Stat Server Deployment Guide

Stat Server 8.5.1

12/30/2021

Table of Contents

Stat Server 8.5 Deployment Guide

Welcome to the *Framework 8.5 Stat Server Deployment Guide*. This manual 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.

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.

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

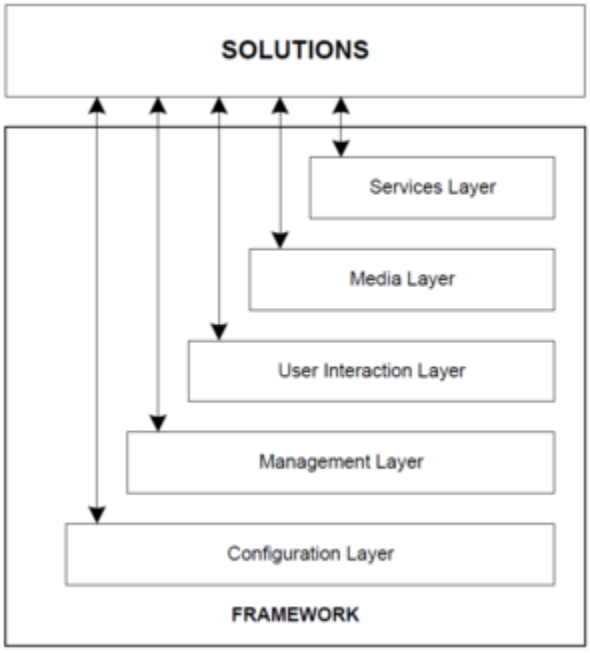
- Computer-telephony integration (CTI) concepts, processes, terminology, and applications.
- Network design and operation.
- Your own network configurations.

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

Stat Server Overview

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.

The Genesys Statistics Server (Stat Server) is part of the Genesys Management Framework Services Layer:



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
- Switches
- Campaigns
- Calling lists

Refer to Stat Server Object Types in the *Framework Stat Server User's Guide* for more information.

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.

New in This Release

This section describes the new or changed functionality that was introduced in release 8.5.1.

Release 8.5.112.10:

- Stat Server supports the following new Agent Workbin actions:
  - InteractionAbandonedDuringOffering
  - InteractionAccepted
  - InteractionAnswered
  - InteractionCleared
  - InteractionDistributed
  - InteractionEntered
  - InteractionReleased
  - InteractionWait

Release 8.5.112.07:

- Stat Server supports protected mode in addition to normal mode during configuration reloads.
- The following new configuration option is added in the **[statserver]** section:
  - config-reload-delay-if-primary
- Stat Server supports Kernel Virtual Machine (KVM) on Windows and Linux.
- Stat Server supports Microsoft SQL Server 2017.

Release 8.5.110.20:

- Stat Server supports changes of the interaction cost for budget-based routing.

Release 8.5.110.18:

- Stat Server supports the SinceLogin time profile for blended agents.
- Stat Server supports the durable InteractionAgentPartyInProgress action for Tenants.
- Stat Server supports the new DoNotDisturb action on a media channel.
- The following new configuration options are added in the **[statserver]** section:
  - interaction-agent-party-in-progress-on-tenant-max-number
  - interaction-agent-party-in-progress-on-tenant-media-list
  - old-stats-remove

Release 8.5.110.05:

- Enhanced tracking of a multimedia interaction associated with the virtual queue (VQ).
  Stat Server continues to collect VQ tracking details using the `EventCustom` event, but now does so from all connected Interaction Servers that may participate in processing of the same interaction. This enhancement addresses cases where URS dispatches routing events through one Interaction Server and then offers the interaction with follow-up handling using another Interaction Server.

Release 8.5.110.03:

- Stat Server supports the budget-based routing.
- The following new configuration options, configured in the new **[budget]** section, have been added to support the budget-based routing:
  - **`Agent`** or **`Place`** object:
    <media type name>
  - **`MediaType`** object:
    default-agent-budget

    default-cost

    default-cost-consult

    default-cost-inbound

default-cost-internal

default-cost-outbound

default-cost-unknown

- **Tenant** object:
  default-agent-budget

  default-cost

  default-cost-consult

  default-cost-inbound

  default-cost-internal

  default-cost-outbound

  default-cost-unknown

  enabled

  interaction-cost-key

Release 8.5.109:

- In addition to Warm Standby, Stat Server supports the Hot Standby redundancy mode for Stat Server high availability (HA) pairs on Windows and Linux platforms. See Hot Standby for more information.
- The following new configuration options, configured in the new **[ha]** section, have been added to support Hot Standby redundancy:
  - addp-remote-timeout
  - addp-timeout
  - addp-trace
  - chunk-size
  - chunk-timeout
  - connect-timeout
  - session-expiration-period
  - session-expiration-timeout
- Stat Server maps the content of the `attribute_reason_desc` from Interaction Server events to the value of the `ReasonValue` key in a `UserData` key-value list on Action for the `CurrentState` or in a `Reasons` key-value list for the `CurrentStateReasons`. See CurrentState Categories for more information.

Release 8.5.108:

- Stat Server supports the overload protection method of reducing CPU consumption.
- The following new configuration options are added in the **[overload]** section:
  - allow-new-connections-during-overload

  - allow-new-requests-during-overload
  - cpu-cooldown-cycles
  - cpu-poll-timeout
  - cpu-threshold-high
  - cpu-threshold-low
  - cut-debug-log
  - protection
  - qos-default-overload-policy
  - qos-recovery-enable-lms-messages
- Stat Server supports the new `DynamicOverloadPolicy` option in the **[<stat type>]** section.
- Stat Server supports the retrospective InteractionAbandoned action for Tenants.
- Stat Server supports Windows Server 2016 and Windows Server 2016 Hyper-V.

Release 8.5.107:

- Stat Server supports Sliding, Selection, and SinceLogin aggregation intervals in the the statistical category Formula.
- Stat Server logs operational statistics and current execution context into a special StatFile log file to help with diagnostics and troubleshooting.
- Stat Server supports the durable InteractionWait action for Tenants. The action is intended to track an interaction until it is actually handled by an agent.
- Stat Server supports the durable DND action on `RegDN`.
- Stat Server supports Red Hat Enterprise Linux 7.
- The following new configuration options are added in the **[statserver]** section:
  - interaction-wait-on-tenant-max-number
  - interaction-wait-on-tenant-media-list
  - stat-file-show-clients-list
  - stat-file-show-options
  - stat-file-timeout

Release 8.5.106:

- Stat Server supports the statistical category Formula for the `Growing` time profile.
- Stat Server supports the Ever Growing time profile for a limited set of the float-valued categories.
- Stat Server supports the new DialRemoteRelease action on `Campaign`/`CallingList` statistical objects.
- Stat Server supports enable-thread, throttle-period, and throttle-threshold options.

Release 8.5.104:

- Stat Server supports new Routable and NotRoutable actions on an agent or place.

- New DistinquishBy feature is introduced.
- Stat Server supports new ApplyFilterAtActionEndOnly stat type option.
- Stat Server supports new ExternalServiceRequested and ExternalServiceResponded actions for Tenants.
- Stat Server supports the `ActorType` and `RequestEnvelope` System attributes on selected actions.
- The following new configuration options are added in the **[statserver]** section:
  - consult-acw-mode
  - suppress-user-data

Release 8.5.103:

- New GroupBy feature is introduced.
- Support of the time zone specification in a growing `TimeProfile` definition.
- Stat Server supports the following new actions for Tenants:
  - InteractionAccepted
  - InteractionAnswered
  - InteractionReleased
- Stat Server supports new InteractionWait action on StagingArea.
- Stat Server supports new `UserEventReceived` action for a regular DN and mediation DN.
- Stat Server supports the `AgentID` System attribute on selected actions.
- Stat Server supports the `GlobalUserData` key-value list in filters and formulas.
- The following new configuration options are added in the **[statserver]** section:
  - interaction-wait-on-sa-max-number
  - interaction-wait-on-sa-media-list

Release 8.5.102:

- Stat Server supports the following new actions for Tenants:
  - InteractionCreated
  - InteractionAbandonedDuringOffering
  - InteractionAccepted1
  - InteractionDeleted
  - InteractionPastAcceptServiceLevel
  - InteractionPastCompletionServiceLevel
- Stat Server calculates regular statistics on StagingArea for the following new actions:
  - InteractionPastAcceptServiceLevel
  - InteractionPastCompletionServiceLevel
- The `ExpectedWaitTime2` statistical category is now applicable to voice media.

- The new `backup-file-aggregates-store` configuration options is added in the `statserver` section.
- Stat Server supports System key-value list in filters.
- Stat Server supports `SuspendAll`/`ResumeAll`/`PeekAll` batch requests from a client.

Release 8.5.101:

- Stat Server supports the following new actions for a regular DN:
  - CallObserving...
  - CallOutboundOriginated
  - CallOutboundReceived
- Stat Server calculates regular statistics on StagingArea for the following new actions:
  - InteractionAbandonedDuringOffering
  - InteractionAccepted
  - InteractionAnswered
  - InteractionReleased
- Absolute paths are now mandatory for the `[java-config]/jvm-path` option. The JVM will not be initialized if the option is set to a relative path.
- Stat Server supports JDK 1.8.
- Stat Server includes functionality for the prevention of database update losses due to commit failures or loss of database server connection.
  Two new database-related options are added:
  - db-timeout
  - db-txn-max-retries

Release 8.5.100:

- Stat Server loads configuration data asynchronously. Therefore, during the initialization Stat Server is able to:
  - Immediately process changes of its configuration options.
  - Dynamically recognize configuration changes such as creating, deleting or editing of configuration objects.
  - Dynamically change the list of its connections.
- After a long disconnect from Configuration Server, when the history log expires, Stat Server is able to re-read its configuration. Therefore, Stat Server processes all configuration changes that were done during its disconnect to Configuration Server.
  This new functionality is essential for large environments, where Stat Server initialization takes a long time.
- The `reconnect-timeout` option in the `statserver` section now is also applicable to the Stat Server connection to Configuration Server.
- Stat Server calculates regular statistics on StagingArea for the following new actions:
  - InteractionCreated

- InteractionEntered
- InteractionDistributed
- InteractionCleared
- InteractionDistributedToQueue
- InteractionDeleted
  Stat Server Java Extensions do not need to be loaded for calculation of these statistics.

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.
- 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).

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.

Note: 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*.

Application 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_851.apd. You should import this file into your configuration environment before configuring Stat Server application.

In the `templates` directory of the *Real-Time Metrics Engine* CD—`Stat_Server_851.xml` file—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 listing of configuration options is located in the RTME Options Reference.) 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.

> **Important**
>
> If you specify more than one database access point, Stat Server will use only the first one.

Configuring Stat Server as Part of a SIP Cluster

Stat Server supports SIP Cluster Solution.

In a SIP Cluster environment Genesys recommends to set Stat Server options **[statserver]**/reg-dns-chunk-delay=1 and **[statserver]**/reg-dns-chunk-volume=100.

Adding SIP Server Applications to Stat Server Configuration

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

- If a particular Stat Server 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 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 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.

Configuring Stat Server in High Availability mode

A high-availability (HA) architecture implies the existence of redundant applications: a primary and a backup. These applications are configured so that if one fails, the other can take over its operations without significant loss of data or impact to business operations.

Stat Server supports both types of redundancy in Genesys software:

- Warm Standby
- Hot Standby. For more information see the Hot Standby (HA) topic in Stat Server User's Guide and HA specific configuration options in RTME Options Reference Manual.

Refer to Redundancy Types in Genesys Security Deployment Guide for more information about redundancy types.

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.

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.
Starting with release 8.5.109, Stat Server supports hot standby mode.

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.

Configuring Stat Server on Slow Networks

For a big configuration environment on a slow network we recommend to:

- Set up an ADDP connection between Stat Server and Configuration Server with longer timeout.
- Set the `packet-size` Configuration Server option to lower value to decrease a maximum size of packets sent by Configuration Server. (See *Framework 8.5 Configuration Options Reference Manual* on the Management Framework page for more information).

Java Sections

> **Important**
>
> For Stat Server release 8.x, Java functionality is reserved for use in conjunction with Genesys-provided reports for Outbound Contact and eServices (formerly known as Multimedia).
> Starting with release 8.5.1, Stat Server is extended to include support for the Orchestration Server (ORS) Java Extension.

Upon startup, Stat Server reads the enable-java configuration option to determine whether SSJE (Stat Server Java Extension) functionality is enabled. Adding Java to the value of the debug-level configuration option enables Stat Server to log messages that are related to Java extension functionality. If the value of the `enable-java` option is `true`, Stat Server processes the information specified in the following Stat Server sections (See the RTME Options Reference for details):

- [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 jvm-path 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`.

- 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`).
- Stat Server loads the Java libraries indicated by the java-libraries-dir configuration option of the [java-config] section.
- 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.
- 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.

  For 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`

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 Statistical Type Sections 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 the java-extensions-dir option.
- stat-type-name is the name of the stat type residing in SSJE.

- 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.

How to Configure Logging Level and Agent Assignment Reset Delay for the OCC Extension

OCC Extension supports the ability to reset agent assignments-related actions both by the explicit request from Outbound Contact Server and upon Outbound Contact Server disconnect. This functionality is required for proper functioning of agent assignments in scenarios involving switchovers between a primary/backup Outbound Contact Server pair.

Configure logging levels and agent assignment reset delays in the [OCCExtension] section of the Stat Server application by setting the following options:

- java-extension-jar
- print-level
- assignment-reset-delay

Troubleshooting Tips

If Windows 2008 does not have the appropriate 32-bit or 64-bit Microsoft Visual C++ 2010 Redistributable Package, Stat Server is unable to load JVM shared library.

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.

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

For information about manipulating Stat Server behavior via configuration options, refer to the RTME Options Reference Guide.

Factors other than Stat Server's own configuration that have an impact on Stat Server output:

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 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 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 `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. 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 `GroupPlaces` objects as well, to determine if origination DNs have been configured therein (in the `Configuration` tab, `Advanced` settings, 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 of the 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.

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.

To Calculate ExpectedWaitTime2

As part of Stat Server's calculation of the ExpectedWaitTime2 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. Starting with Stat Server release 8.5.102, voice is a valid value of the media-type option.

Refer to the ExpectedWaitTime2 statistical category in the Framework Stat Server User's Guide to learn how Stat Server calculates it.

Stat Server Administration

This section provides information for administrators regarding Stat Server.

Installing a Stat Server Application

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

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

Installing Stat Server 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, OCC, and ORS) if the Extension installation packages were also deployed. Follow the installation steps (described below) for each Extension, starting with Step 2.

On Windows

Installing Stat Server 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 Starting and Stopping a Stat Server Application 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. 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, OCC, and ORS 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, OCC, and ORS Java

Extensions, which are delivered in five `.jar` files:

- eServiceContactStat.jar
- eServiceInteractionStat.jar
- eServiceSystemStat.jar
- OCCStatExtension.jar
- ORSStatExtension.jar

You deploy these files in three 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

Installing the eServices Extensions 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`, `eServiceInteractionStat.jar`, and `eServiceSystemStat.jar` files in the `\java\ext` subdirectory of your installed application.

On UNIX

Installing the eServices Extensions 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, eServiceInteractionStat.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

Installing the Outbound Contact Extension 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\StatServer\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

Installing the Outbound Contact Extension 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 the ORS Extension

You can install the `ORSStatExtension` Java Extension, which is used to monitor Orchestration Server performance, on Windows and/or UNIX platforms.

On Windows

Installing the Orchestration Server Extension On Windows

1. In the `\ext\ors\` 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\StatServer\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 `ORSStatExtension.jar` in the `\java\ext` subdirectory of your installed application.

On UNIX

Installing the Orchestration Server Extension On UNIX

1. On the *Real-Time Metrics Engine CD*, navigate to the `/ext/ors/` subdirectory.
2. Run the `install.sh` script from the command line by typing: `sh install.sh`.
3. When prompted, specify the full destination path, indicating the root folder of the Stat Server installation.
   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 `ORSStatExtension.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 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`.

Starting and Stopping a Stat Server Application

This topic 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 Configuring a Stat Server Application and Installing a Stat Server Application respectively.

What Must Be Running Prior to Start

You can start a Stat Server application in several ways. 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), 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)

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.

> **Important**
>
> 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.

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. 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 [-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.
- `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.

> **Important**
>
> 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 [-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.
- `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.

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

For example, to start a Stat Server application 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.)

> **Tip**
>
> 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`.

> **Tip**
> 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`.

Optimizing Performance

Review the recommendations provided in this topic to optimize Stat Server performance.

Hardware-Related Recommendations

- Consider the following formula, which approximates Stat Server memory, in megabytes, for a typical large contact center:
  `MemoryReqd = 100 + (NStatistics × 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):
  `NStatistics` = 28 statistics/report layout × 30,000 objects = 840,000 statistics
  `MemoryReqd` = 100 + (840,000 × 0.0012) = 1,108 MB
  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.

Software-Related Recommendations

- 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:
  - management-port
  - 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 one of the host on which Stat Server is deployed.
  - Set the ixn-id-in-status-table option 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 the debug-level configuration option. Specify only the debugging log level that you need.

Antivirus

Antivirus software can affect system performance and call response time, but it prevents viruses from hiding in software. Genesys recommends to keep antivirus software enabled on hosts where Stat Server is running.

Performance of all applications on a particular host should be analyzed. There might be more vulnerable than Stat Server applications. Some of those vulnerable applications can be considered to be moved to a different host. Genesys does not recommend excluding Stat Server from the antivirus scanning, but in case of a significant overhead, you can consider to disable the following:

- The folder in which Stat Server is running.
- Any logs folders.
- The `statserv.exe` process on Windows.
- The `statserv_32`, `statserv_64`, `statserv_32_unstripped`, or `statserv_64_unstripped` process on Linux.

The antivirus software must not restrict any ports that are used by the Genesys applications.

Application Files

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

- java
- sql_scripts

> **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.

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. 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.
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.
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

File Name	Description
	Windows operating systems.
StatServer.lms	File storing Stat Server–specific log messages.
java subfolder	Subfolder. See Contents of the java subfolder 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 Contents of the sql_scripts subfolder for the contents of each subfolder.

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.
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.

Contents of the java Subfolder

File Name	Description
ssjcldr.class	Java class loader; a member of the Stat Server Java

File Name	Description
	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 • ORSStatExtension.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_851.apd and Stat_Server_851.xml files.

Physical Data Models for Stat Server Tables

In addition to the information on this page, there is also information on the following:

- Table Schema by RDBMS
- Table and Column Descriptions

Stat Server stores data if the status-table, qinfo-table, login-table, and/or voice-reasons-table configuration options are enabled.

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 7.x 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 7.x uses this table and makes this data available for custom reporting.

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

> **Important**
>
> 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

This page provides Stat Server table schema for the supported RDBMSs.

> **Warning**
> 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.

Note: The IxnID and/or StartLocalTime and EndLocalTime fields might not be in the STATUS table, depending on the SQL script that was used for the table initialization.

DB2 Stat Server Database

Table Schema for a DB2 Stat Server Database

STATUS table	
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 table	
QueueDBID	INTEGER
ConnID	NUMERIC(20)
Status	INTEGER
StartTime	INTEGER
Duration	INTEGER

QINFO table	
EndTime	INTEGER

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

Microsoft SQL Stat Server Database

Table Schema for an Microsoft SQL Stat Server Database

STATUS table	
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 table	
QueueDBID	int
ConnID	numeric(20)
Status	int
StartTime	int
Duration	int
EndTime	int

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

Oracle Stat Server Database

Table Schema for an Oracle Stat Server Database

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

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

LOGIN table	
ID	NUMBER(20)

LOGIN table	
APP_DBID	INTEGER
SWITCHDBID	INTEGER
DNDBID	INTEGER
QUEUEDBID	INTEGER
AGENTDBID	INTEGER
PLACEDBID	INTEGER
Status	INTEGER
TIME	INTEGER
LOGINID	CHAR(255)

Table and Column Descriptions

The Stat Server database contains four tables:

<tabber>

LOGIN=

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

Field Name	Description
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`.
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.

Field Name	Description
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 (200 characters for DB2) , 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.

|-| QINFO=

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 below describes this table's fields, which are presented in their order of appearance.

Field Name	Description
QueueDBID	The queue's DBID.
ConnID	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.
Status	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: **Call Status** / **Code** / **Duration** Diverted from queue / 1 / Time in queue Abandoned within queue / 2 / Time in queue Diverted from / 3 / Time in queue

<table>
<tr><th>Field Name</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>queue (answered while ringing)</td><td></td><td>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 from a virtual queue (diverted to an agent's DN from another virtual queue). This status is based on the CallCleared retrospective, instantaneous action.</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>
<tr><td>Duration</td><td>The duration, in seconds, of the status displayed in the Status field.</td></tr>
<tr><td>EndTime</td><td>A sequence number representing the date and time when the status displayed in the Status field (of this table) ended. The sequence begins with</td></tr>
</table>

Field Name	Description
	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.

|-| STATUS=

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 and medias 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-voicerelated 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.

> ### Tip
> The `StartLocalTime`, `EndLocalTime`, and `IxnID` fields appear only if the appropriate script was run to create the STATUS table.

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

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).

<table>
<tr><th>Field Name</th><th>Description</th></tr>
<tr><td>Status</td><td>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:

<table><tr><td>4</td><td>WaitForNextCall (Ready)</td></tr><tr><td>5</td><td>OffHook</td></tr><tr><td>6</td><td>CallDialing</td></tr><tr><td>7</td><td>CallRinging</td></tr><tr><td>8</td><td>NotReadyForNextCall</td></tr><tr><td>9</td><td>AfterCallWork</td></tr><tr><td>13</td><td>CallOnHold</td></tr><tr><td>16</td><td>ASM_Engaged</td></tr><tr><td>17</td><td>ASM_Outbound</td></tr><tr><td>18</td><td>CallUnknown</td></tr><tr><td>19</td><td>CallConsult</td></tr><tr><td>20</td><td>CallInternal</td></tr><tr><td>21</td><td>CallOutbound</td></tr><tr><td>22</td><td>CallInbound</td></tr><tr><td>23</td><td>LoggedOut</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 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.</td></tr>
<tr><td>Duration</td><td>The duration, in seconds, of the status displayed in the Status field in this table.</td></tr>
<tr><td>EndTime</td><td>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-end-time-at-end-only set to yes), this field holds a 0 (zero) value if the status does not complete before the update of long-running statuses.</td></tr>
<tr><td>ConnID</td><td>An identification number that T-Server assigns to</td></tr>
</table>

Field Name	Description
	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`.

|-| VOICE_REASONS=

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.

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.

Determining the Purge Criteria

This topic 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.

> **Tip**
> 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.thetimenow.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.

Appendix A: StatFile

Starting with the 8.5.107 release, Stat Server logs operational statistics and the current execution context into a special `StatFile` log file.

The `StatFile` is configured in two places: the **[log]** section and the **[statserver]** section of the Stat Server application.

Parameters in the **[log]** section define the logging of the `StatFile` as a file (expiration, segment size, time format). See Framework Configuration Options Reference Manual for a complete description of configuration options in the **[log]** section.

The `StatFile` is generated when the `standard`, `interaction`, `trace`, `debug`, or `all` verbosity level is configured with at least one physical file (different from `stdout`, `stderr`, `network`, `memory`). The resultant physical file has a special suffix -1536 to distinguish `StatFile` from all other logs.

If **[log]**`/verbose=alarm`, the `StatFile` is NOT generated.

In the **[statserver]** section there are three new configuration options for the `StatFile`:

- stat-file-show-clients-list
- stat-file-show-options
- stat-file-timeout

> **Warning**
> The main goal of the `StatFile` is to provide information for Genesys engineers to improve problem troubleshooting/resolution. The structure of the `StatFile` can change without further notice.

Appendix B: Log Events

Overview

The following log events have been added since the most recent update to the Framework Combined Log Events Help. Refer to the Combined Log Events Help for information about the log events that do not appear in this appendix.

02-10067

Level	Standard
Text	Application failed to restore session with Config Server
Attributes	None
Description	An application failed to restore a session with Configuration Server.
Alarm Advisory	None
Actions	No action is required.

02-10068

Level	Standard
Text	Name uniqueness violation for <agent or place> 'name: string' (tenant 'name: string') (old dbid=<positive integer value>, new dbid=<positive integer value>)
Attributes	Object type: agent or place Object name Tenant name Old object DBID New object DBID
Description	Reports a situation when Stat Server detects the presence of two agent or place objects with the same type and name but different DBIDs.
Alarm Advisory	Remove duplicates in the Configuration Server database.

Actions	No action is required.

02-10069

Level	Standard
Text	Building virtual agent groups on tenant '[name1]', [Percentage] percent complete
Attributes	[name1] Name of a tenant as defined in the Configuration Server database [Percentage] Positive integer value in the range [0-100]
Description	Reports progress of building Virtual Agent Groups on a tenant during the Stat Server initialization.
Alarm Advisory	None
Actions	No action is required.

02-10070

Level	Standard
Text	Overload detected on '[thread]' ([Percentage] current CPU usage)
Attributes	[thread] Thread ID [Percentage] Positive integer value in the range [0-100]
Description	Stat Server switches from NORMAL to OVERLOAD state.
Alarm Advisory	None
Actions	No action is required.

02-10071

Level	Standard
Text	Overload ended on '[thread]' ([Percentage] current CPU usage)
Attributes	[thread]

	Thread ID [Percentage] Positive integer value in the range [0-100]
Description	Stat Server switches to NORMAL state.
Alarm Advisory	None
Actions	No action is required.

02-10072

Level	Standard
Text	Overload recovery started on '[thread]' ([Percentage] current CPU usage)
Attributes	[thread] Thread ID [Percentage] Positive integer value in the range [0-100]
Description	Stat Server switches from OVERLOAD to RECOVERY state.
Alarm Advisory	None
Actions	No action is required.

02-10073

Level	Standard
Text	Overload recovery failed on '[thread]' ([Percentage] current CPU usage)
Attributes	[thread] Thread ID [Percentage] Positive integer value in the range [0-100]
Description	Stat Server switches from RECOVERY to OVERLOAD state.
Alarm Advisory	None
Actions	No action is required.

02-10074

Level	Standard
Text	Overload protection on '[thread]' activated
Attributes	[thread] Thread ID
Description	Overload protection is activated.
Alarm Advisory	None
Actions	No action is required.

02-10075

Level	Standard
Text	Overload protection on '[thread]' deactivated
Attributes	[thread] Thread ID
Description	Overload protection is deactivated.
Alarm Advisory	None
Actions	No action is required.

02-10077

Level	Standard
Text	Budget model is enabled on tenant '[name]'
Attributes	[name] Name of a tenant as defined in the Configuration Server database
Description	The budget model is enabled on the specified Tenant.
Alarm Advisory	None
Actions	No action is required.

02-10078

Level	Standard
Text	Budget model is disabled on tenant '[name]'

Attributes	[name] Name of a tenant as defined in the Configuration Server database
Description	The budget model is disabled on the specified Tenant.
Alarm Advisory	None
Actions	No action is required.

02-10080

Level	Standard
Text	Invalid filter: '[name]' = '[value]'
Attributes	[name] Filter name [value] Filter value
Description	Invalid filter is detected.
Alarm Advisory	None
Actions	No action is required.

02-10081

Level	Standard
Text	Invalid time-range: '[name]' = '[value]'
Attributes	[name] Time-range name [value] Time-range value
Description	Invalid time-range is detected.
Alarm Advisory	None
Actions	No action is required.

02-10082

Level	Trace
Text	Large kv-list ([number] bytes) in CurrentState value

Attributes	[number] A number of bytes in the sent key-value list.
Description	Large key-value lists (over 65K) in the `CurrentState` statistic's value are detected.
Alarm Advisory	None
Actions	No action is required.

02-10084

Level	Standard
Text	Normal configuration monitoring mode started
Attributes	None
Description	Normal configuration monitoring mode is started.
Alarm Advisory	None
Actions	No action is required.

02-10085

Level	Standard
Text	Normal configuration monitoring mode stopped (duration='[number]' sec)
Attributes	[number] Duration (in seconds) of how long Stat Server was in that mode.
Description	Normal configuration monitoring mode is stopped.
Alarm Advisory	None
Actions	No action is required.

02-10086

Level	Standard
Text	Protected configuration monitoring mode started
Attributes	None
Description	Protected configuration monitoring mode is started.
Alarm Advisory	None
Actions	No action is required.

02-10087

Level	Standard
Text	Protected configuration monitoring mode stopped (duration='[number]' sec)
Attributes	[number] Duration (in seconds) of how long Stat Server was in that mode.
Description	Protected configuration monitoring mode is stopped.
Alarm Advisory	None
Actions	No action is required.

Appendix C: Configuration Options Template

Stat Server application template—Stat_Server_851.tpl in the templates subfolder—includes the following configuration options:

[statserver]
debug-level=Init,Client:6,Server,Action,Status,Mngmnt
DefaultAgentSPT=...
DefaultDNSPT=...
DefaultRPSPT=...
ignore-off-hook-on-position=no
management-port=3031
reconnect-timeout=10
reg-delay=3
status-table=off
local-time-in-status-table=off
qinfo-table=off
login-table=off
voice-reasons-table=off
time-format=%m/%d/%Y %H:%M:%S
auto-backup-interval=15
backup-file-name=ssbackup.000
check-stuck-calls=no
show-queued-interactions=no

[TimeRanges]
Range0-10=00-10
Range0-120=0-120
Range0-5=00-05

[TimeProfiles]
OneMinute,Growing = 0:00+0:01
OneHour,Growing = 0:00+1:00
SlidingDay,Sliding = 86400:3600
SlidingHour,Sliding = 3600:600

[log]
verbose=trace
all=stdout

[log-filter]
default-filter-type=skip

[common]
rebind-delay=0

[AbandCallsPercentage]
Category=RelativeNumberPercentage

MainMask=CallAbandoned
Objects=Queue, RoutePoint, GroupQueues
RelMask=CallDistributed, CallAbandoned
Subject=DNAction

[AverAbandCallTime]
Category=AverageTime
MainMask=CallAbandoned
Objects=Queue, RoutePoint, GroupQueues
RelMask=CallAbandoned
Subject=DNAction

[AverConsultDNStatusTime]
Category=AverageTime
MainMask=CallConsult
Objects=RegDN, Agent, Place, GroupAgents, GroupPlaces
RelMask=CallConsult
Subject=DNStatus

[AverConsultPlaceStatusTime]
Category=AverageTime
MainMask=CallConsult
Objects=Agent, Place, GroupAgents, GroupPlaces
RelMask=CallConsult
Subject=PlaceStatus

[AverConsultStatusTime]
Category=AverageTime
MainMask=CallConsult
Objects=Agent, Place, GroupAgents, GroupPlaces
RelMask=CallConsult
Subject=AgentStatus

[AverDistribCallTime]
Category=AverageTime
MainMask=CallDistributed
Objects=Queue, RoutePoint, GroupQueues
RelMask=CallDistributed
Subject=DNAction

[AverHandleDNStatusTime]
Category=AverageTime
MainMask=CallInbound, CallOutbound, OfflineWorkType1
Objects=RegDN, Agent, Place, GroupAgents, GroupPlaces
RelMask=CallInbound, CallOutbound
Subject=DNStatus

[AverHandlePlaceStatusTime]
Category=AverageTime
MainMask=CallInbound, CallOutbound, OfflineWorkType1
Objects=Agent, Place, GroupAgents, GroupPlaces
RelMask=CallInbound, CallOutbound
Subject=PlaceStatus

[AverHandleStatusTime]

Category=AverageTime
MainMask=CallInbound, CallOutbound, OfflineWorkType1
Objects=Agent, Place, GroupAgents, GroupPlaces
RelMask=CallInbound, CallOutbound
Subject=AgentStatus

[AverInboundDNStatusTime]
Category=AverageTime
MainMask=CallInbound
Objects=RegDN, Agent, Place, GroupAgents, GroupPlaces
RelMask=CallInbound
Subject=DNStatus

[AverInboundPlaceStatusTime]
Category=AverageTime
MainMask=CallInbound
Objects=Agent, Place, GroupAgents, GroupPlaces
RelMask=CallInbound
Subject=PlaceStatus

[AverInboundStatusTime]
Category=AverageTime
MainMask=CallInbound
Objects=Agent, Place, GroupAgents, GroupPlaces
RelMask=CallInbound
Subject=AgentStatus

[AverOutboundDNStatusTime]
Category=AverageTime
MainMask=CallOutbound
Objects=RegDN, Agent, Place, GroupAgents, GroupPlaces
RelMask=CallOutbound
Subject=DNStatus

[AverOutboundPlaceStatusTime]
Category=AverageTime
MainMask=CallOutbound
Objects=Agent, Place, GroupAgents, GroupPlaces
RelMask=CallOutbound
Subject=PlaceStatus

[AverOutboundStatusTime]
Category=AverageTime
MainMask=CallOutbound
Objects=Agent, Place, GroupAgents, GroupPlaces
RelMask=CallOutbound
Subject=AgentStatus

[CurrentAgentState]
Category=CurrentState
MainMask=*
Objects=Agent
Subject=AgentStatus

[CurrentDNState]

Category=CurrentState
MainMask=*
Objects=RegDN
Subject=DNStatus

[CurrentGroupState]
Category=CurrentState
MainMask=*
Objects=GroupAgents, GroupPlaces
Subject=GroupStatus

[CurrentPlaceState]
Category=CurrentState
MainMask=*
Objects=Place
Subject=PlaceStatus

[CurrMaxCallWaitingTime]
Category=CurrentMaxTime
MainMask=CallWait
Objects=Queue, RoutePoint, GroupQueues
Subject=DNAction

[CurrNumberACWStatuses]
Category=CurrentNumber
MainMask=AfterCallWork
Objects=GroupAgents, GroupPlaces
Subject=AgentStatus

[CurrNumberConsultStatuses]
Category=CurrentNumber
MainMask=CallConsult
Objects=GroupAgents, GroupPlaces
Subject=AgentStatus

[CurrNumberDialingStatuses]
Category=CurrentNumber
MainMask=CallDialing
Objects=GroupAgents, GroupPlaces
Subject=AgentStatus

[CurrNumberHoldStatuses]
Category=CurrentNumber
MainMask=CallOnHold
Objects=GroupAgents, GroupPlaces
Subject=AgentStatus

[CurrNumberInboundStatuses]
Category=CurrentNumber
MainMask=CallInbound
Objects=GroupAgents, GroupPlaces
Subject=AgentStatus

[CurrNumberInternalStatuses]
Category=CurrentNumber

MainMask=CallInternal
Objects=GroupAgents, GroupPlaces
Subject=AgentStatus

[CurrNumberNotReadyStatuses]
Category=CurrentNumber
MainMask=NotReadyForNextCall
Objects=GroupAgents, GroupPlaces
Subject=AgentStatus

[CurrNumberOutboundStatuses]
Category=CurrentNumber
MainMask=CallOutbound
Objects=GroupAgents, GroupPlaces
Subject=AgentStatus

[CurrNumberRingingStatuses]
Category=CurrentNumber
MainMask=CallRinging
Objects=GroupAgents, GroupPlaces
Subject=AgentStatus

[CurrNumberWaitingCalls]
Category=CurrentNumber
MainMask=CallWait
Objects=Queue, RoutePoint, GroupQueues
Subject=DNAction

[CurrNumberWaitStatuses]
Category=CurrentNumber
MainMask=WaitForNextCall
Objects=GroupAgents, GroupPlaces
Subject=AgentStatus

[DistribCallsPercentage]
Category=RelativeNumberPercentage
MainMask=CallDistributed
Objects=Queue, RoutePoint, GroupQueues
RelMask=CallDistributed, CallAbandoned
Subject=DNAction

[ServiceFactor]
Category=TotalNumberInTimeRangePercentage
MainMask=CallDistributed
Objects=Queue, RoutePoint, GroupQueues
Subject=DNAction

[TotalAfterCallWorkDNStatusTime]
Category=TotalTime
MainMask=OfflineWorkType1
Objects=RegDN, Agent, Place, GroupAgents, GroupPlaces
Subject=DNStatus

[TotalAfterCallWorkPlaceStatusTime]
Category=TotalTime

MainMask=OfflineWorkType1
Objects=Agent, Place, GroupAgents, GroupPlaces
Subject=PlaceStatus

[TotalAfterCallWorkStatusTime]
Category=TotalTime
MainMask=OfflineWorkType1
Objects=Agent, Place, GroupAgents, GroupPlaces
Subject=AgentStatus

[TotalLoginTime]
Category=TotalTime
MainMask=*, ~LoggedOut,~NotMonitored
Objects=Agent, GroupAgents
Subject=AgentStatus

[TotalNotReadyDNStatusTime]
Category=TotalTime
MainMask=NotReadyForNextCall
Objects=RegDN, Agent, Place, GroupAgents, GroupPlaces
Subject=DNStatus

[TotalNotReadyPlaceStatusTime]
Category=TotalTime
MainMask=NotReadyForNextCall
Objects=Agent, Place, GroupAgents, GroupPlaces
Subject=PlaceStatus

[TotalNotReadyStatusTime]
Category=TotalTime
MainMask=NotReadyForNextCall
Objects=Agent, Place, GroupAgents, GroupPlaces
Subject=AgentStatus

[TotalNumberCallsAband]
Category=TotalNumber
MainMask=CallAbandoned
Objects=Queue, RoutePoint, GroupQueues
Subject=DNAction

[TotalNumberCallsDistrib]
Category=TotalNumber
MainMask=CallDistributed
Objects=Queue, RoutePoint, GroupQueues
Subject=DNAction

[TotalNumberConsultCalls]
Category=TotalNumber
MainMask=CallConsult
Objects=RegDN, Agent, Place, GroupAgents, GroupPlaces
Subject=DNAction

[TotalNumberInboundCalls]
Category=TotalNumber
MainMask=CallInbound

Objects=RegDN, Agent, Place, GroupAgents, GroupPlaces
Subject=DNAction

[TotalNumberInternalCalls]
Category=TotalNumber
MainMask=CallInternal
Objects=RegDN, Agent, Place, GroupAgents, GroupPlaces
Subject=DNAction

[TotalNumberOutboundCalls]
Category=TotalNumber
MainMask=CallOutbound
Objects=RegDN, Agent, Place, GroupAgents, GroupPlaces
Subject=DNAction

[TotalTalkDNStatusTime]
Category=TotalTime
MainMask=CallInbound, CallOutbound, CallInternal, CallConsult, CallUnknown
Objects=RegDN, Agent, Place, GroupAgents, GroupPlaces
Subject=DNStatus

[TotalTalkPlaceStatusTime]
Category=TotalTime
MainMask=CallInbound, CallOutbound, CallInternal, CallConsult, CallUnknown
Objects=Agent, Place, GroupAgents, GroupPlaces
Subject=PlaceStatus

[TotalTalkStatusTime]
Category=TotalTime
MainMask=CallInbound, CallOutbound, CallInternal, CallConsult, CallUnknown
Objects=Agent, Place, GroupAgents, GroupPlaces
Subject=AgentStatus

[IxnQ_Cleared]
Category=TotalNumber
MainMask=InteractionCleared
Objects=StagingArea
Subject=DNAction

[IxnQ_Cleared_Time]
Category=TotalTime
MainMask=InteractionCleared
Objects=StagingArea
Subject=DNAction

[IxnQ_Created]
Category=TotalNumber
MainMask=InteractionCreated
Objects=StagingArea
Subject=DNAction

[IxnQ_Deleted]
Category=TotalNumber
MainMask=InteractionDeleted
Objects=StagingArea

Subject=DNAction

[IxnQ_Deleted_Time]
Category=TotalTime
MainMask=InteractionDeleted
Objects=StagingArea
Subject=DNAction

[IxnQ_Distributed]
Category=TotalNumber
MainMask=InteractionDistributed
Objects=StagingArea
Subject=DNAction

[IxnQ_Distributed_Time]
Category=TotalTime
MainMask=InteractionDistributed
Objects=StagingArea
Subject=DNAction

[IxnQ_Distributed_To_Queue]
Category=TotalNumber
MainMask=InteractionDistributedToQueue
Objects=StagingArea
Subject=DNAction

[IxnQ_Entered]
Category=TotalNumber
MainMask=InteractionEntered
Objects=StagingArea
Subject=DNAction

[IxnQ_InteractionAbandonedDuringOffering]
Category=TotalNumber
MainMask=InteractionAbandonedDuringOffering
Objects=StagingArea
Subject=DNAction

[IxnQ_InteractionAbandonedDuringOffering_Time]
Category=TotalTime
MainMask=InteractionAbandonedDuringOffering
Objects=StagingArea
Subject=DNAction

[IxnQ_InteractionAccepted]
Category=TotalNumber
MainMask=InteractionAccepted
Objects=StagingArea
Subject=DNAction [IxnQ_InteractionAccepted_Time]
Category=TotalTime
MainMask=InteractionAccepted
Objects=StagingArea
Subject=DNAction

[IxnQ_InteractionAnswered]

Category=TotalNumber
MainMask=InteractionAnswered
Objects=StagingArea
Subject=DNAction

[IxnQ_InteractionAnswered_Time]
Category=TotalTime
MainMask=InteractionAnswered
Objects=StagingArea
Subject=DNAction

[IxnQ_InteractionReleased]
Category=TotalNumber
MainMask=InteractionReleased
Objects=StagingArea
Subject=DNAction

[IxnQ_InteractionReleased_Time]
Category=TotalTime
MainMask=InteractionReleased
Objects=StagingArea
Subject=DNAction