the log file is created in the working directory. Changes Take Effect: Immediately **Note:** Log events of debug level are never sent to Message Server nor are they stored in the Log Database.

Log File Extensions

You can use the following file extensions to identify log files that Stat Server creates for various types of output:

- `*.log`—Assigned to log files when you configure output to a log file. For example, if you set `standard = statservlog`, Stat Server prints log messages into a text file called `statservlog.<time_stamp>.log`.
- `*.qsp`—Assigned to temporary (spool) files when you configure output to the network, but the network is temporarily unavailable. For example, if you set `standard = network`, Stat Server prints log messages into a file called `statserv.<time_stamp>.qsp` during the time the network is unavailable.
- `*.snapshot.log`—Assigned to files containing the output snapshot when you configure output to a log file. The file contains the last log messages that Stat Server generates before abnormal termination. For example, if you set `standard = statservlog`, Stat Server prints the last log message into a file called `statserv.<time_stamp>.snapshot.log` in case of failure. If Stat Server terminates normally, the snapshots logs are deleted.

Note: Provide `*.snapshot.log` files to Genesys Customer Care when reporting a problem.

- `*.memory.log`—Assigned to log files that contain the memory output snapshot when you configure output to memory and redirect the most recent memory output to a file. For example, if you set `standard = memory` and `memory = statserv`, Stat Server prints the latest memory output to a file called `statserv.<time_stamp>.memory.log`.

Examples

This section presents three examples of a `log` section that you might configure for a Stat Server application that is operating:

- In production mode
- In a debugging lab mode
- In a troubleshooting lab mode

Example 1: Production Mode [log] Section

```
[log]
verbose=standard
standard=network,statservlogfile
```

With this configuration, Stat Server generates only log events of the `Standard` level and sends them to the standard output, to Message Server, and to a file named `statservlogfile`, which Stat Server creates in its working directory. Genesys recommends that you use this or a similar configuration in a production environment.

Warning! Directing log output to the console (by using the `stdout` or `stderr` settings) can affect application performance. Avoid using these log output settings in a production environment.

Example 2: Lab Mode [log] Section

```
[log]
verbose=all
all=stdout,/usr/local/genesys/statservlogfile
trace=network
```

With this configuration, Stat Server generates log events of the `standard`, `interaction`, `trace`, and `debug` levels and sends them to the standard output and to a file named `statservlogfile`, which Stat Server creates in the `/usr/local/genesys/` directory. In addition, Stat Server sends log events of the `standard`, `interaction`, and `trace` levels to Message Server. Use this configuration to test new interaction scenarios in a laboratory environment. Be sure to appropriately set the `debug-level` option in the `statserver` section.

Example 3: Failure-Troubleshooting [log] Section

```
[log]
verbose=all
standard=network
all=memory
memory=statservlogfile
memory-storage-size=32 MB
```

With this configuration, Stat Server generates log events of the standard level and sends them to Message Server. It also generates log events of all levels and sends them to the memory output. The most current log is stored to a file named statservlogfile, which the application creates in its working directory. An increased memory storage enables Stat Server to save more log information generated before a failure. Use this configuration when trying to reproduce an application failure. The memory log file would contain the snapshot of Stat Server's log at the moment of failure. This should help you and Genesys Customer Care identify the reason for the failure. Be sure to appropriately set the debug-level option in the statserver section.

Note: If you are operating Stat Server on Unix and do not specify any files in which to store the memory output snapshot, the core file that Stat Server produces before terminating contains the most current Stat Server log. Provide the Stat Server's core file to Genesys Customer Care when reporting problems.

log-extended Section

This section must be named log-extended.

Table 13: Extended Log Options

Option	Description
level-reassign-disable	When this option is set to true, the original (default) log level of all log events in the [log-extended] section are restored. This option is useful when you want to use the default levels and keep the customizations. Default Value: false Valid Values: true, false Changes Take Effect: Immediately Defined: Options tab of Application object

Table 13: Extended Log Options (Continued)

Option	Description
level-reassign-<eventID>	Specifies one of five log levels for log event *`<eventID>`*, which may differ from its default level, or disables logging of the named event altogether. This option is useful if you want to change the behavior of what Stat Server logs for the specified log event ID. If no value is specified, then the named log event retains its default level. You can deactivate these options with the `level-reassign-disable` configuration option, described below. Default Value: Default value of log event *`<eventID>`*. Refer to the *Common Log Events Help* or `statserver.lms` (located in the directory where Stat Server is installed) for a listing of each of Stat Server's the default levels. Valid Values: `alarm` The log level of log event *`<eventID>`* is set to `alarm`. `standard` The log level of log event *`<eventID>`* is set to `standard`. `interaction` The log level of log event *`<eventID>`* is set to `interaction`. trace The log level of log event *`<eventID>`* is set to `trace`. debug The log level of log event *`<eventID>`* is set to `debug`. none Log event *`<eventID>`* is not recorded in a log. Changes Take Effect: Immediately

Warning! Use caution when making these changes in a production environment.

Depending on the log configuration, changing the log level to a higher priority might cause the log event to be logged more often or to a greater number of outputs. This could affect system performance.

Likewise, changing the log level to a lower priority may cause the log event to be not logged at all, or not logged to specific outputs, thereby losing important information. The same applies to any alarms associated with that log event.

In addition to the precautionary message above, take note of the following:

- Logs can be customized only by release 7.6 or later applications.
- When the log level of a log event is changed to any level except none, it is subject to the other settings in the `[log]` section at its new level. If set to none, it is not logged and therefore not subject to any log configuration.

- Changing the log level of a log using this feature changes only its priority; it does not change how that log is treated by the system. For example, increasing the priority of a log to `Alarm` level does not mean that an alarm will be associated with it.
- Each application in a high availability (HA) pair can define its own unique set of log customizations, but the two sets are not synchronized with each other. This can result in different log behavior depending on which application is currently in primary mode.
- This feature is not the same as a similar feature in Universal Routing Server, version 7.2 or later. In this Framework feature, the priority of log events are customized. In the URS feature, the priority of debug messages only are customized. Refer to the *Universal Routing Server Reference Manual* for more information about the URS feature.
- You cannot customize any log event that is not in the unified log record format. Log events of the `Alarm`, `Standard`, `Interaction`, and `Trace` levels feature the same unified log record format.

Example

This is an example of using customized log level settings, subject to the following log configuration:

```
[log]
verbose=interaction
all=stderr
interaction=log_file
standard=network
```

Before the log levels of the log are changed:

- Log event `20009`—with default level `trace`—is output to `stderr`.
- Log event `20018`—with default level `standard`—is output to `stderr` and the log file, and sent to Message Server.
- Log event `20022`—with default level `debug`—is output to `stderr`.

Extended log configuration section:

```
[log-extended]
level-reassign-20009=none
level-reassign-20018=interaction
level-reassign-20022=standard
```

After the log levels are changed:

- Log event `20009` is disabled and is not logged.
- Log event `20018` is output to `stderr` and to the log file.
- Log event `20022` is output to `stderr` and to the log file, and sent to Message Server.

log-filter Section

This section must be called `log-filter`. Table 14 describes the one option you can configure in this section. Refer to the “Hide Selected Data in Logs” chapter in the *Genesys Security Deployment Guide* for more information about this feature.

Table 14: log-filter Options

Option	Description
default-filter-type	Specifies the default manner in which KVList information (including `UserData`, `Extensions`, and `Reasons`) is presented in the Stat Server log. Stat Server applies the value of this option to all KVList pairs, but the presentation of specific pairs in `AttributeUserData` can be overridden by options that are explicitly defined within the `[log-filter-data]` section (see page 92). Default Value: `skip` Valid Values: `copy` The keys and values of KVList information are copied to the log. `hide` The keys of the KVList information are copied to the log; the values are replaced with strings of asterisks. `hide-first,<n>` The keys of KVList information are copied to the log; the first `<n>` characters of the value are replaced with asterisks. If `<n>` exceeds the number of characters in the value, the number of asterisks will be equal to the number of characters in the value. `hide-last,<n>` The keys of KVList information are copied to the log; the last `<n>` characters of the value are replaced with asterisks. If `<n>` exceeds the number of characters in the value, the number of asterisks will be equal to the number of characters in the value. `tag[(<tag-prefix>,<tag-postfix>)]` KVList information is tagged with the prefix specified by `<tag-prefix>` and the postfix specified by `<tag-postfix>`. If the two parameters are not specified, the default tags `<# and #>` are used as prefix and postfix, respectively. To use the default tags, you can use any of the following values: • `tag` • `tag()` • `tag(,)` To define your own tags, replace the two parameters in the value with your tags.

Table 14: log-filter Options (Continued)

Option	Description
default-filter-type (continued)	`unhide-first,<n>` The keys of KVList information are copied to the log; all but the first `<n>` characters of the value are replaced with asterisks. If `<n>` exceeds the number of characters in the value, the value of the key appears, with no asterisks. `unhide-last,<n>` The keys of KVList information are copied to the log; all but the last `<n>` characters of the value are replaced with asterisks. If `<n>` exceeds the number of characters in the key, the value of the key appears, with no asterisks. `skip` KVList information is not copied to the log. Changes Take Effect: Immediately
filtering	Enables (`true`) or disables (`false`) log filtering at the Application level. Default Value: `true` Valid Values: `true`, `false` Changes Take Effect: Immediately, if application is subscribed to notifications that this option has been changed.

Example

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

Here is an example of a log with the default log filter settings:

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

Refer to the *Genesys Security Deployment Guide* for additional examples.

log-filter-data Section

This section must be called `log-filter-data`.The options in this section define the treatment of filtering data in log output on a key-by-key basis. Log files can contain a significant amount of information about your configuration and operations. The options in this section enable you to prevent unauthorized users from seeing particular data in the output of log messages. Table 15 describes the options you configure in this section.

Table 15: log-filter-data Options

<table>
<tr><th>Option</th><th colspan="2">Description</th></tr>
<tr><td><key name></td><td colspan="2">Specifies the manner in which the specified KVList pair, defined by this option’s name, is presented in the Stat Server log. Setting this option supersedes the default manner of KVList presentation, which is defined by the value of the `default-filter-type` option in the `[log-filter]` section for the given KVList pair.
If no value is specified for this option, no additional processing of this data element is performed.
Default Value: No default value
Valid Values:</td></tr>
<tr><td></td><td>`copy`</td><td>The key and value of the given KVList pair in the `Attribute UserData` section are copied to the log.</td></tr>
<tr><td></td><td>`hide`</td><td>The given KVList key is copied to the log; the KVList value is replaced with a string of asterisks.</td></tr>
<tr><td></td><td>`hide-first,<n>`</td><td>The key of the given KVList pair in the `AttributeUserData` section is copied to the log; the first `<n>` characters of the value are replaced with asterisks. If `<n>` exceeds the number of characters in the value, the number of asterisks will be equal to the number of characters in the value.</td></tr>
<tr><td></td><td>`hide-last,<n>`</td><td>The key of the given KVList pair in the `AttributeUserData` section is copied to the log; the last `<n>` characters of the value are replaced with asterisks. If `<n>` exceeds the number of characters in the value, the number of asterisks will be equal to the number of characters in the value.</td></tr>
<tr><td></td><td>`tag[(<tag-prefix>,<tag-postfix>)]`</td><td>The KVList pair in the `AttributeUserData` section is tagged with the prefix specified by `<tag-prefix>` and the postfix specified by `<tag-postfix>`. If the two parameters are not specified, the default tags `<# and #>` are used as prefix and postfix, respectively.
To use the default tags, you can use any of the following values:
• `tag`
• `tag()`
• `tag(,)`
To define your own tags, replace the two parameters in the value with your tags.</td></tr>
</table>

Table 15: log-filter-data Options (Continued)

Option	Description	
<key name> (continued)	unhide-first,<n>	The key of the given KVList pair in the AttributeUserData section is copied to the log; all but the first <n> characters of the value are replaced with asterisks. If <n> exceeds the number of characters in the value, the value of the key appears, with no asterisks.
	unhide-last,<n>	The key of the given KVList pair in the AttributeUserData section is copied to the log; all but the last <n> characters of the value are replaced with asterisks. If <n> exceeds the number of characters in the value, the value of the key appears, with no asterisks.
	skip	The KVList pair in the AttributeUserData section is not copied to the log.
	Changes Take Effect: Immediately	

Example

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

Here is an example of the log with the PASSWORD option set to hide. Note that the value of PASSWORD has been replaced with a series of asterisks (****):

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

Refer to the *Genesys Security Deployment Guide* for additional examples.

sml Section

This section must be called sml, which stands for System Management Layer, and must be defined on the Options tab of the Stat Server Application object. Table 16 describes one option that you can configure in this section. Other options that the *Framework Configuration Options Reference Manual* describe are not supported in the 8.*x* releases of Stat Server.

Table 16: SML Options

Option	Description
suspending-wait-timeout	Specifies a timeout, in seconds, after the `Stop Graceful` command is issued to an application within the Management Layer., during which the status of the application should change to `Suspending` if the application supports graceful shutdown. If the status of the application does not change to `Suspending` before the timeout expires, it is assumed that the application does not support graceful shutdown, and it is stopped ungracefully. **Note:** Stat Server does not support graceful shutdown. Default Value: `10` Valid Values: `5–600` Changes Take Effect: Immediately

common Section

This section must be named `common`. Table 17 describes the options that you configure in this section.

Table 17: common Options

Option	Description
enable-ipv6	Specifies that Stat Server is to use TCP/IP v6 for relaying packets of information across network boundaries to and from the Management Layer. Default Value: `0` (for backward compatibility) Valid Values: `0` (off), `1` (`on`) Changes Take Effect: Immediately **Note:** This option is supported for use on Linux, Solaris 8+, Windows Vista, and Windows Server 2008+ operating systems only. Refer to the *Framework Deployment Guide* for more information about this component.
rebind-delay	Specifies the delay, in seconds, between socket-bind operations that are being executed by Stat Server. Use this option if Stat Server has not been able to occupy a configured port successfully. **Warning!** Use this option only when requested to do so by Genesys Customer Care. Default Value: `10` Valid Values: `0–600` Changes Take Effect: After restart

Chapter

8 Installing a Stat Server Application

You must configure a Stat Server Application object in Configuration Server before installing the Stat Server application. Read Chapter 2 for this configuration and other important information. You need not uninstall prior releases of Stat Server in order to install a newer release. This chapter, nonetheless, provides uninstallation procedures, as well as installation procedures, to address the case where you want to permanently remove Stat Server from your machine.

This chapter contains the following topics:

Installing Stat Server Following Manual Configuration

This section describes how to install Stat Server on UNIX and Windows platforms if you manually configured a Stat Server Application object within Genesys Administrator/GAX.

On UNIX

1. On the Real-Time Metrics Engine 8.5 product CD in the appropriate statserver/*operating_system/* directory, locate the install.sh shell script.
2. Run this script from the command line by typing: install.sh.

3. When prompted, specify the host name of the computer on which you want to install Stat Server.
4. When prompted, specify:
   a. The host name of the computer on which Configuration Server is running.
   b. The port that Stat Server will use to connect to Configuration Server.
   c. The user name used to log in to Configuration Server.
   d. The password used to log in to Configuration Server.
5. Specify whether Stat Server should use a client-side port for TCP/IP connection to Configuration Server. If yes, specify the client-side port number and, optionally, either the IP address that Stat Server will use for its connection or `Enter` to ignore.

   Refer to the *Genesys Security Deployment Guide* for more information about client-side port definition and configuration.
6. The installation displays the list of `Application` objects of `StatServer` type configured for this host. Type the number of the Stat Server `Application` you want installed.
7. Specify the full destination path into which you want Stat Server installed.
8. If prompted for which version of the product to install, (32- or 64-bit), select the version appropriate for your operating system.

As soon as the installation process completes, a message announces that installation was successful. The process creates a directory with the name specified during the installation, and places Stat Server in it. The installation routine then prompts you to install each of the Stat Server Java Extensions (MCR and OCC) if the Extension installation packages were also deployed. Follow the steps described for each Extension, starting with Step 2 on page 98.

On Windows

1. From the Real-Time Metrics 8.5 CD, go to the `\statserver\windows` subdirectory.
2. Locate and double-click `setup.exe` to start installation.
3. If the installation routine detects previously installed Stat Server applications on your machine, you are prompted to either install a new instance or perform maintenance on one of the existing applications. Select the former.
4. Specify the parameters for connecting to the Configuration Server where your Stat Server `Application` object has been configured.
5. Specify whether Stat Server should use a client-side port for TCP/IP connection to Configuration Server. If so, specify the client-side port number and, optionally, the IP address that Stat Server will use for its connection.

The installation routine automatically adds these parameters (`transport-port` and `transport-address`) to:

- The `Command–Line Arguments` text box on the `Start Info` tab of the Stat Server `Application Properties` dialog box, so that Stat Server can be started from the Management Layer. (Refer to Chapter 9 for information about command-line parameters.)
- The `startServer` batch file, so that you can start Stat Server using its startup files.

Refer to the *Genesys Security Deployment Guide* for more information about client-side port definition and configuration.

6. Select your Stat Server application.
7. Specify the destination directory into which you want Stat Server installed.
8. Specify a Solution Name if the `Cluster Mode` checkbox was selected.
9. Click `Install` and `Finish` to complete the installation.

The installation routine installs your Stat Server application automatically as a Windows service.

If you run the Stat Server installation package from the *Real-Time Metrics Engine* CD, Stat Server automatically installs the MCR and OCC Stat Server Java Extensions as well.

Manually Installing the Java Extensions

Before installing a Stat Server Java Extension, you must both have configured a Stat Server `Application` object and installed the Stat Server application on your machine. On the *Real-Time Metrics Engine* CD, Genesys provides the installation packages for eServices and OCC Java Extensions, which are delivered in four `.jar` files:

- `eServiceContactStat.jar`
- `eServiceInteractionStat.jar`
- `eServiceSystemStat.jar`
- `OCCStatExtension.jar`

You deploy these files in two separate installations.

Installing the eServices Extensions

You can install the three `eService` Java Extensions, which are used for eServices, on Windows and/or UNIX platforms.

On Windows

1. In the `\ext\mcr\` subdirectory of your deployed Stat Server installation package, locate and double-click `setup.exe.`

2. If the installation routine detects one or more previously installed extension on your machine, you are prompted to either install a new instance or perform maintenance on the existing extension. Select the former.
3. When prompted, specify the root folder of the Stat Server installation (for example, `C:\Program Files\GCTI\Stat Server\StatServer_1`), and click `Next`.

The installation routine deploys the `eServiceContactStat.jar`, `eService InteractionStat.jar`, and `eServiceSystemStat.jar` files in the `\java\ext\` subdirectory of your installed application.

On UNIX

1. On the *Real-Time Metrics Engine* CD, navigate to the `/ext/mcr/` subdirectory.
2. Run the `install.sh` script from the command line by typing:

```
sh install.sh
```

3. When prompted, specify the full destination path where you want the MCR extension deployed on your machine.

   If the installation routine detects one or more installed extensions in the specified path, it prompts you to overwrite them or exit.

The installation routine deploys the `eServiceContactStat.jar`, `eService InteractionStat.jar`, and `eServiceSystemStat.jar` files in the `/java/ext` subdirectory of the path that you specified.

Installing the Outbound Contact Extension

You can install the `OCCStatExtension` Java Extension, which is used for the Outbound Contact solution, on Windows and/or UNIX platforms.

On Windows

1. In the `\ext\occ\` subdirectory of your deployed Stat Server installation package, locate and double-click `setup.exe`.
2. When prompted to specify the destination folder, indicate the root folder of the Stat Server installation (for example, `C:\Program Files\GCTI\Stat Server\StatServer_1`) and click `Next`.

   Note: Select this folder carefully. The default choice provided by the installation routine likely differs from your intended destination.

The installation routine deploys `OCCStatExtension.jar` in the `\java\ext` subdirectory of your installed application.

On UNIX

1. On the *Real-Time Metrics Engine* CD, navigate to the `/ext/occ/` subdirectory.
2. Run the `install.sh` script from the command line by typing:

```
sh install.sh
```

3. When prompted, specify the full destination path where you want the OCC extension deployed on your machine.

   If the installation routine detects one or more installed extensions in the specified path, it prompts you to overwrite them or exit.

The installation routine deploys `OCCStatExtension.jar` in the `/java/ext` subdirectory of the path that you specified.

Installing Stat Server Silently

You can deploy Stat Server silently using InstallShield Silent, a third-party installation program that Genesys provides to facilitate the electronic software distribution for both server and GUI applications on Windows platforms. “Silent” installations eliminate the need for interactive dialog during the installation process. Instead, you create a single response file filled with the necessary parameters that InstallShield Silent references during subsequent silent installations.

For instructions on how to deploy applications silently, refer to the *Framework Deployment Guide.*

Uninstalling the Stat Server Application

To uninstall a Stat Server application, you must first stop it. Refer to “Stopping a Stat Server Application” on page 104 for this information. Uninstalling the Stat Server application differs from uninstalling its `Application` object in Configuration Server From the Control Panel

1. Open `Add/Remove Programs.`
2. Locate and select the desired `Genesys Stat Server` application.
3. Click `Remove.`

Chapter

9 Starting and Stopping a Stat Server Application

This chapter contains procedures for starting and stopping a Stat Server application on the supported platforms. Start procedures assume that you have properly configured and installed Stat Server. If not, refer to Chapters 2 and 8 respectively.

This chapter contains these sections:

As part of the invocation of a Stat Server solution, you can also start several Stat Server applications simultaneously. Refer to the next chapter for this information.

What Must Be Running Prior to Start

You can start a Stat Server application in several ways. Depending on the desired mode of operation, Genesys recommends that you start a Stat Server applications with certain other Genesys applications already running.

For starting Stat Server from Genesys Administrator/Genesys Administrator Extension (GAX) in regular or cluster mode, have the following up and running:

- Configuration Server
- Solution Control Server
- Local Control Agent
- Genesys Administrator/GAX

If you have configured the Stat Server application to write to a database, also have running:

- RDBMS
- DB Server (if `[db-direct-connection]/enable` is set to `no`)

For starting Stat Server in cluster mode, though not mandatory, you should also have the following applications running:

- SIP Server Cluster
- SIP Proxy

And, if your environment uses Stat Server Java extensions, set up Java Runtime Environment (JRE)

Starting a Stat Server Application

You can start a Stat Server application in any of the following ways:

- From Genesys Administrator.
- On UNIX.
- From the Windows command line .
- As a Windows Service.

Note: Prior to opening statistics at startup, Stat Server now checks that the binary format of the backup file is compatible with the running instance of Stat Server.

You can start a Stat Server application that has been configured as a node of a clustered solution in either standalone or cluster mode. When a Stat Server application that was originally configured as a node is run standalone—not as part of the cluster—the application uses the configuration that is defined wholly within the application itself.

Using Genesys Administrator

1. From the `Provisioning` view within Genesys Administrator, locate and select your Stat Server `Application` object.
2. In the `Tasks` pane, select `Start Application`. (Also, right-clicking your `Application` object displays the shortcut menu that contains this menu item.)
3. In the confirmation dialog box, select `Yes`.

Your Stat Server application starts.

For information about how to use Genesys Administrator, refer to the *Genesys Administrator Help*.

On UNIX

1. Go to the directory where you have installed the Stat Server application.
2. Review the contents of the `run.sh` script to ensure that it either includes or excludes the `-cluster` parameter depending on the mode in which Stat Server is to operate. Edit the file, if necessary.
3. At the command line, type:

   ```
   ./run.sh
   ```

   Or, type the name of the Stat Server executable followed by the appropriate command-line parameters using the following syntax:

   ```
   ./statserv -host hostname -port portno -app application
   [-cluster Solution] [-transport-port trnsportno] [-transport-
   address IPaddress ]
   ```

   where:

   - *hostname* refers to the name of the host on which Configuration Server is running.
   - *portno* refers to the communication port that client applications must use to connect to Configuration Server.
   - *application* refers to the name of the Stat Server `Application` object as defined to the Configuration Server.
   - *Solution* is the name of the Stat Server solution to which the Stat Server application belongs. Specifying this parameter is mandatory if Stat Server is to operate in cluster mode.
   - *trnsportno* is the port number that Stat Server uses for TCP/IP connection to Configuration Server. Specifying this parameter is optional.
   - *IPaddress* is the IP address that Stat Server uses for TCP/IP connection to Configuration Server. Specifying this parameter is optional.

Note: If the host or application name contains spaces or hyphens (–), enclose it in double quotation marks.

For example, to start Stat Server with parameters specifying the host as `cs-host`, port as `2020`, and name as `Stat Server 03`, type:

```
./statserv -host "cs-host" -port 2020 -app "Stat Server 03"
```

On Windows, from the Command Line

Start a Stat Server application from the `Start` menu or open a console window, go to the directory where Stat Server is installed, and type the following command:

```
./statserv.exe -host hostname -port portno -app application
[-cluster Solution] [-transport-port trnsportno] [-transport-address
IPaddress ]
```

where:

- `hostname` refers to the name of the host on which Configuration Server is running.
- `portno` refers to the communication port that client applications must use to connect to Configuration Server.
- `application` refers to the name of the Stat Server `Application` object as defined to the Configuration Server.
    - `Solution` is the name of the Stat Server solution to which the Stat Server application belongs. Specifying this parameter is mandatory if Stat Server is to operate in cluster mode.
- `trnsportno` is the port number that Stat Server uses for TCP/IP connection to Configuration Server. Specifying this parameter is optional.
- `IPaddress` is the IP address that Stat Server uses for TCP/IP connection to Configuration Server. Specifying this parameter is optional.

Note: If the host or application name contains spaces or hyphens (–), enclose it in double quotation marks.

For example, to start a Stat Server application in regular mode with parameters specifying the host as `cs-host`, port as `2020`, and name as `Stat Server 03`, from the Stat Server working directory, type:

```
statserv.exe -host "cs-host" -port 2020 -app "Stat Server 03"
```

As a Windows Service

1. Open the Windows `Control Panel` and double-click the `Services` icon. The `Services` dialog box opens.
2. Select your Stat Server service from the list and click `Start`. (If you did not install Stat Server as a Windows Service, your application does not appear for selection in the `Services` list box.)

Note: Since you can install the Local Control Agent (LCA) as a Windows Service with the user interface disabled, all servers started through Genesys Administrator, in this case, are started without a console, unless you specifically select the `Allow Service to Interact with Desktop` check box for both LCA and Stat Server.

Stopping a Stat Server Application

You can stop a Stat Server application from running in any of the following ways:

- From the Genesys Administrator.

- Manually on UNIX.
- Manually on Windows.
- Via the Windows `Control Panel`.

Note: Be sure that the `Auto Restart` checkbox is cleared for the Stat Server `Application` in the Genesys Administrator to prevent Stat Server from self-starting.

Using Genesys Administrator

1. From the `Provisioning` view within Genesys Administrator, locate and select your Stat Server `Application` object.
2. In the `Tasks` pane, select `Stop Application`. (Also, right-clicking your `Application` object displays the shortcut menu that contains this menu item.)
3. In the confirmation dialog box, select `Yes`.

Your Stat Server application stops.

For information about how to use Genesys Administrator, refer to the *Genesys Administrator Help*.

On UNIX

Stop a Stat Server application on UNIX using any one of the following methods:

- On the command line, type `kill -SIGTERM` *`processid`* where *`processid`* is Stat Server's UNIX process ID.
- Press `^C` from the active Stat Server window.
- If you are using LCA and SCS, you can stop Stat Server from running on UNIX using Genesys Administrator.

On Windows

If Stat Server is running as an application—not as a Windows Service—switch to its console window and press `Ctrl+Break` to stop it.

If you are running Stat Server as a Windows NT Service, you should stop it only from the Services Control Manager. To stop Stat Server running as a Windows NT Service:

1. Open the `Control Panel` and double-click the `Services` icon. The `Services` dialog box opens.
2. Select your Stat Server service from the list and click `Stop`.

Genesys

Chapter

10 Starting and Stopping a Stat Server Solution

This chapter applies to Stat Server deployed in cluster mode and contains procedures for starting and stopping a Stat Server solution on the supported platforms. Starting a Stat Server solution starts all of the applications configured within the solution. Likewise, stopping the Stat Server solution stops all of the solution's applications. Start procedures assume that you have properly configured the solution as well as configured and installed the applications that are defined as the solution's components. If not, refer to Chapter 4, "Configuring a Stat Server Cluster Solution," beginning on page 21.

You can also start and stop each Stat Server application that is a component of a Stat Server solution independent of starting the solution; however, this method of sequential invocation is not recommended. All clustered Stat Server applications should be started simultaneously in order to achieve the objectives of the cluster. Refer to Chapter 9, "Starting and Stopping a Stat Server Application," beginning on page 101 for this information.

This chapter contains these sections:

What Must Be Running Prior to Start

Genesys recommends that you start a Stat Server solution with certain applications already running; namely:

- Configuration Server
- SIP Cluster (if Cluster mode selected)

In addition, to start a Stat Server solution from the Genesys Administrator or Genesys Administrator Extension (GAX), have the following up and running:

- Solution Control Server
- Local Control Agent
- Genesys Administrator / GAX

If you have configured the Stat Server application to write to a database, also have running:

- RDBMS
- DB Server (if compatibility mode with work via DB Server selected)

Starting a Stat Server Solution

Starting a solution entails starting all of the solution's components in the order that is specified by each components startup priority. You can start a Stat Server solution in any of the following ways:

- From Genesys Administrator
- From the Windows command line

3. Using Genesys Administrator

1. Open Genesys Administrator, navigate to the `Solutions` folder, and locate your Stat Server `Solution` object.

2. In the `Solutions` pane, right-click your Stat Server solution and select `Start Solutions.`

3. Click `Yes` in the confirmation box that appears.

Your Stat Server solution starts. For information about how to use Genesys Administrator, refer to *Framework 8.5 Genesys Administrator Help.*

On Windows, from the Command Line

You can also start a Stat Server solution from the Windows command line using the Framework `mlcmd.exe` command-line utility. This utility is located in the folder in which Solution Control Server was installed. Refer to Appendix A of the *Framework 8.5 Management Layer User's Guide* for information about how to use this utility, its parameters, and its output.

Stopping a Stat Server Solution

Stopping a solution entails stopping all of the solution's components. You can stop a Stat Server solution in any of the following ways:

- Using Genesys Administrator/GAX (This is the recommended approach.)

- Manually on Windows

Note: Be sure that the `autorestart` property is cleared for the Stat Server `Application` to prevent Stat Server from self-starting.

Using Genesys Administrator

1. Open Genesys Administrator, navigate to the `Solutions` folder, and locate your Stat Server `Solution` object.
2. In the `Solutions` pane, right-click your Stat Server solution and select `Stop Solutions.`

   Note: Stat Server does not support the ability to gracefully shut down. If, at this step, you select `Stop Solutions Gracefully,` the Management Layer performs an abrupt shut down of the Stat Server solution. For this scenario, consider configuring the `suspending-wait-timeout` option, described on page 94.

3. Click `Yes` in the confirmation box that appears.

Your Stat Server solution stops.

On Windows

If the Stat Server solution is running as a Windows application switch to its console window and press `Ctrl+Break` to stop it.

Chapter

11 Optimizing Performance

Review the recommendations provided in this chapter to optimize Stat Server performance. This chapter contains the following sections:

Hardware-Related Recommendations

When you are planning to deploy Stat Server to your environment, follow these recommendations:

For Stat Server Operating in Regular Mode

For Stat Server operation in regular mode:

- Consider the following formula, which approximates Stat Server memory, in megabytes, for a typical large contact center:

  $$MemoryReqd = 100 + (NStatistics \times 0.0012)$$

  where *NStatistics* represents the number of opened statistics and 0.0012 refers to approximately 1.2 KB of memory per statistic. This formula applies to Stat Server memory calculation of core statistics. Java Extension clients might request additional memory of which Stat Server is unaware.

  For example, Stat Server on a computer with 1.5 GB of memory should be more than ample to handle CC Analyzer requests of 30,000 active `Agent` or `Place` objects that originate from the Genesys-provided Agent and Place reports):

  $$\begin{aligned} NStatistics &= 28 \text{ statistics/report layout} \times 30{,}000 \text{ objects} \\ &= 840{,}000 \text{ statistics} \\ MemoryReqd &= 100 + (840{,}000 \times 0.0012) \\ &= 1{,}108 \text{ MB} \end{aligned}$$

  For smaller contact centers, you can reduce the constant (100) to a smaller value.

Install Stat Server on a computer with sufficient physical memory to avoid swapping.

- Consider distributing the total number of required statistics for Solution Reporting and real-time interaction processing for all solutions over a number of Stat Server applications.
- Install Stat Server and source event server on the same computer or connect them through a fast LAN. If you are using several T-Server applications, position Stat Server nearest to the one.
- Do not install Stat Server on the same computer as Configuration Server.
- Do not install real-time, third-party applications on the computer that is running Stat Server.
- If you want to store Stat Server data, consider dedicating a separate Stat Server application whose sole purpose is to write data to the Stat Server database.
- For large contact centers, consider allocating approximately 5 MB of space for each day Stat Server writes data to a database. This recommendation applies only if you configure Stat Server with a database access point and enable your Stat Server application to write data to a database by setting corresponding configuration options.

For Stat Server Operating in Cluster Mode

For Stat Server operation in cluster mode:

- Install Stat Server and source event server on the same computer or connect them through a fast LAN. If you are using several SIP Server applications, position Stat Server nearest to the one.
- The hardware should be powerful enough to handle the entire SIP cluster and Stat Server cluster. The computer should have about at least 32GB of memory and 32 cores.
- Suggestion: Dedicate one node per processor.
- For better performance, in addition to the T-Controller connection, you consider configuring more than one Stat Server application each connecting to the same Interaction Proxy in order to better distribute call loads amongst all Stat Server nodes.

Software-Related Recommendations

For Stat Server Operating in Regular Mode

- For Stat Server applications that write to the Stat Server database, configure options only for the tables that you need by setting the following configuration options:
  - `login-table`
  - `qinfo-table`
  - `status-table`
  - `voice-reasons-table`

- Review the configuration options that are related to write operations to this database:
  - For Oracle, Microsoft SQL, and DB2 relational database management systems (RDBMSs), set the `enable-binding` option to `Yes`.
  - Set the `local-time-in-status-table` configuration option to `No` if you do not need a translation of UTC time to the time zone of the host on which Stat Server is deployed.
  - Set `ixn-id-in-status-table` to `No` for Solution Reporting and other clients that employ only an interaction's connection ID.

You can improve Stat Server performance further by tuning Stat Server configuration options:

- Specify only the debugging log level that you need by setting the `debug-level` configuration option appropriately.

For Stat Server Operating in Cluster Mode

Stat Server operating in cluster mode supports:

- 60,000 administered agents (aka regular DNs)
- 200,000 configured agents
- 40,000 simultaneously logged-in agents
- 20,000 skill-based virtual agents groups
- 10,000 virtual queues
- 20,000 route points
- 2,000 agents per agent group
- 40 calls per second
- 4.46 million open statistics
- 60,000 simultaneous calls

One Stat Server instance sustains the load of one T-Controller up to 150 calls per second.

The number of Stat Server instances within a solution need not correlate with the number of SIP Server nodes in a SIP cluster. Instead, this number depends on the following:

- Call volume
- Number of Stat Server clients
- Number of opened statistics

and other factors. Refer to SIP Cluster documentation for other information pertinent to Stat Server performance within the cluster.

You can operate multiple Stat Server solutions on the same host with the restriction that each Stat Server cluster must designate a different Stat Server solution.

Genesys

Chapter

12 Application Files

The Stat Server installation routine creates a root application folder with two subfolders:

- `java`
- `sql_scripts`

Tables 18, 19, and 20 in this chapter describe the files comprising each folder.

Warning! Do not attempt to run the SQL scripts manually because of the potential for data loss. They are intended only for Stat Server's internal use and advanced database administrators.

Table 18: Contents of the Root Folder

File Name	Description
common.lms	File storing log messages common to all Genesys components.
dbclient_db2.exe, dbclient_oracle.exe, dbclient_msql.exe (Windows) dbclient_db2_32, bclient_db2_64, dbclient_oracle_32, dbclient_oracle_64 (Unix)	The dbclient executable.
ip_description.xml	File storing installation package content.
read_me.html	File containing general information about the installation package.
startServer.bat (Windows) run.sh (Unix)	Batch file containing the Stat Server executable and command-line parameters used to start Stat Server.

Table 18: Contents of the Root Folder (Continued)

File Name	Description
statserv.exe (Windows) statserv (Unix)	Application executable. Where Stat Server supports both the 32- and 64-bit memory models for a particular platform, Stat Server uses your selection during the installation to define this file.
statserv_32.exe (Windows) statserv_32 (Unix)	Application executable for 32-bit platforms. This file and statserv_64 appear only on those platforms that support both memory models.
statserv_64.exe (Windows) statserv_64 (Unix)	Application executable for 64-bit platforms. This file and statserv_32 appear only on those platforms that support both memory models.
statserv.pdb	File for advanced troubleshooting of Stat Server on Windows operating systems.
StatServer.lms	File storing Stat Server–specific log messages.
java subfolder	Subfolder. See Table 20 for folder contents.
sql_scripts subfolder	Subfolder containing three subfolders, holding SQL scripts for each of the following RDBMS types: • DB2 • Oracle • Microsoft SQL See Table 19 for the contents of each subfolder.

Table 19: Contents of the sql_scripts/[dbtype] Subfolder

File Name	Description
login_[dbtype].sql	SQL script that creates the LOGIN table (and indexes and procedures, as necessary) for the RDBMS type.
qinfo_[dbtype].sql	SQL script that creates the QINFO table (and indexes and procedures, as necessary) for the RDBMS type.
status_[dbtype].sql	SQL script that creates the STATUS table (and indexes and procedures, as necessary) for the RDBMS type.
status_ixnid_[dbtype].sql	A variation of the status_[dbtype] script that creates the STATUS table with one additional field, IxnID.
status_ltime_[dbtype].sql	A variation of the status_[dbtype] script that creates the STATUS table with two additional fields, StartLocalTime and EndLocalTime, to store the start and end times in the local time zone.

Table 19: Contents of the sql_scripts/[dbtype] Subfolder (Continued)

File Name	Description
status_ltime_ixnid_[dbtype].sql	A variation of the `status_[dbtype]` script that creates the `STATUS` table with three additional fields, `IxnID`, `StartLocalTime`, and `EndLocalTime`, to store the start and end times in the local time zone.
voice_reasons_[dbtype].sql	SQL script that creates the `VOICE_REASONS` table (and indexes and procedures, as necessary) for the RDBMS type.

Table 20: Contents of the java Subfolder

File Name	Description
ssjcldr.class	Java class loader; a member of the Stat Server Java host environment
statserver.jar	Library that is part of the Stat Server Java SDK, which, in turn, is part of the Stat Server Java host environment
statserver_impl.jar	A member of the Stat Server Java host environment
kvlists.jar	Library that is part of the Stat Server Java SDK, which, in turn, is part of the Stat Server Java host environment. Stat Server uses this file in conjunction with Stat Server Java extensions.
kv65_adapter.jar	Library that is part of the Stat Server Java SDK, which, in turn, is part of the Stat Server Java host environment
ext folder	Directory to store the Genesys solution-specific extensions, such as: • `eServiceContactStat.jar` • `eServiceInteractionStat.jar` • `eServiceSystemStat.jar` • `OCCStatExtension.jar`
lib folder	Directory to store the Genesys' solution-specific libraries, such as: • `dsw_api_java.jar` • `dsw_extension_core.jar` • `dsw_transformers.jar`

The `templates` subfolder includes `Stat_Server_850.apd` and `Stat_Server_850.xml` files.

Appendix

A Physical Data Models for Stat Server Tables

This appendix describes the database tables to which Stat Server stores data if the status-table, qinfo-table, login-table, and/or voice-reasons-table configuration options are enabled. (These configuration options are described in Table 5 beginning on page 49.) The information in this appendix is divided among the following topics:

Note: The appendix applies to Stat Server applications that operate in regular mode only.

Introduction

Stat Server stores status data about places and agents in the STATUS table and data about queues in the QINFO table. Stat Server also maintains information about agent login and logout events in its LOGIN table. These tables are independent and do not reference each other. Genesys Info Mart and custom reporting use these tables.

The VOICE_REASONS table stores hardware and software reasons for agents to change or continue the Ready and NotReady states and the AfterCallWork work mode, when handling voice interactions. Genesys Info Mart uses this table and makes this data available for custom reporting.

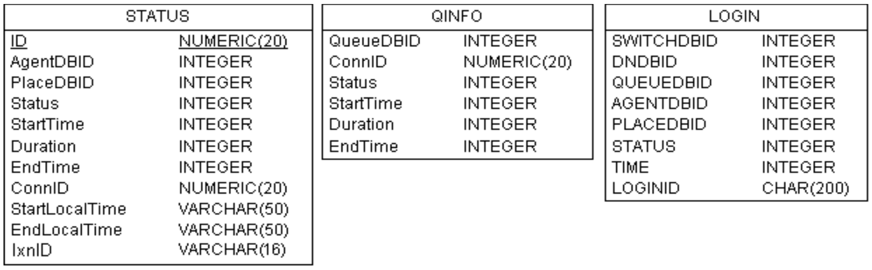
DBID refers to the database identifier that the Configuration Layer assigns to a telephony object when an enterprise is configured.

Note: Stat Server, while functioning in backup mode, does not write data to its database, even if configured to do so. This enables the primary or backup Stat Server, while functioning as the primary application, to store data to the same database.

Table Schema by RDBMS

Figures 3 through 5 depict Stat Server table schema for the supported RDBMSs.

Note: Data from the `VOICE_REASONS` table is not available for custom reporting directly from the Stat Server database. Therefore, the structure of the `VOICE_REASONS` table is not provided in this guide.

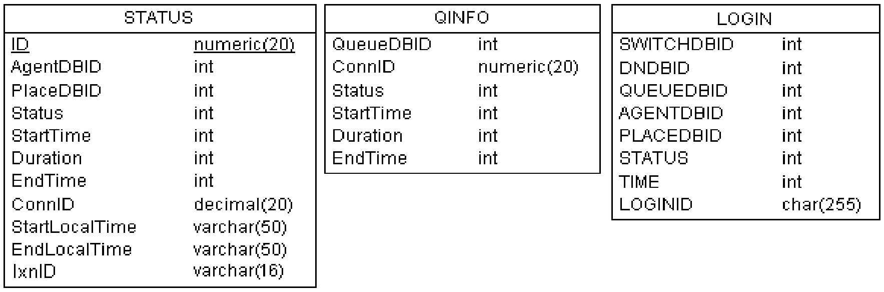


STATUS	
ID	NUMERIC(20)
AgentDBID	INTEGER
PlaceDBID	INTEGER
Status	INTEGER
StartTime	INTEGER
Duration	INTEGER
EndTime	INTEGER
ConnID	NUMERIC(20)
StartLocalTime	VARCHAR(50)
EndLocalTime	VARCHAR(50)
IxnID	VARCHAR(16)

QINFO	
QueueDBID	INTEGER
ConnID	NUMERIC(20)
Status	INTEGER
StartTime	INTEGER
Duration	INTEGER
EndTime	INTEGER

LOGIN	
SWITCHDBID	INTEGER
DNDBID	INTEGER
QUEUEDBID	INTEGER
AGENTDBID	INTEGER
PLACEDBID	INTEGER
STATUS	INTEGER
TIME	INTEGER
LOGINID	CHAR(200)

Figure 3: Table Schema for a DB2 Stat Server Database

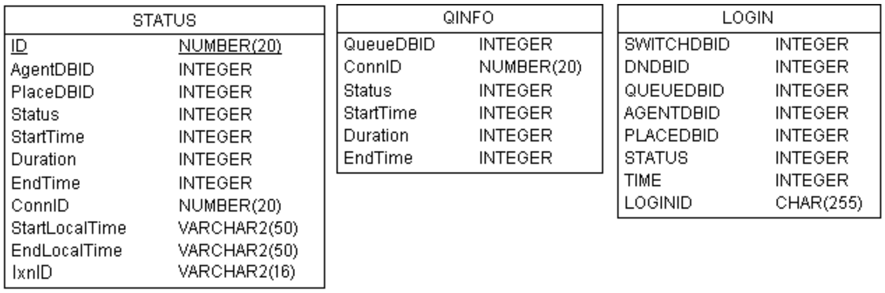


STATUS	
ID	numeric(20)
AgentDBID	int
PlaceDBID	int
Status	int
StartTime	int
Duration	int
EndTime	int
ConnID	decimal(20)
StartLocalTime	varchar(50)
EndLocalTime	varchar(50)
IxnID	varchar(16)

QINFO	
QueueDBID	int
ConnID	numeric(20)
Status	int
StartTime	int
Duration	int
EndTime	int

LOGIN	
SWITCHDBID	int
DNDBID	int
QUEUEDBID	int
AGENTDBID	int
PLACEDBID	int
STATUS	int
TIME	int
LOGINID	char(255)

Figure 4: Table Schema for a Microsoft SQL Stat Server Database



STATUS	
ID	NUMBER(20)
AgentDBID	INTEGER
PlaceDBID	INTEGER
Status	INTEGER
StartTime	INTEGER
Duration	INTEGER
EndTime	INTEGER
ConnID	NUMBER(20)
StartLocalTime	VARCHAR2(50)
EndLocalTime	VARCHAR2(50)
IxnID	VARCHAR2(16)

QINFO	
QueueDBID	INTEGER
ConnID	NUMBER(20)
Status	INTEGER
StartTime	INTEGER
Duration	INTEGER
EndTime	INTEGER

LOGIN	
SWITCHDBID	INTEGER
DNDBID	INTEGER
QUEUEDBID	INTEGER
AGENTDBID	INTEGER
PLACEDBID	INTEGER
STATUS	INTEGER
TIME	INTEGER
LOGINID	CHAR(255)

Figure 5: Table Schema for an Oracle Stat Server Database

Table and Column Descriptions

The Stat Server database contains four tables:

- LOGIN, described on page 121
- QINFO, described on page 122
- STATUS, described on page 124
- VOICE_REASONS, described on page 126

The LOGIN Table

The LOGIN table contains the history of login and logout activity for resources on both voice and multimedia channels. Stat Server writes to this table if the login-table configuration option is set to yes.

Stat Server detects login activity, for T-Server and SIP Server clients, upon receipt of an EventAgentLogin TEvent; Stat Server detects logout upon receipt of EventAgentLogout.

For medias reported through Interaction Server, the pair of EventAgentLogin and EventMediaAdded events are used in Stat Server logic to determine agent readiness to process interactions on a particular media channel. The EventMediaRemoved and EventAgentLogout pair are the triggering logout events.

Stat Server also writes login and logout records in LOGIN table even if EventMediaAdded and EventMediaRemoved events were not received from Interaction Server, but media channel was present in attr_media_list of Interaction event EventAgentLogin.

Table 21 describes the LOGIN table's fields, which are presented in order of appearance.

Table 21: Field Descriptions for the LOGIN Table

Field Name	Description
SWITCHDBID	The DBID of the switch at whose DN the agent has logged in or out.
DNDBID	The DBID of the DN at which the agent has logged in or out. This value is 0 (zero) if the agent has logged in to or logged off a media channel.
QUEUEDBID	The DBID of the ACD queue where the agent has logged in or out.
AGENTDBID	The DBID of the agent who has logged in or out.
PLACEDBID	The DBID of the place where the agent has logged in or out.
STATUS	1 if the agent has logged in. 0 if the agent has logged out.
TIME	Time, in seconds since 1 January 1970 UTC (Universal Time Coordinated), when the related login or logout event occurred.
LOGINID	The login ID of the resource for this record. The initial size of this field, as defined in the login.sql script for your RDBMS, is 255 characters, but you can adjust it as appropriate for your environment. Where the agent has logged in to or logged off a media channel, this field stores the media type. Stat Server gathers this information from the MediaType attribute of the triggering TEvent.
ID	Auto-generated primary key. Used only with Oracle RAC and if the identity-in-login-table option is set to yes.
APP_DBID	DBID of Stat Server application. Used only with Oracle RAC and if the identity-in-login-table option is set to yes.

The QINFO Table

The QINFO table contains the history of voice interaction activities from the perspective of one or more mediation DNs that are registered to the Stat Server application. Stat Server writes to this table if the qinfo-table configuration option is set to `yes`. Table 22 describes this table's fields, which are presented in their order of appearance.

Table 22: Field Descriptions for the QINFO Table

<table>
<tr><th>Field Name</th><th>Description</th></tr>
<tr><td>QueueDBID</td><td>The queue's DBID.</td></tr>
<tr><td>ConnID</td><td>An identifier that T-Server assigns to the connected call. The value in this field is `0` (zero) if the status is not related to the call.
In multi-site scenarios, if the first transfer connection ID differs from the current connection ID associated with the call, the value stored in this field is the first transfer connection ID. Prior to Stat Server release 7.0.3, this field stored the current connection ID.</td></tr>
<tr><td>Status</td><td>The status of the transition of a call through a queue whose DBID is displayed in the `QueueDBID` field (of this table). The possible values of `1`–`9` indicate the following statuses and durations:
<table>
<tr><th>Call Status</th><th>Code</th><th>Duration</th></tr>
<tr><td>Diverted from queue</td><td>1</td><td>Time in queue</td></tr>
<tr><td>Abandoned within queue</td><td>2</td><td>Time in queue</td></tr>
<tr><td>Diverted from queue (answered while ringing)</td><td>3</td><td>Time in queue plus time spent ringing</td></tr>
<tr><td>Diverted from queue (abandoned while ringing)</td><td>4</td><td>Time in queue plus time spent ringing</td></tr>
<tr><td>Party changed from queue (for consultation calls only)</td><td>5</td><td>Time in queue plus time spent ringing until party changed</td></tr>
<tr><td>Diverted from queue (forwarded)</td><td>6</td><td>Time in queue</td></tr>
<tr><td>Call cleared[*] (for virtual queues only)</td><td>7</td><td>Time in queue</td></tr>
<tr><td>Call cleared after being stuck on a distribution DN</td><td>8</td><td>Time in queue</td></tr>
<tr><td>Call cleared after being stuck while ringing at an agent's DN</td><td>9</td><td>Time at DN</td></tr>
</table>
</td></tr>
<tr><td>StartTime</td><td>A sequence number representing the date and time when the status displayed in the `Status` field (of this table) began. The sequence begins with January 1, 1970, 12:01 AM UTC and increments every second. For example, `878159351` represents October 29, 1997, 13:09:11. Each new second represents an increment of 1 in the sequence.</td></tr>
</table>

Table 22: Field Descriptions for the QINFO Table (Continued)

Field Name	Description
Duration	The duration, in seconds, of the status displayed in the Status field.
EndTime	A sequence number representing the date and time when the status displayed in the Status field (of this table) ended. The sequence begins with January 1, 1970, 12:01 AM, UTC, and increments each second. For example, 878159351 represents October 29, 1997, 13:09:11. Each new second represents an increment of 1 in the sequence.

*. Indicates that a call was cleared from a virtual queue (diverted to an agent's DN from another virtual queue). This status is based on the CallCleared retrospective, instantaneous action. (Refer to the *Framework 8.5 Stat Server User's Guide* for a description of this action.)

The STATUS Table

The STATUS table contains the history of status changes for agent and place resources. This table also stores the current status for such resources that persist more than 600 seconds. Stat Server determines place status by the highest ranking action (as defined by Stat Server status priority tables) that occurs at the DNs it includes.

Starting with the 7.6.1 release, Stat Server supports status reporting for multimedia DNs—DNs capable of handling multiple simultaneous interactions. By setting the multimedia-activity-in-status-table configuration option to yes, Stat Server selectively accounts for non-voice-related actions on multimedia DNs in the status records that are written to this table.

Starting with the 8.0 release, Stat Server records the interaction IDs of multimedia interactions when the ixn-id-in-status-table configuration option is set to yes.

Note: The StartLocalTime, EndLocalTime, and IxnID fields appear only if the appropriate script was run to create the STATUS table. Refer to Table 19 on page 116 for descriptions of the scripts.

Stat Server writes to this table only if the status-table configuration option is set to yes. Table 23 describes this table's fields, which are presented in their order of appearance.

Table 23: Field Descriptions for the STATUS Table

Field Name	Description
ID	A unique key field used for internal purposes. Upon reaching 4,294,967,296 (that is, 2^{32}), Stat Server restarts the counter reusing all values starting from 1, provided that no records are associated with the IDs to be reused. **Warning!** To store new records after the number of records in the STATUS table reaches 4,294,967,296, clear the STATUS table. To keep previous records, back up this table's data into a backup database prior to clearing the table.
AgentDBID	The database ID (DBID) of an agent, logged into the place (which DBID is displayed in the Place DBID), or 0 (zero).
PlaceDBID	The DBID of a place or 0 (zero).
Status	The status of the place whose DBID appears in the PlaceDBID field or the status of the agent whose DBID appears in the AgentDBID field. If agent is logged into a place, he or she shares the status of the place and this status is written to the table. Agent status is written when the agent is not logged into any place. The following lists STATUS field values and their significance: 4 WaitForNextCall (Ready) 5 OffHook 6 CallDialing 7 CallRinging 8 NotReadyForNextCall 9 AfterCallWork 13 CallOnHold 16 ASM_Engaged 17 ASM_Outbound 18 CallUnknown 19 CallConsult 20 CallInternal 21 CallOutbound 22 CallInbound 23 LoggedOut
StartTime	A sequence number representing the date and time when the status displayed in the Status field (of this table) began. The sequence begins with January 1, 1970, 12:01 AM UTC and increments each second. For example, 878159351 represents October 29, 1997, 13:09:11. Each new second is represented by an increment of 1 in the sequence.
Duration	The duration, in seconds, of the status displayed in the Status field in this table.

Table 23: Field Descriptions for the STATUS Table (Continued)

Field Name	Description
EndTime	A sequence number representing the date and time when the status displayed in the Status field (of this table) ended. The sequence begins with January 1, 1970, 12:01 AM, UTC, and increments each second. For example, 878159351 represents October 29, 1997, 13:09:11. Each new second is represented by an increment of 1 in the sequence. Beginning with the 7.1 release, if Stat Server is configured not to set status end times during updates (status-table-update-endtime-at-end-only=yes), this field holds a 0 (zero) value if the status does not complete before the update of long-running statuses.
ConnID	An identification number that T-Server assigns to the connected call. The value in this field is 0 (zero) if the status is not related to a voice interaction. In multi-site scenarios, if the first transfer connection ID differs from the current connection ID associated with the call, the value stored in this field is the first transfer connection ID. Prior to Stat Server release 7.0.3, this field stored the current connection ID.
StartLocalTime	A string containing a user-defined format for the local time of status start. The format of the start local time is controlled by the time-format option. This field is populated if the local-time-in-status-table configuration option has been enabled.
EndLocalTime	A string that contains a user-defined format for the local time of status end. The format of the end local time is controlled by the time-format option. This field is populated if the local-time-in-status-table configuration option has been enabled.
IxnID	A string that contains the number that Interaction Server assigns to an interaction. The value of this field is null if the ixn-id-in-status-table configuration option is set to off or if the associated status for this record originated from a source other than Interaction Server. In conjunction with a yes setting for the ixn-id-in-status-table configuration option, it is also recommended, although not required, that you set multimedia-activity-in-status-table to true.

The VOICE_REASONS Table

Stat Server writes to the VOICE_REASONS table if the voice-reasons-table configuration option is set to yes in the Stat Server application. This table contains the history of hardware and software reasons for each agent to change or continue the Ready and NotReady states and the AfterCallWork work mode when handling voice interactions. (Hardware reasons are reported by the switch whereas software reasons are established at a software level by a request from a software application, such as an agent desktop.)

Stat Server retrieves Reasons information from data that is attached to the EventAgentReady and EventAgentNotReady TEvents for a DN assigned to a place that has a logged-in agent. Stat Server inserts reason records into the table retroactively—a record is added only after the Reasons value or work mode has changed or the DN state associated with the reason has ended.

The data from the Stat Server's VOICE_REASONS table is not available for custom reporting off the Stat Server database directly; therefore, no description of the VOICE_REASONS table structure is provided in this guide.

Reasons data is available to users of Genesys Info Mart releases 7.2–7.6. Refer to the *Genesys Info Mart Operations Guide* for information about Reasons data in the Info Mart database.

Genesys

Appendix

B Manually Purging Data from the Stat Server Database

Stat Server provides no utility to periodically purge unwanted data from the Stat Server database and Genesys provides no defined procedure for implementing the purge. What data to purge and the purge operation itself are left to your discretion.

The steps, however, are relatively simple:

1. Back up your Stat Server data.
2. Determine your purge criteria—for example, the date beyond which to purge data.
3. For time-related purge criteria, determine the UTC-equivalent integer for the targeted date beyond which you want to purge data.
4. Write and execute an SQL script to purge data based on your criteria.

Note: The appendix applies to Stat Server applications that operate in regular mode only.

Determining the Purge Criteria

This appendix provides one approach, based on time, for trimming down the data stored in the Stat Server database. You may want to purge data based on other criteria, such as deactivated resources or status. In addition, you may wish to apply different purge rules to each of the `STATUS`, `QINFO`, and `LOGIN` tables. Tailor the suggestions provided in this appendix to meet your business need.

Time-Related Fields in the Stat Server Database

Data in the Stat Server database is time-stamped in accordance with the time that Stat Server detected events from other servers. (The `UseSourceTimeStamps` feature does not pertain to data stored in the Stat Server database.) The `STATUS` table holds the following time-related fields to measure when the status of a particular agent or place changes:

- `StartTime` (and `StartLocalTime`)
- `EndTime` (and `EndLocalTime`)

The `QINFO` table holds:

- `StartTime`
- `EndTime`

Finally, the `LOGIN` table holds the `Time` time-related field.

Except for the `LocalTime` fields in the `STATUS` table, all time fields are based on Coordinated Universal Time (UTC), which measures the seconds from January 1, 1970, 12:01 AM. To purge data prior to a particular date, you must have the equivalent UTC integer value of your targeted date.

Note: Some `EndTime` fields may hold `0` values for incompleted statuses. Basing a purge operation solely on this field is not advisable.

Determining the UTC Equivalent for a Selected Date

To determine the number of seconds between your targeted date and January 1, 1970, calculate the number of days between these two dates, and multiply the result by 86,400—the number of seconds in one day. There are numerous websites, such as `http://www.timeanddate.com`, that can help you to calculate the difference between two dates, or you can query your own RDBMS, using its date-diff functions.

Designing a Purge Script

`QINFO`, `LOGIN`, and `STATUS` are independent tables in the Stat Server database; there are no fields joining these tables; no parent-child inter-relationships exists between them. Therefore, when deleting records, you need not be concerned about maintaining data integrity *in between* these tables, such as the integrity that is preserved by cascade-update and -delete operations for some databases. The absence of data in one Stat Server table has no impact on the content or significance of data in another Stat Server table.

One consideration to weigh in your purge script's design, however, is that of performance. If the volume of unwanted rows is large, executing one delete statement to purge this data will certainly impact RDBMS performance.

Therefore, you should break up the operation so that the RDBMS purges data into whatever you determine to be manageable chunks.

The following generalized SQL statement deletes data:

```
DELETE FROM StatServerTable WHERE criteria ;
```

To delete rows from the `LOGIN` table for resources that logged in prior to July 30, 2001, issue the following query against the database:

```
DELETE FROM LOGIN WHERE Time < 996451200 ;
```

[996,451,200 = 11,533 days (between 1/1/70 and 7/30/01) * 86,400 sec/day]

This assumes that the volume of data in your database prior to July 30, 2001 is of a manageable enough size to be purged by one `DELETE` statement without adversely impacting performance.

Genesys

Supplements

Related Documentation Resources

The following resources provide additional information that is relevant to this software. Consult these additional resources as necessary.

Management Framework

- The *Framework 8.5 Deployment Guide,* which will help you configure and install other Framework components.
- The *Framework 8.5 Stat Server User's Guide,* for information about the makeup of a statistic and the manner in which Stat Server connects and provides data to its clients.
- *Genesys Administrator Help,* for information about configuring Genesys applications using Genesys Administrator.

SIP Cluster

- *SIP Cluster Solution Guide,* which serves as a central location for the descriptions and deployment instructions of all components of a SIP Cluster.
- *SIP Proxy Deployment Guide,* which describes the interface for SIP communication between SIP devices and SIP Server components.
- *SIP Voicemail Deployment Guide,* which describes the options and instructions for configuring a SIP voicemail server.

Note: SIP Cluster is available as a Restricted release at the time of this publication.

Genesys

- The *Genesys Resource Capacity Planning Guide,* which explains how the Genesys model has been expanded to serve agents conducting contact center interactions across several media types.
- The *Genesys Security Deployment Guide,* which will help you install the Genesys Security Pack and manage security certificates for the hosts in your contact center.
- The *Reporting Technical Reference* series, which describes the stat type definitions provided by Genesys solutions.
- *Genesys Technical Publications Glossary,* available on the Genesys Documentation website and which provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms used in this document.
- *Genesys Migration Guide*, which ships on the Genesys Documentation Library DVD, and which provides documented migration strategies for Genesys product releases. Contact Genesys Customer Care for more information.
- Release Notes and Product Advisories for this product, which are available on the Genesys Documentation website.

Information about supported hardware and third-party software is available on the Genesys Customer Care website in the following documents:

- *Genesys Supported Operating Environment Reference Guide*
- *Genesys Supported Media Interfaces Reference Manual*

Consult these additional resources as necessary:

- *Genesys Interoperability Guide,* which provides information on the compatibility of Genesys products with various Configuration Layer Environments; Interoperability of Reporting Templates and Solutions; and Gplus Adapters Interoperability.
- *Genesys Licensing Guide,* which introduces you to the concepts, terminology, and procedures relevant to the Genesys licensing system.

For additional system-wide planning tools and information, see the release-specific listings of System-Level Documents on the Genesys Documentation website (docs.genesys.com).

Genesys product documentation is available on the:

- Genesys Customer Care website at `http://genesys.com/customer-care`.
- Genesys Documentation website at `http://docs.genesys.com/`.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at `orderman@genesys.com`.

Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

81fr_dep_statserver_12-2013_v8.1.201.00

You will need this number when you are talking with Genesys Customer Care about this product.

Screen Captures Used in This Document

Screen captures from the product graphical user interface (GUI), as used in this document, may sometimes contain minor spelling, capitalization, or grammatical errors. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Type Styles

Table 24 describes and illustrates the type conventions that are used in this document.

Table 24: Type Styles

Type Style	Used For	Examples
Italic	• Document titles • Emphasis • Definitions of (or first references to) unfamiliar terms • Mathematical variables Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on page 136).	Please consult the *Genesys Migration Guide* for more information. Do *not* use this value for this option. A *customary and usual* practice is one that is widely accepted and used within a particular industry or profession. The formula, $x + 1 = 7$ where x stands for . . .

Table 24: Type Styles (Continued)

Type Style	Used For	Examples
Monospace font (Looks like `teletype` or `typewriter text`)	All programming identifiers and GUI elements. This convention includes: • The *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages. • The values of options. • Logical arguments and command syntax. • Code samples. Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.	Select the `Show variables on screen` check box. In the `Operand` text box, enter your formula. Click `OK` to exit the `Properties` dialog box. T-Server distributes the error messages in `EventError` events. If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls. Enter `exit` on the command line.
Square brackets ([ ])	A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.	`smcp_server -host [/flags]`
Angle brackets (< >)	A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise. **Note:** In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.	`smcp_server -host <confighost>`

Index