Framework 8.1

T-Server for Cisco Unified Communications Manager

Deployment Guide

About Genesys

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

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

Notice

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

Your Responsibility for Your System's Security

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

Trademarks

Genesys and the Genesys logo are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other company names and logos may be trademarks or registered trademarks of their respective holders.

The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from VARs

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

Customer Care from Genesys

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

Ordering and Licensing Information

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

Released by

Genesys Telecommunications Laboratories, Inc. www.genesys.com

Document Version: 81fr_dep-ts_cisco_09-2016_v8.1.203.00

Table of Contents

List of Procedures

Preface

Welcome to the *Framework 8.1 T-Server for Cisco Unified Communications Manager Deployment Guide*. This document introduces you to the concepts, terminology, and procedures relevant to T-Servers® in general and provides detailed reference information about T-Server for Cisco Unified Communications Manager. The reference information includes, but is not limited to, configuration options, limitations, and switch-specific functionality. You must configure the configuration objects and options described in this document in the Framework Configuration Layer.

Use this document only after you have read through the *Framework 8.1 Deployment Guide*, and the Release Note for your T-Server.

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

This preface contains the following sections:

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on page 295.

About T-Server for Cisco Unified Communications Manager

T-Server is the Genesys software component that provides an interface between your telephony hardware and the rest of the Genesys software components in your enterprise. It translates and keeps track of events and requests that come from, and are sent to, the CTI (computer-telephony

integration) link in the telephony device. T-Server is a TCP/IP-based server that can also act as a messaging interface between T-Server clients. It is the critical point in allowing your Genesys solution to facilitate and track the contacts that flow through your enterprise.

Note that the T-Server name has changed over the course of previous releases for various reasons (including, but not limited to, changes in vendor name or in Genesys policy). The former names include:

- T-Server for Cisco CallManager.

The current name is T-Server for Cisco Unified Communications Manager.

Intended Audience

This guide is intended primarily for system administrators, both those who are new to T-Server and those who are familiar with it.

- If you are new to T-Server, read the *Framework 8.1 Deployment Guide* and the Release Note mentioned earlier, and then read all of the sections of this document that apply to your software and its accompanying components. Refer back to the *Framework 8.1 Deployment Guide* as needed.
- If you are an experienced T-Server user—someone with computer expertise, who is used to installing, configuring, testing, or maintaining Genesys software—you may find it more time efficient to go to the Index to see what is new or different in T-Server release 8.1. If you take that approach, please also read Release Notes and refer to other related resources, such as the *Genesys Events and Models Reference Manual* and *Platform SDK 8.x .NET (or Java) API Reference* for technical details of T-Library functions.

In general, this document assumes that you have a basic understanding of, and familiarity with:

- Computer-telephony integration concepts, processes, terminology, and applications.
- Network design and operation.
- Your own network configurations.
- Your telephony hardware and software.
- Genesys Framework architecture and functions.
- Configuration Manager interface and object-managing operations.

Based on your specific contact center environment and your responsibilities in it, you may need to be familiar with a much wider range of issues as you deploy T-Server.

Reading Prerequisites

You must read the *Framework 8.1 Deployment Guide* before using this *T-Server Deployment Guide*. That book contains information about the Genesys software you must deploy before deploying T-Server.

Making Comments on This Document

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

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

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

Contacting Genesys Customer Care

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

Before contacting Customer Care, please refer to the *Genesys Care Support Guide for On-Premises* for complete contact information and procedures.

Document Change History

This section lists topics that are new or that have changed significantly since the first release of this document.

New in Document Version 8.1.203.00

The following topics have been added or changed since the previous release of this document:

- Added "Call Participant Info" on page 143.
- Added "Providing AttributeDNIS in EventDialing" on page 161.
- Added the following configuration options:
  - clean-calls-on-all-links-up
  - delay-dialing

  - java-home
  - party-changed-from-external-release
  - recording-filename-pop
  - reg-failed-delay
  - reg-failed-retries
  - use-external-establish-from-other-link
  - use-ringing-for-net-alerting
- Updated the following configuration options:
  - recording-filename
  - TraceFileSize

New in Document Version 8.1.202.00

The following topics have been added or changed since the previous release of this document:

- Added "Transport Layer Security" on page 174.
- Added support of Hunt Groups in Broadcast mode (parallel ringing). See "Hunt Groups" on page 153 for details.
- Added "Agent State for Out-Of-Service Agent DNs" on page 140.
- Added "Disabling the Default MOH Treatment" on page 157.
- Updated "Whisper Coaching and Extra Instance of Intercom Call" on page 173.
- Added the following configuration options:
  - force-moh-on-ms-down
  - moh-server-music (DN level)
  - password
  - tls-cert-path
  - tls-capf-host
  - tls-capf-port
  - tls-tftp-host
  - tls-tftp-port
  - tls-instance-id
  - tls-auth-code
  - out-of-service-action
  - out-of-service-action-delay
- Removed options:
  - enable-pickup-jtapi-workaround

New in Document Version 8.1.201.00

The following topics have been added or changed since the previous release of this document:

- Added "ANI Modification For Outbound Calls" on page 140.
- Added "Do Not Disturb" on page 151.
- Added the `free-form-terminal-id` option.

For configuration options additions, see "Changes from 8.0 to 8.1" on page 289.

New in Document Version 8.1.1

The following topics have been added or changed since the previous release of this document:

- Updated Table 10, "T-Server Device and CUCM DN Types," on page 135.
- Added "Hunt Groups" on page 153.
- Added "Single-Step Conference" on page 168.
- Added Chapter 13, "Integration with Genesys Media Server" on page 197.
- Added Chapter 14, "SIP Server In Front Deployment" on page 201.

Genesys

Part

1 T-Server Deployment

Part One of this *T-Server Deployment Guide* familiarizes the reader with T-Server in general. It addresses architectural, functional, and procedural information common to all T-Servers.

The information in Part One is divided into the following chapters:

- Chapter 1, "T-Server Fundamentals," on page 19, describes T-Server, its place in the Framework 8 architecture, T-Server redundancy, and multi-site issues. It stops short of providing configuration and installation information.
- Chapter 2, "T-Server General Deployment," on page 33, presents configuration and installation procedures for all T-Servers.
- Chapter 3, "High-Availability Deployment," on page 47, addresses high availability (HA).
- Chapter 4, "Multi-Site Support," on page 59, details the variations available for T-Server implementations across geographical locations.
- Chapter 5, "Starting and Stopping T-Server Components," on page 117, describes how, and in what order, to start up T-Server among other Framework components. It also provides possible stopping commands.

New for All T-Servers in 8.1

Before looking at T-Server's place in Genesys solutions and in the architecture of the Genesys Framework, note the following general changes that have been implemented in the 8.1 release of T-Server:

- T-Server no longer connects to applications that have disabled status in the configuration environment.
- The default value of the `background-processing` configuration option has been changed to `true`. See "background-processing" on page 230 for details.

- T-Server now supports the Unresponsive Process Detection feature. The following configuration options enable this feature:
  - "heartbeat-period" on page 225
  - "hangup-restart" on page 226

  For more information, refer to the *Framework 8.1 Management Layer User's Guide.*
- T-Server now supports IPv6. For more information, refer to the *Framework 8.1 Deployment Guide.*
- T-Server now supports vSphere 4 Hypervisor.
- T-Server now supports Acresso FLEXNet Publisher v11.9 license manager.

Notes:
- Configuration option changes common to all T-Servers are described in "Changes from Release 8.0 to 8.1" on page 255.
- For information about the new features that are available in your T-Server in the initial 8.1 release, see Part Two of this document.

Genesys

Chapter

1 T-Server Fundamentals

This chapter provides general information about T-Server features and functionality and about its configuration and installation. For reference information about your specific T-Server and about options for all T-Servers, see "Part Two: Reference Information."

This chapter has various levels of information, some of it intended for people who have configured, installed, and used previous releases of T-Server, and some of it aimed at those less familiar with such T-Server operations. That means some sections will not necessarily be relevant for you.

- If you are an experienced user of T-Server, start with "New for All T-Servers in 8.1" on page 17, and then move to the chapters comprising Part Two of this document, where specific information about your T-Server is available.
- If you are new to T-Server, begin with "Learning About T-Server." Once you have read through that and subsequent sections, you are ready for the other chapters in Part One that go into detail about T-Server configuration and installation.

Generally, this chapter presents overview information that applies to all T-Servers (and Network T-Servers) and their deployment. This chapter is divided into the following sections:

Learning About T-Server

The *Framework 8.1 Deployment Guide* provides you with a high-level introduction to the role that T-Server plays in the Genesys Framework. If you have already looked through that guide, you may recall that T-Server is the most important component of the Framework Media Layer (the other two components are Load Distribution Server (LDS) and HA Proxy). The Media Layer enables Genesys solutions to communicate with various media, including traditional telephony systems, voice over IP (VoIP), e-mail, and the Web. This layer also provides the mechanism for distributing interaction-related business data, also referred to as *attached data,* within and across solutions.

Framework and Media Layer Architecture

Figure 1 illustrates the position Framework holds in a Genesys solution.

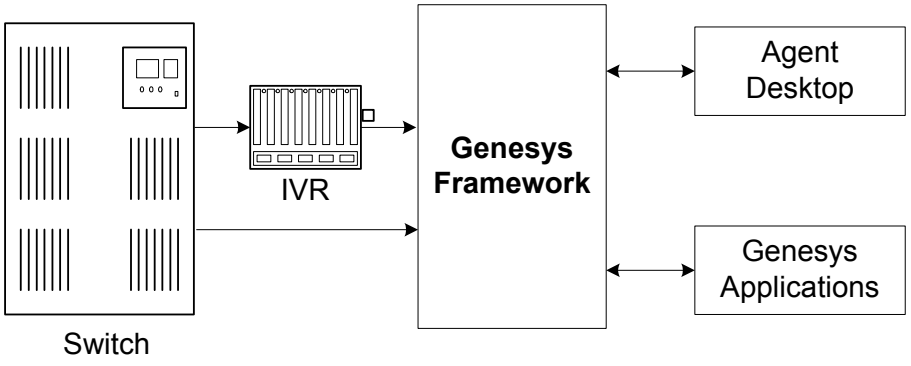

Figure 1: Framework in a Genesys Solution

Moving a bit deeper, Figure 2 presents the various layers of the Framework architecture.

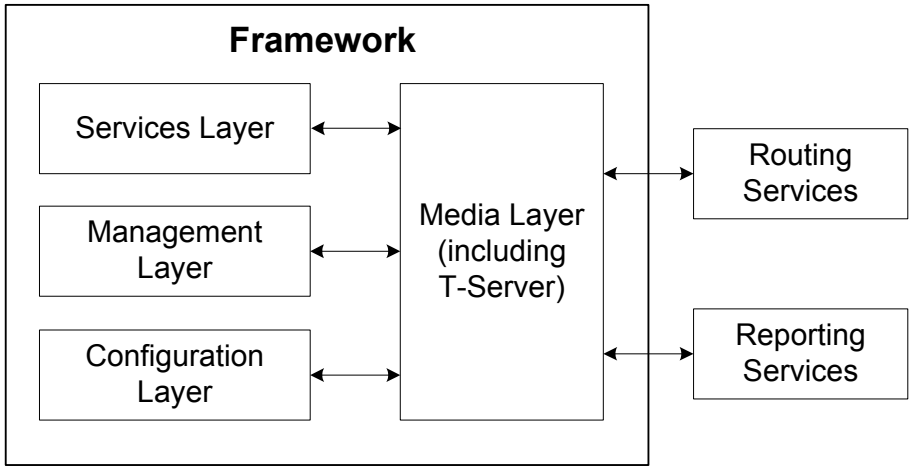

Figure 2: The Media Layer in the Framework Architecture

T-Server is the heart of the Media Layer—translating the information of the media-device realm into information that Genesys solutions can use. It enables your contact center to handle the computer-based form of the interactions that arrive and it translates the information surrounding a customer contact into reportable and actionable data.

Figure 3 presents the generalized architecture of the Media Layer.

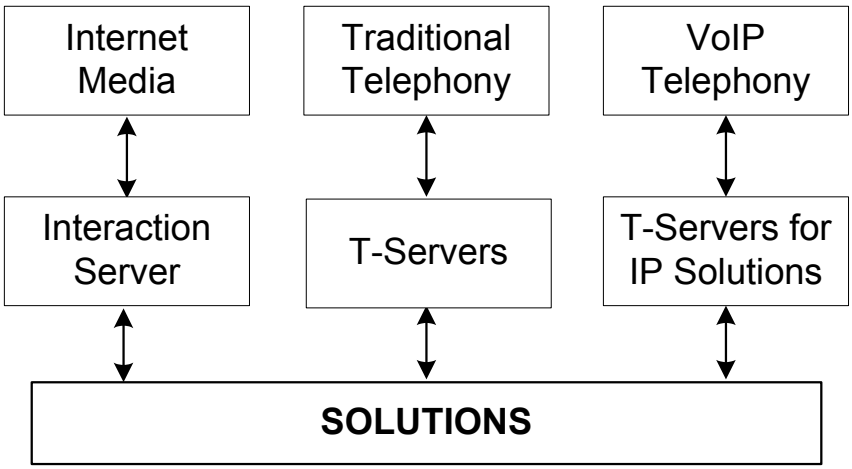

Figure 3: Media Layer Architecture

In addition to being the most important component of the Media Layer, T-Server plays the most significant role in making information about telephony traffic and its data available to Framework as a whole.

One or more components in practically every solution are T-Server clients. Solutions comprise a number of different Genesys software packages, from collections of components for various types of routing to those that allow for outbound dialing to still others. Framework in general, and T-Server in particular, enable these solutions to function in your enterprise.

T-Server has several typical clients: Stat Server, Interaction Concentrator, Universal Routing Server, and agent desktop applications. T-Server gets the information it needs about the enterprise from Configuration Server. Additionally, if you use the Management Layer, T-Server provides its ongoing status and various other log messages to server components of the Management Layer (for instance, allowing you to set alarms).

T-Server Requests and Events

This section outlines the roles that T-Server plays in a contact center. While it is possible to describe roles for all T-Servers, at a detailed level, T-Server's functionality depends on the hardware to which it is connected. (For example, when connected to a traditional switch, it performs CTI functions, but when connected to a VOIP-based telephony device, it controls IP traffic.) The CTI connection is only for the switch.

Details of T-Server Functionality

T-Server is a TCP/IP server that enables intelligent communication between media-specific protocols (such as the various CTI protocols, including CSTA and ASAI) and TCP/IP-based clients of T-Server. Applications that are clients to T-Server use the T-Library format to transmit requests to T-Server through a TCP/IP socket. T-Server can then either translate those requests to CTI protocol for switch use or relay them directly to other TCP/IP clients.

T-Server performs three general functions in the contact center: Bridging, Messaging, and Interaction Tracking.

Bridging

T-Server acts as a platform-independent interface between media devices and business applications. In the case of a telephony device, for instance, it receives messages from and sends commands to the telephony equipment using either CTI links provided by the switch manufacturer or interface protocols provided by telephony network vendors.

On the client-application end, T-Server offers three models (call model, agent model, and device model) unified for all switches. The core functionality (such as processing an inbound call, an agent login, or a call-forwarding request) translates into a unified application programming interface (API) called T-Library, so that applications do not need to know what specific switch model they are dealing with. On the other hand, T-Library accommodates many

functions that are unique to a specific switch, so that client applications are able to derive the maximum functionality offered by a particular switch.

Refer to the *Genesys Events and Models Reference Manual* for complete information on all T-Server events and call models and to the `TServer.Requests` section of the *Platform SDK 8.x .NET* (or *Java*) *API Reference* for technical details of T-Library functions.

Messaging

In addition to translating requests and events for the client application involved in an interaction, T-Server:

- Provides a subscription mechanism that applications can use to receive notifications about interaction-related and non-interaction-related events within the contact center.
- Broadcasts messages of major importance (such as a notification that the link is down) to all clients.
- Broadcasts messages originated by a T-Server client to other T-Server clients.

The subscription mechanism consists of two parts, the DN subscription and event-type masking. Applications must register for a DN or a set of DNs to receive notifications about all events that occur in association with each registered DN. For example, when two softphone applications are registered for the same DN, and the first application initiates a call from the DN, T-Server notifies both applications that the call is initiated from the DN.

Client applications can also specify one or more types of events, and T-Server will filter out events of the non-specified types and only send events of the requested types. For example, if agent supervisors are interested in receiving agent-related events, such as `AgentLogin` and `AgentLogout`, they have to mask `EventAgentLogin` and `EventAgentLogout`, provided that a particular T-Server supports these events.

The combination of each client's subscription for DNs and masking of event types defines what messages T-Server distributes to what client.

Interaction Tracking

T-Server maintains call information for the life of the call (or other T-Server-supported media type) and enables client applications to attach user data to the call. Call information includes:

- A unique identifier, connection ID, that T-Server assigns when creating the call.
- Automatic Number Identification (`ANI`) and Dialed Number Identification Service (`DNIS`), if reported by the CTI link.
- User data that a client application (such as an Interactive Voice Response unit or Genesys Universal Routing Server) provides.

Difference and Likeness Across T-Servers

Although Figure 3 on page 21 (and other figures) depicts T-Server that works with telephony systems as a single product, this is a simplification. Because almost every traditional telephony device has its own characteristics and communication protocols, Genesys makes different T-Servers for different telephony systems. (That means your T-Server will not work with another switch.) Thus, all T-Servers play a common role in the architecture, but their specific features differ from implementation to implementation, based on the media device in use.

Despite their switch-based differences, T-Servers for telephony systems are similar to one another in at least one important respect: they are all built with a certain amount of shared software code. This shared code is rolled into a single unit and is called T-Server Common Part (TSCP). TSCP is the central, common component for all T-Servers and has its own Release Note, which is accessible via a hyperlink from your T-Server's Release Note.

Note: This document separates common-code features based on TSCP into separate sections and chapters, such as the "T-Server Common Configuration Options" chapter. These are the options for all T-Servers that TSCP makes available for configuration.

T-Server Functional Steps During a Sample Call

The following example, Figure 4, outlines some basic steps that T-Server might take when a call arrives from outside the contact center. In this scenario, T-Server starts tracking the call even before it is delivered to the agent. T-Server then informs the selected agent that a call has arrived. When the switch delivers the call to the agent's extension, T-Server presents account information, collected at an Interactive Voice Response (IVR) unit, to the agent at the agent desktop application.

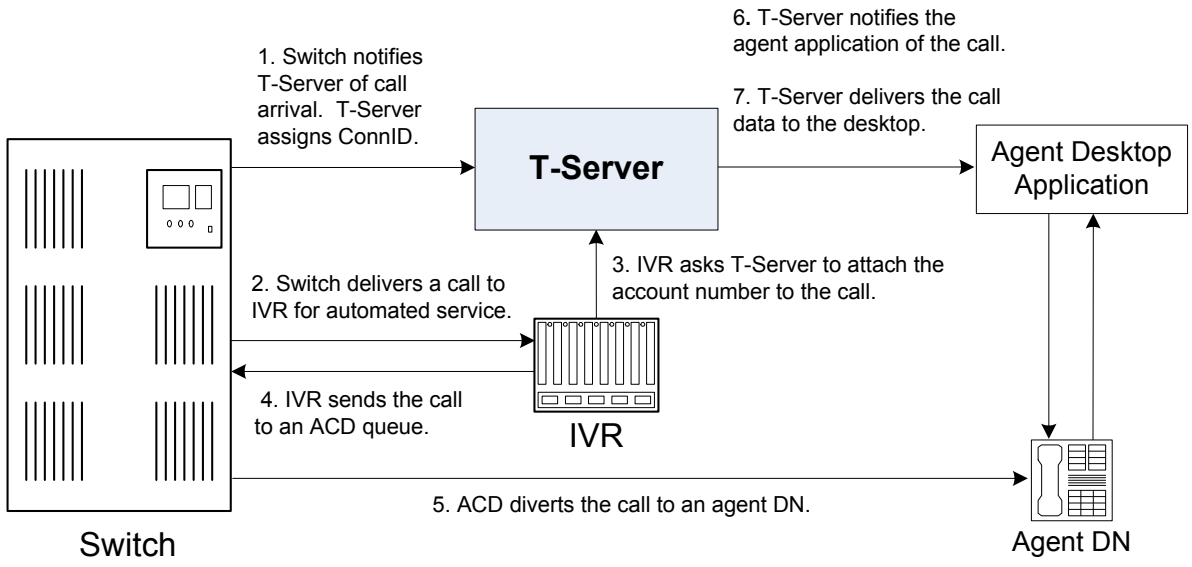

Figure 4: Functional T-Server Steps

Step 1

When the call arrives at the switch, T-Server creates a call in its internal structure. T-Server assigns the call a unique identifier, connection ID.

Step 2

The switch delivers the call to an Interactive Voice Response (IVR) unit, which begins automated interactions with the caller.

Step 3

IVR acquires user information from the caller through prompts and requests T-Server to attach that information to the call. T-Server updates the call with the user information.

Step 4

IVR sends the call to an ACD (Automated Call Distribution) queue.

Step 5

The ACD unit distributes the call to an available agent logged in to a particular DN (directory number).

Step 6

T-Server notifies the agent desktop application that the call is ringing on the agent DN. The notification event contains call data including `ANI`, `DNIS`, and account information that the IVR has collected.

Step 7

The agent desktop application presents the account information, including the name of the person whose account this is, on the agent's screen, so that the agent answering the call has all the relevant information.

These seven steps illustrate just a small part of T-Server's bridging, messaging, and interaction-processing capabilities.

Advanced Disconnect Detection Protocol

Since the 6.0 release of T-Server, the Advanced Disconnect Detection Protocol (ADDP) has replaced the Keep-Alive Protocol (KPL) as the method to detect failures for certain T-Server connections, including connections between two T-Servers and between a T-Server and its clients.

Notes: Starting with release 7.5, the KPL backward-compatibility feature is no longer supported.

ADDP applies only to connections between Genesys software components.

With ADDP, protocol activation and initialization is made on the client's side and you can change these parameters. No additional messages are sent when there is existing activity over the connection. T-Server client applications and the remote T-Server (if any) must be listening to the socket and respond promptly to the polling signal for the connection to be preserved.

If you are going to enable ADDP, you must do it using the `protocol`, `addp-timeout`, `addp-remote-timeout`, and `addp-trace` configuration options. When configuring a timeout, consider the following issues:

- The configured timeout must be at least twice as long as the maximum network latency.
- There may be an interval when T-Server does not check for network activity.
- If the link connection fails but the client is not notified (for example, because the host is turned off, or because a network cable is unplugged), the maximum reaction time to a link-connection failure is equal to double the configured timeout plus the established network latency.

Also keep in mind that the T-Server receiving the polling signal may not respond immediately, and that a delay occurs after the polling signal, while the response travels from one T-Server to another. If you do not account for these contingencies when configuring a timeout, the connection that ADDP is monitoring will be dropped periodically.

Redundant T-Servers

T-Servers can operate in a high-availability (HA) configuration, providing you with redundant systems. The basics of each T-Server's redundant capabilities differ from T-Server to T-Server. One basic principle of redundant T-Servers is the standby redundancy type, which dictates how quickly a backup T-Server steps in when the primary T-Server goes down.

The Framework Management Layer currently supports two types of redundant configurations: `warm standby` and `hot standby`. All T-Servers offer the `warm standby` redundancy type and, starting with release 7.1, the `hot standby` redundancy type is implemented in T-Servers for most types of switches. (See Table 1.)

Instructions for configuring T-Server redundancy are available in Chapter 3, "High-Availability Configuration and Installation." Specifics on your T-Server's HA capabilities are outlined in Part Two of this document.

Note: IVR Server and some Network T-Servers can be configured for load sharing or `warm` or `hot standby`; however, they do not support any combination of these redundancy types. Details of your component's HA capabilities are discussed in Part Two of this document.

Support for Hot Standby Redundancy in Various T-Servers

Use Table 1 to determine whether your T-Server supports the `hot standby` redundancy type. The table also indicates whether HA Proxy components are required for this support, and, if so, how many are required per pair of redundant T-Servers (or per link, if so noted).

Table 1 only summarizes `hot standby` redundancy support in various T-Servers. For detailed, up-to-date information on the subject, see the *Genesys Supported Operating Environment Reference Guide* located on the Genesys Care website.

Table 1: T-Server Support of the Hot Standby Redundancy Type

T-Server Type	Hot Standby Supported	HA Proxy Required	Number of HA Proxy Components
Aastra MXONE CSTA I	Yes	No	—
Alcatel A4200/OXO	Yes	No	—
Alcatel A4400/OXE	Yes	No	—
Aspect ACD	Yes	No	—
Avaya Communication Manager	Yes	No[a]	—
Avaya INDeX	Yes	No	—
Avaya TSAPI	Yes	No	—
Cisco UCCE	Yes	No	—
Cisco Unified Communications Manager	Yes	No	—
DataVoice Dharma	Yes	No	—
Digitro AXS/20	Yes	No	—
EADS Intecom M6880	Yes	No	—
EADS Telecom M6500	Yes	No	—
eOn eQueue	Yes	No	—
Fujitsu F9600	Yes	No	—
Huawei C&C08	Yes	No	—
Huawei NGN	Yes	No	—
Mitel MiTAI	Yes	No	—
NEC NEAX/APEX	Yes	No	—
Nortel Communication Server 2000/2100	Yes	Yes[b], No[c]	1 per link
Nortel Communication Server 1000 with SCCS/MLS	Yes	No	—
Philips Sopho iS3000	Yes	No[d]	1
Radvision iContact	No	—	—
Samsung IP-PCX IAP	Yes	No	—

Table 1: T-Server Support of the Hot Standby Redundancy Type (Continued)

T-Server Type	Hot Standby Supported	HA Proxy Required	Number of HA Proxy Components
Siemens Hicom 300/HiPath 4000 CSTA I	Yes	No	—
Siemens HiPath 3000	Yes	No	—
Siemens HiPath 4000 CSTA III	Yes	No	—
Siemens HiPath DX	Yes	No	—
SIP Server	Yes	No	—
Spectrum	Yes	No	—
Tadiran Coral	Yes	No	—
Teltronics 20-20	Yes	Yes	1
Tenovis Integral 33/55	Yes	No	—
Network T-Servers[e]			
AT&T	No	—	—
Concert	No	—	—
CRSP	No	—	—
DTAG	No	—	—
GenSpec	No	—	—
ISCP	No	—	—
IVR Server, using network configuration	Yes	—	—
KPN	No	—	—
MCI	No	—	—
NGSN	No	—	—
Network SIP Server	No	—	—
Sprint	No	—	—
SR3511	No	—	—
Stentor	No	—	—

a. With release 7.1, T-Server for Avaya Communication Manager no longer uses HA Proxy for its support of `hot standby`. Earlier releases of this T-Server require two HA Proxies to support `hot standby`.

b. For T-Server for Nortel Communication Server 2000/2100 in high-availability (`hot standby`) configuration, Genesys recommends that you use link version SCAI14 or above with call-progress and noncontroller-released messages enabled. See the switch-specific information in Part 2 of this *Deployment Guide* for additional information on HA configurations.

c. Starting with release 7.5, T-Server for Nortel Communication Server 2000/2100 supports HA without HA Proxy when operating in Dual CTI Links mode. See the switch-specific information in Part 2 of this *Deployment Guide* for additional information on HA configurations.

d. Starting with release 6.5.3, T-Server for Philips Sopho iS3000 supports HA both with and without HA Proxy.

e. Although they do not support high availability per se, Network T-Servers do support a load-sharing schema.

Multi-Site Support

Multi-site configuration implies the existence of two or more switches that belong to the same enterprise or service provider, and that share the Genesys Configuration Database. (In some cases this may include isolated partitions on a given switch served by different T-Servers.) The main goal of T-Server support for multi-site operations is to maintain critical information about a call as it travels from one switch to another.

For instructions on installing and configuring a multi-site environment, including information on the Inter Server Call Control (ISCC) features, please see Chapter 4, “Multi-Site Support,” on page 59.

Agent Reservation

T-Server provides support for clients to invoke the agent reservation function, `TReserveAgent()`. This function allows a server application that is a client of T-Server to reserve a DN along with an agent, a `Place`, or both, so that no other T-Server client can route calls to it during a specified reservation interval. Alternatively, when clients use the ISCC feature (see “ISCC Call Data Transfer Service” on page 61), they can use an agent reservation embedded in an ISCC request. (To do so, clients have to specify a certain `Extensions` attribute in an ISCC request when initiating an ISCC transaction. See page 68 for the list of ISCC requests.)

The reservation does not currently prevent the reserved objects from receiving direct calls or calls distributed from ACD Queues; agent reservation is intended as a way of synchronizing the operation of several clients. See `RequestReserveAgent` in the *Platform SDK 8.x .NET (or Java) API Reference* for more details on this function from the client’s point of view.

In addition to invoking the `TReserveAgent` function, you can customize the Agent Reservation feature by configuring options in the T-Server `Application` object. See the "agent-reservation Section" on page 238 in the "T-Server Common Configuration Options" chapter in Part Two for more details.

Starting with version 8.1, T-Server supports Agent Reservation failure optimization, to ensure that only agent reservation requests of the highest priority are collected. T-Server responds immediately with the `EventError` message to existing or new reservation requests of a lower priority while collecting the agent reservation requests of the highest priority only. This functionality is controlled with the `collect-lower-priority-requests` configuration option (see page 238).

Client Connections

The number of connections T-Server can accept from its clients depend on the operating system that T-Server runs. Table 2 illustrates the number of client connections that T-Server support.

Table 2: Number of T-Server's Client Connections

Operating System	Number of Connections
AIX 32-bit mode (versions 5.3)	32767
AIX 64-bit mode (versions 5.3, 6.1, 7.1)	32767
HP-UX 32-bit mode (versions 11.11)	2048
HP-UX 64-bit mode (versions 11.11, 11i v2, 11i v3)	2048
HP-UX Itanium (version 11i v3)	2048
Linux 32-bit mode (versions RHEL 4.0, RHEL 5.0)	32768
Linux 64-bit mode (version RHEL 5.0)	32768
Solaris 32-bit mode (version 9)	4096
Solaris 64-bit mode (versions 9, 10)	65536
Windows Server 2003, 2008	4096

See the *Genesys Supported Operating Environment Reference Guide* on the Genesys Documentation website for more detailed information and a list of all supported operating systems.

Next Steps

Now that you have gained a general understanding of the roles and features available with T-Servers, you are ready to learn how T-Servers are installed and configured. That information is presented in the next few chapters of this *Deployment Guide.* So unless you are already familiar with T-Server deployment and operation procedures, continue with Chapter 2, "T-Server General Deployment," on page 33. Otherwise, you may want to jump to Part Two of this *Deployment Guide*, where you will find information about your specific T-Server.

Genesys

Chapter

2 T-Server General Deployment

This chapter contains general information for the deployment, configuration, and installation of your T-Server. You may have to complete additional configuration and installation steps specific to your T-Server and switch. You will find these steps in Part Two of this document.

This chapter contains these sections:

Note: You *must* read the *Framework 8.1 Deployment Guide* before proceeding with this T-Server guide. That book contains information about the Genesys software you must deploy before deploying T-Server.

Prerequisites

T-Server has a number of prerequisites for deployment. Read through this section before deploying your T-Server.

Software Requirements

Framework Components

You can only configure T-Server after you have deployed the Configuration Layer of Genesys Framework. This layer contains DB Server, Configuration

Server, and Configuration Manager. If you intend to monitor or control T-Server through the Management Layer, you must also install and configure components of this Framework layer, such as Local Control Agent (LCA), Message Server, Solution Control Server (SCS), and Solution Control Interface (SCI), before deploying T-Server.

Refer to the *Framework 8.1 Deployment Guide* for information about, and deployment instructions for, these Framework components.

Media Layer and LCA

To monitor the status of components in the Media Layer through the Management Layer, you must load an instance of LCA on every host running Media Layer components. Without LCA, Management Layer cannot monitor the status of any of these components. If you do not use the Management Layer, LCA is not required.

Supported Platforms

Refer to the *Genesys Supported Operating Environment Reference Guide* for the list of operating systems and database systems supported in Genesys releases 7.x and 8.x. You can find this document on the Genesys Customer Care website.

For UNIX-based (UNIX) operating systems, also review the list of patches Genesys uses for software product builds, and upgrade your patch configuration if necessary. A description of patch configuration is linked to installation `read_me.html` files for the Genesys applications that operate on UNIX, and is available within the installation packages.

Security

Starting with release 7.5, T-Server supports the Genesys Transport Layer Security (TLS) and can be configured for secure data exchange with the other Genesys components that support this functionality.

The Genesys TLS is not supported on all operating systems that T-Server itself supports. For information about the supported operating systems, see the *Genesys 8.x Security Deployment Guide.*

Hardware and Network Environment Requirements

Hosting

Genesys recommends that you or your IT specialist assign host computers to Genesys software before you start Genesys installation. Remember the following restrictions:

- Do not install all the Genesys server applications on the same host computer.
- When installing a few server applications on the same host computer, prevent them (except for Configuration Server) from using the swap area.

Installation Privileges

During deployment, be sure to log in with an account that will permit you to perform administrative functions—that is, one that has root privileges.

Server Locations

Refer to the "Network Locations for Framework Components" chapter of the *Framework 8.1 Deployment Guide* for recommendations on server locations.

Supported Platforms

Refer to the *Genesys Supported Media Interfaces Reference Manual* for the list of supported switch and PBX versions. You can find this document on the Genesys Customer Care website.

Licensing Requirements

All Genesys software is licensed—that is, it is not shareware. Genesys products are protected through legal license conditions as part of your purchase contract. However, the level of technical license-control enforcement varies across different solutions and components.

Before you begin to install T-Server, remember that, although you may not have had to use technical licenses for your software when you deployed the Configuration and Management Layers in their basic configurations, this is not the case with the Media Layer.

T-Server requires seat-related DN technical licenses to operate even in its most basic configuration. Without appropriate licenses, you cannot install and start T-Server. If you have not already done so, Genesys recommends that you install License Manager and configure a license file at this point. For complete information on which products require what types of licenses, and on the installation procedure for License Manager, refer to the *Genesys Licensing Guide* available on the Genesys Documentation Library DVD.

The sections that follow briefly describe the T-Server license types.

Note: Starting with release 7.2, the licensing requirements for T-Server have changed from previous releases. Please read this section carefully and refer to the *Genesys Licensing Guide* for complete licensing information.

Licensing Basic Implementations

A stand-alone T-Server serving a single site requires licenses to register all DNs it monitors. DNs that agents use in day-to-day contact center operations, such as Extensions and ACD Positions, have to be registered using licenses that control agent seats.

Note: Configure all seat DNs that agents use (Extensions and ACD Positions) in the Configuration Layer. This enables detailed call monitoring through Genesys reporting, and generally allows you to control access to individual DNs.

Licensing HA Implementations

T-Servers operating with the `hot standby` redundancy type require a special CTI HA technical license, which allows for high-availability implementations, in addition to regular T-Server licenses. Neither T-Server in a redundant pair configured for `hot standby` starts if this license is unavailable. Moreover, the primary and backup T-Servers must use the same licenses to control the same pool of DNs. If your T-Servers are configured with the `hot standby` redundancy type, order licenses for CTI HA support.

Licensing Multi-Site Implementations

T-Servers performing multi-site operations require licenses that allow for such operations, in addition to regular T-Server licenses. If some of your T-Servers are configured for multi-site routing while others are not, either order licenses for multi-site support for all T-Servers or install an additional License Manager to handle the T-Servers involved in multi-site routing.

Note: You do not need licenses for multi-site support if some T-Server clients include the local location as the `location` attribute value in their requests for routing within the same site.

Configuring License Files

You need a license to configure and install Media Layer components. Genesys recommends that, if you have not already done so, at this point you:

1. Install License Manager.
2. Configure license files.

Note: If you use the `<port>@<server>` format when entering the name of the license server during installation, remember that some operating systems use `@` as a special character. In this case, the installation routine is unable to write license information for T-Server to the Configuration Layer or the `run.sh` file. Therefore, when you use the `<port>@<server>` format, you must manually modify the command-line license parameter after installing T-Server.

For information about which products require what types of licenses and for the installation procedure for License Manager, refer to the *Genesys Licensing Guide* available on the Genesys Documentation Library DVD.

About Configuration Options

Configuring T-Server is not a onetime operation. It is something you do at the time of installation and then in an ongoing way to ensure the continued optimal performance of your software. You must enter values for T-Server configuration options on the `Options` tab of your T-Server `Application` object in Configuration Manager. The instructions for configuring and installing T-Server that you see here are only the most rudimentary parts of the process. You must refer extensively to the configuration options chapters located in Part Two of this book. Pay particular attention to the configuration options specific to your own T-Server.

Configuration options common to all T-Servers, independent of switch type, are described in Chapter 12, "T-Server Common Configuration Options," on page 229. *T-Server-specific* configuration options are described in a separate chapter. T-Server also supports unified Genesys log options, as described in the "Common Configuration Options" chapter.

Options that configure values for the TSCP software in your T-Server are common to all T-Servers. Options based on the custom features of your switch apply to your T-Server only. Familiarize yourself with both types of options. You will want to adjust them to accommodate your production environment and the business rules that you want implemented there.

Deployment Sequence

This is the recommended sequence to follow when deploying T-Server.

Task Summary: T-Server Deployment Sequence

Objective	Related Procedures and Actions
1. Deploy Configuration Layer objects and ensure Configuration Manager is running.	See the *Framework 8.1 Deployment Guide* for details.
2. Deploy Network objects (such as Host objects).	See the *Framework 8.1 Deployment Guide* for details.
3. Deploy the Management Layer.	See the *Framework 8.1 Deployment Guide* for details.
4. Deploy T-Server.	See"Deployment of T-Server" on page 38.
5. Test your configuration and installation.	See Chapter 5, "Starting and Stopping T-Server Components," on page 117.

Note: If, during the installation procedure for any of the Genesys applications, the script warns you that Configuration Server is unavailable and that the configuration cannot be updated, continue with the installation. Following the installation, you must complete the information on the `Start Info` tab to ensure that T-Server will run.

Deployment of T-Server

Deploying T-Server manually requires that you configure a number of different objects in the Configuration Layer prior to setting up your T-Server objects and then install T-Server. This section describes the manual deployment process.

Configuration of Telephony Objects

This section describes how to manually configure T-Server telephony objects if you are using Configuration Manager. For information about configuring T-Server telephony objects using Genesys Administrator, refer to the *Framework 8.1 Genesys Administrator Help.*

Recommendations

Genesys recommends registering (configuring) only those entities you plan to use in the current configuration. The more data there is in the Configuration Database, the longer it takes for the CTI setup to start, and the longer it will take to process configuration data. Remember that adding configuration objects to the Genesys Configuration Database does not cause any interruption in contact center operation.

Depending on how much work is required to manually configure all applications and objects, consider registering more `Person` objects first, with a set of privileges that lets them perform configuration tasks.

Switching Offices

Your telephony network may contain many switching offices, but you should only configure those that are involved with customer interactions.

Using Configuration Manager, be sure to register a `Switching Office` object that accommodates your `Switch` object under `Environment`. Until you have done this, you cannot register a `Switch` object under `Resources` (single-tenant environment) or a `Tenant` (multi-tenant environment).

Note: The value for the switching office name must not have spaces in it.

Switches

1. Configure a `Switch` object for each switch on your telephony network. Assign each `Switch` object to the appropriate T-Server `Application` object.
2. If implementing the multi-site configuration, specify access codes for all switches on the network so that the call-processing applications can route and transfer calls between switches.

   Two types of access codes exist in a Genesys configuration:

   - Default access codes that specify how to reach this switch from any other switch in the Genesys environment.

- Switch-to-switch access codes that specify how to reach a particular switch from any other switch. Use this type when either a nondefault dial number or routing type is required between any two locations. When a switch-to-switch access code is configured, its value has a higher priority than that of a default access code.

See Chapter 4, "Multi-Site Support," on page 59, for step-by-step instructions.

Note: When the numbering plan uses unique directory number (DN) assignment across sites and multi-site routing is not used, you do not have to configure access codes.

DNs and Agent Logins

Note: Starting with release 7.2, the requirements for configuring DNs in the Configuration Layer have changed. Refer to Part Two of this guide for information about the requirements on configuring specific DN types for your T-Server.

For each T-Server for which you are configuring DNs, you must configure all DNs that agents and their supervisors use in day-to-day contact center operation—so-called *seat-related DNs*—such as Extensions and ACD Positions. Otherwise, T-Server does not register such DNs.

1. To configure Telephony objects within each switch, consult the switch documentation. Information specific to your T-Server in Part Two of this document contains tables that indicate how to set DN types in the Genesys Configuration Database depending on the switch DN types and configuration.
2. Check the numbering plan for different types of DNs, to see if you can save time by registering Ranges of DNs. Usually, DNs of the same type have consecutive numbers, which will make an otherwise tedious configuration task easy. `Agent Login` objects almost always have consecutive numbers, which means you can register them through the Range of Agent Logins feature as well.
3. If you plan to use Virtual Queues and Virtual Routing Points in the contact center operation, Genesys recommends registering them after you have outlined the call-processing algorithms and identified your reporting needs.

Note: Remember that CTI applications, not the switch, generate telephony events for DNs of these types.

Warning! When setting the `Register` flag for a DN, make sure you select the value according to your T-Server. The `Register` flag values are as follows:

- `False`—T-Server processes this DN locally, and never registers it on the switch.
- `True`—T-Server always registers this DN on the switch during T-Server startup or CTI link reconnect.
- `On Demand`—T-Server registers this DN on the switch only if a T-Server client requests that it be registered.

Multi-Site Operations

See the section, "Configuring Multi-Site Support" on page 102, for information on setting up DNs for multi-site operations.

Configuration of T-Server

Use the *Framework 8.1 Deployment Guide* to prepare accurate configuration information. You may also want to consult *Configuration Manager Help* and/or *Genesys Administrator Help,* which contains detailed information about configuring objects.

Recommendations

Genesys recommends using an Application Template when you are configuring your T-Server application. The Application Template for your particular T-Server contains the most important configuration options set to the values recommended for the majority of environments. When modifying configuration options for your T-Server application later in the process, you can change the values inherited from the template rather than create all the options by yourself.

Procedure: Configuring T-Server

Start of procedure

1. Follow the standard procedure for configuring all `Application` objects to begin configuring your T-Server `Application` object. Refer to the *Framework 8.1 Deployment Guide* for instructions.
2. In a `Multi-Tenant` environment, specify the `Tenant` to which this T-Server belongs on the `General` tab of the `Properties` dialog box.

3. On the Connections tab:
   - Add all Genesys applications to which T-Server must connect.

   Note: For multi-site deployments you should also specify T-Server connections on the Connections tab for any T-Servers that may transfer calls directly to each other.

4. On the Options tab, specify values for configuration options as appropriate for your environment.

   Note: For T-Server option descriptions, see Part Two of this document.

5. In a multi-site environment, you must complete additional T-Server configuration steps to support multi-site operations; see Chapter 4, "Multi-Site Support," on page 59.

End of procedure

Next Steps

- See "Installation of T-Server" on page 43.

Procedure: Configuring multiple ports

Purpose: To configure multiple ports in T-Server for its client connections.

Start of procedure

1. Open the T-Server Application Properties dialog box.
2. Click the Server Info tab.
3. In the Ports section, click Add Port.
4. In the Port Properties dialog box, on the Port Info tab:
   a. In the Port ID text box, enter the port ID.
   b. In the Communication Port text box, enter the number of the new port.
   c. In the Connection Protocol box, select the connection protocol, if necessary.
   d. Select the Listening Mode option.

   Note: For more information on configuring secure connections between Framework components, see *Genesys 8.x Security Deployment Guide*.

e. Click OK.

5. Click OK to save the new configuration.

End of procedure

Installation of T-Server

The following directories on the Genesys 8.1 Media product DVD contain T-Server installation packages:

- media_layer/<switch>/<platform> for UNIX installations, where <switch> is your switch name and <platform> is your operating system.
- media_layer\<switch>\windows for Windows installations, where <switch> is your switch name.

Procedure: Installing T-Server on UNIX

Note: During installation on UNIX, all files are copied into the directory you specify. No additional directories are created within this directory. Therefore, do not install different products into the same directory.

Start of procedure

1. In the directory to which the T-Server installation package was copied, locate a shell script called install.sh.
2. Run this script from the command prompt by typing sh and the file name. For example: sh install.sh.
3. When prompted, confirm the host name of the computer on which T-Server is to be installed.
4. When prompted, specify the host and port of Configuration Server.
5. When prompted, enter the user name and password to access Configuration Server.
6. When prompted, select the T-Server application you configured in "Configuring T-Server" on page 41 from the list of applications.
7. Specify the destination directory into which T-Server is to be installed, with the full path to it.
8. If the target installation directory has files in it, do one of the following:
   - Type 1 to back up all the files in the directory (recommended).
   - Type 2 to overwrite only the files in this installation package. Use this option only if the installation being upgraded operates properly.

- Type `3` to erase all files in this directory before continuing with the installation.

The list of file names will appear on the screen as the files are copied to the destination directory.

9. If asked which version of the product to install, the 32-bit or the 64-bit, choose the one appropriate to your environment.
10. If asked about the license information that T-Server is to use: specify either the full path to, and the name of, the license file, or the license server parameters.
11. As soon as the installation process is finished, a message appears announcing that installation was successful. The process places T-Server in the directory with the name specified during the installation.

End of procedure

Next Steps

- To verify manual installation, go to "Verifying the installation of T-Server" on page 45.
- To test your configuration and installation, go to Chapter 5, "Starting and Stopping T-Server Components," on page 117, and try it out.
- To configure and install redundant T-Servers, see Chapter 3, "High-Availability Deployment," on page 47.
- To install T-Servers for a multi-site environment, proceed to Chapter 4, "Multi-Site Support," on page 59.

Procedure: Installing T-Server on Windows

Start of procedure

1. In the directory to which the T-Server installation package was copied, locate and double-click `Setup.exe` to start the installation.
2. When prompted, specify the connection parameters to the Configuration Server associated with this T-Server.
3. When prompted, select the T-Server `Application` you configured in "Configuring T-Server" on page 41 from the list of applications.
4. Specify the license information that T-Server is to use: either the full path to, and the name of, the license file, or the license server parameters.
5. Specify the destination directory into which T-Server is to be installed.
6. Click `Install` to begin the installation.
7. Click `Finish` to complete the installation.

By default, T-Server is installed as a Genesys service (Windows `Services`) with `Automatic` startup type.

End of procedure

Next Steps

- To verify manual installation, go to "Verifying the installation of T-Server" on page 45.
- To test your configuration and installation, go to Chapter 5, "Starting and Stopping T-Server Components," on page 117, and try it out.
- To configure and install redundant T-Servers, see Chapter 3, "High-Availability Deployment," on page 47.
- To install T-Servers for a multi-site environment, proceed to Chapter 4, "Multi-Site Support," on page 59.

Procedure: Verifying the installation of T-Server

Purpose: To verify the completeness of the manual installation of T-Server to ensure that T-Server will run.

Prerequisites

- Procedure: Installing T-Server on UNIX, on page 43
- Procedure: Installing T-Server on Windows, on page 44

Start of procedure

1. Open the `Properties` dialog box for a corresponding `Application` object in Configuration Manager.
2. Verify that the `State Enabled` check box on the `General` tab is selected.
3. Verify that the `Working Directory`, `command-line`, and `Command-Line Arguments` are specified correctly on the `Start Info` tab.
4. Click `Apply` and `OK` to save any configuration updates.

End of procedure

Next Steps

At this point, you have configured and installed T-Server using Configuration Manager. If you want to test your configuration and installation, go to Chapter 5, "Starting and Stopping T-Server Components," on page 117, and try

it out. Otherwise, if you want to configure and install redundant T-Servers, see Chapter 3, “High-Availability Deployment,” on page 47. If you want to install T-Servers for a multi-site environment, proceed to Chapter 4, “Multi-Site Support,” on page 59.

Genesys

Chapter

3 High-Availability Deployment

This chapter describes the general steps for setting up a high-availability (HA) environment for your T-Server. The high-availability architecture implies the existence of redundant applications, a primary and a backup. These are monitored by a management application so that, if one application fails, the other can take over its operations without any significant loss of contact center data.

Every switch/T-Server combination offers different high-availability options. The Framework Management Layer currently supports two types of redundant configurations: `warm standby` and `hot standby`. All T-Servers offer the `warm standby` redundancy type and, starting with release 7.1, the `hot standby` redundancy type is implemented in T-Servers for most types of switches. Some T-Servers support a switch's ability to provide two CTI links to two T-Servers or even one CTI link to two T-Servers. Other T-Servers require Genesys's HA Proxy in order to support the `hot standby` redundancy type. See Table 1 on page 28 and the T-Server-specific information later in this document for details on your T-Server.

This chapter describes the redundant architecture and how to configure T-Server so that it operates with either type. Information in this chapter is divided into the following sections:

Warm Standby Redundancy Type

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

Warm Standby Redundancy Architecture

Figure 5 illustrates the `warm standby` architecture. The standby server recognizes its role as a backup and does not process client requests until the Management Layer changes its role to primary. When a connection is broken between the primary server and the Local Control Agent (LCA, not shown in the diagram) running on the same host, a failure of the primary process is reported, and the switchover occurs; or, if the host on which the T-Server is running fails, the switchover also occurs. (See the *Framework 8.1 Deployment Guide* for information on LCA.) As a result:

1. The Management Layer instructs the standby process to change its role from backup to primary.
2. A client application reconnects to the new primary.
3. The new primary (former backup) starts processing all new requests for service.

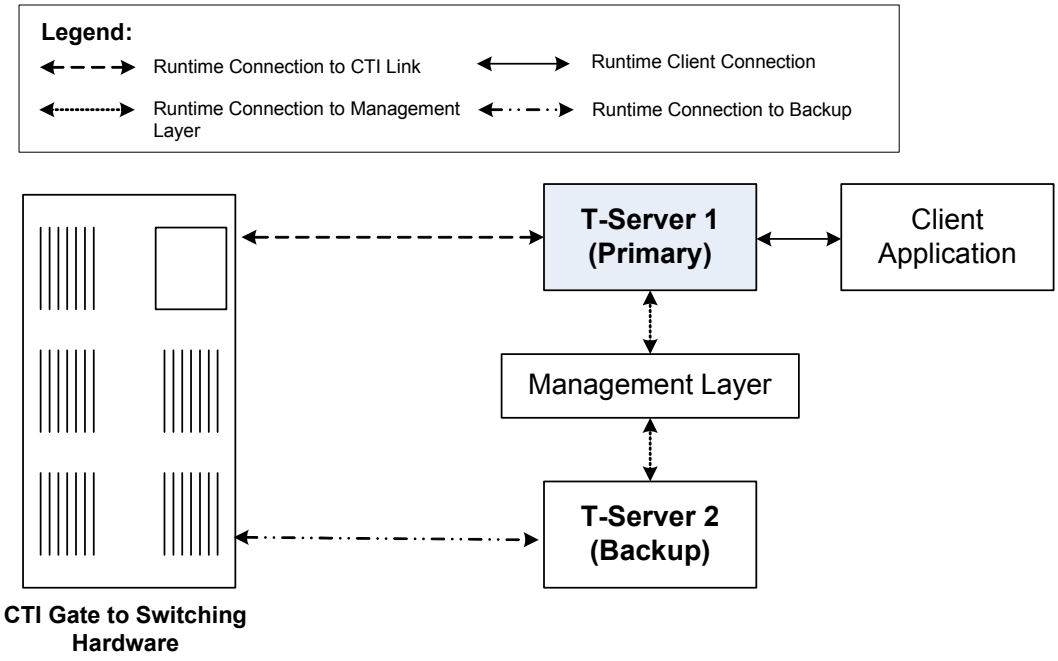

Figure 5: Warm Standby Redundancy Architecture

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

Note: You can find full details on the role of the Management Layer in redundant configurations in the *Framework 8.1 Deployment Guide*.

Hot Standby Redundancy Type

Genesys uses the expression *hot standby* to describe the redundancy type in which a backup server application remains initialized, clients connect to both the primary and backup servers at startup, and the backup server data is synchronized from the primary server. Data synchronization and existing client connections to the backup guarantee higher availability of a component. (See Figure 6 on page 50.)

Starting with release 7.1, the `hot standby` redundancy type is implemented in T-Servers for most types of switches. However, for some switches, you must compensate for the lack of link redundancy by using an additional Genesys component called *HA Proxy*.

Hot Standby Redundancy Architecture

Figure 6 illustrates the switch-independent side of a `hot standby` implementation. Here, T-Servers start simultaneously and connect to the switch. At T-Server startup, the Management Layer assigns the role of the primary server to T-Server 1, and the role of backup to T-Server 2. T-Server clients register with both T-Servers, but only the primary T-Server handles client requests other than the registration requests. The internal T-Server information, such as a DN status, `ConnID`, `UserData`, and `Call Type`, is synchronized between the primary and backup T-Servers. Therefore, the backup T-Server has the same information as the primary T-Server.

If T-Server 1 fails, the Management Layer makes T-Server 2 the new primary server, and it starts processing client requests. The Management Layer attempts to restart T-Server 1, and if it is successful, it makes T-Server 1 the new backup server.

The details of `hot standby` redundancy implementation between T-Servers and their switches vary depending on switch support for multiple CTI links. If your T-Server supports `hot standby` (see Table 1 on page 28), refer to Part Two for detailed information on the available `hot standby` schema.

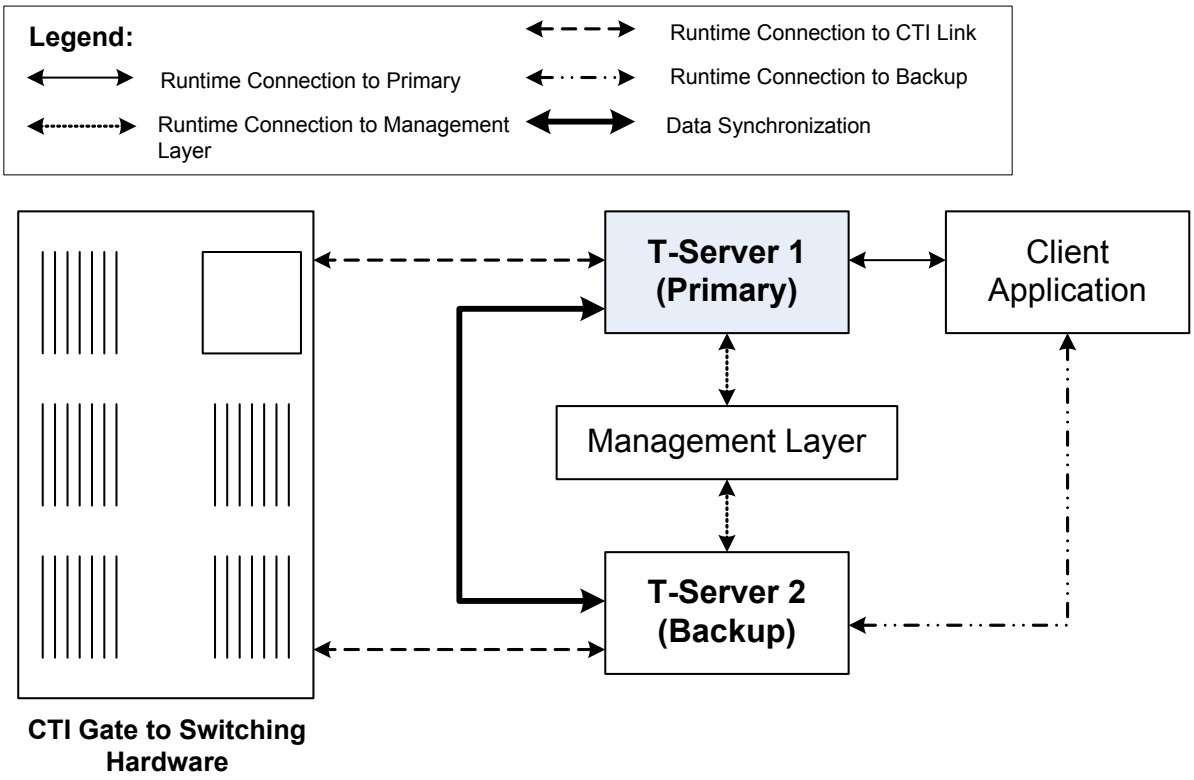

Figure 6: Hot Standby Redundancy Architecture

Benefits of Hot Standby Redundancy

The `hot standby` redundancy type provides the following benefits over the `warm standby` type:

- Using `hot standby` ensures the processing of interactions in progress if a failure occurs. After the primary T-Server (T-Server 1) fails, T-Server 2 handles all new interactions and takes over the processing of interactions that are currently in progress.
- T-Servers perform one-way (from primary to backup) synchronization of call-associated data, including, but not limited to:
  - Connection IDs.
  - Attached user data.
  - Inter Server Call Control (ISCC; formerly called External Routing) call references to another site in a multi-site environment (to support the ISCC/COF feature).

> **Note:** Refer to "ISCC Call Data Transfer Service" on page 61 for ISCC feature descriptions.

- When mirrored links are not available, HA Proxy helps T-Server synchronize the current states of agents, calls, parties, and devices between the primary and backup T-Servers.

However, keep the following `hot standby` limitations in mind:

- Client requests sent during the failure and switchover may be lost.
- Routing requests sent by the switch during the failure and switchover may be lost.
- T-Server does not synchronize interactions that begin before it starts, including incomplete ISCC-related transactions.
- Some T-Library events might be duplicated or lost.
- Reference IDs from client requests can be lost in events.

Prerequisites

This section presents basic requirements and recommendations for configuring and using redundant T-Servers.

Requirements

You must install the Management Layer if you are installing redundant T-Server applications. In particular, install Local Control Agent (LCA) on each computer that runs T-Server.

Warning! Genesys strongly recommends that you install the backup and primary T-Servers on different host computers.

Synchronization Between Redundant T-Servers

When T-Servers operate in a high-availability environment, the backup T-Server must be ready to take on the primary role when required. For this purpose, both T-Servers must be running and must have the same information. When you configure redundant T-Servers to operate with the `hot standby` type, the primary T-Server uses the connection to the backup to deliver synchronization updates. Genesys recommends that you enable the Advanced Disconnect Detection Protocol (ADDP), described in Chapter 1, for this connection. Do so using the configuration options in the "Backup-Synchronization Section" section. Refer to the "T-Server Common Configuration Options" chapter for option descriptions.

Configuration Warnings

When configuring T-Servers to support either the `warm standby` or `hot standby` redundancy type, remember:

1. When at least one of the two T-Servers that operate in a redundant mode is running, do not change a redundancy type, host, or port in either T-Server configuration.
2. When both the primary and backup T-Servers are running, do not remove the backup T-Server `Application` object from the configuration.

You are responsible for the option synchronization in the configuration of the primary and backup T-Servers; Configuration Server does not synchronize either options or their values in different T-Server `Application` objects. That is, you must configure both T-Servers to have the same options with the same values. If you change a value in one T-Server configuration, you must change it in the other T-Server configuration manually. The log options in the primary T-Server can differ from those in the backup T-Server configuration. The link configuration options in the primary T-Server can also differ from those in the backup T-Server configuration.

Warm Standby Deployment

This section describes how to configure redundant T-Servers to work with the `warm standby` redundancy type, including details on their connections and settings.

General Order of Deployment

The general guidelines for T-Server `warm standby` configuration are:

1. Configure two T-Server `Application` objects as described in "Configuration of T-Server" on page 41.
2. Make sure the `Switch` object is configured for the switch these T-Servers should serve, as described in "Configuration of T-Server" on page 41.
3. Modify the configuration of the primary and backup T-Servers as instructed in the following sections.

After completing the configuration steps, ensure that both T-Servers are installed (see page 54).

Modification of T-Servers for Warm Standby

Modify the configuration of both the primary and backup T-Server Application objects as described in the following sections.

Note: Starting with release 7.5, you can configure multiple ports for any application of type server. When multiple ports are configured for a server in a warm standby redundancy pair, the number of ports, their Port IDs, and the Listening Mode settings of the primary and backup servers must match respectively.

Procedure: Modifying the primary T-Server configuration for warm standby

Start of procedure

1. Stop both the primary and backup T-Servers, if they are already running.
2. Open the Configuration Manager main window.
3. Open the Properties dialog box of the Application object for the T-Server that you want to configure as a primary server.
4. Click the Switches tab.
5. Ensure that it specifies the Switch that this T-Server Application should serve. If necessary, select the correct Switch using the Browse button.
6. Click Apply to save the configuration changes.
7. Click the Server Info tab.
8. Specify the T-Server Application you want to use as the backup server. Use the Browse button next to the Backup Server field to locate the backup T-Server Application object.
9. Select Warm Standby as the Redundancy Type.
10. Click Apply to save the configuration changes.
11. Click the Start Info tab.
12. Select Auto-Restart.
13. Click Apply and OK to save the configuration changes.

End of procedure

Next Steps

- Procedure: Modifying the backup T-Server configuration for warm standby, on page 54

Procedure:
Modifying the backup T-Server configuration for warm standby

Start of procedure

1. Make sure the two T-Servers are *not* running.
2. Open the Configuration Manager main window.
3. Open the `Properties` dialog box of the `Application` object for the T-Server that you want to configure as a backup server.
4. Click the `Switches` tab.
5. Using the `Browse` button, select the same `Switch` object you associated with the primary T-Server `Application` object.
6. Click `Apply` to save the configuration changes.
7. Click the `Start Info` tab.
8. Select `Auto-Restart`.
9. Click `Apply` and `OK` to save the configuration changes.

End of procedure

Warm Standby Installation of Redundant T-Servers

The installation of a redundant T-Server is the same as that for the stand-alone T-Server. If you have not installed the primary and backup T-Servers yet, follow the instructions in "Installation of T-Server" on page 43 for both installations.

Hot Standby Deployment

This section describes how to configure redundant T-Servers to work with the `hot standby` redundancy type, including details on their connections and settings.

General Order of Deployment

The general guidelines for T-Server `hot standby` configuration are:

1. Configure two T-Server `Applications` objects as described in "Configuring T-Server" on page 41.

2. Make sure the Switch object is configured for the switch these T-Servers should serve, as described in "Configuration of Telephony Objects" on page 38.
3. Modify the configuration of the primary and backup T-Servers as instructed in the following sections.

After completing the configuration steps, ensure that both T-Servers are installed (see page 58).

Table 1 on page 28 summarizes hot standby redundancy support in various T-Servers. For detailed, up-to-date information on the subject, see the *Genesys Supported Media Interfaces Reference Manual* located on the Genesys Customer Care website.

Modification of T-Servers for Hot Standby

Modify the configuration of both the primary and backup T-Server Application objects for hot standby redundancy as described in the following sections.

Note: Starting with release 7.5, you can configure multiple ports for any application of type server. When multiple ports are configured for a server in a hot standby redundancy pair, the number of ports, their Port IDs, and the Listening Mode settings of the primary and backup servers must match respectively.

Procedure: Modifying the primary T-Server configuration for hot standby

Start of procedure

1. Stop both primary and backup T-Servers if they are already running.
2. Open the Configuration Manager main window.
3. Open the Properties dialog box of the Application object for the T-Server that you want to configure as a primary server.
4. Click the Switches tab.
5. Ensure that it specifies the Switch that this T-Server Application should serve. If necessary, select the correct Switch using the Browse button.
6. Click Apply to save the configuration changes.
7. Click the Server Info tab.

8. In the Ports section, select the port to which the backup server will connect for HA data synchronization and click Edit Port.

   Note: For information on adding multiple ports, see "Configuring multiple ports" on page 42.

   a. In the Port Properties dialog box, on the Port Info tab, select the HA sync check box.
   b. Click OK.

   Note: If the HA sync check box is not selected, the backup T-Server will connect to the *default* port of the primary T-Server.

9. Specify the T-Server Application you want to use as the backup server. Use the Browse button next to the Backup Server field to locate the backup T-Server Application object.
10. Select Hot Standby as the Redundancy Type.
11. Click Apply to save the configuration changes.
12. Click the Start Info tab.
13. Select Auto-Restart.
14. Click Apply to save the configuration changes.
15. To enable ADDP between the primary and backup T-Servers, click the Options tab. Open or create the backup-sync section and configure corresponding options.

    Note: For a list of options and valid values, see the "Backup-Synchronization Section" section of "T-Server Common Configuration Options" chapter in Part Two of this document.

16. Click Apply and OK to save the configuration changes.

End of procedure

Next Steps

- Procedure: Modifying the backup T-Server configuration for hot standby, on page 57

Procedure:
Modifying the backup T-Server configuration for hot standby

Start of procedure

1. Make sure the two T-Servers are *not* running.
2. Open the Configuration Manager main window.
3. Open the `Properties` dialog box of the `Application` object for the T-Server that you want to configure as a backup server.
4. Click the `Switches` tab.
5. Using the `Browse` button, select the same `Switch` object you associated with the primary T-Server `Application`.
6. Click the `Server Info` tab.
7. In the `Ports` section, select the port to which the primary server will connect for HA data synchronization and click `Edit Port`.

   Note: For information on adding multiple ports, see "Configuring multiple ports" on page 42.

   a. In the `Port Properties` dialog box, on the `Port Info` tab, select the `HA sync` check box.
   b. Click `OK`.

   Note: If the `HA sync` check box is not selected, the primary T-Server will connect to the *default* port of the backup T-Server.

8. Click `Apply` to save the configuration changes.
9. Click the `Start Info` tab.
10. Select `Auto-Restart`.
11. Click the `Options` tab.
12. Modify the values for all necessary configuration options. Genesys recommends that you set all configuration options for the backup T-Server to the same values as for the primary T-Server; the only exceptions are the log options and the `server-id` option.
13. Click `Apply` and `OK` to save the configuration changes.

End of procedure

Hot Standby Installation of Redundant T-Servers

The installation of a redundant T-Server is the same as that for the stand-alone T-Server. If you have not installed the primary and backup T-Servers yet, follow instructions in "Installation of T-Server" on page 43 for both installations.

Next Steps

At this point, you have learned how to configure and install redundant T-Servers. Go to Chapter 5, "Starting and Stopping T-Server Components," on page 117, to test your configuration and installation, or continue with Chapter 4, "Multi-Site Support," on page 59, for more possibilities.

Genesys

Chapter

4 Multi-Site Support

This chapter contains general information about multi-site environments, as well as information on deploying a multi-site environment for your T-Server.

This chapter is divided into the following sections:

Note: Each switch/T-Server combination offers different multi-site options. For details describing your specific switch/T-Server environment, refer to Chapter 12, "T-Server Common Configuration Options," on page 229.

The following instructions apply to both local and remote switches and T-Servers. Because different vendor switches can be installed at the local and remote locations, this chapter covers several, but not all, possible configurations. To help determine which sections of this chapter apply to your situation, refer to Table 3 on page 77 and Table 4 on page 82.

For more information on your specific switch/T-Server environment, refer to the multi-site topics in Part Two of this guide.

Multi-Site Fundamentals

A multi-site configuration has two or more switches that belong to the same enterprise or service provider and that share the Genesys Configuration Database. (In some cases, this may include isolated partitions on a given switch served by different T-Servers.) The main goal of T-Server support for multi-site operations is to maintain critical information about a call as it travels from one switch to another.

T-Server supports multi-site operations using its *Inter Server Call Control* (*ISCC;* formerly called External Routing), which supports the following functions:

- **Call matching—**To link instances of a call distributed across multiple sites and to re-attach essential data associated with the call (`ConnID`, `UserData`, `CallType`, and `CallHistory`). The following T-Server features support this capability:
  - ISCC Call Data Transfer Service (active external routing)—when requested by a T-Server client by specifying the desired destination in the location parameter, and also with various ISCC strategies performed by direct dial or by using the Transfer Connect Service. See "ISCC Transaction Types" on page 68 and "Transfer Connect Service Feature" on page 80.
  - Inter Server Call Control/Call Overflow (ISCC/COF) feature (passive external routing)—applicable when calls are overflowed to another site either directly or manually (see page 81).
  - Number Translation feature (see page 85).
  - Network Attended Transfer/Conference (NAT/C) feature (see page 93).

  Note: When ISCC detects call instance reappearance on a given site, the call is assigned a unique `ConnID` and the user data is synchronized with the previous call instances. This ensures that `ConnIDs` assigned to different instances of the same call on a given site are unique.

- **Call data synchronization between associated call instances (ISCC Event Propagation)—**To provide the most current data to call instances residing on remote T-Servers. The following T-Server features support this capability:
  - User Data propagation (see page 96)
  - Party Events propagation (see page 97)

  Note: ISCC automatically detects topology loops and prevents continuous updates.

Note: In distributed networks, Genesys recommends using call flows that prevent call topology loops and multiple reappearances of the same call instance. This approach ensures that all T-Servers involved with the call report the same `ConnID`, and also optimizes telephony trunk allocation by preventing trunk tromboning.

The T-Server configuration contains information about other T-Servers with which it will communicate. T-Server uses this information to connect with the other T-Servers. During this "handshake" process, T-Servers exchange information about the following parameters:

- Protocol type
- Switch type
- Server name
- Location name (switch name)
- T-Server role (primary or backup)

To complete the handshake process, T-Servers exchange messages about the current condition of the links to their switches. After the handshake process is complete, T-Server is ready to support a multi-site operation.

ISCC Call Data Transfer Service

Because ISCC supports active external routing, T-Servers that serve different switches (usually on different sites) can exchange call data when a call is passed from one switch to another. With this functionality, T-Server provides its clients with the following additional information about each call received from another switch:

- The connection identifier of the call (attribute `ConnID`).
- Updates to user data attached to the call at the previous site (attribute `UserData`).
- The call type of the call (attribute `CallType`)—In multi-site environments the `CallType` of the call may be different for each of its different legs. For example, one T-Server may report a call as an `Outbound` or `Consult` call, but on the receiving end this call may be reported as `Inbound`.
- The call history (attribute `CallHistory`)—Information about transferring/routing of the call through a multi-site contact center network.

Note: Load-sharing IVR Servers and Network T-Servers cannot be designated as the destination location for ISCC, except when `cast-type` is set to `dnis-pool`. Consult the *Universal Routing Deployment Guide* for specific configuration details.

Figure 7 shows the steps that occur during a typical external routing (ISCC) transaction. Note that the location where a call is initially processed is called the *origination location,* and the location to which the call is passed is called the *destination location.*

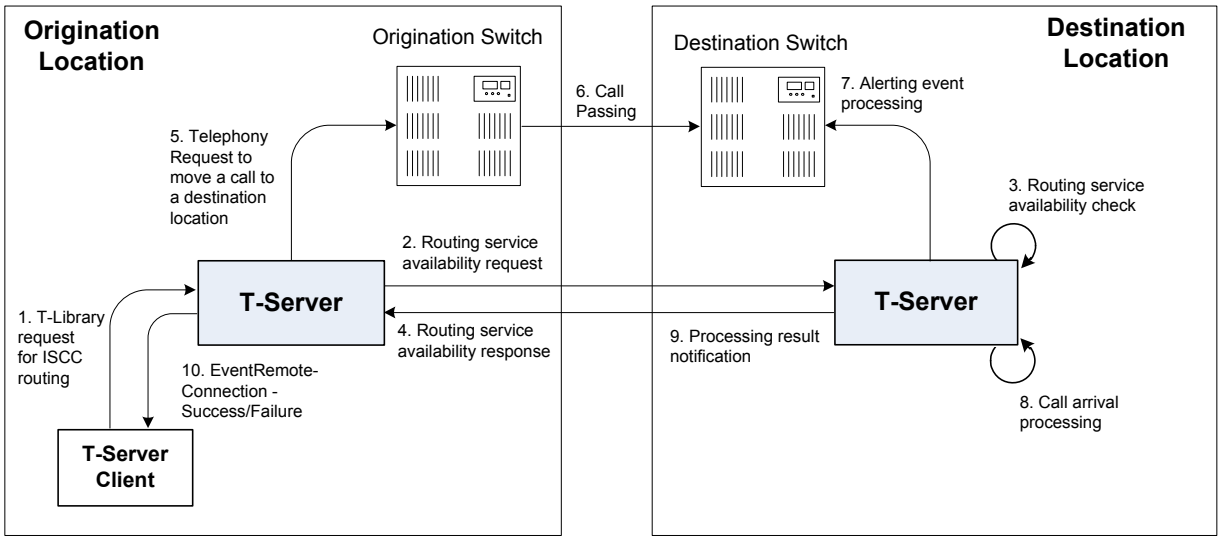

Figure 7: Steps in the ISCC Process

ISCC Call Flows

The following section identifies the steps (shown in Figure 7) that occur during an ISCC transfer of a call.

Step 1

A client connected to the T-Server at the origination location requests this T-Server to pass a call with call data to another location. For this purpose, the client must specify the `location` parameter (Attribute `Location`) when calling a corresponding T-Library function. ISCC processes the following T-Library requests:

- `TInitiateConference`
- `TInitiateTransfer`
- `TMakeCall`
- `TMuteTransfer`
- `TRouteCall`
- `TSingleStepTransfer`

Step 2

Upon receiving a client's request, the origination T-Server checks that the:

1. Connection to the destination T-Server is configured in the origination T-Server `Properties` dialog box.
2. The connection to the destination T-Server is active.
3. The destination T-Server is connected to its link.
4. The origination T-Server is connected to its link.

If these four conditions are met, the origination T-Server determines the transaction type that will be used for passing call data to another location in this transaction. The following possibilities exist:

- The client can request what *ISCC transaction type* (or simply *transaction type*) to use by specifying an appropriate key-value pair in the `Extensions` attribute of the request. The key-value pair must have a key equal to `iscc-xaction-type` and either an integer value as specified in the `TXRouteType` enumeration (see the *Platform SDK 8.x .NET (or Java) API Reference*) or a string value equal to one of the following: `default`, `route`, `direct` (or `direct-callid`), `direct-network-callid`, `direct-notoken`, `direct-ani`, `direct-uui`, `direct-digits`, `reroute`, `dnis-pool`, `pullback`, or `route-uui`.
- If the client does not specify the transaction type in the request, or specifies the `default` transaction type, T-Server checks the Switch configuration for the transaction type configured in the `Access Code` (or `Default Access Code`) properties:
  - If the `Route Type` property of the `Access Code` is set to any value other than `default`, T-Server uses the specified value as the transaction type.
  - If the `Route Type` property of the `Access Code` is set to the `default` value, T-Server uses the first value from the list specified in the `cast-type` configuration option configured for the destination T-Server. If no value has been specified for the `cast-type` option, the default value of `route` is used as the transaction type.

> **Note:** For more information on Access Codes and Default Access Code, see "Switches and Access Codes" on page 104.

After the origination T-Server determines the requested transaction type, it determines if the destination T-Server supports this transaction type.

You must list the transaction types T-Server supports in the `cast-type` configuration option.

The origination T-Server issues a request for routing service availability and sends it to the destination T-Server. The T-Server request contains data that should be passed along with the call to the destination location. This data includes the transaction type, `ConnID`, `UserData`, `CallType`, and `CallHistory`.

The timer specified by the request-tout configuration option is set when the origination T-Server sends the request. If either the specified timeout expires or the call is abandoned before the origination T-Server receives a response from the destination T-Server, the operation is considered failed. In this scenario, the origination T-Server:

1. Generates a request to the destination T-Server to cancel the request for routing service.
2. Sends EventError to the client that requested the service.
3. Deletes information about the request.

Step 3

The destination T-Server receives the request for routing service availability and checks the requested type of routing. Depending on the ISCC transaction type, it stores the request information and, when appropriate, allocates access resources for the coming call. For example, an External Routing Point is allocated when the transaction type is route, and an Access Resource of type dnis is allocated when the transaction type is dnis-pool.

Note: The resource-allocation-mode and resource-load-maximum configuration options determine how resources are allocated. For the option descriptions, refer to Chapter 12, "T-Server Common Configuration Options," on page 229.

If resources are unavailable, the request is queued at the destination location until a resource is free, or the origination T-Server cancels the request. If the request is canceled, the destination T-Server deletes all information about the request.

If resources are unavailable because of incorrect configuration, the destination T-Server returns an error event to the origination T-Server.

Step 4

If resources are available, the destination T-Server generates a positive response and the timer is started for the interval specified by the timeout configuration option of the destination T-Server.

Step 5

If the origination T-Server receives a negative response, it sends an EventError message to the client and clears all data about the request.

If the origination T-Server receives the confirmation about routing service availability, it processes the client's request and sends a corresponding message to the switch. The timer on the origination T-Server is also started for the interval specified by the timeout configuration option of the destination T-Server.

Step 6

The origination switch processes the T-Server request and passes the call to the destination switch.

Step 7

If the call arrives at the destination switch, the switch generates an alerting event.

The destination T-Server waits for the call no longer than the interval specified by the timeout configured on the destination T-Server. If the call is not received at the destination location within this interval, the destination T-Server issues a failure notification to the origination T-Server, deletes all data about the request, and, when appropriate, frees the resources previously allocated for the request.

If either the specified timeout expires or the call is abandoned before the origination T-Server receives a response from the destination T-Server, the operation is considered failed. In this case, the origination T-Server:

1. Generates a request to the destination T-Server to cancel the request for routing service.
2. Responds to the client that requested the service in one of the following ways:
   - If the origination T-Server has already sent a response to the request the client sent in Step 1 on page 62, the origination T-Server supplements its response with `EventRemoteConnectionFailed`.
   - If the origination T-Server has not yet sent a response to the client, the origination T-Server sends `EventError`.
3. Deletes information about the request.

Step 8

If the destination T-Server matches the arrived call, it updates the `ConnID`, `UserData`, `CallType`, and `CallHistory` attributes with the data received in the request for routing service availability. The connection ID is updated as follows:

The arrived call is assigned the `ConnID` that is specified in the request for routing service availability, but only if this `ConnID` does not coincide with the `ConnID` of a call that has existed at the destination site. If two such `ConnIDs` are identical, the arrived call is assigned a new unique `ConnID`.

For `direct-*` transaction types (where the asterisk stands for a `callid`, `uui`, `ani`, or `digits` extension), the call reaches the destination DN directly.

For the transaction types `route` and `route-uui`, the call first arrives at an External Routing Point from which it is routed to the destination DN. The call info is updated when the call reaches the External Routing Point. An External

Routing Point is considered free when the first alerting event (`EventQueued` or `EventRouteRequest`) is distributed.

Please keep the following issues in mind when using the ISCC feature:

- If routing from a dedicated External Routing Point to the destination DN fails, T-Server considers the transaction failed. However, the `ConnID`, `UserData`, `CallType`, and `CallHistory` attributes are updated. Then, T-Server attempts to route the call to one of the Default DNs configured for this External Routing Point.
- If the destination T-Server did not receive a request for routing service availability, but a call arrives at an External Routing Point, T-Server considers the call to be unexpected and routes the call to the DN specified by the `dn-for-unexpected-calls` configuration option. When no alternative targets are defined, the call remains at the External Routing Point until diverted by the switch or abandoned by the caller.

For `reroute` and `pullback` transaction types, the call returns to the network location. For the `dnis-pool` transaction type, the call reaches the destination DN directly.

Step 9

If, in Step 8 on page 65, the call does not arrive within the configured timeout, or the transaction fails, the destination T-Server sends a notification of failure to the origination T-Server.

Otherwise, the destination T-Server notifies the origination T-Server that the routing service was successful and deletes all information about the request.

Step 10

The origination T-Server notifies the client that the routing service was successful (or failed) and deletes all information about the request.

Client-Controlled ISCC Call Flow

The following section identifies the steps that occur during a client-controlled ISCC transfer of a call.

Step 1

A client, such as Universal Routing Server (URS), that is connected to the T-Server at the origination location detects a call to be delivered to another destination location.

Step 2

The client chooses a destination location and the target DN for the call. Then, it sends the `TGetAccessNumber` request to the destination T-Server for routing service availability, indicating the target DN and other call context (`ConnID`, `UserData`, and `CallHistory` attributes).

Step 3

The destination T-Server receives the request for routing service availability. Depending on the ISCC transaction type, it stores the request information, including the call context. When appropriate, it allocates access resources for the coming call, such as External Routing Point.

If resources are unavailable, the request is queued at the destination T-Server until an appropriate ISCC resource is free or the client cancels the request. If the request is canceled, the destination T-Server deletes all information about the request.

If resources are unavailable because of incorrect configuration, the destination T-Server returns an `EventError` message to the client.

Step 4

The destination T-Server replies to the client with the `EventAnswerAccessNumber` message, which contains the allocated ISCC resource.

Step 5

The client requests that the origination T-Server delivers the call to the destination location using the allocated access resource.

Step 6

The origination T-Server receives and processes the client's request, and then sends a corresponding message to the switch.

Step 7

The call arrives at the destination switch and is reported to the destination T-Server via CTI. The call is matched by means of ISCC, based on the specified `cast-type` setting and allocated resource, and then the call is assigned a requested call context (such as `ConnID` or call data). Upon successful transaction completion, the destination T-Server notifies the client by sending `EventRemoteConnectionSuccess`.

The destination T-Server waits for the call no longer than the interval specified by the timeout that is configured on the destination T-Server. If the call is not received at the destination location within this interval, the destination T-Server issues a failure notification to the client by sending

`EventRemoteConnectionFailed`, deletes all data about the request, and, when appropriate, frees the resources previously allocated for the request.

The destination T-Server notifies the client whether the routing service succeeded or failed by sending either the `EventRemoteConnectionSuccess` or `EventRemoteConnectionFailure`, respectively.

ISCC Transaction Types

As switches of different types provide calls with different sets of information parameters, a single mechanism for passing call data between the switches is not feasible in some cases. Therefore, the ISCC feature supports a number of mechanisms for passing call data along with calls between locations. This section describes ISCC transaction type principles, identifies which transaction types are supported for each T-Server, and defines each transaction type (beginning with "direct-ani" on page 69).

It is important to distinguish the two roles that T-Servers play in an external routing (ISCC) transaction—namely *origination T-Server* and *destination T-Server*.

- The origination T-Server initiates an ISCC transaction. It prepares to send the call to another T-Server and coordinates the process.
- The destination T-Server receives call data from an origination T-Server and matches this data to a call that will arrive at some time in the future.

The distinction between these roles is important because the range of telephony-hardware functionality often requires T-Servers to support two entirely different sets of ISCC transactions based on which of the two roles they play. For instance, it is very common for a particular T-Server to support many types of ISCC transactions when it takes on the origination role, but fewer when it takes on the role of a destination T-Server.

The ISCC transaction type `reroute` is a good example. Most T-Servers support `Reroute` as origination T-Servers, but very few support `Reroute` as destination T-Servers.

Determining and Configuring Transaction Type Support

You can find descriptions of these transaction types starting on page 69. Use Table 3 on page 77 to identify the transaction types your destination T-Server supports. A blank table cell indicates that T-Server does not support a certain transaction type.

You can configure the transaction types specific to your T-Server as values of the `cast-type` configuration option specified in the ISCC configuration section `extrouter`. Refer to Chapter 12, "T-Server Common Configuration Options," on page 229 for the option description.

ISCC Transaction Type General Principles

Generally, since most of the ISCC implementation is done at the T-Server Common Part (TSCP) code level, all T-Servers support certain ISCC transaction types. Any T-Server can act as the origination T-Server for the following transaction types:

- `direct-ani`, page 69
- `direct-notoken`, page 71
- `dnis-pool`, page 72
- `pullback`, page 73
- `reroute`, page 74
- `route` (aliased as `route-notoken`), the default transaction type, page 75

The following transaction types are unevenly supported for both the origination and destination T-Server roles:

- `direct-callid` (aliased as `direct`), page 70
- `direct-digits` (reserved for Genesys Engineering)
- `direct-network-callid`, page 70
- `direct-uui`, page 71
- `route-uui`, page 76

The `reroute` and `pullback` transaction types are supported only for selected T-Servers in the *destination* role. However, if you implement this support, other transaction types require additional configuration and testing—even those that would normally be supported by default.

direct-ani

With the transaction type `direct-ani`, the `ANI` call attribute is taken as the parameter for call matching. Properly configured switches and trunks can keep the `ANI` attribute when a call is transferred over the network. T-Server can use this network feature for call matching.

Warning! Depending on the switch platform, it may be possible to inherit the `ANI` attribute after routing a call to a remote destination, and after performing a single-step transfer and other telephone actions. However, ISCC only works properly in scenarios where the `ANI` attribute on the destination T-Server is represented by exactly the same digit string as on the origination T-Server.

Typically, the `ANI` attribute represents the original call identifier (customer phone number), which guarantees that the attribute remains unique. However, you can use the `non-unique-ani` resource type to block ISCC from matching calls based on an ANI that is known to be non-unique. (See "Configuring access resources for non-unique ANI" on page 113 for details.)

direct-callid

With the transaction type `direct-callid`, the call reaches the destination DN directly from another location, and the `CallID` of the call is taken as the attribute for call matching. When a call arrives at the final destination, the destination T-Server identifies its `CallID`, and updates the call info if the `CallID` matches.

Use this transaction type when the destination switch has the capability to assign to an incoming call the same network-wide unique `CallID` that the origination switch has already assigned to that call.

Notes: The `direct-callid` transaction type is used only in conjunction with the `TRouteCall` and `TSingleStepTransfer` function calls. It is applied only to the call that is in progress, and does not apply to functions that involve in the creation of a new call, such as `TMakeCall`.

For T-Server for Nortel Communication Server 2000/2100, the `direct-callid` transaction type is also applied to the `TMuteTransfer` function.

direct-network-callid

With the transaction type `direct-network-callid`, the call reaches the destination DN directly from another location, and the `NetworkCallID` of the call is taken as the attribute for call matching. When a call arrives at the final destination, the destination T-Server identifies its `NetworkCallID`, and updates the call info if the `NetworkCallID` matches.

Use this transaction type when the destination switch has the capability to assign to an incoming call the same network-wide unique `NetworkCallID` that the origination switch has already assigned to that call.

Note: To support this transaction type, you must configure `Target Type` and `ISCC Protocol Parameters` fields of the corresponding `Switch Access Code` in the Configuration Layer. For information about settings that are specific for your T-Server type, refer to Part Two of this document.

direct-uui

With the transaction type `direct-uui`, so-called user-to-user information (UUI) is taken as the attribute for call matching. Some switches make it possible to send a small data packet along with a call. T-Server can use this data to recognize a call passed from one switch to another. The destination T-Server generates a local unique value for `UUI`, and then notifies the origination T-Server. The origination T-Server uses a provided value to mark the call coming from the origination location. The destination T-Server receives a call and checks whether it is marked with an exact `UUI` value. If so, the call is considered to be matched.

On the Avaya Communication Manager and the Aspect ACD, UUI is referred to as "user-to-user information." On the Siemens Hicom 300 switch with CallBridge, UUI is referred to as "Private User Data." On the Alcatel A4400/OXE switch, UUI is referred to as "correlator data."

Note: To support this transaction type, you must configure your switches to pass the UUI provided by your T-Server. You must also ensure that the trunks involved do not drop this data.

direct-notoken

With the transaction type `direct-notoken`, T-Server expects a call to arrive from another location to the destination DN specified in the request for routing service availability. When a call reaches the specified DN, T-Server processes the call as the expected externally-routed call.

Notes: This matching criterion is weak because any call that reaches the specified DN is considered to be the expected call. Genesys recommends that you use this transaction type only in a contact center subdivision that can only be reached from within the contact center (such as the second line of support, which customers cannot contact directly).

When using `direct` transaction types, Network T-Servers and load-sharing IVR Servers are not meant to act as destination T-Servers for call routing. Using Network T-Server with these transaction types requires special architecture.

dnis-pool

With the `dnis-pool` transaction type, T-Server reserves one of its DNIS access resources and waits for the call that has the same `DNIS` attribute as the name of the reserved DNIS access resource.

If the arrived call is matched successfully, the destination T-Server may update the value of the `DNIS` attribute of the call (along with `ConnID`, `UserData`, `CallType`, and `CallHistory`) with the value of the `DNIS` attribute of the original call. This occurs when the value of the `DNIS` attribute of the original call is specified as a value of the key-value pair `_ISCC_TRACKING_NUMBER_` in the `Extensions` attribute of the original client request.

The DNIS matching can be based on any number of digits out of all the digits that comprise the `DNIS` attribute. The number of digits that T-Server should use for DNIS matching is specified for the destination switch as the `ISCC Protocol Parameters` property of the Switch Access Code. The value syntax should be as follows:

`dnis-tail=<number-of-digits>`

For example, if this property is set to the `dnis-tail=7` value, ISCC matches only the last seven digits of a `DNIS`.

You must configure DNIS access resources in the switch; otherwise, ISCC fails to use this transaction type and sends `EventError` in response to the client application request.

Note: The `dnis-pool` transaction type is typically used for networks that employ a “behind the SCP” architecture, such as network IVR. Network T-Server for GenSpec and IServer are two examples of this, but other Network T-Servers might also be used in this architecture.

In Load-Balancing Mode

When T-Server uses load balancing for call routing with the `dnis-pool` transaction type, the following processes occur:

1. A client of the origination T-Server sends a request to pass a call to the location with a DNIS access resource specified in the key-value pair `iscc-selected-dnis`.
2. The origination T-Server distributes the request for a routing service to all destination T-Servers.
3. The destination T-Servers receive the request and check that the specified DNIS is not being used by another routing service request.
4. The origination T-Server expects to receive a positive response from each destination T-Server. If the origination T-Server receives a negative response from at least one T-Server, it sends an `EventError` to the client and clears all data about the request. If the origination T-Server receives the confirmation about routing service availability from all destination T-Servers, it processes the client's request and sends a corresponding message to the switch.
5. The origination switch processes the T-Server request and passes the call to the destination switch.
6. The call arrives at the destination switch, which generates an alerting event to one of the corresponding load-balanced destination T-Servers.
7. That destination T-Server processes the call and notifies the origination T-Server that the routing service was successful and deletes all information about the request.
8. The origination T-Server sends a routing service request cancellation to all other destination T-Servers.
9. The origination T-Server notifies the client that the routing service has been successful and deletes all information about the request.

pullback

`Pullback` is used in the following scenario, for those T-Servers that support it:

1. A call arrives at Site A served by a Network T-Server.
2. At Site A, a Network T-Server client requests to pass the call by means of ISCC routing to Site B served by a premise T-Server. Any transaction type except `reroute` or `pullback` can be specified in this request.
3. The call arrives at Site B and is either answered by an agent or delivered to a routing point.
4. A client of the premise T-Server at Site B sends a `TRouteCall` or `TSingleStepTransfer` request to transfer the call to the network.

5. The Site B premise T-Server notifies the Network T-Server about this request.
6. The network T-Server receives the notification and issues an `EventRouteRequest` to obtain a new destination.
7. After receiving the new destination information, the Network T-Server disconnects the call from its current premise location at Site B and attempts to route the call to the new destination.
8. The Site B premise T-Server stops tracking the call, which has disconnected from the premise's agent DN or routing point and is delivered to the network.
9. The network T-Server completes routing the call to its new destination.

Note: The transaction type `pullback` can only be used to return a call from a premise T-Server to the Network T-Server that serves the site from which the call was previously transferred.

reroute

`Reroute` is used in the following scenario, for those T-Servers that support it:

1. A call arrives at Site A served by a Network T-Server.
2. At Site A, a Network T-Server client requests to pass the call by means of ISCC to Site B served by a premise T-Server. Any transaction type except `reroute` or `pullback` can be specified in this request.
3. An agent at Site B answers the call.
4. A client of the premise T-Server at Site B sends a `TSingleStepTransfer` or `TRouteCall` request to transfer the call elsewhere (to a PSTN, to an agent, or to a routing point).
5. The Site B premise T-Server notifies the Network T-Server about this request and releases the call leg that resides at the agent's phone (using `TReleaseCall`) or at the Routing Point (using `TRouteCall` with the parameter `RouteTypeCallDisconnect`).
6. The Network T-Server receives the notification and reroutes the call to the requested destination by sending `EventRouteRequest` and attaching the call's user data.

Notes: The transaction type `reroute` can only be used to return a call from a premise T-Server to the Network T-Server that serves the site from which the call was previously transferred.

To perform multi-site operations that are initiated with `TRouteCall` and for which the `reroute` transaction type is requested, the origination T-Server must support the `RouteTypeCallDisconnect` subtype of `TRouteCall`.

route

With the transaction type `route` (aliased as `route-notoken`), a call from the origination location reaches a dedicated External Routing Point, and from there, it is routed to a destination DN.

To control configured External Routing Points, T-Server must register these DNs with the switch. Failure to register implies that the External Routing Point is not available for ISCC purposes. Client applications can register External Routing Points via T-Server for monitoring purposes only.

Point-to-Point (One-to-One)

In the Point-to-Point access mode, only one trunk line is used to access an External Routing Point (for example, VDN, CDN) at the destination site. See Figure 8.

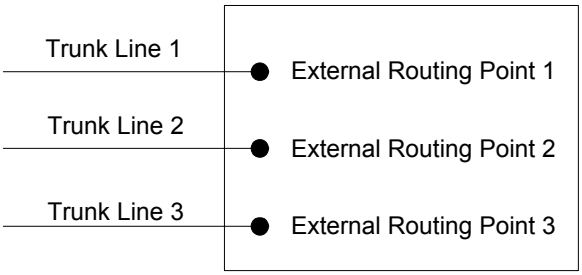

Figure 8: Point-to-Point Trunk Configuration

Note: Dedicated DNs of the `External Routing Point` type must be configured in a switch. See “Configuring Multi-Site Support” on page 102.

Multiple-to-Point (Multiple-to-One)

In the Multiple-to-Point access mode, trunk lines are assigned to the destination switch’s trunk group, from which calls are routed to the final destination. See Figure 9 on page 76.

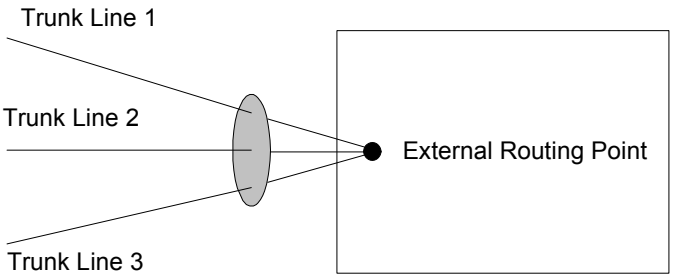

Figure 9: Multiple-to-Point Trunk Configuration

With this configuration, all calls reach the same External Routing Point. The `DNIS` attribute of a specific call differs from that of other calls and uniquely identifies the trunk from which the call arrived.

Note: To switch to this operating mode, you must configure the `route-dn` configuration option for T-Server.

route-uui

The `route-uui` transaction type employs the dedicated External Routing Point feature of the `route` transaction type (page 75) and the UUI matching feature of the `direct-uui` transaction type (page 71). This transaction type accommodates those switches that require a designated External Routing Point even though they use UUI for tracking.

Note: To support this transaction type, you must configure your switches to pass the UUI provided by your T-Server. You must also ensure that the trunks involved do not drop this data.

T-Server Transaction Type Support

Table 3 shows which transaction types are supported by a specific T-Server. Use this table to determine the transaction types that are available for use with your T-Server. This applies both to the `cast-type` you specify in the configuration options for your T-Server, and to any client-designated route-type requests specified for transfers of calls. A blank table cell indicates that T-Server does not support a certain transaction type.

Table 3: T-Server Support of Transaction Types

T-Server Type	Transaction Type										
	route		re-route	direct-callid	direct-uui / route-uui	direct-no-token	direct-ani	direct-digits	direct-network-callid	dnis-pool	pull-back
	one-to-one	multiple-to-one									
Aastra MXONE CSTA I	Yes			Yes[a]		Yes	Yes[a]				
Alcatel A4200/OXO	Yes			Yes		Yes	Yes				
Alcatel A4400/OXE	Yes			Yes[a,b,c]	Yes[d]	Yes	Yes[a]		Yes[e]		
Aspect ACD	Yes	Yes		Yes[c]		Yes[f]	Yes[f]				
Avaya Communication Manager	Yes				Yes	Yes	Yes				
Avaya INDeX	Yes					Yes	Yes[b]				
Avaya TSAPI	Yes				Yes	Yes	Yes				
Cisco UCCE	Yes					Yes	Yes				
Cisco Unified Communications Manager	Yes			Yes		Yes	Yes				
DataVoice Dharma	Yes			Yes		Yes	Yes				
Digitro AXS/20	Yes			Yes		Yes					
EADS Intecom M6880	Yes			Yes		Yes	Yes				
EADS Telecom M6500	Yes			Yes		Yes	Yes				
eOn eQueue	Yes			Yes		Yes					
Fujitsu F9600	Yes					Yes					

Table 3: T-Server Support of Transaction Types (Continued)

T-Server Type	Transaction Type										
	route		re-route	direct-callid	direct-uui / route-uui	direct-no-token	direct-ani	direct-digits	direct-network-callid	dnis-pool	pull-back
	one-to-one	multiple-to-one									
Huawei C&C08	Yes			Yes							
Huawei NGN	Yes					Yes	Yes				
Mitel MiTAI	Yes					Yes	Yes		Yes[g]		
NEC NEAX/APEX	Yes			Yes		Yes	Yes				
Nortel Communication Server 2000/2100	Yes			Yes[f]		Yes[f]	Yes[f]				
Nortel Communication Server 1000 with SCCS/MLS	Yes			Yes		Yes	Yes		Yes		
Philips Sopho iS3000	Yes			Yes		Yes	Yes				
Radvision iContact	Yes		Yes								Yes
Samsung IP-PCX IAP	Yes			Yes		Yes					
Siemens Hicom 300/HiPath 4000 CSTA I	Yes			Yes	Yes[d]	Yes	Yes				
Siemens HiPath 3000	Yes			Yes		Yes					
Siemens HiPath 4000 CSTA III	Yes				Yes[d]	Yes	Yes				

Table 3: T-Server Support of Transaction Types (Continued)

T-Server Type	Transaction Type										
	route		re-route	direct-callid	direct-uui / route-uui	direct-no-token	direct-ani	direct-digits	direct-network-callid	dnis-pool	pull-back
	one-to-one	multiple-to-one									
Siemens HiPath DX	Yes				Yes[h]	Yes	Yes[i]				
SIP Server	Yes		Yes		Yes[j]	Yes					Yes
Spectrum	Yes	Yes		Yes		Yes[f]	Yes[f]				
Tadiran Coral	Yes			Yes		Yes	Yes				
Teltronics 20-20	Yes			Yes		Yes	Yes				
Tenovis Integral 33/55	Yes			Yes		Yes	Yes				
Network T-Servers											
AT&T											
Concert											
CRSP											Yes
DTAG			Yes								
GenSpec	Yes	Yes	Yes							Yes	
IVR Server, using network configuration	Yes	Yes	Yes							Yes	Yes
KPN			Yes								
ISCP											
MCI											
NGSN	Yes									Yes	Yes
Network SIP Server	Yes					Yes	Yes			Yes	
Sprint	Yes										

Table 3: T-Server Support of Transaction Types (Continued)

T-Server Type	Transaction Type										
	route		re-route	direct-callid	direct-uui / route-uui	direct-no-token	direct-ani	direct-digits	direct-network-callid	dnis-pool	pull-back
	one-to-one	multiple-to-one									
SR-3511											
Stentor											

a. Not supported in the case of function `TRouteCall` on a Virtual Routing Point: a Routing Point can be simulated using a hunt group with calls being deflected or transferred from the hunt-group member when routing. When a two-step (typically mute) transfer is used on such a hunt-group member, `CallID` and `ANI` usually change; thus, the `direct-callid` and `direct-ani` types do not work.

b. Not supported in the case of function `TSingleStepTransfer` when the T-Server service is simulated using a two-step transfer to the switch. In this case, `CallID` and `ANI` change; thus, the `direct-callid` and `direct-ani` types do not work.

c. Not supported if two T-Servers are connected to different nodes.

d. There are some switch-specific limitations when assigning CSTA correlator data UUI to a call.

e. Supported only on ABCF trunks (Alcatel internal network).

f. To use this transaction type, you must select the `Use Override` check box on the `Advanced` tab of the DN `Properties` dialog box.

g. Supported only for `TRouteCall` requests made from a Native Routing Point.

h. Not supported if a `TMakeCall` request is made.

i. Not supported if a `TInitiateTransfer` or `TInitiateConference` request is made from an outgoing call on a device.

j. SIP Server supports the `direct-uui` type.

Transfer Connect Service Feature

The Transfer Connect Service (TCS) feature supports transfer connect services available on some telephony networks. When this feature is enabled, ISCC passes user data to remote locations to which calls are transferred or conferenced using transfer connect services.

Procedure:
Activating Transfer Connect Service

Start of procedure

1. Open the T-Server `Application's Properties` dialog box.
2. Click the `Options` tab.
3. Set the `tcs-use` configuration option to `always`.
4. Set the `tcs-queue` configuration option to the number of a DN on the origination switch.

   ISCC uses this DN as an intermediate step when sending calls to the remote location. The DN that is configured as `tcs-queue` receives attached data indicating the Feature Access Code (FAC) needed to reach the remote site. After a call is directed to the DN with data, a monitoring application takes the data and generates the required DTMF (dual-tone multifrequency) tones to redirect the call through the network to the remote location.
5. When you are finished, click `Apply`.
6. Click `OK` to save your changes and exit the `Properties` dialog box.

End of procedure

Note: With T-Server for Avaya Communication Manager, you can use `TRouteCall` with `RouteTypeOverwriteDNIS` to initiate the playing of DTMF tones. This is done through the use of another intermediate DN (typically, an announcement port configured to give the silent treatment), to which the call is routed. When the call is established on this DN, T-Server requests that the digits sent in the `DNIS` field of the `TRouteCall` be played by using the `ASAI-send-DTMF-single` procedure.

ISCC/Call Overflow Feature

The Inter Server Call Control/Call Overflow (ISCC/COF) feature of T-Server, that supports *passive external routing*, is specifically designed to handle calls delivered between sites without an explicitly defined destination location. Such scenarios include contact center overflows and manual call transfers.

An *overflow situation* occurs when a call comes into a contact center where all agents are currently busy. In this situation, the switch can transfer (overflow) the incoming call to another site where there is an available agent.

T-Server uses two methods to handle call overflow and manual transfer scenarios. The first method is based on `NetworkCallID` matching and the second method is based on `ANI/OtherDN` matching.

When connected to each other via switch-specific networks, switches of some types can pass additional information along with transferred calls. This information may contain the `NetworkCallID` of a call, which is a networkwide unique identifier of the call.

When connected via a regular PSTN, switches of all types can send the `ANI` and/or `OtherDN` attributes to the destination switch during any call transfer operation.

While all T-Servers support the ISCC/COF feature using the `ANI` and/or `OtherDN` attributes, only a few support this feature using the `NetworkCallID` attribute. Table 4 shows the T-Server types that provide the `NetworkCallID` of a call.

Table 4: T-Server Support of NetworkCallID for ISCC/COF Feature

T-Server Type	Supported NetworkCallID Attribute
Alcatel A4400/OXE[a]	Yes
Aspect ACD	Yes
Avaya Communication Manager[a,b]	Yes
Avaya TSAPI[a,b]	Yes
Cisco UCCE	Yes
Mitel MiTAI[a]	Yes
Nortel Communication Server 2000/2100[a]	Yes
Nortel Communication Server 1000 with SCCS/MLS[a]	Yes
SIP Server[a]	Yes
Spectrum	Yes

a. Supported only if the `match-flexible` configuration parameter is used.

b. ISCC/COF is cross-compatible between T-Server for Avaya Communication Manager and T-Server for Avaya TSAPI.

The ISCC/COF feature can use any of the three attributes (`NetworkCallID`, `ANI`, or `OtherDN`) as criteria for matching the arriving call with an existing call at another location. Consequently, the attribute that is used determines what

`ConnID`, `UserData`, `CallType`, and `CallHistory` are received for the matched call from the call's previous location.

Warning! Depending on the switch platform, it may be possible to inherit the ANI attribute after routing a call to a remote destination, and after performing a single-step transfer and other telephone actions. However, ISCC/COF works properly only in scenarios where the `ANI` attribute on the destination T-Server is represented by exactly the same unique digit string as on the origination T-Server.

Typically, the `ANI` attribute represents the original call identifier (customer phone number), which guarantees that the attribute remains unique.

Note: When the ISCC/COF feature is in use, the Number Translation feature becomes active. For more information on feature configuration, see "Number Translation Feature" on page 85.

ISCC/COF Call Flow

Figure 10 shows the sequence of steps that occur in an ISCC/COF scenario when a call is made or transferred by an agent at Site A to a DN at Site B, or when a call is overflowed from Site A to Site B.

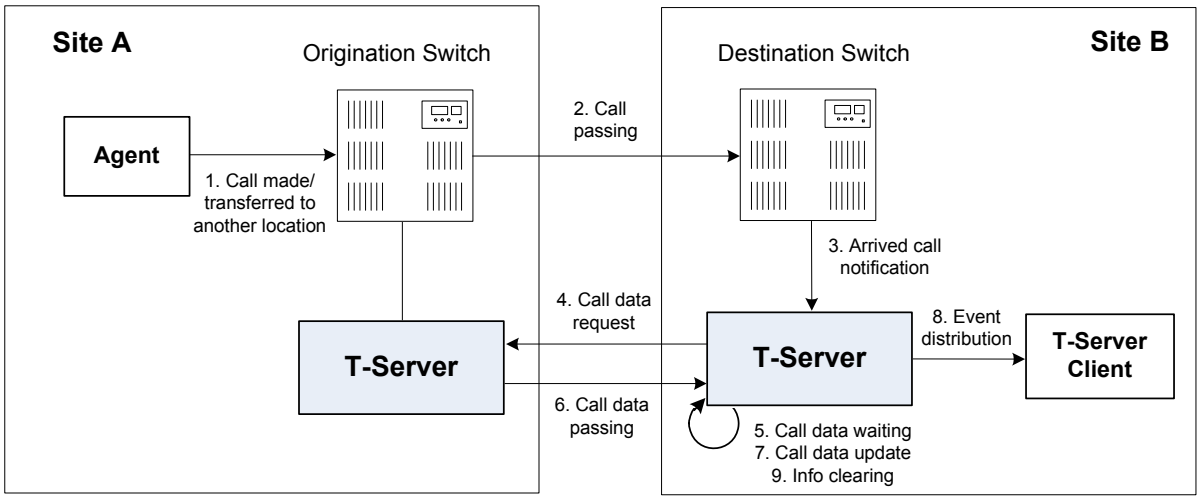

Figure 10: Steps in the ISCC/COF Process

Step 1

An agent makes or transfers a call manually to another location or a call is overflowed from Site A (origination location) to Site B (destination location).

Step 2

Switch A (the origination switch) passes the call to Switch B (the destination switch).

Step 3

The call reaches the destination switch, which notifies the destination T-Server about the arrived call.

Step 4

The destination T-Server verifies with remote locations whether the call overflowed at any of them.

To determine which calls to check as possibly having overflowed, T-Server relies on the `Switch` object and the presence of DNs on the `Switch` configured as the `Access Resource` type with the `Resource Type` set either to `cof-in` (COF-IN DNs) or to `cof-not-in` (COF-NOT-IN DNs):

T-Server skips an arriving call when one of following conditions is met:

- The call arrives at a DN configured as an Enabled COF-NOT-IN DN.
- COF-IN DNs are configured, but the call arrives at a DN other than one of the configured COF-IN DNs or to a COF-IN DN which is Disabled.

In all other cases, the call is checked for overflow.

To determine which location the call arrived from, T-Server checks the call type and checks whether the call has the `NetworkCallID`, `ANI`, or `OtherDN` attribute:

- If the call is not an inbound call, the request for call data is sent to all remote locations *except* those whose Switch Access Code has the ISCC Call Overflow `Parameters` property set to `inbound-only=true`.
- If the call of any type has the `NetworkCallID` attribute, the destination T-Server sends a request for call data to the remote locations of the same switch type as the destination location if their Switch Access Codes have the ISCC Call Overflow `Parameters` property set to `match-callid`.
- If the call of any type has the `ANI` or `OtherDN` attribute, the request for call data is sent to remote locations whose Switch Access Code has the ISCC Call Overflow `Parameters` property set to `match-ani`.

Step 5

The destination T-Server waits (suspending events related to that call) for the call data from the remote T-Server for the time interval specified in the `cof-ci-req-tout` configuration option. Within this interval, T-Server holds any events related to the call. In addition, the `cof-ci-defer-delete` option on the origination T-Server establishes the time interval only after which that T-Server deletes the call information. And the `cof-ci-wait-all`, if set to `true`,

forces the origination T-Server to wait for responses related to possible call overflow situations before updating call data.

Step 6

The T-Server at the location from which the call was transferred or overflowed sends call data to the requesting T-Server.

Step 7

If a positive response to the call-data request is received, T-Server updates `ConnID`, `UserData`, `CallType`, and `CallHistory`, distributes all suspended events related to that call, and deletes all information regarding the transaction (Step 9).

Step 8

If the timeout set by `cof-ci-req-tout` expires, T-Server distributes all suspended events, and starts the timeout specified by the `cof-rci-tout` option. If a positive response is received within the timeout set by `cof-rci-tout`, T-Server updates the `ConnID`, `UserData`, `CallType`, and `CallHistory`, and notifies client applications by distributing `EventPartyChanged`.

Step 9

T-Server deletes all information regarding the transaction when one of these results occurs:

- The first positive response to the call-data request is received.
- Negative responses from all queried locations are received.
- The timeout specified by the `cof-rci-tout` option expires.

Number Translation Feature

The Number Translation feature of T-Server extends the ISCC/COF and `direct-ani` transaction type functions to provide more flexibility for handling calls distributed across multiple sites. T-Server translates the input string (ANI string) into a number defined by the translation rules. This processing is called number translation. T-Servers participating in handling calls at multiple sites exchange the translated numbers in order to match the call instances.

The translation process involves two algorithms, one for rule selection and the other for the actual translation. Through the first algorithm, T-Server selects a rule that will be used for number translation. Through the second algorithm, T-Server translates the number according to the selected rule definition. See “Number Translation Rules” on page 86 for more information on configuring rules for your environment.

Number translation occurs as follows:

1. The switch reports a number, typically via `AttributeANI`.
2. T-Server evaluates all configured inbound rules to determine which one is the best fit for the received number. The best fit is determined by comparing the length of, and the specific digits in, the input number with the inbound pattern of each configured rule. See "Rule Examples" on page 91 for specific examples.
3. T-Server translates the number according to the selected rule.

To enable T-Server to translate numbers, you must perform specific configuration tasks that are associated with translation. See "Configuring Number Translation" on page 93.

Number Translation Rules

T-Server uses the number translation rules that you define in the T-Server configuration object in two ways:

- Rule selection—To determine which rule should be used for number translation
- Number translation—To transform the number according to the selected rule

Using ABNF for Rules

The number translation rules must conform to the following syntax, represented using Augmented Backus-Naur Form (ABNF) notation. For more information about ABNF, see RFC 2234, "Augmented BNF for Syntax Specifications: ABNF."

Note: The following notation explanations begin with the highest level notation. Each explanation includes the name of a component notation and a basic definition of each component that it contains. Some components require more detailed definitions, which are included later in this section.

Common Syntax Notations

Syntax notations common to many of these rules include:

- `*`—Indicates that `0` (zero) to an infinite number of the item following this symbol are acceptable.
- `1*`—Indicates that one repetition is required. For T-Server, only one instance is acceptable.
- `/`—Indicates that any of the items mentioned, or a combination of those items, is acceptable.

Component Notations

Component notations include:

- `dialing-plan = *dialing-plan-rule`

  where:

  - `dialing-plan-rule` represents the name of the rule. Each rule must have a unique name. There are no other naming restrictions, and you do not need to model your names according to the examples in this chapter.

  The rules are represented as separate options in the configuration. Also, fields from a rule are represented as parameters in a single option string.

- `rule = [name] in-pattern [out-pattern]`

  where:

  - `[name]` is the name for the rule option, for example, `rule-01`. In ABNF notation, the brackets `[]` indicate that 0 or 1 instance of the component is required. However, for T-Server, a name is required.
  - `in-pattern` is the part of the rule to which T-Server looks when attempting to match the input number.
  - `[out-pattern]` is the part of the rule that instructs T-Server on how to translate the input number into the required format. The brackets indicate that either 0 or 1 instance is required. You must create an `out-pattern` for number translation rules.

- `name = *( ALPHA / DIGIT / "-")`

  where:

  - `ALPHA` indicates that letters can be used in the name for the rule option.
  - `DIGIT` indicates that numbers can be used in the name for the rule option.
  - `"-"` indicates that a dash (-) can also be used in the option name, for example, `rule-01`.

- `in-pattern = 1*(digit-part / abstract-group)`

  where:

  - `digit-part` represents numbers. T-Server uses this when selecting the most appropriate rule from the entire dialing plan.
  - `abstract-group` represents one or more letters with each letter representing one or more numbers. T-Server uses this when transforming a dial string.

  For example, `[1-9]` is the `digit-part` (representing a range of numbers) and `ABBB` is the `abstract-group` for `in-pattern=[1-9]ABBB`.

- `out-pattern = 1*(symbol-part / group-identifier) *param-part`

  where:

  - `symbol-part` represents digits, symbols, or a combination. Symbols are rarely used. They are not used in the United States.

- `group-identifier` are letters that represent groups of numbers. A letter in the `out-pattern` represents one or more digits, based on the number of times the letter is used in the `in-pattern`.
- `*param-part` represents an additional parameter, such as `phone-context`. Reminder: an asterisk means that 0 to an infinite number of these are acceptable.

For example, in `rule-04; in-pattern=1AAABBBCCC;out-pattern=91ABC, 91` is the `symbol-part`; `A, B,` and `C` are `group-identifiers` in the `out-pattern`, each representing three digits, since there are three instances of each in the `in-pattern`.

Note: Prefix an `out-pattern` value with a plus sign (+) for the inbound rule when the output must be in a global format (E.164 format).

- `digit-part = digits / range / sequence`

  where:
  - `digits` are numbers 0 through 9.
  - `range` is a series of digits, for example, 1-3.
  - `sequence` is a set of digits.
- `symbol-part = digits / symbols`

  where:
  - `digits` are numbers 0 through 9.
  - `symbols` include such characters as `+, -,` and so on.
- `range = "[" digits "-" digits "]" group-identifier`

  where:
  - `"[" digits "-" digits "]"` represents the numeric range, for example, `[1-2]`.
  - `group-identifier` represents the group to which the number range is applied.

    For example, `[1-2]` applies to group identifier `A` for `in-pattern=[1-2]ABBB`. When T-Server evaluates the rule to determine if it matches the number, it examines whether the first digit of the number, identified as `group-identifier A`, is `1` or `2`.
- `sequence = "[" 1*(digits [","] ) "]" group-identifier`

  where:
  - `"[" 1*(digits [","] ) "]"` represents a sequence of digits, separated by commas, and bracketed. T-Server requires that each digit set have the same number of digits. For example, in `[415,650]` the sets have three digits.
  - `group-identifier` represents the group to which the number sequence is applied.

For example, in `in-pattern=1[415,650]A*B, [415,650]` applies to `group-identifier A`. When T-Server evaluates the rule to determine if it matches the number, it examines whether the three digits (`group-identifier A`) following the 1 in the number are `415` or `650`.

- `abstract-group = fixed-length-group / flexible-length-group / entity`

  where:

  - `fixed-length-group` specifies a group composed of a specific number of digits and determined by how many times the group identifier is included in the `in-pattern`. For example, for `in-pattern=1AAABBBCCCC`, there are three digits in group `A` and `B` but four in group `C`.

    When you create an `out-pattern`, you include the group identifier only once because the `in-pattern` tells T-Server how many digits belong in that group. For example, `rule-04` (see page 91) is `in-pattern=1AAABBBCCCC; out-pattern=91ABC`.
  - `flexible-length-group` specifies a group composed of 0 or more digits in the group represented by the `group-identifier`. For example, in `in-pattern=1[415,650]A*B, *B` represents the flexible length group containing the remaining digits in the number.
  - `entity` represents digits defined for a specific purpose, for example, country code.

  The component `abstract-group` is used only for the `in-pattern`.

- `fixed-length-group = 1*group-identifier`

  See the earlier explanation under `abstract-group`.

- `flexible-length-group = "*" group-identifier`

  See the earlier explanation under `abstract-group`.

- `entity = "#" entity-identifier group-identifier`

  where:

  - "`#`" indicates the start of a Country Code `entity-identifier`.
  - `entity-identifier` must be the letter `C` which represents Country Code when preceded by a pound symbol (`#`). Any other letter following the # causes an error.
  - `group-identifier` represents the Country Code group when preceded by `#C`.

  The `entity` component is a special group that assumes some kind of predefined processing, such as the Country Code detection.

- `param-part = ";" param-name "=" param-value`

  where:

  - "`;`" is a required separator element.
  - `param-name` is the name of the parameter.
  - "`=`" is the next required element.

- `param-value` represents the value for `param-name`.

- `param-name = "ext" / "phone-context" / "dn"`

  where:
  - `"ext"` refers to extension.
  - `"phone-context"` represents the value of the `phone-context` option configured on the switch.
  - `"dn"` represents the directory number.
- `param-value = 1*ANYSYMBOL`

  where:
  - `ANYSYMBOL` represents any number, letter, or symbol with no restrictions.
- `group-identifier = ALPHA`
- `entity-identifier = ALPHA`
- `digits = 1*DIGIT`
- `symbols = 1*("-" / "+" / ")" / "("/ ".")`

Recommendations for Rule Configuration

The configuration of rules for inbound numbers usually depends on the settings in the corresponding PBX. These settings often define the form in which the PBX notifies its client applications about the number from which an inbound call is coming.

As a general guideline, configure rules that define how to process calls from:

- Internal numbers.
- External numbers within the same local dialing area.
- External numbers within the same country.
- International numbers.

Rules for inbound numbers, typically for North American locations, might look like this:

1. Two rules to transform internal numbers (extensions):

   ```
   name=rule-01;in-pattern=[1-9]ABBB;out-pattern=AB
   name=rule-02;in-pattern=[1-9]ABBBB;out-pattern=AB
   ```
2. A rule to transform local area code numbers (in `333-1234` format in this example):

   ```
   name=rule-03;in-pattern=[1-9]ABBBBBB;out-pattern=+1222AB
   ```
3. A rule to transform U.S. numbers (in `+1(222)333-4444` format):

   ```
   name=rule-04;in-pattern=1AAAAAAAAAA;out-pattern=+1A
   ```
4. A rule to transform U.S. numbers without the `+1` prefix (in `(222)333-4444` format):

   ```
   name=rule-05;in-pattern=[2-9]ABBBBBBBBB;out-pattern=+1AB
   ```

5. A rule to transform U.S. numbers with an outside prefix (in `9 +1(222)333-4444` format):

   ```
   name=rule-06;in-pattern=91AAAAAAAAAA;out-pattern=+1A
   ```

6. A rule to transform international numbers with an IDD (international dialing digits) prefix (in `011 +44(111)222-3333` format):

   ```
   name=rule-07;in-pattern=011*A;out-pattern=+A
   ```

7. A rule to transform international numbers without an IDD prefix (in `+44(111)222-3333` format):

   ```
   name=rule-08;in-pattern=[2-9]A*B;out-pattern=+AB
   ```

Rule Examples

This section provides examples of six rules that are configured as options in the Genesys Configuration Database. It also provides examples of how T-Server applies rules to various input numbers.

Rules

rule-01 `in-pattern=[1-8]ABBB;out-pattern=AB`

rule-02 `in-pattern=AAAA;out-pattern=A`

rule-03 `in-pattern=1[415,650]A*B;out-pattern=B`

rule-04 `in-pattern=1AAABBBCCCC;out-pattern=91ABC`

rule-05 `in-pattern=*A913BBBB;out-pattern=80407913B`

rule-06 `in-pattern=011#CA*B;out-pattern=9011AB`

Examples

Here are examples of how T-Server applies configured above rules to various input numbers.

Example 1 T-Server receives input number `2326`.

As a result of the rule selection process, T-Server determines that the matching rule is `rule-01`:

```
name=rule-01;in-pattern=[1-8]ABBB;out-pattern=AB
```

The matching count for this rule is `1`, because Group `A` matches the digit `2`.

As a result of the parsing process, T-Server detects two groups: Group `A = 2` and Group `B = 326`.

T-Server formats the output string as `2326`.

Example 2 T-Server receives input number `9122`.

As a result of the rule selection process, T-Server determines that the matching rule is `rule-02`:

```
name=rule-02;in-pattern=AAAA;out-pattern=A
```

The matching count for this rule is `0`; however, the overall length of the input number matches that of the `in-pattern` configuration.

As a result of the parsing process, T-Server detects one group: Group `A = 9122`.

T-Server formats the output string as `9122`.

Example 3 T-Server receives input number `16503222332`.

As a result of the rule selection process, T-Server determines that the matching rule is `rule-03`:

```
name=rule-03;in-pattern=1[415,650]A*B;out-pattern=B
```

The matching count for this rule is `4`, because the first digit matches and all three digits in Group `A` match.

As a result of the parsing process, T-Server detects two groups: Group `A = 650` and Group `B = 3222332`.

T-Server formats the output string as `3222332`.

Example 4 T-Server receives input number `19253227676`.

As a result of the rule selection process, T-Server determines that the matching rule is `rule-04`:

```
name=rule-04;in-pattern=1AAABBBCCCC;out-pattern=91ABC
```

The matching count for this rule is `1`, because the first digit matches.

As a result of parsing process, T-Server detects three groups: Group `A = 925`, Group `B = 322`, and Group `C = 7676`.

T-Server formats the output string as `919253227676`.

Example 5 T-Server receives input number `4089137676`.

As a result of rule selection process, T-Server determines that the matching rule is `rule-05`:

```
name=rule-05;in-pattern=*A913BBBB;out-pattern=80407913B
```

The matching count for this rule is `3`, because three digits match.

As a result of the parsing process, T-Server detects two groups: Group `A = 408` and Group `B = 7676`.

T-Server formats the output string as `804079137676`.

Example 6 T-Server receives input number `011441112223333`.

As a result of the rule selection process, T-Server determines that the matching rule is `rule-06`:

```
name=rule-06;in-pattern=011#CA*B;out-pattern=9011AB
```

The matching count for this rule is `3`, because three digits match.

As a result of the parsing process, T-Server detects two groups: Group `A = 44` and Group `B = 1112223333`.

T-Server formats the output string as `9011441112223333`.

Procedure:
Configuring Number Translation

Purpose: To configure the Number Translation feature in T-Server to provide more flexibility for handling calls distributed across multiple sites.

Overview

- The Number Translation feature becomes active when the ISCC/COF feature and/or the `direct-ani` transaction type are used.
- This configuration procedure must be completed within the T-Server `Application` object corresponding to your T-Server.

Start of procedure

1. Open the T-Server `Application's` Properties dialog box.
2. Click the `Options` tab.
3. Create a new section called `extrouter` or open an existing section with this name.
4. Create a new option called `inbound-translator-<n>`. This option points to another section that describes the translation rules for inbound numbers.
5. In this section, create one configuration option for each rule. Specify the rule name as the option name. The values of these options are the rules for the number translation.

   For the option description and its valid values, see Chapter 12, "T-Server Common Configuration Options," on page 229.
6. When you are finished, click `Apply`.
7. Click `OK` to save your changes and exit the `Properties` dialog box.

End of procedure

Network Attended Transfer/Conference Feature

The Network Attended Transfer/Conference (NAT/C) feature is designed to enable agents working in multi-site contact centers to consult with each other before making call transfers or conferences, regardless of whether both agents work at the same or different sites. It also enables the agent who requests a consultation to maintain his or her conversation with the customer while the system is looking for an available agent and setting up the consultation call.

The NAT/C feature does not rely on the call transfer capabilities of the local switch.

There are two modes in which the network attended transfer/conference can be performed: *direct* and *URS-controlled*. Figure 11 shows the sequence of steps that occur in *URS-controlled* mode, when Agent A, who is handling a customer call, requests a consultation with another agent, and URS (Universal Routing Server) selects Agent B, who is working at another site. The *direct* mode is similar to the *URS-controlled* mode, with the difference that URS is not involved in the process (Step 2 and Step 3 are omitted).

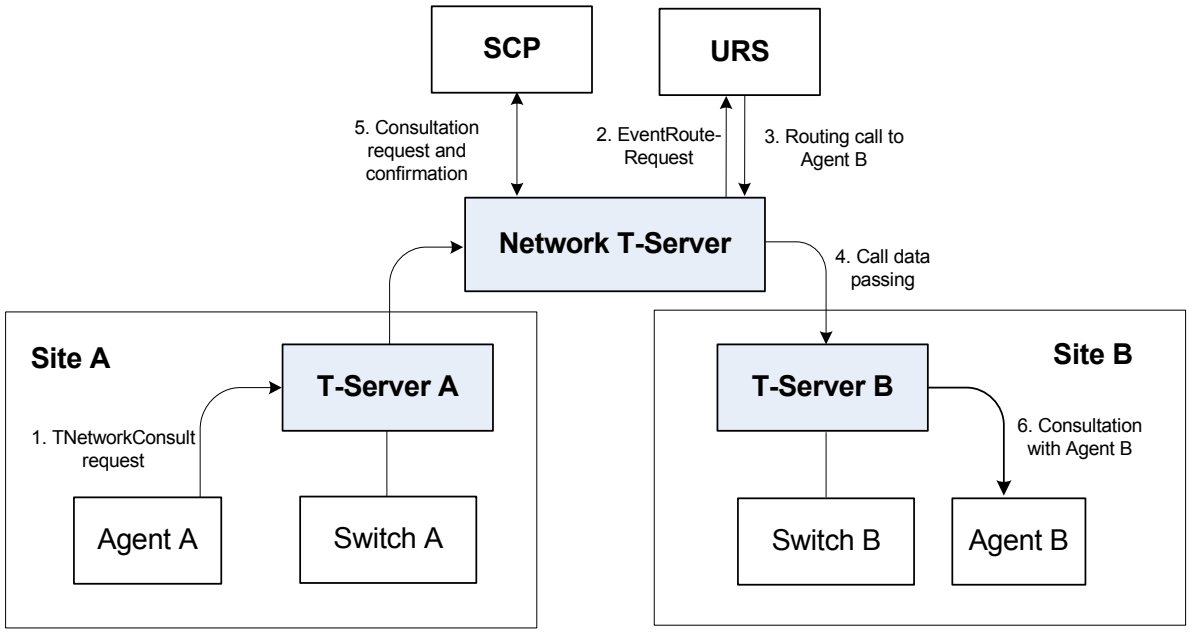

Figure 11: Steps in the NAT/C Process in URS-Controlled Mode

Step 1

Agent A makes a request for a consultation with another agent. A `TNetworkConsult` request is relayed to the Network T-Server. Depending on the parameter settings of the `TNetworkConsult` request, the NAT/C feature will operate in either *direct* or *URS-controlled* mode. For more information, see the *Platform SDK 8.x .NET (or Java) API Reference.*

Step 2

(*URS-controlled* mode only.) The Network T-Server sends `EventRouteRequest` to URS.

Step 3

(*URS-controlled* mode only.) URS locates an available agent at Site B and instructs the Network T-Server to route the call to Agent B. The Network

T-Server confirms the initiation of the network transfer by sending `EventNetworkCallStatus` to T-Server A, which then relays it to Agent A.

Step 4

The Network T-Server proceeds to obtain the access number from T-Server B, and passes the call data to T-Server B. (See "ISCC Call Data Transfer Service" on page 61 for details.)

Step 5

The Network T-Server instructs the Service Control Point (SCP) to initiate a new voice path with Agent B. Once the connection is confirmed, the Network T-Server distributes `EventNetworkCallStatus` to both T-Server A and T-Server B, which then relay it to Agent A and Agent B respectively, to indicate that the consultation call is being established.

The Network T-Server also distributes `EventRouteUsed` to URS to confirm successful routing of the call to the selected agent.

Step 6

At this point, the customer is on hold, and Agent A is consulting with Agent B. Agent A can do one of the following:

- End the consultation and retrieve the original customer call
- Alternate between Agent B and the customer
- Set up a conference call with Agent B and the customer
- Transfer the customer call to Agent B

Note: All T-Servers support NAT/C requests with `AttributeHomeLocation` provided that this attribute identifies a network location that is capable of processing such requests. Refer to the *Network T-Server Deployment Guides* to determine whether a specific Network T-Server can process these requests.

Event Propagation Feature

The Event Propagation feature complements the ISCC and ISCC/COF features by distributing updated user data and party-related events to remote T-Servers. This feature is used when a call is being made, transferred, or conferenced to another location, and when, as a result, one or more instances of the call reside at one location while other call instances reside at another location. In this scenario, when a client at one location makes changes to user data, updated user data is passed *(propagated)* to T-Servers at other locations.

The Event Propagation feature consists of User Data update propagation and Party Events propagation.

User Data Propagation

User data propagation takes place when a client at one location makes changes to user data associated with a call that was made, transferred, conferenced, or routed to other locations. The remote clients involved with the call are notified about the changes with `EventAttachedDataChanged`.

When T-Server receives a local update to user data (that is, when a client of this T-Server has changed the call's user data), T-Server determines if parties at remote locations are involved with the call and, if so, sends (propagates) the updated user data to the T-Servers at remote locations.

When T-Server receives a remote update to user data (that is, when a client of a remote T-Server has changed the call's user data and the remote T-Server has used the Event Propagation feature to send the updated user data), T-Server:

1. Updates the user data of the corresponding local call.
2. Determines if parties at other remote locations are involved with the call and, if so, propagates the updated user data to T-Servers at other remote locations.

The locations to which user data is propagated are selected based on a call distribution topology. That is, the updated user data is passed directly to the location to which a call was sent and to the location from which the call was received, excluding the location from which the update was received.

For example, consider a call made from location A to location B, and then conferenced from location B to location C. The three instances of the call reside at different locations: the first instance is at location A, the second instance is at location B, and the third instance is at location C. The Event Propagation feature is employed in the following scenarios:

- When T-Server at location A receives a local update to user data, it notifies T-Server at location B (to which it sent the call) about changes to the call's user data. Thus, T-Server at location B receives a remote update to user data and, in turn, notifies T-Server at location C (to which it sent the call) about these changes.

  Although T-Server at location C receives a remote update to user data, it does not pass the notification to any other T-Servers, because it did not send the call to any other locations. As mentioned earlier, T-Servers at locations B and C update the user data of the corresponding local calls and notify their clients about the changes with `EventAttachedDataChanged`.
- When T-Server at location B receives a local update to user data, it notifies T-Server at location C (to which it sent the call) and T-Server at location A (from which it received the call) about changes to the call's user data. Thus, T-Servers at locations C and A receive a remote update to user data.

Because T-Server at location C did not send the call to any other locations, and T-Server at location A originated the call, neither of these T-Servers passes the notification to any other T-Servers. T-Servers at locations C and A update the user data of the corresponding local calls and notify their clients about the changes with `EventAttachedDataChanged`.

- When T-Server at location C receives a local update to user data, it notifies T-Server at location B (from which it received the call) about changes to the call's user data. Thus, T-Server at location B receives a remote update to user data and, in turn, notifies T-Server at location A (from which it received the call) about these changes.

  Although T-Server at location A receives a remote update to user data, it does not pass the notification to any other T-Servers, because it originated the call. T-Servers at locations B and A update the user data of the corresponding local calls and notify their clients about the changes with `EventAttachedDataChanged`.

When a call is distributed between location A and location C using location B, and is then deleted on location B, propagation between locations A and C still occurs through the transit node at location B.

Party Events Propagation

Party events propagation takes place when a transfer or a conference is completed for a call that was made to or from one or more remote locations, or when a conference party is removed from the conference.

In these cases, the Event Propagation feature distributes party events, such as `EventPartyChanged`, `EventPartyAdded`, and `EventPartyDeleted`, to remote locations involved with the call, according to appropriate call model scenarios.

For example, consider a call made from DN 1 to DN 2 on location A. A `TInitiateConference` request is then issued for DN 2 to transfer the call to external DN 3 on location B. That transfer is made by means of ISCC routing. When this conference is completed on location A, the Event Propagation feature sends `EventPartyChanged` to location B and distributes this event to involved client applications that are connected to location B and registered for DN 3. After that, if a party of the conference is removed from the conference (for example, a party on DN 2), the Event Propagation feature sends `EventPartyDeleted` to location B and distributes this event to client applications registered for DN 3.

If a call involved in the propagation has no local parties but has two or more remote parties, the party events propagation is processed in the same manner as the propagation of user data updates.

For a complete event flow in such scenarios, refer to the *Genesys Events and Models Reference Manual.*

Switch Partitioning

A multi-site environment with switch partitioning or intelligent trunks can be defined as a configuration of multiple virtual switches (or `Switch` objects) that are defined in Configuration Manager under a single `Switching Office` object representing a physical switch. Each `Switch` object has its own instance of a T-Server application. All T-Server applications connect to the switch via the same or different CTI link or a gateway. (See Figure 12.)

When the Event Propagation feature is active, updated user data and party-related events—`EventPartyChanged`, `EventPartyDeleted`, and `EventPartyAdded`—are propagated to T-Servers that are involved in call transactions, such as transfer or conference. However, with switch partitioning, the call instances may reside at one partition or at different partitions.

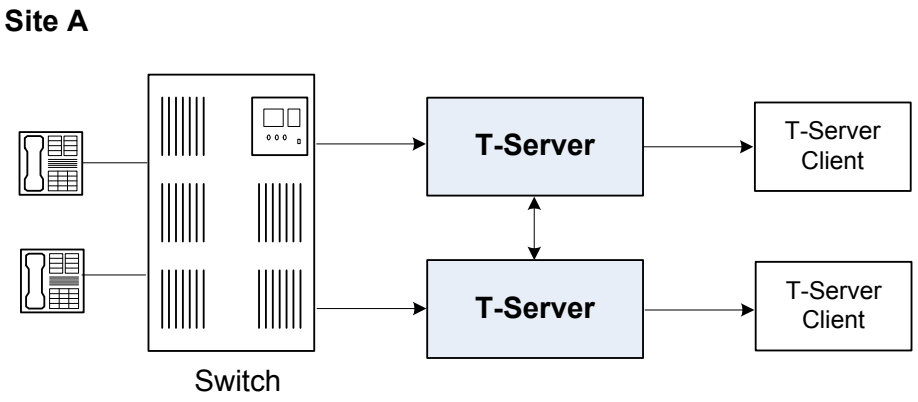

Figure 12: Switch Partitioning Architecture

Starting with version 8.0, in addition to `ConnIDs` and `UserData`, T-Server can synchronize the `CallType` attribute. Each T-Server is required to register all DNs it monitors. In a multi-partitioned environment, when configured, calls between partitions are reported as internal `(CallTypeInternal)`. In a non-partitioned environment, such calls are reported as inbound `(CallTypeInbound)` and/or outbound `(CallTypeOutbound)`, depending on the direction of a call. In order for T-Servers to report calls between specified partitions as internal, registered DNs of these partitions must be assigned to a Switch (T-Server), Switching Office, or Tenant, using the `dn-scope` configuration option. If DNs that are involved in calls are not in the T-Server scope, those DNs will be reported as inbound or outbound.

In addition, T-Server supports `LocalCallType` and `PropagatedCallType` attributes, which depend on the `propagated-call-type` configuration option setting for reporting. See the option description on page 234.

To control race conditions that may occur in the switch-partitioned environment, use the `epp-tout` configuration option (see page 249).

Notes: Because of possible delays in TCP/IP connections, a sequence of events sent for the same call by two or more T-Servers to clients may appear in an unexpected order. For example, in a simple call scenario with two partitions, `EventRinging` and `EventEstablished` messages may both arrive before `EventDialing`.

Genesys switch partitioning does not apply to hardware partitioning functionality that is supported on some switches.

Table 5 shows the T-Server types that support switch partitioning.

Table 5: T-Server Support for Switch Partitioning

T-Server Type	Supported
Alcatel A4400/OXE	Yes
Avaya Communication Manager	Yes
Avaya TSAPI	Yes
Cisco Unified Communications Manager	Yes
SIP Server	Yes

Event Propagation Configuration

The basic Event Propagation feature configuration includes a setting of specific configuration options at a T-Server `Application` level. The advanced feature configuration allows you to customize the feature at a `Switch` level.

When determining whether to notify other T-Servers of changes to user data, or to distribute party events, T-Server checks:

1. Call topology (what location a call came from and to what location the call was then transferred or conferenced).
2. Outbound parameters of the `Switch` this T-Server relates to (whether propagation parameters are configured for the access codes this switch uses to reach the switch at the location a call came from and the switch at the location to which the call was then transferred or conferenced).

Warning! The direction of user-data or party-events propagation does not necessarily match the direction of call distribution. Therefore, the access code used to deliver the call can differ from the access code used for the purpose of Event Propagation.

If one of the T-Servers along the call distribution path has the Event Propagation feature disabled, that T-Server does not distribute events to remote locations.

Procedure: Activating Event Propagation: basic configuration

Purpose: To activate the Event Propagation feature for User Data updates and call-party–associated events (Party Events) distribution.

Start of procedure

1. Open the T-Server `Application's Properties` dialog box.
2. Click the `Options` tab.
3. Open the `extrouter` section.
4. Set the `event-propagation` option to the `list` value.

   This setting enables User Data propagation. If you need to enable Party Events propagation, perform Step 5.
5. Set the `use-data-from` option to the `current` value.

   This setting enables Party Events propagation.

   For the option description and its valid values, see Chapter 12, "T-Server Common Configuration Options," on page 229.
6. When you are finished, click `Apply`.
7. Click `OK` to save your changes and exit the `Properties` dialog box.

End of procedure

Next Steps

- For advanced feature configuration, do the following procedure:

  Procedure: Modifying Event Propagation: advanced configuration, on page 100

Procedure: Modifying Event Propagation: advanced configuration

Purpose: To modify access codes for advanced Event Propagation configuration.

Prerequisites

- Procedure: Activating Event Propagation: basic configuration, on page 100

Overview

You can set Event Propagation parameters using:

- The `Default Access Code` properties of the `Switch` that receives an ISCC-routed call (the destination switch).
- The `Access Code` properties of the `Switch` that passes an ISCC-routed call (the origination switch).

If you do not set up Event Propagation parameters for a given `Access Code`, T-Server uses corresponding settings configured for the `Default Access Code` of the destination switch.

The procedures for modifying `Default Access Codes` and `Access Codes` are very similar to each other.

Start of procedure

1. Among configured `Switches`, select the `Switch` that the configured T-Server relates to.
2. Open the `Switch's Properties` dialog box and click either the `Default Access Codes` tab or the `Access Codes` tab.
3. Select a configured `Default Access Code` or configured `Access Code` and click `Edit`.

> **Note:** If no `Default Access Code` is configured, see page 105 for instructions. If no `Access Codes` are configured, see page 106 for instructions.

4. In the `Switch Access Code Properties` dialog box that opens, specify a value for the `ISCC Protocol Parameters` field as follows:
   - To enable distribution of both user data associated with the call and call-party–associated events[1], type:

     `propagate=yes`

     which is the default value.
   - To enable distribution of user data associated with the call and disable distribution of call-party–associated events, type:

     `propagate=udata`
   - To disable distribution of user data associated with the call and enable distribution of call-party–associated events, type:

1. The following are call-party–associated events: `EventPartyChanged`, `EventPartyDeleted`, and `EventPartyAdded`.

```
propagate=party
```

- To disable distribution of both user data associated with the call and call-party–associated events, type:

```
propagate=no
```

5. Click `OK` to save configuration updates and close the `Switch Access Code Properties` dialog box.
6. Click `Apply` and `OK` to save configuration updates and close the `Switch Properties` dialog box.

End of procedure

ISCC Transaction Monitoring Feature

This feature allows T-Server clients to monitor ISCC transactions that occur during the call data transfer between T-Servers in a multi-site environment.

In order to be able to monitor ISCC messaging, a T-Server client must subscribe to the ISCC Transaction Monitoring. Once a subscription request is confirmed, a client will receive updates about all multi-site operations of this T-Server.

The `TTransactionMonitoring` request is used to instruct T-Server to start, stop, or modify a client's subscription to Transaction Monitoring feature notifications by setting the `TSubscriptionOperationType` parameter to `SubscriptionStart`, `SubscriptionStop`, or `SubscriptionModify` respectively. The transaction status is reported in `EventTransactionStatus` messages to the subscribed clients.

To determine whether the Transaction Monitoring feature is supported by a specific T-Server, a T-Server client may query T-Server's capabilities. For more information about support of this feature, see *Genesys Events and Models Reference Manual* and *Platform SDK 8.x .NET (or Java) API Reference*.

Configuring Multi-Site Support

Prior to configuring T-Server to support multi-site operation, you must read the "Licensing Requirements" on page 35, as well as previous sections of this chapter on multi-site deployment. In particular, Table 3 on page 77 shows which transaction types are supported by a specific T-Server, while Table 4 on page 82 shows whether your T-Server supports the `NetworkCallID` attribute for

the ISCC/COF feature. Use this information as you follow the instructions in this chapter.

Note: Before attempting to configure a multi-site environment, Genesys recommends that you plan the changes you want to make to your existing contact centers. You should then gather the configuration information you will need (such as the name of each T-Server application, port assignments, and switch names), and use Configuration Manager to create and partially configure each T-Server object. Review multi-site option values in the "extrouter Section" on page 239 and determine what these values need to be, based on your network topology.

For T-Server to support multi-site operation, you must create and configure three types of objects in the Configuration Layer:

1. Applications
2. Switches, including Access Codes
3. DNs

You must configure these objects for origination and destination locations. Multi-site support features activate automatically at T-Server startup. See "DNs" on page 110 for details.

Applications

Ensure that T-Server Application objects, and their corresponding Host objects, exist and are configured for origination and destination locations.

Once you've done that, use Configuration Manager to add this configuration to a T-Server Application.

Procedure: Configuring T-Server Applications

Purpose: To configure T-Server Application objects for multi-site operation support.

Start of procedure

1. Open the T-Server Application's Properties dialog box.
2. Click the Connections tab and click Add to add a connection to the appropriate T-Server. The Connection Info Properties dialog box displays.
3. Use the Browse button to search for the T-Server you want to connect to, and fill in the following values:
   - Port ID

- Connection Protocol
- Local Timeout
- Remote Timeout
- Trace Mode

4. Click the Options tab. Create a new section called extrouter or open an existing section with this name.

> **Note:** If you do not create the extrouter section, T-Server uses the default values of the corresponding configuration options.

5. Open the extrouter section. Configure the options used for multi-site support.

> **Note:** For a list of options and valid values, see "extrouter Section" on page 239, in the "T-Server Common Configuration Options" chapter in Part Two of this document.

6. When you are finished, click Apply.
7. Repeat this procedure for all T-Servers for origination and destination locations that are used for multi-site operations.

End of procedure

Next Steps

- See "Switches and Access Codes."

Switches and Access Codes

Ensure that Switching Office and Switch objects are configured for both origination and destination locations.

You configure Access Codes to a destination switch in the origination Switch's Properties dialog box. The only exception is the Default Access Code, which is configured at the destination Switch's Properties dialog box.

You can configure two types of switch Access Codes in the Switch's Properties dialog box:

- A Default Access Code (for inbound calls)—Specifies the access code that other switches can use to access this switch when they originate a multi-site transaction.
- An Access Code (for outbound calls)—Specifies the access code that this switch can use when it originates a multi-site transaction to access another switch.

When the origination T-Server processes a multi-site transaction, it looks for an access code to the destination switch. First, T-Server checks the Access Code of the origination Switch:

- If an access code to the destination switch is configured with the target type Target ISCC and with any transaction type except Forbidden, T-Server uses this access code to dial the destination switch.
- If the access code to the destination switch is not configured on the Access Code tab of the origination switch, the origination T-Server checks the Default Access Code tab of the destination switch. If an access code is configured there with the target type Target ISCC and with any transaction type except Forbidden, T-Server uses this access code to dial the destination switch.
- If no access code with the required properties is found, T-Server rejects the transaction.

Note: When migrating from previous releases of T-Servers to 8.1, or when using T-Servers of different releases (including 8.1) in the same environment, see "Compatibility Notes" on page 109.

Procedure: Configuring Default Access Codes

Purpose: To configure the Default Access Codes (one per Switch object) to be used by other switches to access this switch when they originate a multi-site transaction.

Prerequisites

- Ensure that Switching Office and Switch objects are configured for both origination and destination locations.

Start of procedure

1. Among configured Switches, select the Switch that the configured T-Server relates to.
2. Open the Switch Properties dialog box and click the Default Access Codes tab.
3. Click Add to open the Access Code Properties dialog box.

4. In the Code field, specify the access code used by remote switches to reach a DN at this switch. An access code is used as a prefix to the remote switch numbers.

   Note: If no prefix is needed to dial to the configured switch, you can leave the Code field blank.

5. In the Target Type field, select Target ISCC.
6. In the Route Type field, select a value corresponding to the transaction type you want to use (given that it is supported for your switch type).
7. When you are finished, click Apply.

End of procedure

Next Steps

- See "Configuring Access Codes."

Procedure: Configuring Access Codes

Purpose: To configure the Access Codes (one or more per Switch object) that this switch can use when it originates a multi-site transaction to access another switch.

Prerequisites

- Ensure that Switching Office and Switch objects are configured for both origination and destination locations.

Start of procedure

1. Among configured Switches, select the Switch that the configured T-Server relates to.
2. Open the Switch Properties dialog box and click the Access Codes tab.
3. Click Add to open the Access Code Properties dialog box.
4. In the Switch field, specify the switch that this switch can reach using this access code. Use the Browse button to locate the remote switch.

5. In the Code field, specify the access code used to reach a DN at the remote switch from this switch. An access code is used as a prefix to the remote switch numbers.

   Note: If no prefix is needed to dial from one switch to another, you can leave the Code field blank.

6. In the Target Type field, select Target ISCC.

   When you select Target ISCC as your target type, the Properties dialog box changes its lower pane to the Sources pane. It is here that you enter the extended parameters for your access codes, by specifying the ISCC Protocol and ISCC Call Overflow Parameters.

   To set these parameters, locate the two drop-down boxes that appear below the Target Type field in the Sources pane of that Properties dialog box.

   a. In the ISCC Protocol Parameters drop-down box, enter the appropriate ISCC Protocol parameter, as a comma-separated list of one or more of the following items shown in Table 6:

Table 6: Target Type: ISCC Protocol Parameters

ISCC Protocol Parameters	Description
dnis-tail=<number-of-digits>	Where number-of-digits is the number of significant DNIS digits (last digits) used for call matching. 0 (zero) matches all digits.
propagate=<yes, udata, party, no>	Default is yes. For more information, see "Modifying Event Propagation: advanced configuration" on page 100.
direct-network-callid=<>	For configuration information, see Part Two of this document. (Use Table 4 on page 82 to determine if your T-Server supports the direct-network-callid transaction type.)

   b. In the ISCC Call Overflow Parameters drop-down box, enter call overflow parameters, as a comma-separated list of one or more of the following items shown in Table 7:

Table 7: Target Type: ISCC Call Overflow Parameters

ISCC Call Overflow Parameters	Description
`match-callid`	Matches calls using network `CallID`.
`match-ani`	Matches calls using ANI. **Note:** When using `match-ani`, the `match-flexible` parameter must be set to `false`.
`match-flexible`	Supports flexible call matching based on the following values: Default Value: `true` Valid Values: `true`, `false`, and `[matching-context-type]`, where `[matching-context-type]` is the switch-specific value, which must be the same as the value of the `default-network-call-id-matching` configuration option of the corresponding T-Server.
`inbound-only=<boolean>`	Default is `true`. Setting `inbound-only` to `true` disables COF on consultation and outbound calls.

7. In the `Route Type` field, select a value corresponding to the transaction type you want to use (given that it is supported for your switch type). Table 8 contains cross-reference information on transaction types that the Configuration Layer and T-Server use.

Table 8: Route Type and ISCC Transaction Type Cross-Reference

Route Type Field Value	ISCC Transaction Type
Default	The first value from the list of values specified in the `cast-type` option for the T-Server at the destination site
Direct	`direct-callid`
Direct ANI	`direct-ani`
Direct Digits	`direct-digits`
Direct DNIS and ANI	Reserved

Table 8: Route Type and ISCC Transaction Type Cross-Reference (Continued)

Route Type Field Value	ISCC Transaction Type
Direct Network Call ID	`direct-network-callid`
Direct No Token	`direct-notoken`
Direct UUI	`direct-uui`
DNIS Pooling	`dnis-pooling`
Forbidden	External routing to this destination is not allowed
ISCC defined protocol	Reserved
PullBack	`pullback`
Re-Route	`reroute`
Route	`route`

8. When you are finished, click `Apply`.

End of procedure

Next Steps

- After configuring a switch for multi-site support, proceed with the configuration of DNs assigned to this switch.

Compatibility Notes

When migrating from previous releases of T-Servers to 8.1, or when using T-Servers of different releases (including 8.1) in the same environment, keep in mind the following compatibility issues:

- The `Target External Routing Point` value of the `Target Type` field is obsolete and provided only for backward compatibility with T-Servers of releases 5.1 and 6.0. When two access codes for the same switch are configured, one with the `Target ISCC` target type and the other with the `Target External Routing Point` target type, T-Servers of releases 8.x, 7.x, 6.5, and 6.1:
  - Use the `Target ISCC` access code for transactions with T-Servers of releases 8.x, 7.x, 6.5, and 6.1.
  - Use the `Target External Routing Point` access code for transactions with T-Servers of releases 5.1 and 6.0.

When the only access code configured for a switch has the Target External Routing Point target type, T-Server uses this access code for all transactions.

- When the Target External Routing Point value of the Target Type field is configured, you must set the Route Type field to one of the following:
  - Default to enable the route transaction type
  - Label to enable the direct-ani transaction type
  - Direct to enable the direct transaction type

  Note: The direct transaction type in releases 5.1 and 6.0 corresponds to the direct-callid transaction type in releases 6.1 and later.

  - UseExtProtocol to enable the direct-uui transaction type
  - PostFeature to enable the reroute transaction type

  These values are fully compatible with the transaction types supported in T-Server release 5.1.
- For successful multi-site operations between any two locations served by release 5.1 T-Servers, identical Route Type values must be set in the Switch's Access Code Properties dialog boxes for both the origination and destination switches.

DNs

Use the procedures from this section to configure access resources for various transaction types.

Procedure: Configuring access resources for the route transaction type

Purpose: To configure dedicated DNs required for the route transaction type.

Prerequisites

- Ensure that Switching Office and Switch objects are configured for both origination and destination locations.

Start of procedure

1. Under a configured Switch, select the DNs folder. From the main menu, select File > New > DN to create a new DN object.
2. On the General tab of the DN's Properties dialog box, specify the number of the configured DN as the value of the Number field. This value must correspond to the Routing Point number on the switch.
3. Select External Routing Point as the value of the Type field.
4. If a dialable number for that Routing Point is different from its DN name, specify the number in the Association field.
5. Click the Access Numbers tab. Click Add and specify these access number parameters:
   - Origination switch.
   - Access number that must be dialed to reach this DN from the origination switch.

   In determining an access number for the Routing Point, T-Server composes it of the values of the following properties (in the order listed):
   a. Access number (if specified).
   b. Switch access code from the switch of the origination party to the switch to which the Routing Point belongs, concatenated with its Association (if the Association value is specified).
   c. Switch access code from the switch of the origination party to the switch to which the Routing Point belongs, concatenated with the number for the DN.
   d. Default access code of the switch to which the Routing Point belongs, concatenated with its Association (if the Association value is specified).
   e. Default access code of the switch to which the Routing Point belongs, concatenated with the number for the DN.

   Note: If option use-implicit-access-numbers is set to true, the access number composed of switch access code and DN can be used for external transfers of calls originating at switches for which an access number is not specified.

6. When you are finished, click Apply.

End of procedure

Procedure: Configuring access resources for the dnis-pool transaction type

Purpose: To configure dedicated DNs required for the `dnis-pool` transaction type.

Start of procedure

1. Under a configured `Switch`, select the `DNs` folder. From the main menu, select `File` > `New` > `DN` to create a new `DN` object.
2. On the `General` tab of the DN's `Properties` dialog box, specify the number of the configured DN as the value of the `Number` field. This value must be a dialable number on the switch.
3. Select `Access Resource` as the `Type` field and type `dnis` as the value of the `Resource Type` field on the `Advanced` tab.
4. Click the `Access Numbers` tab. Click `Add` and specify these `Access Number` parameters:
   - Origination switch.
   - Access number that must be dialed to reach this DN from the origination switch.

   An access number for the access resource is determined in the same manner as for the `route` access resource.
5. When you are finished, click `Apply`.

End of procedure

Procedure: Configuring access resources for direct-* transaction types

Start of procedure

You can use any configured DN as an access resource for the `direct-*` transaction types. (The `*` symbol stands for any of the following: `callid`, `uui`, `notoken`, `ani`, or `digits`.)

You can select the `Use Override` check box on the `Advanced` tab to indicate whether the override value should be used instead of the number value to dial to the DN. You must specify this value if the DN has a different DN name and dialable number. In fact, this value is required for T-Servers for some switch types—such as Aspect ACD, Nortel Communication Server 2000/2100, and Spectrum.

End of procedure

Procedure:
Configuring access resources for ISCC/COF

Purpose: To configure dedicated DNs required for the ISCC/COF feature.

Start of procedure

Note: Use Table 4 on page 82 to determine if your T-Server supports the ISCC/COF feature.

1. Under a configured `Switch`, select the `DNs` folder. From the main menu, select `File > New > DN` to create a new `DN` object.
2. On the `General` tab of the DN `Properties` dialog box, enter the name of the configured DN in the `Number` field.

   Note: The name of a DN of type `Access Resource` must match the name of a DN in your configuration environment (typically, a DN of type `Routing Point` or `ACD Queue`), so T-Server can determine whether the calls arriving at this DN are overflowed calls.

3. Select `Access Resource` as the value for the `Type` field.
4. On the `Advanced` tab, type `cof-in` or `cof-not-in` as the value for the `Resource Type` field.

   Note: Calls coming to DNs with the `cof-not-in` value for the `Resource Type` are never considered to be overflowed.

5. When you are finished, click `Apply`.

End of procedure

Procedure:
Configuring access resources for non-unique ANI

Purpose: To configure dedicated DNs required for the `non-unique-ani` resource type.

The `non-unique-ani` resource type is used to block `direct-ani` and `COF/ani` from relaying on ANI when it matches configured/enabled resource digits. Using `non-unique-ani`, T-Server checks every ANI against a list of `non-unique-ani` resources.

Start of procedure

1. Under a configured Switch, select the DNs folder. From the main menu, select File > New > DN to create a new DN object.
2. On the General tab of the DN Properties dialog box, specify the ANI digits that need to be excluded from normal processing.
3. Select Access Resource as the value for the Type field.
4. On the Advanced tab, specify the Resource Type field as non-unique-ani.
5. When you are finished, click Apply.

End of procedure

Procedure: Modifying DNs for isolated switch partitioning

Purpose: To modify DNs that belong to a particular partition where switch partitioning is used.

This configuration instructs T-Server to select an External Routing Point that has the same partition as the requested destination DN.

Note: When a target DN is not configured or has no configured partition name, T-Server allocates a DN of the External Routing Point type that belongs to any partition.

Start of procedure

1. Under a Switch object, select the DNs folder.
2. Open the Properties dialog box of a particular DN.
3. Click the Annex tab.
4. Create a new section named TServer.
5. Within that section, create a new option named epn. Set the option value to the partition name to which the DN belongs.
6. Repeat Steps 1–5 for all DNs, including DNs of the External Routing Point type, that belong to the same switch partition.
7. When you are finished, click Apply.

End of procedure

Configuration Examples

This section provides two configuration examples and describes how the configuration settings affect T-Server's behavior.

Multiple Transaction Types

This example demonstrates the difference in how ISCC directs a call when you specify two different transaction types (`route` and `direct-ani`).

In this example, you configure an origination and a destination switch for as described in "Switches and Access Codes" on page 104.

1. Among configured `Switches`, select the origination `Switch`.
2. Open the `Switch Properties` dialog box and click the `Default Access Codes` tab.
3. Click `Add` to open the `Access Code Properties` dialog box.
4. Set the `Access Code` field to `9`.
5. When you are finished, click `Apply`.
6. Among configured `Switches`, select the destination `Switch`.
7. Under the destination `Switch`, configure a DN as described in "Configuring access resources for the route transaction type" on page 110.
8. Set the DN `Number` field to `5001234567`.
9. Click the `Advanced` tab of this DN's `Properties` dialog box.
10. Select the `Use Override` check box and enter `1234567` in the `Use Override` field.
11. When you are finished, click `Apply` or `Save`.
12. Use a T-Server client application to register for this new DN with the destination T-Server and, therefore, with the switch.
13. Request to route a call from any DN at the origination switch to the destination DN you have just configured:
    - If you are using the `route` ISCC transaction type, the client requests that T-Server deliver a call to a destination location using the DN number `5001234567`. ISCC requests that the switch dial one of the external routing points at the destination location, using the value either of the `Access Number` field or of the `Access Code` field, which is `9`, concatenated with the external routing point at the destination location. The call is routed to the DN number `5001234567`.
    - If you are using the `direct-ani` ISCC transaction type, the client requests that T-Server deliver a call to a destination location using the DN number `1234567`, which is the `Use Override` value. ISCC requests that the switch dial `91234567`, which is a combination of the `Switch Access Code` value and the `Use Override` value. The destination T-Server is waiting for the call to directly arrive at DN number `5001234567`.

Call Overflow Methods

This section demonstrates how to indicate which overflow methods a switch supports.

In this example, for T-Server to use ANI/OtherDN matching in call overflow and manual transfer scenarios, set the `ISCC Call Overflow Parameters` to:

```
match-ani, inbound-only=true
```

when configuring `Switch Access Codes` as described on page 106.

With this setting, the switch's location is queried for call data each time the destination T-Server receives an inbound call with the `ANI` or `OtherDN` attribute.

For T-Server to use `NetworkCallID` matching in call overflow and manual transfer scenarios, set the `ISCC Call Overflow Parameters` to (for example):

```
match-callid, inbound-only=false
```

when configuring `Switch Access Codes` as described on page 106.

With this setting, the switch's location is queried for call data each time the destination T-Server receives a call of any type (including inbound) with the `NetworkCallID` attribute.

Next Steps

Continue with Chapter 5, "Starting and Stopping T-Server Components," on page 117 to test your configuration and installation.

Chapter

5 Starting and Stopping T-Server Components

This chapter describes methods for stopping and starting T-Server, focusing on manual startup for T-Server and HA Proxy for all switches. It includes these sections:

Command-Line Parameters

You can start and stop Framework components using the Management Layer, a startup file, a manual procedure, or the Windows Services Manager.

With all these methods, command-line parameters are usually required for a server application in addition to an executable file name.

Common command-line parameters are as follows:

`-host`	The name of the host on which Configuration Server is running.
`-port`	The communication port that client applications must use to connect to Configuration Server.
`-app`	The exact name of an Application object as configured in the Configuration Database.

`-l`	The license address. Use for the server applications that check out technical licenses. Can be either of the following: • The full path to, and the exact name of, the license file used by an application. For example, `-l /opt/mlink/license/license.dat`. • The host name and port of the license server, as specified in the `SERVER` line of the license file, in the `port@host` format. For example, `-l 7260@ctiserver`. **Note:** Specifying the License Manager's host and port parameter eliminates the need to store a copy of a license file on all computers running licensed applications.
`-V`	The version of a Framework component. Note that specifying this parameter does not start an application, but returns its version number instead. You can use either uppercase or lowercase.
`-nco X/Y`	The Nonstop Operation feature is activated; `X` exceptions occurring within `Y` seconds do not cause an application to exit. If the specified number of exceptions is exceeded within the specified number of seconds, the application exits or, if so configured, the Management Layer restarts the application. If the `-nco` parameter is not specified, the default value of `6` exceptions handled in `10` seconds applies. To disable the Nonstop Operation feature, specify `-nco 0` when starting the application.
`-lmspath`	The full path to log messages files (the common file named `common.lms` and the application-specific file with the extension `*.lms`) that an application uses to generate log events. This parameter is used when the common and application-specific log message files are located in a directory other than the application's working directory, such as when the application's working directory differs from the directory to which the application is originally installed. Note that if the full path to the executable file is specified in the startup command-line (for instance, `c:\gcti\multiserver.exe`), the path specified for the executable file is used for locating the `*.lms` files, and the value of the `lmspath` parameter is ignored.
`- transport-port <port number>`	`<port number>` is the port number that a client will use for its TCP/IP connection to Configuration Server. See the Client-Side Port Definition section in the *Genesys 8.x Security Deployment Guide* for more information.
`- transport-address <IP address>`	`<IP address>` is the IP address that a client will use for its TCP/IP connection to Configuration Server. See the Client-Side Port Definition section in the *Genesys 8.x Security Deployment Guide* for more information.

Note: In the command-line examples in this document, angle brackets indicate variables that must be replaced with appropriate values.

Starting and Stopping with the Management Layer

Procedure: Configuring T-Server to start with the Management Layer

Start of procedure

1. Open the T-Server `Application's Properties` dialog box.
2. Click the `Start Info` tab.
3. Specify the directory where the application is installed and/or is to run as the `Working Directory`.
4. Specify the name of the executable file as the `command-line`.
5. Specify command-line parameters as the `Command–Line Arguments`.

   The command-line parameters common to Framework server components are described on page 117.
6. When you are finished, click `Apply`.
7. Click `OK` to save your changes and exit the `Properties` dialog box.

End of procedure

Note: Before starting an application with the Management Layer, make sure the startup parameters of the application are correctly specified in the application's `Properties` dialog box in Configuration Manager.

After its command-line parameters are correctly specified in the `Properties` dialog box, you can start and stop T-Server from Solution Control Interface (SCI), which is the graphical interface component of the Management Layer. (The starting procedure for SCI is described in the *Framework 8.1 Deployment Guide*.) *Framework 8.0 Solution Control Interface Help* provides complete instructions on starting and stopping applications.

You can also use the Management Layer to start a T-Server that has failed. To enable T-Server's autorestart functionality, select the corresponding check box in the `Application's Properties` dialog box.

Note that when you start (or restart) an application via the Management Layer, the application inherits environment variables from Local Control Agent (LCA), which executes the startup command. Therefore, you must also set the environment variables required by the application for the account that runs LCA.

Warning! *Stopping* an application via the Management Layer is not considered an application failure. Therefore, the Management Layer does not restart applications that it has stopped unless an appropriate alarm condition and alarm reaction are configured for these applications.

Starting with Startup Files

Startup files are files with the extension `run.sh` (on UNIX) or `startServer.bat` (on Windows), which installation scripts create and place into the applications' directories during the installations. These files are created for all Framework server applications except:

- Configuration Server (primary or backup) running on Windows.
- Backup Configuration Server running on UNIX.
- DB Server running on Windows.
- LCA running on either Windows or UNIX.

When using a startup file, verify that the startup parameters the installation script inserted in the startup file are correct. Use the following instructions for UNIX and Windows to start those application for which startup files are created. See the appropriate sections in "Starting Manually" on page 121 to identify which applications should be running for a particular application to start.

Procedure: Starting T-Server on UNIX with a startup file

Start of procedure

1. Go to the directory where an application is installed.
2. Type the following command line:
   ```
   sh run.sh
   ```

End of procedure

Procedure:
Starting T-Server on Windows with a startup file

Start of procedure

To start T-Server on Windows with a startup file, use either of these methods:

- Go to the directory where an application is installed and double-click the `startServer.bat` icon.

Or

- From the `MS-DOS` window, go to the directory where the application is installed and type the following command-line:

  ```
  startServer.bat
  ```

End of procedure

Starting Manually

When starting an application manually, you must specify the startup parameters at the command prompt, whether you are starting on UNIX or Windows. At the command prompt, command-line parameters must follow the name of the executable file. On the `Shortcut` tab of the Program `Properties` dialog box, command-line parameters must also follow the name of the executable file.

The command-line parameters common to Framework server components are described on page 117.

If an `Application` object name, as configured in the Configuration Database, contains spaces (for example, `T-Server Nortel`), the `Application` name must be surrounded by quotation marks in the command-line:

```
-app "T-Server Nortel"
```

You must specify the rest of the command-line parameters as for any other application.

The following sections provide general instructions for starting HA Proxy and T-Server manually. Along with these instructions, refer to Table 9, which lists T-Servers and HA Proxy executable file names for supported switches for Windows and UNIX operating systems.

Table 9: T-Server and HA Proxy Executable Names

T-Server Type	T-Server Executable File Name		HA Proxy Executable File Name	
	UNIX	Windows	UNIX	Windows
Aastra MXONE CSTA I	md110_server	md110_server.exe	Not Applicable	
Alcatel A4200/OXO	a4200_server	a4200_server.exe	Not Applicable	
Alcatel A4400/OXE	a4400_server	a4400_server.exe	Not Applicable	
Aspect ACD	aspect_server	aspect_server.exe	Not Applicable	
Avaya Communication Manager	avayacm_server	avayacm_server.exe	Not Applicable[a]	
Avaya INDeX	Not Applicable	index_server.exe	Not Applicable	
Avaya TSAPI	avayatsapi_server	avayatsapi_server.exe	Not Applicable	
Cisco UCCE	CiscoUCCE_server	CiscoUCCE_server.exe	Not Applicable	
Cisco Unified Communications Manager	ciscocm_server	ciscocm_server.exe	Not Applicable	
DataVoice Dharma	Dharma_server	Dharma_server.exe	Not Applicable	
Digitro AXS/20	digitro_server	digitro_server.exe	Not Applicable	
EADS Intecom M6880	intecom_server	intecom_server.exe	Not Applicable	
EADS Telecom M6500	m6500_server	m6500_server.exe	Not Applicable	
eOn eQueue	eon_server	eon_server.exe	Not Applicable	
Fujitsu F9600	Not Applicable	F9600_server.exe	Not Applicable	
Huawei C&C08	cc08_server	cc08_server.exe	Not Applicable	
Huawei NGN	huaweingn_server	huaweingn_server.exe	Not Applicable	
Mitel MiTAI	Not Applicable	mitel_server.exe	Not Applicable	
NEC NEAX/APEX	neax_server	neax_server.exe	Not Applicable	
Nortel Communication Server 2000/2100	ncs2000_server	ncs2000_server.exe	ha_proxy_dms	ha_proxy_dms.exe

Table 9: T-Server and HA Proxy Executable Names (Continued)

T-Server Type	T-Server Executable File Name		HA Proxy Executable File Name	
	UNIX	Windows	UNIX	Windows
Nortel Communication Server 1000 with SCSS/MLS	succession_server	succession_server.exe	Not Applicable	
Philips Sopho iS3000	iS3000_server	iS3000_server.exe	ha_proxy_iS3000	ha_proxy_iS3000.exe
Radvision iContact	nts_server	nts_server.exe	Not Applicable	
Samsung IP-PCX IAP	samsung_server	samsung_server.exe	Not Applicable	
Siemens Hicom 300/HiPath 400 CSTA I	rolmcb4_server	rolmcb4_server.exe	Not Applicable	
Siemens HiPath 3000	HiPath3000_server	HiPath3000_server.exe	Not Applicable	
Siemens HiPath 4000 CSTA III	HiPath4000_server	HiPath4000_server.exe	Not Applicable	
Siemens HiPath DX	HiPathDX_server	HiPathDX_server.exe	Not Applicable	
SIP Server	sip_server	sip_server.exe	Not Applicable	
Spectrum	spectrum_server	spectrum_server.exe	Not Applicable	
Tadiran Coral	Coral_server	Coral_server.exe	Not Applicable	
Teltronics 20-20	Teltronics2020_server	Teltronics2020_server.exe	ha_proxy_teltronics2020	ha_proxy_teltronics2020.exe
Tenovis Integral 33/55	Tenovis_server	Tenovis_server.exe	Not Applicable	
Network T-Servers				
AT&T	nts_server	nts_server.exe	Not Applicable	
Concert	nts_server	nts_server.exe	Not Applicable	
CRSP	nts_server	nts_server.exe	Not Applicable	
DTAG	dtag_server	dtag_server.exe	Not Applicable	
GenSpec	genspec_server	genspec_server.exe	Not Applicable	

Table 9: T-Server and HA Proxy Executable Names (Continued)

T-Server Type	T-Server Executable File Name		HA Proxy Executable File Name	
	UNIX	Windows	UNIX	Windows
ISCP	nts_server	nts_server.exe	Not Applicable	
IVR Server, using network configuration	nts_server	nts_server.exe	Not Applicable	
KPN	kpn_server	kpn_server.exe	Not Applicable	
MCI	mci800_server	mci800_server.exe	Not Applicable	
NGSN	nts_server	nts_server.exe	Not Applicable	
Network SIP Server	tsip_server	tsip_server.exe	Not Applicable	
Sprint	sprint_server	sprint_server.exe	Not Applicable	
SR3511	sr3511_server	sr3511_server.exe	Not Applicable	
Stentor	stentor_server	stentor_server.exe	Not Applicable	

a. For releases prior to 7.1, this T-Server has an HA Proxy available: `ha_proxy_g3tcp` (UNIX) or `ha_proxy_g3tcp.exe` (Windows).

HA Proxy

If you do not use HA Proxy in your Genesys implementation, proceed to "T-Server" on page 125.

If one or more HA Proxy components are required for the T-Server connection, start HA Proxy before starting T-Server.

Before starting HA Proxy, be sure that the following components are running:

- DB Server that provides access to the Configuration Database
- Configuration Server

The command-line parameters common to Framework server components are described on page 117.

Procedure:
Starting HA Proxy on UNIX manually

Start of procedure

1. Go to the directory where HA Proxy is installed and type the following command-line:

   ```
   ha_proxy_<switch> -host <Configuration Server host>
   -port <Configuration Server port> -app <HA Proxy Application>
   ```

2. Replace `ha_proxy_<switch>` with the correct HA Proxy executable name, which depends on the type of the switch used.

   Table 9 on page 122 lists HA Proxy executable names for supported switches.

End of procedure

Procedure:
Starting HA Proxy on Windows manually

Start of procedure

1. Start HA Proxy from either the `Start` menu or the `MS-DOS` window. If using the `MS-DOS` window, go to the directory where HA Proxy is installed and type the following command-line:

   ```
   ha_proxy_<switch>.exe -host <Configuration Server host> -port
   <Configuration Server port> -app <HA Proxy Application>
   ```

2. Replace `ha_proxy_<switch>.exe` with the correct HA Proxy executable name, which depends on the type of the switch used.

   Table 9 on page 122 lists HA Proxy executable names for supported switches.

End of procedure

T-Server

Before starting T-Server, be sure that the following components are running:

- DB Server that provides access to the Configuration Database
- Configuration Server
- License Manager

Note: If an HA Proxy component is required for the T-Server connection, HA Proxy must be started before T-Server.

The command-line parameters common to Framework server components are described on page 117.

Procedure: Starting T-Server on UNIX manually

Start of procedure

1. Go to the directory where T-Server is installed and type the following command-line:

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

2. Replace `<switch>_server` with the correct T-Server executable name, which depends on the type of the switch used.

   Table 9 on page 122 lists T-Server executable names for supported switches.

End of procedure

Procedure: Starting T-Server on Windows manually

Start of procedure

1. Start T-Server from either the `Start` menu or the `MS-DOS` window. If using the `MS-DOS` window, go to the directory where T-Server is installed and type the following command-line:

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

2. Replace `<switch>_server.exe` with the correct T-Server executable name, which depends on the type of the switch used.

   Table 9 on page 122 lists T-Server executable names for supported switches.

End of procedure

Verifying Successful Startup

After executing the startup command, you might want to check whether it was successful.

If you used the Management Layer to start either T-Server or HA Proxy, check whether Solution Control Interface displays `Started` or `Service Unavailable` status for the corresponding application. Refer to the "Troubleshooting" section of the *Framework 8.0 Management Layer User's Guide* if the startup command does not result in either `Started` or `Service Unavailable` status for some period of time.

If you start your T-Server or HA Proxy with startup files or manually, and if you have configured logging to console or a log file, check the log for messages similar to the following:

- T-Server log file: `Link connected`
- HA Proxy log file: `Link connected`

Stopping Manually

The following stopping procedures apply to Genesys server applications, such as DB Server, Configuration Server, Message Server, Local Control Agent, Solution Control Server, HA Proxy, T-Server, and Stat Server.

Procedure: Stopping T-Server on UNIX manually

Start of procedure

To stop a server application from its console window on UNIX, use either of these commands:

- `Ctrl+C`
- `kill <process number>`

End of procedure

Procedure: Stopping T-Server on Windows manually

Start of procedure

To stop a server application on Windows, use either of these commands:

- To stop a server application from its console window on Windows, use the Ctrl+C command.
- To stop a server application on Windows, use the End Task button on the Windows Task Manager.

End of procedure

Starting and Stopping with Windows Services Manager

When starting an application installed as a Windows Service, make sure the startup parameters of the application are correctly specified in the ImagePath in the Application folder in the Registry Editor. The ImagePath must have the following value data:

```
<full path>\<executable file name> -service <Application Name as
Service> -host <Configuration Server host>
-port <Configuration Server port> -app <Application Name>
-l <license address>
```

where the command-line parameters common to Framework server components are described on page 117 and

`-service` The name of the Application running as a Windows Service; typically, it matches the Application name specified in the `-app` command-line parameter.

Framework components installed as Windows Services with the autostart capability are automatically started each time a computer on which they are installed is rebooted.

You can start Framework components installed as Windows Services with the manual start capability with the Start button in Services Manager.

Note: Use the Windows Services window to change the startup mode from Automatic to Manual and vice versa.

Regardless of a component's start capability, you can stop Framework components installed as Windows Services with the Stop button in Services Manager.

Next Steps

This chapter concludes Part One of this document—the set of general instructions for deploying any T-Server. Refer to subsequent chapters in this guide for detailed reference information and any special procedural instructions that pertain to your particular T-Server.

Genesys

Part

2 T-Server Configuration

Part Two of this *T-Server Deployment Guide* contains reference information specific to your T-Server. However, it also contains information on *all* T-Server options—both those specific to your T-Server and those common to all T-Servers. The information is divided among these chapters:

- Chapter 6, "Switch-Specific Configuration," on page 133, describes compatibility and configuration information specific to this T-Server, including instructions for setting the DN properties and recommendations for the switch configuration.
- Chapter 7, "Supported Functionality," on page 137, describes the features that are supported by this T-Server, including T-Library functionality, and error messages.
- Chapter 8, "HA Configuration and Operation with CUCM JTAPI," on page 193 describes how Cisco Unified Communications Manager (CUCM) operates in high-availability (HA) environment.
- Chapter 9, "Integration with Genesys Media Server," on page 197, describes the deployment configuration of T-Server for Cisco Unified Communications Manager when used with Media Server. The relevant configuration options are described in the Genesys Voice Platform Deployment Guide, release 8.1.5 or later.
- Chapter 10, "SIP Server In Front Deployment," on page 201, describes the deployment configuration for the Genesys Voice Platform in conjunction with CUCM that involves the Genesys SIP Server and Media Server deployed in front of the Cisco switch.
- Chapter 11, "Common Configuration Options," on page 207, describes the log configuration options common to all Genesys server applications.
- Chapter 12, "T-Server Common Configuration Options," on page 229, describes the configuration options common to all T-Server types, including options for multi-site configuration.

- Chapter 13, "T-Server-Specific Configuration Options," on page 257, describes the configuration options specific to this T-Server, including the link-related options—those that address the interface between T-Server and the switch.
- Chapter 14, "Stream Manager Configuration," on page 291,describes the configuration options that enable Stream Manager to work with T-Server for Cisco Unified Communications Manager.

New in T-Server for Cisco Unified Communications Manager

The following new features are available in the 8.1 release of T-Server for Cisco Unified Communications Manager.

Release 8.1.2

- **Providing Call Participant information.** T-Server reports a list of conference parties in `AttributeExtensions` of `EventPartyChanged`, `EventPartyAdded`, and `EventEstablished`. See "Call Participant Info" on page 143 for details.
- **Providing `AttributeDNIS` in `EventDialing`.** T-Server can include `AttributeDNIS` in `EventDialing` for most call flows by setting the `delay-dialing` configuration option to `true`. See "Providing AttributeDNIS in EventDialing" on page 161 for details.
- **Hunt Groups enhancement.** T-Server supports Hunt Groups in Broadcast mode (parallel ringing). See "Hunt Groups" on page 153 for details.
- **Disabling the Default Music On Hold Treatment.** See "Disabling the Default MOH Treatment" on page 157 for details.
- **Agent State enhancement.** T-Server can automatically change the agent state to the specified state (and work mode) and can delay the agent state change when the corresponding agent DN goes out of service (OOS). See "Agent State for Out-Of-Service Agent DNs" on page 140 for details.
- **Transport Layer Security support.** T-Server supports secure communication with Java Telephony API (JTAPI). See "Transport Layer Security" on page 174 for details.
- **Whisper Coaching enhancement.** Intercom DNs no longer require extra licenses by changing the way they are configured in the Genesys Configuration environment. See "Whisper Coaching and Extra Instance of Intercom Call" on page 173.
- **Support of E.164 numbering plan.**
- **Support for ANI modification.** T-Server provides the ability to change the Automatic Number Identification/Calling Line Identification (ANI/CLI) that is displayed on a destination party's phone during call routing. See "ANI Modification For Outbound Calls" on page 140 for details.

- **Support for Do Not Disturb (DND).** T-Server supports DND functionality that could be activated from an agent's phone or by a T-Library request. See "Do Not Disturb" on page 151 for details.
- **Support of a free format for a terminal ID in JTAPI events**. See the `free-form-terminal-id` option that enables this feature.

Release 8.1.1

- **Support for Telephone Display Name**. T-Server can provide the telephone display name for specific events within `AttributeExtensions` if the information is available from Cisco Unified Communications Manager. This functionality is controlled by the new configuration option `use-party-display-name`. See "Telephone Display Name" on page 150 for details.
- **Conference Party Information for Shared Lines**. T-Server fully supports the deletion of a shared line party from a conference call without the client having to know which shared party is active. T-Server internally adjusts the party to be deleted to correspond to the active shared party to ensure that the request is successful.
- **Whisper Intercom Feature**. 3PCC support for the Cisco Talk-back Intercom Feature. This feature allows the monitored Agent to use CTI requests from T-Server to talk back to the Supervisor without being overheard by the customer. See "Whisper Intercom Feature" on page 178 for details.
- **Support for Called Address Redirect Destination in TRouteCall messages.** `TRouteCall` messages can be customized by sending the request to T-Server in special keys within `AttributeExtensions`. This allows calls to be further redirected, for example, to different voice mailboxes depending on the destination digits provided by T-Server and then reported by JTAPI. See "Called Address in TRouteCall Messages" on page 146 for further information.
- **Support for monitoring DNs in Hunt Groups.** T-Server supports Hunt Groups on Cisco Unified Communications Manager. See "Hunt Groups" on page 153 for further information.
- **Support for Querying JTAPI on Call States / Active Call On DN.** This feature provides an enhanced capability to detect and clean up stuck calls after a link disconnection/reconnection. See "Querying JTAPI on Call State/Active Call on DN" on page 162 for details.
- **Support for Media Server.** T-Server supports Media Server to play call treatments.

Note: Configuration option changes that apply to your T-Server are described in "Changes from 8.0 to 8.1" on page 289.

Chapter

6 Switch-Specific Configuration

This chapter presents switch-specific reference information for configuring T-Server for the Cisco Unified Communications Manager (CUCM) switch. This chapter has these sections:

Known Limitations

Several known limitations result from the current T-Server/CUCM interface:

- T-Server does not add the value of `Forwarded` to the `CallState` message in the `EventRinging` event when a call is unconditionally forwarded.
- When digits collection is completed (because the `MAX_DIGITS` limit is reached or the `ABORT/TERM_DIGITS` is entered), the treatment `PlayAnnouncementAndCollectDigits` ends, causing the interruption of the announcement regardless of the setting of the `INTERRUPTABLE` flag for this announcement.
- In Call Pickup scenarios, T-Server distributes additional `EventOnHook` / `EventOffHook` when PickUp button is pressed on the phone set.
- CUCM establishes a voice path for inbound calls immediately upon their being placed in an ACD Queue, or when you have configured a music treatment on a Routing Point. This behavior differs from other PBXs, which establish the voice path only when the call is delivered to the agent. This limitation does not affect the Genesys Call Model.

Configuring the Java Virtual Machine on a T-Server Host

Starting with version 8.1.2, T-Server requires that the Java 2 Standard Edition (J2SE) Runtime Environment (or J2SE SDK) version 1.5.0 or later be installed on the T-Server host. This can be downloaded freely for most Operating Systems from the Oracle web site at `http://www.oracle.com/technetwork/java/index.html`.

Note: Starting with version 8.0, T-Server no longer uses dynamic JVM loading. For this reason, `jvm_lib.so` or DLL files are no longer needed for T-Server to operate. As well, `LD_LIBRARY_PATH` is no longer needed for T-Server to operate.

The following platforms are currently supported, and you must select the appropriate one based on the T-Server host operating system:

- Solaris
- Linux
- Windows

For AIX deployments, download the Java 2 Standard Edition (J2SE) Runtime Environment directly from IBM, at `http://www.ibm.com/developerworks/java/jdk/aix/service.html`.

After installing Java VM, set the following environment variable:

`JAVA_HOME`.

See the Java installation guide to configure this environment variable correctly. Java should run without errors before T-Server is started.

Warning! T-Server will not run if the JVM version is earlier than 1.5.0. See console output for errors.

Configuring the CUCM Switch for T-Server

Note: For specific information about Cisco Unified Communications Manager configuration, refer to the *Cisco Unified Communications Manager System Guide* and the *Cisco Unified Communications Manager Administration Guide*.

The following procedure enables all versions of the CUCM switch to work with the T-Server application:

1. For Cisco UCM, create a new End User in CallManager, associate all of the necessary devices, and place the user in the Standard CTI Enabled Group. The user name and password of this new End User must be entered in the T-Server options `user-login` on page 285 and `password` on page 284, respectively.
2. In both the CUCM and the Genesys Configuration Layer, create or identify the following DN types to be controlled or monitored by Genesys:
   - Agent IP-Phone DNs for standalone Cisco phones/softphones.
   - CTI Routing Points.
3. Associate these DNs with the End User created previously in Step 2. This will allow T-Server to register these devices. Table 10 lists the names for the CUCM DN types and their Genesys equivalents.

   Use the Genesys terminology when configuring DNs in the Configuration Layer. T-Server accepts other DNs, but they are not registered with the CUCM.

   Warning! In a deployment with multiple CUCM links, each DN must be controlled only by a single End User (out of multiple users used by T-Server).

Table 10: T-Server Device and CUCM DN Types

Description	CUCM DN Type	Configuration Layer Device Type
Agent Extension	DN assigned to any phone device type, including all IP phone models, and CTI ports	Extension, Position, or Mixed

Table 10: T-Server Device and CUCM DN Types (Continued)

Description	CUCM DN Type	Configuration Layer Device Type
Routing Point	One DN/line assigned to a CTI Routing Point device [a]	Routing Point
ACD Queue	One DN/line assigned to a CTI Routing Point device [a]	ACD Queue

a. Routing in JTAPI requires that only a single line be configured on a CTI Route Point device. Refer to the Cisco Unified JTAPI Developers Guide for CUCM for further information.

Only T-Server uses agent logins, so therefore you do not need to match the user information on the CUCM switch. T-Server manages the status of the agents who use these logins and enables these agents to log in to the CUCM addresses.

JTAPI and Configuring JTAPI Options

JTAPI is the application programming interface (API) that T-Server uses to communicate with the CUCM switch. JTAPI manages all CTI communication between T-Server and Cisco UCM.

Although you can set various JTAPI options, Genesys does not recommend changing the default values of any of them unless you:

- Want to enable JTAPI logging. These logs contain details of all the CTI messages to and from the CUCM.
- Are instructed to do so by Genesys or Cisco Technical Support.

You can configure these options in one of three ways:

- By configuring options in the `jtapi` section on the `Options` tab for the T-Server `Application` object in the Configuration Layer; see the "jtapi Section" on page 274 for more information.
- By creating a `jtapi.ini` text file and placing it in the same working directory in which you installed T-Server (or in a directory defined by the `CLASSPATH` environment variable).
- For Windows systems that have Cisco JTAPI software installed, by using `jtprefs.exe` to set `jtapi.ini`.

For more information about JTAPI, `jtprefs.exe`, and JTAPI options, see the *Cisco Unified Communications Manager Administration Guide.*

Chapter

7 Supported Functionality

This chapter describes the telephony functionality that T-Server for Cisco Unified Communications Manager supports. It contains the following sections:

ACD Queues

DNs of the `ACD Queue` type are configured as the `Routing Point` type in the Cisco Unified Communications Manager. T-Server manages the ACD Queue call distribution and `agentlogin` functionality internally; however this is transparent to T-Server clients. An Agent DN may log into only one queue. Calls with the longest wait time on an ACD queue are distributed to agents with the longest idle time.

The music played to the caller in the ACD queue is configurable. For more information, see “Music Treatment on ACD Queues” on page 156.

ACD-like Default Routing

When a call arrives to a Routing Point, and the URS is disconnected, the call will be routed by T-Server to the ACD Queue configured in the `default-dn` option. The use of this feature is triggered with the value of the `use-default-route` option.

Note: The following line will be displayed in the log when this feature is used: `Router not present, will use default route to <DN number>`.

Agent Login and Agent States

T-Server internally manages agent login and agent states, because the CUCM switch does not contain this functionality. The agent login IDs for the CUCM switch are arbitrary, and there is no coordination between these IDs and the “users” configured in the switch.

The valid agent-state behavior, transitions, and events for T-Server for CUCM are described in the following books:

- *Genesys Events and Models Reference Manual*
- *Platform SDK 8.x .NET (or Java) API Reference*

Refer to these manuals for technical details of T-Library functions. Except for the following unique behavior:

- When an agent is set to the `NotReady` state, the switch can direct calls to this agent. Genesys Universal Routing Server (URS), however, does not direct calls to an agent in the `NotReady` state.
- After a T-Server restart, all agents are set to the `Logged Out` state.
- When an agent logs in with the `TAgentLogin` request, the agent state is determined by the workmode in the request, as follows:
  - `AgentAutoIn/AgentManualIn: Ready` state (`EventAgentReady, workmode` = same as request, is sent).
  - `AgentAfterCallWork: AfterCallWork` state (`EventAgentNotReady, workmode = AgentAfterCallWork,` is sent).
  - `AgentWalkAway: WalkAway` state (`EventAgentNotReady, workmode = AgentWalkAway,` is sent).
  - `AgentAuxWork: NotReady` state (`EventAgentNotReady, workmode = AgentAuxWork,` is sent).
  - `AgentWorkModeUnknown: NotReady` state (`EventAgentNotReady, no workmode,` is sent).
- T-Server automatically sets the agent to the `AfterCallWork` state after releasing a call if `wrap-up-time` is configured for that agent. See "Agent After Call Work" on page 139.
- T-Server automatically sets the agent to the `WalkAway` state if the agent does not answer a call within the `ring timeout` for that call. See "Redirect On No Answer" on page 162.
- An agent cannot log in to multiple queues.

Agent After Call Work

`AfterCallWork` (ACW) is an agent state that prevents Genesys-routed calls from being delivered to the agent.

ACW is configured in the Configuration Layer using the `<wrap-up-time>` variable, located in the Agent's `Properties` section. The variable is defined in seconds, with the following behavior:

- Automatic: (if `WrapUpTime > 0`)

  After releasing a call, an agent in `Ready` state automatically is placed into the ACW unless the call was released because it was redirected using `TRedirectCall`, or unless the agent has another call in progress. The agent remains in the ACW state for <wrap-up-time> seconds, or until the agent explicitly enters the `Ready` state through a `TAgentReady` request.

- Manual: (if `WrapUpTime = 0`)

  After releasing a call, an agent remains in `Ready` state. In both automatic and manual ACW modes, T-Server can also set the agent to the ACW state if the client sends `TAgentNotReady` with `Workmode=AfterCallWork`. In this case, the agent remains in the ACW state until `TAgentReady` is sent.

Agent State for Out-Of-Service Agent DNs

T-Server supports basic and advanced configurations for changing an agent state when a corresponding agent DN goes out of service. The basic configuration is supported by the `logout-on-out-of-service` option. When enabled (set to `true`), T-Server logs the agent out when the agent DN goes out of service.

Introduced in T-Server 8.1.201.12

The advanced configuration enables changing an agent state to the specified state (and work mode) and the ability to apply a delay to the agent state change when a corresponding agent DN goes out of service. After T-Server changes the state, the agent cannot log in until the corresponding DN goes back in service.

The following configuration options enable this advanced functionality:

- `out-of-service-action`
- `out-of-service-action-delay`

If both `out-of-service-action` and `logout-on-out-of-service` options are configured, the `logout-on-out-of-service` option takes precedence unless the `out-of-service-action` option is set to any value other than `logout`.

ANI Modification For Outbound Calls

T-Server provides the ability to change the Automatic Number Identification/Calling Line Identification (ANI/CLI) that is displayed on a destination party's phone during call routing. The modified ANI/CLI must be specified in the `CPNDigits` key in `AttributeExtensions` of a `TRouteCall` request.

This feature is supported for the following requests:

- `TMakeCall`
- `TInitiateTransfer`
- `TInitiateConference`

Feature Configuration

To enable the feature:

1. Configure a DN of type `Routing Point` with the following parameter:
   - `Advanced` tab > `Switch-specific Type`—Set to `2`.

2. (Optional) Configure the T-Server Application:
   - `Options` tab > `JTAPI` section—Modify the `RouteSelectTimeout` configuration option as necessary.

Note: There is a CUCM service parameter (`CTI New Call Accept Timer`) which should be considered when configuring the `RouteSelectTimeout` option value. This parameter specifies the maximum time in seconds for a CTI application to handle a new incoming call. If the CTI application does not handle the call within the time specified in this parameter, the system drops the call. Genesys recommends setting the `RouteSelectTimeout` option to a value greater than or equal to this service parameter, which will default route or drop the call. This will ensure T-Server always has visibility of the call.

3. Specify ANI replacement digits in the `CPNDigits` key in the `Extensions` attribute of a `TRouteCall` request:
   - Key: `CPNDigits`
     Value: A string representing ANI replacement digits

   You can also specify a KV-pair with the key `CPNDigits` inside a KV-List named `CPNOptions`. When `CPNDigits` are available, T-Server will specify the modified ANI via JTAPI for supported Routing Points (`Switch-specific Type` set to `2`).
4. Complete the following steps on the CUCM switch:
   a. Enable the calling party transformation feature for a particular user.
   b. Assign the privilege to transform calling party numbers during a call to the CTI application (T-Server):
      i. Under User Management, click `End User`. Then, select the User associated with T-Server.
      ii. On the User page, in the Permissions Information section, click `Add To Access Control Group`.
      iii. In the window that appears, select `Standard CTI Allow Calling Number Modification`, and click `Add Selected`. Ensure that `Standard CTI Allow Calling Number Modification` is now listed in the Permissions Information Group window.
      iv. Save the User form. The feature is now committed and appears in the Roles window.

Distribution of Events

EventNetworkReached

T-Server will use `EventNetworkReached` with the key `MODIFIED_CALLING_NUMBER` to communicate modified calling digits to T-Server clients.

Key: MODIFIED_CALLING_NUMBER
Type: String
Value: <modified number>

EventRouteRequest

To distinguish between a Routing Points ability to route with CPNDigits or not, T-Server will add the key SupportsCPN in AttributeExtensions of EventRouteRequest.

Key: SupportsCPN
Type: Integer
Value: 0, 1

If SupportsCPN is set to 1, call routing with CPNDigits on a particular Routing Point is supported. If SupportsCPN is set to 0 or missing, call routing with CPNDigits on a particular Routing Point is not supported.

Note: Attempting to route a call with CPNDigits, which did not previously contain the SupportsCPN key with a value of 1, will result in an EventError.

Feature Limitations

The following limitations are a direct result of CUCM restrictions:

- Treatments are not supported on Routing Points configured with Switch-specific Type set to 2. Sending a treatment for a call on this Routing Point will result in an EventError.
- If the RouteSelectTimeout expires before the service parameter (CTI New Call Accept Timer) for an existing call on a Routing Point, T-Server will send an EventRouteEnd for the call; however, the call will remain on the Routing Point.
- If an inbound call arrives on a Routing Point configured with Switch-specific Type set to 2 and the RouteSelectTimeout expires, the call could remain on the Routing Point without ability to be terminated.
- Routing Points with Switch-specific Type set to 2 must be used only for outbound calls.

Call Parking

T-Server supports the Call Parking feature available with Cisco Unified Communications Manager. This feature allows users to park existing call on an internal parking DN within CUCM. Another user may unpark the call by dialing the appropriate park DN. Attached data is maintained for calls that are parked/unparked.

Call Parking with 3PCC

With release version 8.0.1, T-Server allows a softphone, while on a call, to issue a Park request. T-Server sends this request to CUCM, which parks the call and returns the parked number in the released message. T-Server then passes this to the softphone which displays the location where the call is parked. The softphone issues a `TSingleStepTransfer` request to a special DN (`gcti::park`) to accomplish this action. If a physical phone is used to park a call, the DN where the call is parked is returned in `EventReleased` message using the key-value pairs in `AttributeExtensions` of `PARK_ADDRESS`=<DN number> (Example: `PARK_ADDRESS` = `7000`). To unpark a call, a client must issue a `TMakeCall` request to the DN number.

Note: The Park DN Monitor feature must be enabled for the T-Server user on CUCM in order for the `PARK_ADDRESS` key-value pair to be populated by T-Server. Refer to the *Cisco Unified Communications Manager System Guide* and the *Cisco Unified Communications Manager Administration Guide* for further information on this feature.

Call Participant Info

Starting with release 8.1.202.07, T-Server includes the list of conference parties in `AttributeExtensions` of `EventPartyChanged`, `EventPartyAdded`, and `EventEstablished`, using the following keys:

- `OrigDN-`*n*—String value type. The origination DN number specified by *n*.
- `NumOfOrigDNs`—Integer value type. The number of DNs on an original call (excluding the party already reported as `ThisDN`) and all other DNs.
- `Consult-DN-`*n*—String value type. The consultation DN number specified by *n*.
- `NumOfConsultDNs`—Integer value type. The number of DNs on a consultation call and all other DNs (present only in the `EventPartyAdded` event).

Call Pickup

T-Server supports the Call Pickup feature available with Cisco Unified Communications Manager. This feature allows users to pick up calls within their own group. CUCM automatically dials the appropriate Call Pickup group number when the user activates this feature from a Cisco IP phone. Attached data is maintained for calls that are picked up.

Group Call Pickup

T-Server supports the Group Call Pickup feature available with Cisco Unified Communications Manager. This feature allows users to pick up incoming calls within their own group or in other groups. The users must dial the appropriate Group Call Pickup number when activating this feature from a Cisco IP phone. Attached data is maintained for calls that are picked up.

Call Recording

T-Server supports both regular call recording and emergency call recording.

Regular Call Recording

Call recording is performed by passing an RTP stream through Stream Manager or Media Server. Stream Manager or Media Server acts as a media stream proxy, recording all media packets into a file. Depending on the configuration, Stream Manager or Media Server may perform media mixing, or they may save the RTP packets without any change, which improves call recording performance. (See *Framework 7.6 Stream Manager Deployment Guide* and *Genesys Voice Platform 8.1 Deployment Guide* respectively for details.) Call recording is always enabled on a single call leg, such as a leg with a gateway or a leg with a Cisco phone.

Call recording starts after a call becomes established. It does not result in any changes to the call itself, to event processing, or to any other generated TEvents.

Call recording has the highest priority compared to other operations that can be performed on the call when it is established. That is, when the `EventEstablished` message is generated on the destination DN, the operations on the call are performed in the following order:

1. Call recording, if enabled.
2. Supervisor monitoring, if enabled.

Emergency (Manual) Call Recording

T-Server performs emergency call recording when processing a single-step conference call request to the predefined `gcti::record` number. T-Server recognizes this special request and initiates call recording as follows:

- Performs a single-step conference call and adds the selected call recording unit to the call.
- Creates the file name as configured in the `recording-filename` option.

To stop emergency call recording, the agent must issue the `TDeleteFromConference` request using the `gcti::record` number.

Feature Configuration

Table 11 describes how to configure the call recording functionality.

Table 11: Configuring Call Recording

Objective	Related Procedures and Actions
1. Configure a DN.	To enable call recording on a particular DN, in the `TServer` section on the `Annex` tab of the `DN` object, set the configuration option `record` to `true`.
2. Configure a T-Server `Application` object.	In the `TServer` section on the `Options` tab of the T-Server `Application` object, set the configuration option `recording-filename` to the name of the recorded file—Example: `CUCM/call-$REFCI$-at-$AGENTDN$-on-$DATE$-$TIME$`
3. Configure an `Extension` attribute.	Specify an `Extension` attribute with key `record` in the `TRouteCall` request. The routing strategy will determine whether call recording is needed. Key: `record` Value: `destination` When the extension is set to `destination`, the recording will be initiated on the routing destination DN (agent) and will continue while the agent stays in the call.
4. Configure a Stream Manager or Media Server `Application` object.	**Stream Manager** Set configuration options in the Stream Manager or Media Server `Application` object for precise control of how recording is performed. See the *Framework 7.6 Stream Manager Deployment Guide* for more information. It is recommended to use `pcap` recording mode on Stream Manager for best performance. **Genesys Media Server** Configure and tune the application according to the *Genesys Media Server 8.1 Deployment Guide:* • Configure T-Server-CUCM to Media Server Connector. • Configure Media Server components to their default settings: ◦ Resource Manager ◦ Media Control Platform
5. Configure a recorder device on CUCM.	Refer to CUCM documentation for details.

AttributeUserData in EventAttachedDataChanged

Key: `GSIP_REC_FN`
Value: <actual filename>
The value of this key is equal to the recorded Stream Manager or Media Server

filename. T-Server 8.x is generating this filename to match the Stream Manager or Media Server filename.

Key: GSIP_REC_RECORDER
Value: <recorder>
This key has a value that is equal to the recorder address name that is configured on Cisco Unified Communications Manager.

Recorded filename matching

T-Server may not be able to construct exactly the same file name that Stream Manager or Media Server used for recording.

- Stream Manager or Media Server always adds the codec suffix and file extension (depending on the recording mode), so the exact file name is not available (as T-Server does not control codec selection in this case).
- The recorded file still can be matched by the prefix (directory/call-$REFCI$-at-$AGENTDN$-on) that is known.

If recording-filename is set to call-$REFCI$-at-$AGENTDN$-on-$DATE$ then a 100% match can be achieved (this only applies to Stream Manager).

Called Address in TRouteCall Messages

TRouteCall messages can be customized by sending the request to T-Server in special keys within AttributeExtensions. One of these keys, CALLED_ADDRESS_OPTION, is used to indicate how T-Server specifies the original called party address to JTAPI. By default, the called party address is taken from the AttributeDNIS attribute of the request. However, if the key CALLED_ADDRESS_OPTION is set to CALLED_ADDR_REDIRECT_DEST, T-Server reads the digits from another key called, ORIG_CALLED_PARTY_DIGITS. These digits are then passed to the JTAPI Redirect function as the original called party digits. If the DNIS or the extension are not present in the request, an empty string is passed to the JTAPI Redirect function.

The digits must be in a valid Cisco address in one of the following forms:

- DN[partition]@terminal
- DN[partition]
- DN@terminal
- DN

T-Server will attempt to convert any of the formats listed above to the basic DN format before sending it to JTAPI in the same manner as AttributeOtherDN in RequestMakeCall requests. The digits will be passed to JTAPI “as is”, if the conversion cannot be made.

Calling Search Space Feature

Cisco Unified Communications Manager can use the Calling Search Space feature when routing a call by adding the following key-value pair in `AttributeExtensions` in `TRouteCall` in the routing strategy:

Key: `CALLING_SEARCH_SPACE`
Type: String
Value: `SEARCH_SPACE_CALLINGADDRESS` or `SEARCH_SPACE_ADDRESS`

where:

- `SEARCH_SPACE_CALLINGADDRESS` indicates that the routing strategy is to use the Search Space of the Calling Address. This is the default value.
- `SEARCH_SPACE_ADDRESS` indicates that the routing strategy is to use the Search Space of the Route Point Address.

Refer to the Cisco Unified Communications Manager System and Administration guides for more information about Partitions and Calling Search Spaces.

CUCM Partition

Starting with release 5.0 of CUCM, JTAPI supports addresses that have the same DN but belong to different partitions and treats these DNs as different addresses.

Starting with release 8.0, the following configurations are supported by T-Server:

- Addresses with the same DN, in the same partition, and in different devices are treated as shared lines.
- Addresses with the same DN, in the same partition, and in the same device are not allowed.
- Addresses with the same DN, in different partitions, and in the same device are treated as different addresses.
- Addresses with the same DN, in different partitions, and in different devices are treated as different addresses.

DN Name Mangling

To distinguish between two addresses with the same digits but with different partitions, T-Server will use CUCM partition information as a part of the DN name in the Configuration Layer. This name mangling is also compatible with the existing T-Server Shared Lines feature.

Table 12: DN Name Mangling Examples

Regular Address	6000
Shared Lines Addresses	6001@SEP00012345 6001@SEP00054321
Address with multiple partitions	6002[PART1] 6002[PART2]
Shared Lines with multiple partitions	6003[PART1]@SEP000125675 6003[PART2]@SEP000125675 6003[PART2]@SEP000765123

Partition information in the Configuration Layer must be enclosed in square brackets [ ] and must immediately follow DN digits. Terminal information is appended with an @ symbol as before.

Invalid or Missing Information Handling by T-Server

T-Server will rely on JTAPI exceptions and/or Error events if invalid or missing partition information is used for particular requests. Any information for partition-specific errors will be propagated to clients with an error message.

Single DN Configuration

Starting with release 8.0.1, T-Server supports the configuration of DNs in the Configuration Layer in the form of 6002[PART1] or 6002[PART1]@SEP000125675 even if 6002 has only one partition.

Limitation

The use of Extension Mobility with user-friendly name resolution is not supported across multiple partitions.

Customer Matters Code and Forced Authorization Code

Starting with release 8.0, T-Server extends the processing of the following T-Library requests to support Customer Matters Code (CMC) and Forced Authorization Code (FAC):

- `TMakeCall`
- `TMakePredictiveCall`
- `TInitiateTransfer`
- `TMuteTransfer`
- `TInitiateConference`
- `TRedirectCall`
- `TRouteCall`

T-Server will look for an `AttributeExtension` with the following data:

Key: `CMC`
Type: String
Value: <cmc number>

Key: `FAC`
Type: String
Value:<fac number>

Note: FAC and CMC numbers must not have the # symbol as a terminating sign. T-Server will append it automatically.

Call-Creating Requests

The following requests result in a new call or the creation of a consultation call when a call is created to a DN that requires an FAC, CMC, or both codes:

- `TMakeCall`
- `TMakePredictiveCall`
- `TInitiateTransfer`
- `TMuteTransfer`
- `TInitiateConference`
- `TRedirectCall`
- `TRouteCall`

Display Name Information

T-Server provides the ability to display name information for DNs involved in calls, if such information is available from CUCM. T-Server attaches this information to specific events within `AttributeExtensions`.

Telephone Display Name

The display name key will be in the format: `DISPLAY_NAME-<dn>`, where the value for `<dn>` is derived from the raw digits, and does not include partition or MAC address information. Display name information for all addresses on the call will be included, only if the information is available from CUCM. Additionally, a display name key value is included only if it is non-empty. T-Server will attach the display name information (if available) for the following events:

- `EventDialing`
- `EventRinging`
- `EventEstablished`
- `EventBridged`
- `EventRouteRequest`
- `EventQueued`
- `EventPartyChanged`
- `EventPartyAdded`

For `EventPartyChanged` and `EventPartyAdded` events, it will include the address involved on both branches of the call (main and consultation).

This functionality is enabled using the `use-party-display-name` configuration option.

Called and Calling Parties Display Name

T-Server is able to include display names of the calling and called parties (if available) in the `Extensions` attribute of the following events:

- `EventRinging`
- `EventQueued`
- `EventRouteRequest`
- `EventEstablished`
- `EventNetworkReached`

This functionality is enabled by the `use-called-party-display-name` and `use-calling-party-display-name` configuration options.

Do Not Disturb

T-Server supports the Do Not Disturb (DND) Call Reject feature available with Cisco Unified Communications Manager. DND can be activated from an agent's phone or by a T-Library request. When DND is enabled, no incoming calls with normal priority are presented to the user.

Note: Only devices specifically configured with DND Call Reject are supported by this feature.

Dual-Tone Multi-Frequency Digits

T-Server is able to send Dual-Tone Multi-Frequency (DTMF) digits to the receiving party of a call through the `TSendDTMF` request. `TSendDTMF` accepts strings of fewer than `30` characters, with only the following characters allowed:

`1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `0`, `A`, `B`, `C`, `D`, `#`, `*`.

`EventDTMFSent` is sent in response to `TSendDTMF`. However, this event is not distributed if DTMF digits are sent via some other means (for example, through the IP phone keypad), and it is not distributed from the backup T-Server when T-Server is in a high-availability configuration.

Extension Mobility

T-Server supports the Extension Mobility (EM) feature available with Cisco Unified Communications Manager. This allows users to dynamically declare their EM profile number on a different telephone set.

Extension Mobility Profile with Unique DNs

When Extension Mobility is used and the EM number is different from the user's regular extension, no special configuration is needed in the Configuration Layer. In fact an EM DN should be unique within the T-Server account.

Extension Mobility Profile with the Same DN as Regular Extension DN

When Extension Mobility (EM) is used and the EM number is the same as the user's regular extension number, the telephones in both the original and the new extension location will ring when a call is routed to the user's number.

This results in Shared Lines containing two DNs: an EM number and a regular extension number. Configuration using a user-friendly suffixes method must be used in the EM profile, while the user's regular extension can be configured as the shared line DN by using either the MAC address or the user-friendly-names method.

All limitations related to single DN configuration (without any suffix) for Shared Lines are applicable in cases where the Extension Mobility has the same DN as a regular extension DN. T-Server will generate the standard message `51114 Shared Line DN is not configured correctly in CME` when the EM profile is logged in on another phone. See the configuration of "Shared Lines" on page 164.

Extension Mobility and Shared Lines

Extension Mobility is fully operable together with the Shared Lines feature. If the EM profile has a DN from Shared Lines, the user-friendly configuration method must be used for the EM Shared Line DN. See sample configuration in Figure 13, where EM Shared Line DN=`1111@EM` and `uniqueDN=2222`.

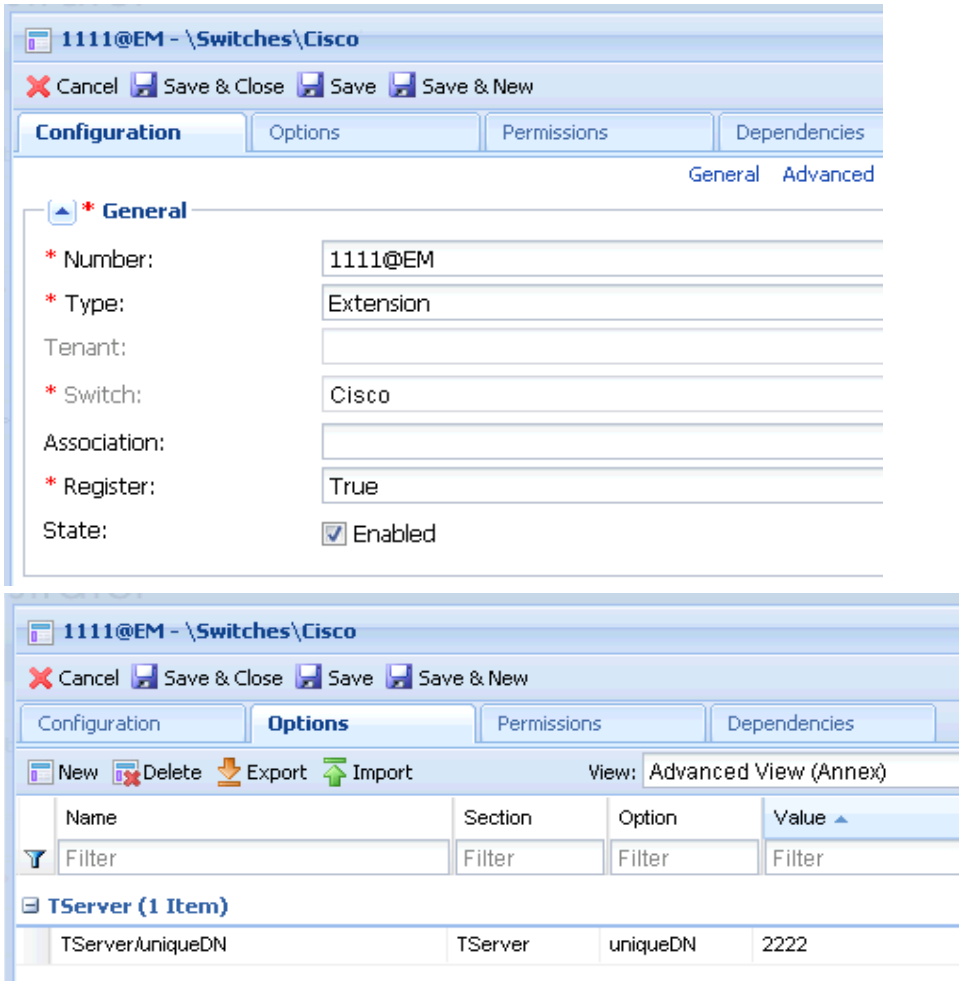

Figure 13: Shared Line DN=1111@EM and uniqueDN=2222

Hunt Groups

T-Server supports Hunt Groups on Cisco Unified Communications Manager (CUCM). This feature is supported through two Application-level configuration options: `HUNTLIST_ENABLED` in the `jtapi` section, and `ignore-cisco-cause-500` in the `globalgroup` section.

T-Server supports the Top Down, Circular, Longest Idle Time, and Broadcast (parallel ringing) type of Hunt Group configuration.

- Top Down—If you choose this distribution algorithm, CUCM distributes a call to idle or available members starting from the first idle or available member of a line group to the last idle or available member.
- Circular—If you choose this distribution algorithm, CUCM distributes a call to idle or available members starting from the (*n*+1)th member of a line group, where the *n*th member is the member to which CUCM most recently extended a call. If the *n*th member is the last member of a line group, CUCM distributes a call starting from the top of the line group.
- Longest Idle Time—If you choose this distribution algorithm, CUCM only distributes a call to idle members, starting from the longest idle member to the least idle member of a line group.

Introduced in T-Server 8.1.201.15

- Broadcast (parallel ringing)—If you choose this distribution algorithm, CUCM distributes a call to all idle and available extensions in the group simultaneously

Note: For this feature to properly function, all `HuntPilot` DNs must be removed from the Configuration Layer.

The following Application-level configuration options are used to configure this feature:

- `HUNTLIST_ENABLED`
- `ignore-cisco-cause-500`

Logging of Network Connection Failures Between JTAPI and T-Server

To perform logging, in a situation where the network connection fails between JTAPI and T-Server, T-Server writes a severe error condition to create log files. A new log file is generated as each previous one becomes full at a specified capacity. Each log file is named according to a pattern controlled by the properties of the `TServerTraceFileBase` and `TServerTraceFileExt` options. The `TServerTraceFileBase` option determines the prefix of each log file name, and the `TServerTraceFileExt` option determines the suffix. The

TServerTraceMaxFiles option determines how many files should be created before the first one is overwritten. And the TServerTraceMaxFileSize option is the maximum file size in bytes for individual files.

The following is an example of how filenames are constructed: TServerTraceFileBase + NN + "." + TServerTraceFileExt, where NN is an increasing number.

The following options are used to configure this feature:

- TServerTraceFileBase
- TServerTraceFileExt
- TServerTraceMaxFiles
- TServerTraceMaxFileSize

Configure these options in the jtapi section on the Options tab for the T-Server Application object in the Configuration Layer.

Music and Announcements

T-Server is able to control the playing of announcements and music on Routing Points and ACD Queues in Cisco Unified Communications Manager (CUCM). Music and announcements can come from three sources:

1. **Cisco Unified Communications Manager Music On Hold (MOH) Server:** The server that provides music to endpoints when an IP phone is placed on hold. The selection of music to be played can be customized in CUCM for a particular Routing Point. Refer also to your Cisco Unified Communications Manager Administration Guide.
2. **Stream Manager:** A Genesys client application that is able to stream media files. For more information about configuring, refer to the Chapter 14, “Stream Manager Configuration,” on page 291. Stream Manager plays files using a codec that is negotiated with the CUCM switch. The list of possible codecs that can be used is configured in the T-Server option audio-codec.
3. **Media Server:** Genesys Media Server (GMS) is a standards-based media processing platform that is able to stream media files and record media streams. For more information about T-Server-CUCM to Media Server Connector, see the *Genesys Media Server 8.1 Deployment Guide*.

T-Server can use Cisco UCM MOH Server to play music on hold, for a call on a Routing Point DN, if the DN is configured in the Configuration Layer.

Procedure:
Configuring a DN to use CUCM MOH Server

Start of procedure

1. Under a `Switch` object, select the DNs folder.
2. Right click to open the `Properties` dialog box of a particular Routing Point DN.
3. Select the `Annex` tab.
4. Create a new section named `TServer`.
5. Within that section, create a new option named `moh-server-music` and an arbitrary string value (for example: `mohserver-music-treatment-01`). T-Server uses CUCM MOH Server instead of Stream Manager or Media Server to play music treatment if the name of the music file, as defined in the strategy, is equal to the value of this option, in this case `mohserver-music-treatment-01`. For all other music files Stream Manager or Media Server will be used.
6. Repeat Steps 1-5 for all Routing Point DNs that require this functionality.
7. Load a Strategy in URS (Universal Routing Server) on the Routing Point DN.

End of procedure

Announcement Treatments on Routing Points

The two announcement treatments, `PlayAnnouncement` and `PlayAnnouncementAndDigits`, include the following parameters:

- `LANGUAGE`: Ignored.
- `MSGID`: Ignored.
- `MSGTXT`: Ignored.
- `PROMPT`: Contains up to 10 sub-prompts. Each of these will contain a music file, and they are played in order.
- `INTERRUPTABLE`: If set to `true`, the caller can interrupt the announcement with a `DTMF` keystroke.
- `ID`: Contains an integer that refers to the file `announcement/<integer>`. For example, a value of ID `1` would refer to the file `announcement/1_alaw.au`, if the G.711 a-law codec is used.
- `DIGITS`: Ignored.
- `USER_ID`: Ignored.
- `USER_ANN_ID`: Ignored.
- `TEXT`: Ignored.

Cisco Unified Communications Manager Music On Hold does not support these treatment types. Only Stream Manager or Media Server can be the source for the announcements.

If the treatment is terminated early because of a problem with Stream Manager, Media Server, or T-Server, T-Server sets the Extension data fields, `ERR_CODE` and `ERR_TEXT`. To determine whether these fields exist, and/or to determine their values, from a routing strategy, use the `ExtensionData` function. Place this function on a normal completion branch (not the error branch) after the treatment.

Refer to the *Universal Routing 8.1 Reference Manual* for more information about using and configuring strategies.

Note: Leave the `Wait For Treatment End` check box selected to enable these treatments to play until completion.

Music Treatment on ACD Queues

Each ACD Queue can play a specific Stream Manager or Media Server music file. The file is specified in the DN `queue-music` option (in the `Options/Annex` > `TServer` section). The file is specified as `<directory/music file name>`, where `<directory>` is a sub-directory off the `Stream Manager` root directory, and `<music file name>` refers to the name of the file, without the codec extension. If Stream Manager or Media Server is not configured, or if it has failed, then music is played by the CUCM MOH Server.

If an ACD Queue does not have the `queue-music` option configured, T-Server plays the Stream Manager or Media Server music file that is specified in the T-Server `queue-music` option.

Note: Leave the `compatible mode` checkbox unselected when configuring the music treatment on an ACD Queue.

Music Treatments on Routing Points (TreatmentMusic)

The music treatment includes the following parameters:

- `MUSIC_DN`: Specifies the music file for Stream Manager or Media Server to play. The format is `<directory>/<music file name>`

  where:

  - `<directory>` is a sub-directory off the `Stream Manager` or `Media Server` root directory
  - `<music file name>` is the name of the file without the codec extension.

  If the music file name is prefixed with a plus sign `+`, the music file loops continuously, otherwise the file is played only once. However, if the `MUSIC_DN` field is blank, `music-on-hold` music is played by default. For

example, `music/in_queue` would refer to the file `music/in_queue_alaw.au` if G.711 a-law codec is used.

- `DURATION`: Specifies the duration, in seconds, that the music plays. Note that this parameter is ignored if `MUSIC_DN` is blank (that is, if `music-on-hold` is used).

This treatment terminates before the music file has finished playing. To continue playing music after the treatment terminates, consider one of the following strategies in Interaction Routing Designer:

- Execute the treatment inside a route-selection treatment block. In this case the treatment continues until a route target is selected.
- Follow the treatment with the `SuspendForTreatmentEnd` function. In this case, the treatment plays music until terminated after `DURATION` seconds.
- Follow the treatment with the `delay` function. In this case, the treatment plays music for `delay` seconds. If `DURATION` is less than `delay`, silence is played for the time difference.

Refer to the *Universal Routing 8.1 Reference Manual* for more information about using and configuring strategies.

Disabling the Default MOH Treatment

Introduced in T-Server 8.1.201.15

T-Server provides the ability to turn off the default MOH (Music On Hold) treatment provided by the CUCM MOH Server when T-Server does not have an active Media Server connection. This feature is controlled by the `force-moh-on-ms-down` configuration option.

T-Server generates `EventTreatmentApplied` for all selected treatments of type `TreatmentMusic` to include the proper treatment type of `TreatmentMusic` and also includes a new `Extensions` attribute key `MOH` to distinguish between music treatment via Media Server and CUCM MOH Server.

`Extensions` attribute

Key: `MOH`

Type: Integer

Values: `0`, `1`

When the `AttributeTreatmentType` in `EventTreatmentApplied` or `EventTreatmentEnd` is of value `TreatmentMusic`, the `MOH` key is used to describe the specific music type—`1` for the CUCM MOH Server treatment and `0` for Media Server treatment music.

Call Recording (RecordUserAnnouncement)

The call recording function is supported by the `RecordUserAnnouncement` treatment. By default, the recorded user's announcement (an audio file) is saved into a single `users` folder for a single Stream Manager or Media Server, with a filename specified in the `RecordUserAnnouncement` treatment (which

must coincide with the Configuration Layer's Tenant's name). The format of the recorded file depends on the audio-codec chosen during the terminal capability negotiation procedure. Multiple Stream Managers or Media Servers can share a single storage for media files, even if they are executed on different hosts, under different operating systems. In this case, issues of file accessibility from the shared storage for all hosts should be resolved on an organizational/administrative level.

When a treatment of type `TreatmentRecordUserAnnouncement` is issued, it is passed to Stream Manager or Media Server. In cases where multiple Stream Managers are connected, the `TreatmentRecordUserAnnouncement` command is chosen among all available Stream Managers or Media Servers in a round-robin fashion.

The `RecordUserAnnouncement` announcement treatment includes the following parameters:

- `USER_ID`: Mandatory
- `ABORT_DIGITS`: Ignored
- `TERM_DIGITS`: Ignored
- `RESET_DIGITS`: Ignored
- `START_TIMEOUT`: Specifies the time during which a user should start talking. If the timeout expires, the treatment is terminated with an error code.
- `TOTAL_TIMEOUT`: Specifies the maximum time allocated to recording the announcement. In cases where the call recording `start_timeout` parameter affects the `total_timeout` parameter, Genesys recommends that the `start_timeout` parameter field be left empty.
- `PROMPT`: Specifies the prompt that is played before recording.

The user's recorded announcement is saved in the SM "users" directory with the name `<USER_ID>-<number>_<codec>.wav` (where `<number>` is the sequence number of the file for `USER_ID`). The user who is executing the SM application must have "write" permission to this directory. The audio codec used in the recoding will be one of the codecs defined on page 293.

When recording calls the following options are used:

- **Stream Manager:**
  - `max-record-file-size`: Specifies the maximum size, in KB, of the audio file used for recording.
  - `max-record-time`: Specifies the maximum recording time, in seconds.
  - `max-record-silence`: Specifies the maximum amount of time, in seconds, that silence can be detected during a recording.
- **Media Server:**
  - `record-content-type`: Specifies the MIME type of the recorded file.
  - `max-record-time`: Specifies the maximum recording time, in seconds.
  - `max-record-silence`: Specifies the maximum amount of time, in seconds, that silence can be detected during a recording.

Additional recording parameters should be specified in the RecordUserAnnouncement treatment. Refer to the *Universal Routing 8.1 Reference Manual* for more information.

File recording ceases when:

- An interruptible treatment is interrupted by DTMF entry.
- The `max-record-time` interval has expired.
- The `max-record-silence` or the `max-record-file-size` limit is reached.
- T-Server issues `EventTreatmentEnd` in these cases.

Busy, Fast Busy, Silence and RingBack Treatments on Routing Points

These treatments continuously loop a pre-defined audio file to a call. You must configure Stream Manager or Media Server to use these treatments. `Music-on-hold` cannot be the source of the audio file. The treatment types are:

- `Busy`: Plays a busy tone. To define the `busy` tone audio file, configure the T-Server `busy-tone` option.
- `Fast Busy`: Plays a fast busy tone. To define the `fast busy` tone audio file, configure the T-Server `fast-busy-tone` option.
- `Silence`: Plays no sound. To define the `silence` audio file, configure the T-Server `silence-tone` option.
- `Ringback`: Plays a ringback tone. To define the `ringback` audio file, configure the T-Server `ring-tone` option.

T-Server is capable of applying a treatment with a `RingBack` type to a new call without a charge to the customer. This treatment uses the Cisco Unified Communications Manager native ringback feature rather than connecting a customer with Stream Manager or Media Server to play a predefined audio file with a ringback sound. This treatment should be the first treatment in a strategy.

Note: The `DURATION` parameter is not used for Cisco Unified Communications Manager.

Refer to the *Universal Routing 8.1Reference Manual* for more information about using and configuring strategies.

Predictive Dialing

T-Server provides support for outbound campaigns in predictive/progressive dialing mode, either by using a Dialogic board in ASM (Active Switching Matrix) mode, or directly through the `TMakePredictiveCall` request.

It is recommended not to apply treatments on a Routing Point during predictive dialing to prevent excessive use of dialogic channels by the Call Progress Detection (CPD) Server because of a premature `CONNECTED` ISDN message. Also, a call should be routed off a Routing Point within a four-second timeout as is configured on Cisco Unified Communications Manager.

Outbound Dialing with TMakePredictiveCall

Outbound Contact Server (OCS) uses a direct interface to T-Server via the T-Library `TMakePredictiveCall` request. T-Server initiates the outbound call, and when the call is connected, diverts the call to an agent.

This is the simplest way to configure the outbound solution; however the Cisco Unified Communications Manager and the gateway are unable to perform Call Progress Detection (CPD). Although, T-Server is able to recognize the call results `busy`, `answered`, `not answered`, and `SIT tone`, it is unable to detect `fax` or `answering machine` responses. Therefore, these latter calls are delivered to agents along with regular phone calls.

Note: `TMakePredictiveCall` requests can be sent on behalf of ACD Queues or Routing Points

Figure 14 illustrates a scenario in which `TMakePredictiveCall` is used for Outbound dialing.

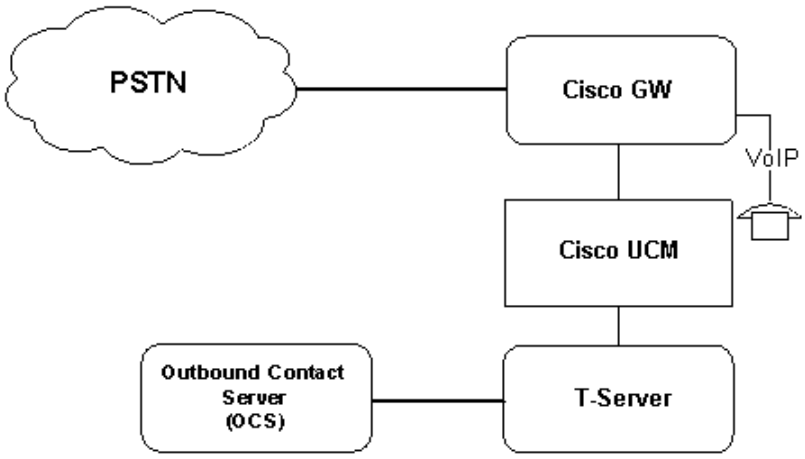

Figure 14: Using TMakePredictiveCall for Outbound Dialing

With this method of predictive dialing, you do not need a Call Progress Detection (CPD) server or Dialogic board.

Outbound Calling with Dialogic Dialer

Outbound Contact Server (OCS) uses the Dialogic dialer to place a call from Dialogic to the ACD Queue or Routing Point on the Cisco Unified Communications Manager switch through the first `T1` trunk. Dialogic then places an outbound call using the PSTN connection on the second `T1` trunk. After a call is connected, Dialogic bridges the two calls using the Active Switching Matrix (ASM). Dialogic stays on the call path for the entire duration of the call.

Figure 15 illustrates the way in which T-Server supports outbound dialing with the Dialogic dialer.

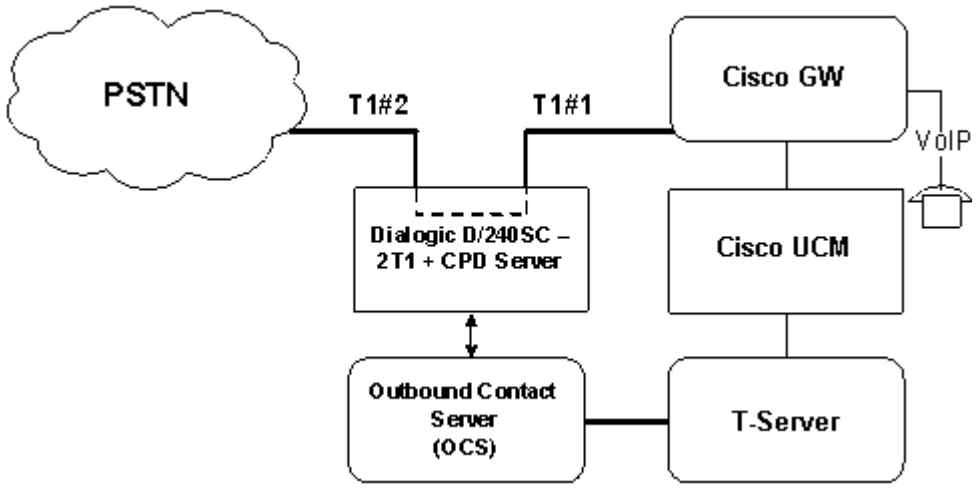

Figure 15: Outbound Calling with Dialogic Dialer

Providing AttributeDNIS in EventDialing

Starting with release 8.1.202.07, T-Server can include `AttributeDNIS` in `EventDialing` for most call flows by setting the restricted option, `delay-dialing`, to `true`. This causes `EventDialing` to be delayed until DNIS info is available.

A few call flows do not use this option and, therefore, do not include DNIS in `EventDialing`:

- Predictive calling
- Unparking a parked call
- Calls between Intercom DNs, such as during supervisor call monitoring

Querying JTAPI on Call State/Active Call on DN

T-Server has an enhanced ability to detect and clean up stuck calls after a link disconnection/reconnection by using a procedure that allows T-Server to query JTAPI for all calls on a specific DN. A `TQueryAddress` request is sent by T-Server to JTAPI *only* if T-Server has a call in memory for a specific address. JTAPI returns `CCM_TEV_ADDRESS_QUERY` containing the number of calls and a list of `CallIDs`. If the query determines that the specified call no longer exists, T-Server releases the call, and distributes all appropriate events. The `address-query-delay` configuration option is used to specify the amount of time T-Server waits after receiving an `EventDNBackInService` message prior to sending a `TQueryAddress` request for a DN.

Note: When the TSCP initiates the check for call cleanup, T-Server will initiate a query to the switch instead of immediately releasing the call.

Redirect On No Answer

This feature is intended to prevent calls from ringing indefinitely at an agent desktop when that agent logged in, but then left his or her desk, without logging out. When this feature is invoked, T-Server allows a call to ring at an agent's phone for only a specified number of seconds before considering the agent as "walked away". When this occurs, T-Server reroutes the call to another address (usually configured to be a Routing Point) and, if the agent is in `Ready` state, sets the agent to `NotReady` (`WalkAway`) state.

T-Server provides the following options for defining the behavior of the Redirect On No Answer feature:

1. Application-level configuration options
2. Agent Login-level configuration options
3. `AttributeExtensions` keys

Application-Level Options

The following configuration options can be set globally for all calls in the T-Server `Application` object:

- `agent-no-answer-action`
- `agent-no-answer-overflow`
- `agent-no-answer-timeout`

Agent Login-Level Options

The following configuration options can be set for individual Agent Login objects to override the Application-level options:

- agent-no-answer-action
- agent-no-answer-overflow
- agent-no-answer-timeout

Note: For any of these Agent Login-level options to override the Application-level options, you must set the agent-no-answer-timeout option.

AttributeExtensions Keys

For all of the No-Answer options, you can specify the corresponding Extensions attribute in the TRouteCall request, to override the configured value for individual calls. This method allows the no-answer behavior to be determined in a routing strategy. The three extensions are:

- Key: NO_ANSWER_TIMEOUT

  Value: A string representing the timeout.
- Key: NO_ANSWER_ACTION

  Value: A string representing what action T-Server performs for an agent. It can be either notready, walkaway, or logout.
- Key: NO_ANSWER_OVERFLOW

  Value: A string representing an overflow DN.

If you do not set a timeout value, the call is redirected to the value specified in the default-dn option on page 242; or if that option is not set, the phone continues to ring without redirection (but the agent is still placed in the NotReady state).

Note: To override the no-answer behavior set by the Agent Login-level and/or Application-level options, you must add the NO_ANSWER_TIMEOUT key to AttributeExtensions in a routing strategy.

Retrieval and Distribution of Modified CLID

Starting with release 8.0, T-Server is able to retrieve extended call information about a modified Calling Number if Address is configured on CUCM using the External Phone Number Mask modifier. This allows an external customer to call back using modified phone numbers rather than the ANI of the original call.

Distribution of Events

T-Server uses the `ModifiedCallingAddress` event to communicate new digits to T-Server clients.

CUCM modifies the `CallingAddress` only for outbound calls. For this reason, T-Server will use `EventNetworkReached` to communicate modified calling digits to T-Server clients. For internal calls, `ModifiedCallingAddress` is the same as the `OriginalCallingAddress`.

T-Server distributes a modified Calling Number only for the Network Reached event. Consequently, the following `AttributeExtensions` is used for extra information:

Key: `MODIFIED_CALLING_NUMBER`
Type: String
Value: <modified number>

Routing Points with Multiple Partitions

When a Routing Point DN is registered on Cisco Unified Communications Manager, T-Server enumerates all the partitions for this Routing Point DN and registers all terminals associated with the partitions.

As JTAPI does not support Multiple Partitioning, Genesys recommends that the Routing Point be created as a single DN in the Configuration Layer without specifying a partition suffix. This DN will receive all Routing Point related events coming from any partition, but events will not have partition-specific information.

Shared Lines

T-Server supports Shared Lines available with Cisco Unified Communications Manager. Incoming calls to Shared Lines arrive directly to the Shared Line DN.

Note: Statistics for Shared Lines DNs may differ from traditional Agent DNs (ACD or Extension DNs).

Configuration for Multiple Extensions within Shared Lines

DNs within Shared Lines in the Configuration Layer can be configured differently:

- Using MAC address suffixes as a part of the DN number.

- Using user-friendly suffixes on multi-line IP phones with a unique DN on a second line.

Shared Lines configuration may have, simultaneously, DNs configured with MAC addresses and DNs with user-friendly suffixes. If Shared Line DN is being used with an Extension Mobility (EM) profile, the user-friendly configuration method should be used for the EM Shared Line DN. Refer to the Extension Mobility section on page 151 for more details.

Notes:
- T-Server does not allow single DN configuration (without a suffix) if the DN number is configured as Shared Lines on CUCM. In this case T-Server will generate the standard message `51114 Shared Line DN is not configured correctly in CME`.
- Agent operations (login, ready, not ready, logout) are not supported for Shared Lines.

Procedure: Configuration using MAC address suffixes

Start of procedure

1. Under a `Switch object`, select and right click the `DNs` folder.
2. Select the `New DN` item from the menu.
3. Enter a number in the DN@MAC format, for example, `5005@SEP00127F73F273` as in Figure 16. The MAC address can be found on the back of your IP phone.

Note: The `uniqueDN` option is not required with MAC address suffixes.

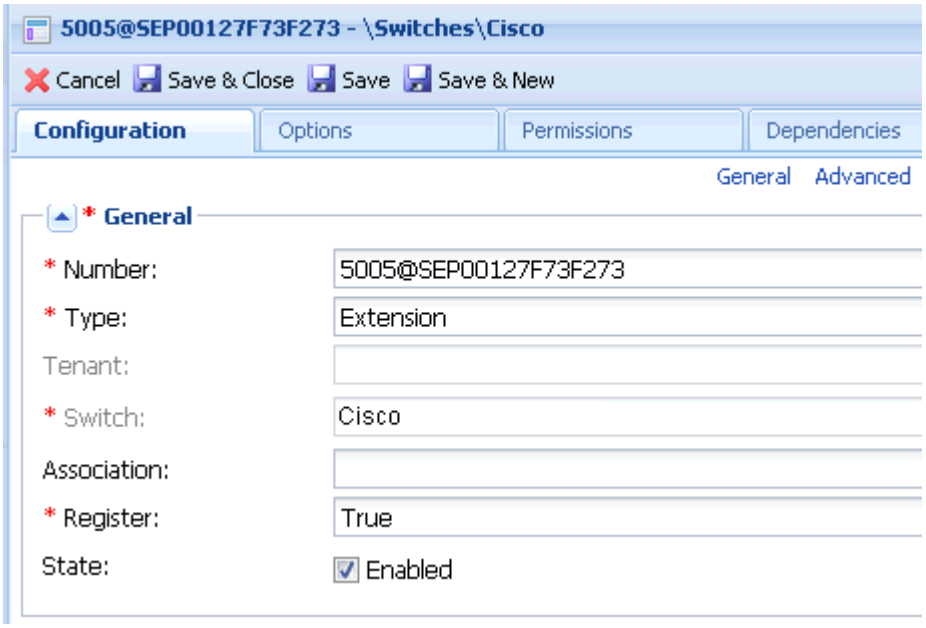

Figure 16: Number in DN@MAC format

End of procedure

Procedure:
Configuration using user-friendly address suffixes

Prerequisites

- Make sure the IP phones where you want to use the user-friendly suffixes for shared lines have at least two lines: one for the shared line DN number and another line for the DN number that is unique within the T-Server user account on Cisco Unified Communications Manager.

Start of procedure

1. Under a `Switch` object, select the DNs folder.
2. Right click `DNs` folder and select `New DN` menu item.
3. Enter a number in the `DN@<user-friendly-name>` format, for example, `5005@phone1` as in Figure 17.

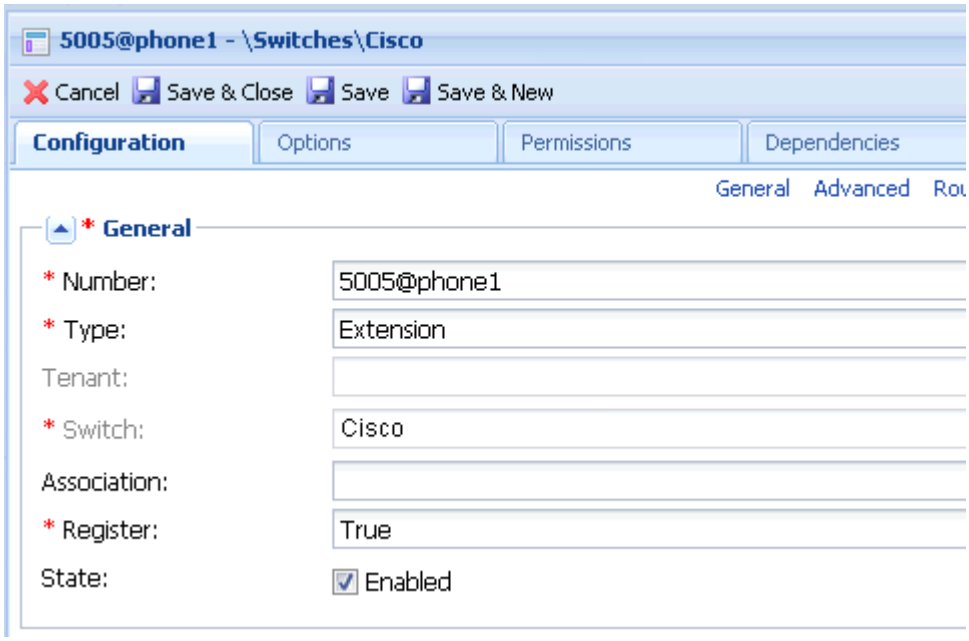

Figure 17: Number in DN@<user-friendly-name> format

4. Click the `Annex` tab.
5. Create a new section named `TServer`. Within that section, create a new option named `uniqueDN`.
6. Set the value of the option to a DN from the second line on the IP phone, for example, 1231 as in Figure 18.

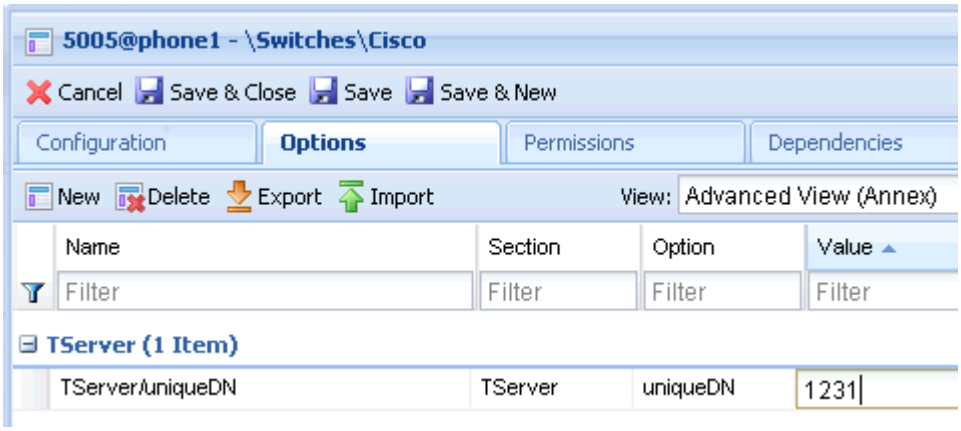

Figure 18: Value for the uniqueDN option

Apart from assisting in resolving MAC addresses in Shared Lines, the second line DN is fully functional and can be used as regular extension DN in the Genesys suite.

End of procedure

Single-Step Conference

T-Server does not support a single-step conference to a regular DN. A single-step conference is supported when T-Server performs emergency call recording if a single-step conference call request is made to a predefined `gcti::record` number.

A single-step conference is also supported for the following Call Monitoring types:

- Intrusion
- Silent Monitoring

Limitation

The support of `TSingleStepConference` is limited in T-Server. Only `TSingleStepConference` to `gcti::record` is allowed.

Socket Mode of Communication

Starting with 7.6, T-Server uses a standard `link-`*`n`*`-name` link configuration option set in Socket mode.

The `TServer` section will contain options `link-1-name`, `link-2-name`, ..., `link-4-name` while separate sections will contain link parameters.

See the following example:

```
[TServer]
application=T-Server
packet-size=60
queue-music=music/in_queue
audio-codec=4

link-1-name=link-1
link-2-name=link-2
link-3-name=link-3
link-4-name=link-4

[link-1]
hostname=localhost
port=7888
protocol=tcp
ccm-host= CTIManager1.acme.com
password=*******
user-login=ccm-user-1
```

```
[link-2]
hostname= localhost
protocol=tcp
ccm-host= CTIManager2.acme.com
port=7890
password=*******
user-login=ccm-user-2

[link-3]
hostname= localhost
protocol=tcp
ccm-host= CTIManager3.acme.com
port=7892
password=*******
user-login=ccm-user-3

[link-4]
hostname= localhost
protocol=tcp
ccm-host= CTIManager4.acme.com
port=7894
password=*******
user-login=ccm-user-4
```

See Table 18 on page 258 for a more detailed explanation of the link options.

Make sure that the `ccm-host`, `port`, and `user-login` options are different for all link sections. Also, make sure that the `port` values are different from T-Server listening ports from the `Server Info` tab in the Configuration Layer and the `sm-port` option in the `TServer` section.

Supervisor Monitoring

Starting with version 8.0, T-Server supports supervisor monitoring. The call supervision functionality enables contact center managers to monitor agents.

Call Supervision Subscription

Subscription is designed for the supervisors. Supervisors can subscribe to monitor one agent. If a subscription is active, T-Server automatically invites the supervisor to all calls where the agent participates. T-Server stops working in this mode as soon as the subscription is cancelled.

Monitoring Session

A Monitoring Session is a period where a supervisor monitors and agent-customer conversation. Monitoring Sessions can be created as a result of an active subscription. A Monitoring Session starts when a supervisor joins the call and it ends when the supervisor disconnects.

Modes of Call Supervision

Call supervision can be carried out in different modes:

- Silent Monitoring: Hiding the supervisor presence from all call participants, including the monitored agent who is the target of the supervisor's attention in this case.
- Whisper Coaching: The supervisor’s presence is hidden from all call participants except the monitored agent. In this scenario, only the monitored agent can hear the supervisor.

Scope of Call Supervision

Call supervision scope defines the part of the call to be monitored by the supervisor. The only scope available in T-Server for this is `agent`. The supervisor follows the monitored agent in this mode. This means that the supervisor leaves the call as soon as agent is gone.

Types of Call Supervision

Call supervision type specifies the number of calls to be monitored. There are two options:

- One Call: Only one call is monitored.
- All Calls: All calls are monitored.

Supervision type plays an important role in the monitoring subscription. If the One Call option is chosen for the subscription, the subscription gets cancelled automatically when the supervisor finishes monitoring the first call on the monitored agent. In the case of All Calls, the supervisor should cancel the subscription manually to stop monitoring the agent's calls.

Intrusion

Intrusion occurs if a Supervisor activates a new call supervision subscription to monitor the agent who is currently on the call. In this case T-Server creates the requested subscription and immediately invites the Supervisor to join the existing call.

Call Supervision Subscription

General approach

Call supervision subscription is implemented through the use of two T-Library requests:

- `TMonitorNextCall`: starts new subscription.
- `TCancelMonitoring`: deletes existing subscription.

Both of these requests use `AttributeThisDN` to point to the supervisor and `AttributeOtherDN` to define the monitored agent. Additionally, the `TMonitorNextCall` request carries the information to specify the mode, scope, and type of the monitoring subscription to be created. This information is transferred in the request as follows:

- `AttributeMonitorNextCallType`: Defines the type of call supervision.
  - `MonitorOneCall`: A subscription is created to monitor one call only and will be terminated automatically.
  - `MonitorAllCalls`: A subscription is created to monitor all calls and is terminated when the supervisor sends a `TCancelMonitoring` request.
- `AttributeExtensions/MonitorMode`: Defines the mode of call supervision.
- `AttributeExtensions/MonitorScope`: Defines the scope of call supervision.

Creating Subscriptions

T-Server creates a new subscription based on the `TMonitorNextCall` request that is submitted by the supervisor. Based on certain circumstances, the request can be either accepted or rejected. Requests are rejected in the following cases:

- Either the supervisor or agent DN is not internal, and monitored by the switch.
- Either the supervisor or the agent DN is not configured in the Configuration Layer.
- The agent DN is already monitored.

If a request is accepted, T-Server creates a new subscription and initializes it with the information that is passed in the `AttributeMonitorNextCallType` request attribute and in the `AttributeExtensions` key-value pairs: `MonitorScope` and `MonitorMode`.

If the monitor mode is missing, the value from the `default-monitor-mode` configuration option will be used.

The `AttributeMonitorNextCallType` is mandatory.

T-Server confirms the creation of a new subscription for both the supervisor and agent by sending `EventMonitoringNextCall` to both destinations. This event always contains `AttributeExtensions` with both monitoring extensions representing the monitoring configuration defined in T-Server for a new subscription.

Monitoring Parameters Used for Subscription Initialization

To specify MonitorMode for a subscription, the following key-value pairs must be attached to the attribute Extensions of the TMonitorNextCall request:

Key: MonitorMode
Type: String
Values:

normal	Silent monitoring (mutes supervisor's connection, possible warning beep).
mute	Silent monitoring (mutes supervisor's connection).
coach	Whisper coaching (only monitored agent can hear the supervisor).
connect	Opens the supervisor's presence (Not supported. Use a two-step conference instead).

The default-monitor-mode option is used when the MonitorMode extension in the TMonitorNextCall request is not specified or is specified incorrectly.

Key: MonitorScope
Type: String
Values: agent (Only possible value)

Canceling Subscriptions

There are two scenarios used in T-Server to cancel active subscriptions:

- Incoming TCancelMonitoring request.
- Completion of a monitoring session created based on the Monitor One Call type of subscription.

T-Server generates EventMonitoringCancelled events for both supervisor and agent to inform them that a subscription is cancelled.

Call Intrusion

Call intrusion is a scenario when T-Server gets RequestMonitorNextCall from the supervisor at the time when an agent is to be monitored, but is already on the call. T-Server's behavior in this scenario is controlled by the intrusion-enabled option. If this option is set to false, T-Server only creates a new subscription. If intrusion-enabled is set to true, T-Server creates a new subscription and immediately invites the supervisor to join the existing call.

Mute Off/Mute On

T-Server is able to switch the supervisor from silent monitoring to coach conferencing (barge-in functionality), and switch from coach conferencing to

silent monitoring. This functionality is provided by implementation of two T-Library requests:

- `TSetMuteOff`—Switch from silent monitoring to coach conferencing.
- `TSetMuteOn`—Switch from coach conferencing to silent monitoring.

Request `TSetMuteOff` will be performed only if the corresponding supervisor DN is in the process of silent monitoring. When the request has completed successfully, `EventMuteOff` is sent.

Request `TSetMuteOn` will be performed only if a supervisor is in Whisper mode. When the request has completed successfully, `EventMuteOn` is sent.

Note: T-Server does not support the supervisor changing from silent monitoring to regular conference.

Whisper Coaching and Extra Instance of Intercom Call

Whisper coaching in T-Server is performed using the CUCM Intercom feature.

The Intercom feature allows one user to call another user and have the call answered automatically, and with one-way media from the caller to the called party, regardless of whether the called party is busy or idle. The called user can press the talk back softkey (unmarked key) on their phone display to start talking to the caller. Only a specially configured Intercom address on the phone can initiate an Intercom call. JTAPI creates a new type of address object named `CiscoIntercomAddress` for Intercom addresses that are configured on the phone.

Only one Intercom DN can call another Intercom DN. For this reason, the DN configuration option `intercomDN` must be configured on both the supervisor phone and the agent phone for the Whisper feature to work. Intercom DNs must be configured only in the `Annex` tabs of the corresponding agent/supervisor DNs.

Note: Starting with T-Server version 8.1.201.20, Extension DNs are no longer required to be created for Intercom DNs. If an Intercom DN is created as a DN of type `Extension`, an additional license will be taken when Stat Server or any other client registers on that Extension representing the Intercom DN.

Figure 19 shows an Intercom DN sample configuration, where DN 5012 is an agent DN configured with the `intercomDN` option set to `3012`, and DN 3012 is an Intercom DN.

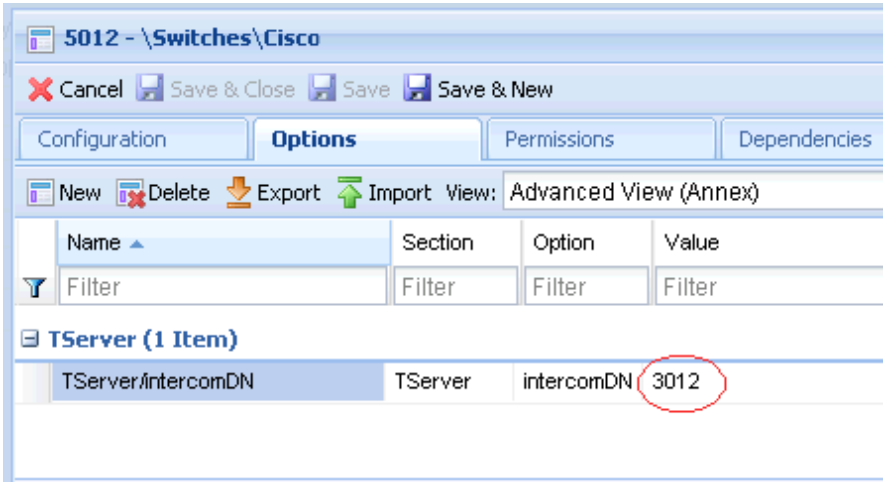

Figure 19: Intercom DN Sample Configuration

A regular `MakeCall` is silently used between the supervisor Intercom DN and the agent Intercom DN. These calls are answered automatically.

Transport Layer Security

Introduced in T-Server 8.1.200.06

CTI-level communication between T-Server and the Genesys Java Telephony API (JTAPI) process and the Cisco CTIManager can now be encrypted. Communication between T-Server and the Cisco CTIManager can traverse multiple network paths. Each link within T-Server communicates over a TCP connection to a Genesys JTAPI process and each JTAPI process communicates over a TCP connection to the Cisco CTIManager. Because T-Server for Cisco UCM supports multiple links, the number of network paths (CTI/TCP connections) is twice that of the number of links (2 links - 4 TCP connections, 3 links - 6 TCP connections).

This feature enables secure communication over TCP sockets originating and terminating between the JTAPI process and the Cisco CTIManager. JTAPI provides the necessary functionality required to provide two-way authentication and secure communication between JTAPI and Cisco CTIManager. This functionality is dependent on client server certificates, and is out of scope of this document.

Securing Communication with JTAPI

Securing communication with JTAPI requires communication to:

1. Cisco TFTP server to obtain the trusted server certificate (using the `tls-tftp-host` and `tls-tftp-port` configuration options).
2. Cisco CAPF server to obtain the client certificates (using the `tls-capf-host` and `tls-capf-port` configuration options).

When T-Server starts, all required certificates are automatically downloaded by the Genesys JTAPI process and stored in the local folder specified by the `tls-cert-path` configuration option. These downloaded certificates are encrypted based on the password defined by the password option.

Each connection/link between JTAPI and CTIManager requires its own unique client certificate. To obtain a client certificate for a particular link, an authorization code and an instance ID are required. Two link-level options, `tls-auth-code` and `tls-instance-id`, represent the authorization code and the instance ID, respectively, for TLS-enabled configurations.

The authorization code is required only for the initial download of the client certificates.

The instance ID provides a method to associate a specific certificate with a specific link and is configured in the Cisco UCM database. There is a one-to-one relationship between a link and an instance ID. Using the same instance ID on different links simultaneously might cause the certificate to be invalidated by Cisco.

Note: The first initial download of the server certificate is considered trusted. For this reason, it is recommended that the initial T-Server run, after configuring TLS, be done in a secure network (environment).

TLS Configuration

Table 13 describes how to enable TLS communication.

Table 13: Configuring TLS

Objective	Key Procedures and Actions
1. Obtain the certificates.	1. 1.Configure the following options in the `link-tls` section: • `password` • `tls-cert-path` • `tls-capf-host` • `tls-capf-port` • `tls-tftp-host` • `tls-tftp-port` 2. Configure the following options in the `link` section specified by the `link-`*n*`-name` option: • `tls-auth-code` • `tls-instance-id` 3. Start T-Server. 4. Check that certificates were obtained and are located in the directory specified in `tls-cert-path`. 5. Stop T-Server. 6. Remove the `tls-auth-code` option from the `link` section.
2. Run T-Server with the secure connection.	1. Ensure that the `link` section contains only the TLS-related option `tls-instance-id`. 2. Start T-Server.

To disable TLS communication, remove one or more of the mandatory TLS options.

User-Data Display to IP Phones

A call with the following KV-data in user-data displays that data on all phones engaged in the phone call:

Key: IP_PHONE_DISPLAY

Type: KV-List

Value: KV-list with the following three fields:

- Key: TITLE
  Type: STRING
  Value: Any string, up to 24 characters. This is displayed at the top of the phone screen.
- Key: TEXT
  Type: STRING
  Value: Any string of fewer than 1024 characters. This is displayed in the center of the phone screen.
- Key: PROMPT
  Type: STRING
  Value: Any string, up to 28 characters. This is displayed at the bottom of the phone screen.

Note: If the data length is longer than the values specified above, T-Server silently "trims" the excess text. Note that the IP phone screen might not be large enough to hold all the text.

Phones on a call will no longer display user-data if the data is deleted.

For a conference call, all phones on the original call, and all phones that later join the call when the data is attached (for instance, those transferred to the call), displays KV-data. However, when a phone leaves the call, the data is no longer displayed on the departed phone.

If a user presses the Exit button (or any "soft" button) on the phone, KV data is no longer displayed.

The display of KV-data has the following limitations:

- If a remote T-Server updates the user-data (using the ISCC user-data propagation feature), the IP phones on the local T-Server do not display the updated data. However, if a local T-Server updates user-data, all the phones on the call displays the updated data.
- If a T-Server switches over to the backup T-Server, the backup T-Server displays user-data only for new calls, not existing calls.
- If the primary T-Server stops or is restarted, the phones on a call retains the KV data when the call terminates.

User data displayed on IP phones feature is applicable to shared lines DNs.

User-Data Display to IP Phones Not on a Call

To send a message to a phone that is not on call, the client must use the request `TPrivateService` with `PrivateServiceID = 3` and the following `KV-data` in `user-data`:

Key: `IP_PHONE_DISPLAY`

Type: `KV-List`

Value: `KV-list` with the following three fields:

- Key: `TITLE`
  Type: STRING
  Value: Any string, up to 24 characters. This is displayed at the top of the phone screen.
- Key: `TEXT`
  Type: STRING
  Value: Any string, up to 1024 characters. This is displayed in the center of the phone screen.
- Key: `PROMPT`
  Type: STRING
  Value: Any string, up to 28 characters. This is displayed at the bottom of the phone screen.

Note: If the data length is longer than the values specified above, T-Server truncates the excess text. Note that the IP phone screen might not be large enough to display all the text.

Voice Monitoring

T-Server supports third-party voice monitoring applications. For more details about voice monitoring, contact Genesys Customer Care.

Whisper Intercom Feature

This feature allows the monitored Agent to use CTI requests from T-Server to talk back to the supervisor without being overheard by the customer. A `TPrivateService` request enables the supervisor to talk over the intercom line while muting the connection to the customer. This request is only effective when the supervisor has already opened the intercom line for talking, that is, only after the supervisor has successfully requested `MuteOff`. If the request is made in any other state, an `Error` will be returned to the client. `ServiceID 5` is used to support this private service. T-Server sends the `join` command to Java when this request is received from the client. If the request is successful, the

customer is held and the call to the supervisor is established. T-Server then sends an `EventACK` event to the client. Speaking with the supervisor is terminated if the agent retrieves the customer call, or if the supervisor requests `MuteOn`.

T-Library Functionality

The tables in this section present the T-Library functionality that is supported in the Cisco Unified Communications Manager (CUCM) switch. The table entries use the following notations:

N—Not supported

Y—Supported

I—Supported, but reserved for Genesys Engineering

E—Event only is supported

In Table 14, when a set of events is sent in response to a single request, the events are listed in an arbitrary order. An asterisk (*) indicates the event that contains the same Reference ID as the request. For more information, refer to the *Genesys Events and Models Reference Manual* and *Platform SDK 8.x .NET (or Java) API Reference* for technical details of T-Library functions.

Table 14 reflects only the switch functionality that is used by Genesys software, and therefore it might not include the complete set of events that the switch offers.

Certain requests listed in Table 14 are reserved for Genesys Engineering and are listed here merely for completeness of information.

Notes describing specific functionality may appear at the end of a table.

Table 14: Supported T-Library Functionality

Feature Request	Request Subtype	Corresponding Event(s)	Supported
General Requests			
TOpenServer		EventServerConnected	Y
TOpenServerEx		EventServerConnected	Y
TCloseServer		EventServerDisconnected	Y
TSetInputMask		EventACK	Y
TDispatch		Not Applicable	Y
TScanServer		Not Applicable	Y
TScanServerEx		Not Applicable	Y

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
Registration Requests			
TRegisterAddress[a]		EventRegistered	Y
TUnregisterAddress[a]		EventUnregistered	Y
Call-Handling Requests			
TMakeCall[b]	Regular	EventDialing	Y
	DirectAgent		N
	SupervisorAssist		N
	Priority		N
	DirectPriority		N
TAnswerCall		EventEstablished	Y
TReleaseCall		EventReleased	Y
TClearCall		EventReleased	N
THoldCall		EventHeld	Y
TRetrieveCall		EventRetrieved	Y
TRedirectCall		EventReleased	Y
TMakePredictiveCall[c]		EventDialing* EventQueued	Y
Transfer/Conference Requests			
TInitiateTransfer[b]		EventHeld EventDialing*	Y
TCompleteTransfer		EventReleased* EventPartyChanged	Y
TInitiateConference[b]		EventHeld EventDialing*	Y
TCompleteConference		EventReleased* EventRetrieved EventPartyChanged EventPartyAdded	Y

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TDeleteFromConference		EventPartyDeleted* EventReleased	Y
TReconnectCall		EventReleased EventRetrieved*	Y
TAlternateCall		EventHeld* EventRetrieved	Y
TMergeCalls	ForTransfer	EventHeld EventReleased* EventRetrieved[d] EventPartyChanged	Y
	ForConference	EventHeld[d] EventReleased* EventRetrieved[d] EventPartyChanged EventPartyAdded	Y
TMuteTransfer[b]		EventHeld EventDialing* EventReleased EventPartyChanged	Y
TSingleStepTransfer[b]		EventReleased* EventPartyChanged	Y
TSingleStepConference[d]		EventRinging* EventEstablished	Y

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
Call-Routing Requests			
TRouteCall[b]	Unknown	EventRouteUsed	Y
	Default		Y
	Label		Y
	OverwriteDNIS		N
	DDD		N
	IDDD		N
	Direct		N
	Reject		Y
	Announcement		N
	PostFeature		N
	DirectAgent		N
	Priority		N
	DirectPriority		N
	AgentID		N
	CallDisconnect		N

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
Call-Treatment Requests			
TApplyTreatment	Unknown	(EventTreatmentApplied+ EventTreatmentEnd)/EventTreatmentNotApplied	N
	IVR		N
	Music[e]		Y
	RingBack[e]		Y
	Silence[e]		Y
	Busy[e]		Y
	CollectDigits		Y
	PlayAnnouncement[e]		Y
	PlayAnnouncementAnd-Digits		Y
	VerifyDigits		Y
	RecordUserAnnouncement		Y
	DeleteUserAnnouncement		N
	CancelCall		N
	PlayApplication		N
	SetDefaultRoute		N
	TextToSpeech		N
	TextToSpeechAndDigits		N
	FastBusy[e]		Y
	RAN		N
TGiveMusicTreatment		EventTreatmentApplied	N
TGiveRingBackTreatment		EventTreatmentApplied	N
TGiveSilenceTreatment		EventTreatmentApplied	N

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
DTMF (Dual-Tone MultiFrequency) Requests			
TCollectDigits		EventDigitsCollected	N
TSendDTMF		EventDTMFSent	Y
Voice-Mail Requests			
TOpenVoiceFile		EventVoiceFileOpened	N
TCloseVoiceFile		EventVoiceFileClosed	N
TLoginMailBox		EventMailBoxLogin	N
TLogoutMailBox		EventMailBoxLogout	N
TPlayVoice		EventVoiceFileEndPlay	N
Agent & DN Feature Requests			
TAgentLogin		EventAgentLogin	Y
TAgentLogout		EventAgentLogout	Y
TAgentSetReady		EventAgentReady	Y
TAgentSetNotReady		EventAgentNotReady	Y
TMonitorNextCall	OneCall	EventMonitoringNextCall	Y
	AllCalls		Y
TCancelMonitoring		EventMonitoringCanceled	Y
TCallSetForward	None	EventForwardSet	Y
	Unconditional		Y
	OnBusy		N
	OnNoAnswer		N
	OnBusyAndNoAnswer		N
	SendAllCalls		N
TCallCancelForward		EventForwardCancel	Y
TSetMuteOff		EventMuteOff	Y

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TSetMuteOn		EventMuteOn	Y
TListenDisconnect		EventListenDisconnected	N
TListenReconnect		EventListenReconnected	N
TSetDNDOn		EventDNDOn	Y
TSetDNDOff		EventDNDOff	Y
TSetMessageWaitingOn		EventMessageWaitingOn	N
TSetMessageWaitingOff		EventMessageWaitingOff	N
		EventOffHook	Y
		EventOnHook	Y
		EventDNBackInService	Y
		EventDNOutOfService	Y
Query Requests			
TQuerySwitch[a]	DateTime	EventSwitchInfo	N
	ClassifierStat		N
TQueryCall[a]	PartiesQuery	EventPartyInfo	Y
	StatusQuery		Y

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TQueryAddress[a]	AddressStatus	EventAddressInfo	Y
	MessageWaitingStatus		Y
	AssociationStatus		N
	CallForwardingStatus		Y
	AgentStatus		Y
	NumberOfAgentsInQueue		N
	NumberOfAvailableAgents-InQueue		N
	NumberOfCallsInQueue		N
	AddressType		Y
	CallsQuery		Y
	SendAllCallsStatus		N
	QueueLoginAudit		Y
	NumberOfIdleTrunks		N
	NumberOfTrunksInUse		N
	DatabaseValue		N
	DNStatus		Y
	QueueStatus		Y

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TQueryLocation[a]	AllLocations	EventLocationInfo	I
	LocationData		I
	MonitorLocation		I
	CancelMonitorLocation		I
	MonitorAllLocations		I
	CancelMonitorAll-Locations		I
	LocationMonitorCanceled		I
	AllLocationsMonitor-Canceled		I
TQueryServer[a]		EventServerInfo	Y
User-Data Requests			
TAttachUserData [Obsolete]		EventAttachedDataChanged	Y
TUpdateUserData		EventAttachedDataChanged	Y
TDeleteUserData		EventAttachedDataChanged	Y
TDeleteAllUserData		EventAttachedDataChanged	Y
ISCC (Inter Server Call Control) Requests			
TGetAccessNumber[b]		EventAnswerAccessNumber	I
TCancelRegGetAccess-Number		EventReqGetAccessNumber Canceled	I
Special Requests			
TReserveAgent		EventAgentReserved	Y
TSendEvent		EventACK	I
TSendEventEx		EventACK	I
TSetCallAttributes		EventCallInfoChanged	I
TSendUserEvent		EventACK	Y
TPrivateService		EventPrivateInfo	Y

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
Network Attended Transfer Requests[f]			
TNetworkConsult		EventNetworkCallStatus	Y
TNetworkAlternate		EventNetworkCallStatus	Y
TNetworkTransfer		EventNetworkCallStatus	Y
TNetworkMerge		EventNetworkCallStatus	Y
TNetworkReconnect		EventNetworkCallStatus	Y
TNetworkSingleStep-Transfer		EventNetworkCallStatus	Y
TNetworkPrivateService		EventNetworkPrivateInfo	Y
ISCC Transaction Monitoring Requests			
TTransactionMonitoring		EventACK	Y
		EventTransactionStatus	E

a. Only the requestor receives a notification of the event associated with this request.

b. This feature request can be made across locations in a multi-site environment. However, if the location attribute of the request contains a value relating to any location other than the local site—except when the response to this request is `EventError`—there will be a second event response that contains the same reference `ID` as the first event. This second event will be either `EventRemoteConnectionSuccess` or `EventRemoteConnectionFailed`.

c. T-Server for Cisco Unified Communications Manager does not use the `extensions` parameter. Any data in this parameter is ignored.

d. Refer to "Single-Step Conference" on page 168 for more details.

e. More detail about these treatments is provided in the section "Music and Announcements" on page 154.

f. All T-Servers support NAT/C requests with `AttributeHomeLocation` provided that this attribute identifies a network location that is capable of processing such requests. Refer to the *Network T-Server Deployment Guides* to determine whether a specific Network T-Server can process these requests.

T-Server Error Messages

The following table presents the complete set of error messages that T-Server distributes in `EventError`, which T-Server generates when it cannot execute a request because of an error condition.

Table 15: T-Server Error Messages

Code	Symbolic Name	Description
40	TERR_NOMORE_LICENSE	No more licenses are available.
41	TERR_NOT_REGISTERED	Client has not registered for the DN.
42	TERR_RESOURCE_SEIZED	Resource is already seized.
43	TERR_IN_SAME_STATE	Object is already in requested state.
50	TERR_UNKNOWN_ERROR	Unknown error code. Request cannot be processed.
51	TERR_UNSUP_OPER	Operation is not supported.
52	TERR_INTERNAL	Internal error.
53	TERR_INVALID_ATTR	Attribute in request operation is invalid.
54	TERR_NO_SWITCH	No connection to the switch.
55	TERR_PROTO_VERS	Incorrect protocol version.
56	TERR_INV_CONNID	`Connection ID` in request is invalid.
57	TERR_TIMEOUT	Switch or T-Server did not respond in time.
58	TERR_OUT_OF_SERVICE	Out of service.
59	TERR_NOT_CONFIGURED	DN is not configured in the Configuration Database.
61	TERR_INV_CALL_DN	`DN` in request is invalid.
96	TEER_CANT_COMPLETE_CONF	Call cannot add new conference party
119	TERR_BAD_PASSWD	`Password` was invalid
122	TERR_CANT_REG_DNS	Cannot register DNs on the switch
128	TERR_BAD_DN_TYPE	Invalid DN type for DN registration

Table 15: T-Server Error Messages (Continued)

Code	Symbolic Name	Description
166	TERR_RES_UNAVAIL	(JTAPI object) resource is not available
168	TERR_INV_ORIG_ADDR	Originating address in request was invalid
177	TERR_TARG_DN_INV	DN target (in route call) was invalid
195	TERR_CFW_DN_INV	Call forwarding address is invalid.
243	TERR_CLNT_NOT_MON	Internal error—client corrupted in T-Server
302	TERR_INV_DTMF_STRING	DTMF string invalid
410	TERR_INAPPR_TRTM	Invalid treatment type
415	TERR_INV_DEST_DN	The destination DN in the request is invalid
470	TERR_PARTY_NOT_ON_CALL	Party in request is not involved in a call
496	TERR_INV_CALL_STATE	Party in request is in the `call` state
506	TERR_RECVD_INV_STATE	Call/Party is in invalid state for this time
700	TERR_INV_LOGIN_REQ	Agent cannot log in at this time
701	TERR_INV_LOGOUT_REQUEST	Agent cannot logout
702	TERR_INV_READY_REQ	Agent cannot go to `ready` state
1605	TERR_INVALIDPARTY	Party in request was invalid on switch
1703	TERR_SOFT_AGENT_WRONG_ID	Wrong Agent ID
1704	TERR_SOFT_AGENT_ID_IN_USE	Agent ID already used
1705	TERR_SOFT_AGENT_PSWD_DOESNT_MATCH	Agent Password does not match
1706	TERR_SOFT_AGENT_ALREADY_LOGGED_IN	AGENT is already logged in
1707	TERR_SOFT_AGENT_NOT_LOGGED_IN	AGENT is not logged in
3002	TERR_PRIVVIOLATION	User doesn’t have security privilege on the switch
3005	TERR_UNSUCC_ROUTECALL	`Routecall` request was unsuccessful

Table 15: T-Server Error Messages (Continued)

Code	Symbolic Name	Description
Network Attended Transfer/Conference Error Messages		
1901	TERR_NATC_UNEXP_CONSULT	Unexpected request `TNetworkConsult`.
1902	TERR_NATC_UNEXP_ALTERNATE	Unexpected request `TNetworkAlternate`.
1903	TERR_NATC_UNEXP_RECONNECT	Unexpected request `TNetworkReconnect`.
1904	TERR_NATC_UNEXP_TRANSFER	Unexpected request `TNetworkTransfer`.
1905	TERR_NATC_UNEXP_MERGE	Unexpected request for `TNetworkMerge`.
1906	TERR_NATC_UNEXP_SST	Unexpected request `TNetworkSingleStepTransfer`.
1907	TERR_NATC_UNEXP_NPS	Unexpected request `TNetworkPrivateService`.
1908	TERR_NATC_UNEXP_MSG	Unexpected message.

Chapter

8 HA Configuration and Operation with CUCM JTAPI

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

HA Configuration of T-Server with 4 JTAPI Links

Here are two different types of configurations using four JTAPI links in an high-availability (HA) environment. The first one, Figure 20, uses four JTAPI links for the primary T-Server, and four different JTAPI links for the backup T-Server for a total of eight JTAPI links. This method avoids the duplication of JTAPI links by the primary and backup T-Server, but is a more costly HA implementation. The second type of configuration, Figure 21, uses four JTAPI links that are shared by both the primary, and the backup T-Server.

Note: In an HA environment, do not specify the backup CTI manager in the JTAPI link configuration. Example: CTI1, CTI2 for primary, and CTI2, CTI1 for backup. If the primary T-Server fails in this scenario,CTI2 will not be available to the backup T-Server after a switchover.

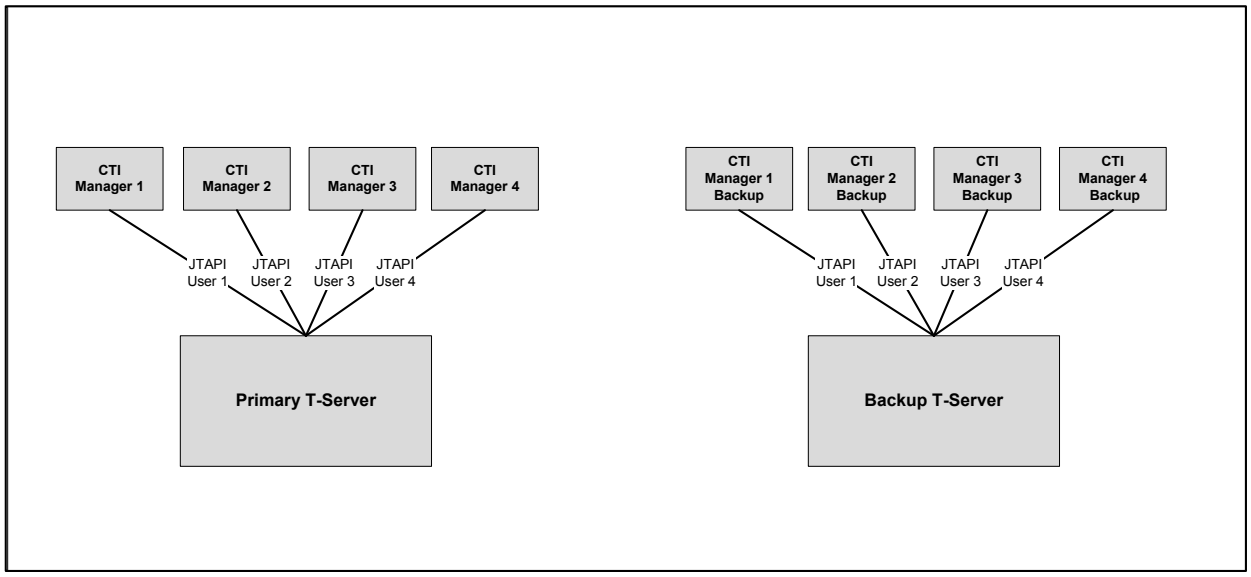

Figure 20: Four JTAPI Links for the Primary and Four JTAPI Links for the Backup T-Server

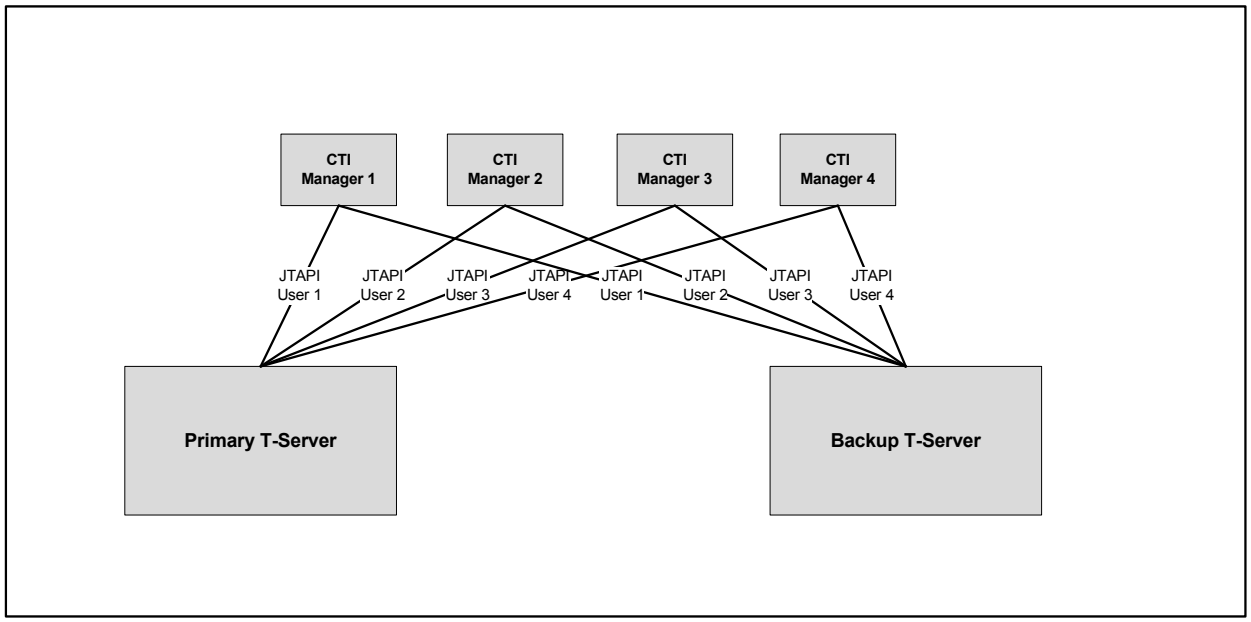

Figure 21: Four JTAPI Links Shared by the Primary and Backup T-Servers

How HA Works When Primary T-Server Fails

When a T-Server failure occurs, the JTAPI link processes get a socket exception (or terminate due to excessive keep alive retries), and the Genesys Solution Control Server switches to backup T-Server. In the configuration shown in Figure 22, all events for both the primary, and backup T-Server go to their own CTI Managers simultaneously. In this case, if a switchover occurs due to a failure with the primary T-Server, the backup T-Server will become the primary T-Server with no data lost.

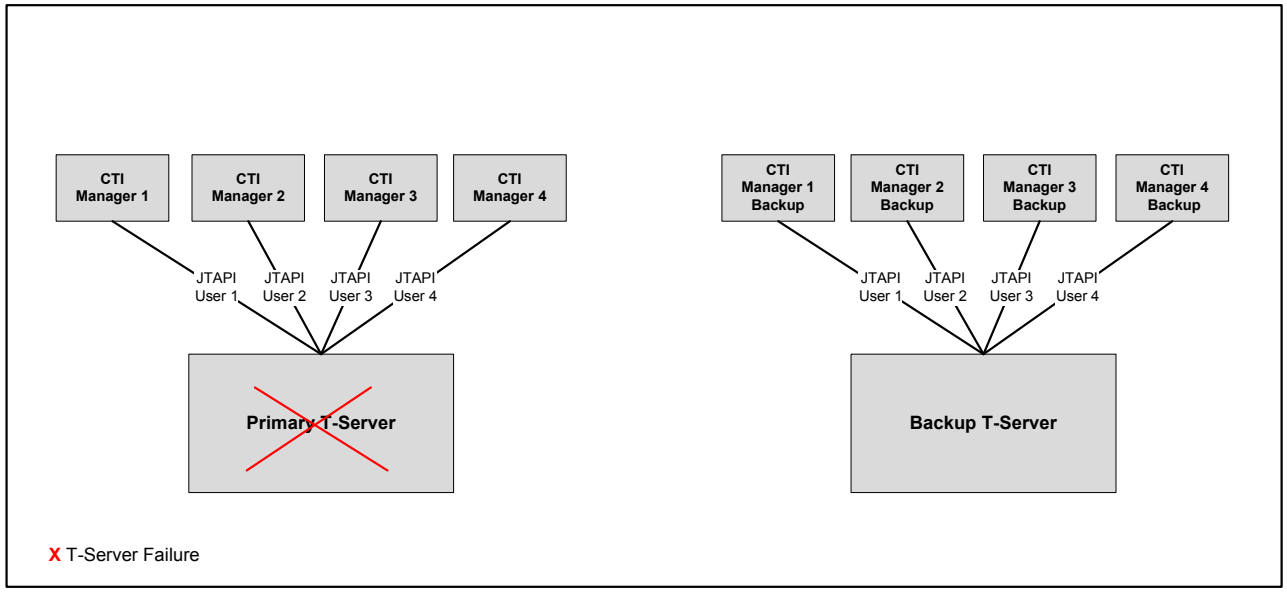

Figure 22: When the Primary T-Server Fails

How HA Works When One JTAPI Link Fails

In Figure 23, when a JTAPI failure (JAVA process) occurs, T-Server gets a socket exception, or a keep-alive timeout, and will generate a link-disconnected message. For single link failure, the primary T-Server will declare all DNs for the affected CTI Manager out of service, try to reconnect to the failed link, and upon a successful reconnect, place the affected DNs back in service. If there is a CTI manager failure, JTAPI will send a `ProviderFailed` event and cleanup the call events. T-Server ignores all cleanup events

(`EventReleased` with a `CTI_MANAGER_FAILURE`) and will send a link-disconnected event, and T-Server will switch to backup.

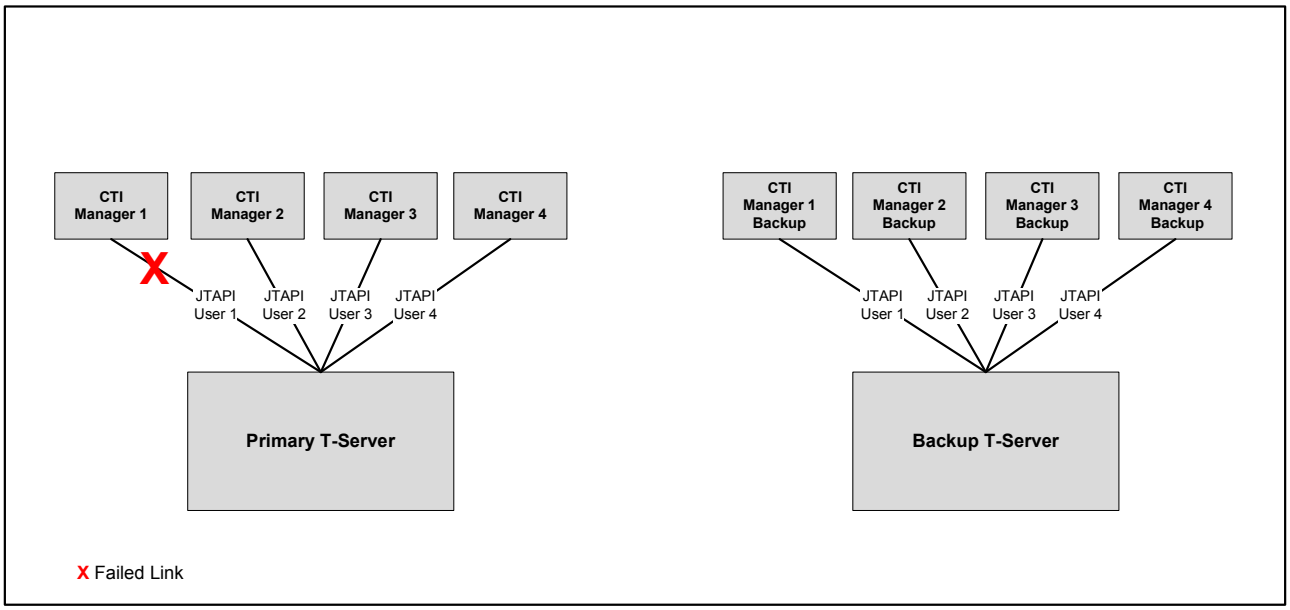

Figure 23: When a Link for the Primary T-Server Fails

How HA Works When All JTAPI Links Fail

The following is a description of how HA works when all JTAPI links fail (or T-Server loses all communication to CUCM).

With a JTAPI failure (JAVA process), T-Server gets a socket exception, or a keep-alive timeout and will generate an out-of-service message for all DNs registered on this link and the LMS message `JTAPILinkFailure` is generated. Then it restarts the link. SCS could request switchover by the fact of `JTAPILinkFailure` message (custom scripts may be used to control SCS switchover conditions).

CTI manager failure - JTAPI will send `ProviderFailed` event and cleanup call events. T-Server ignores all cleanup events (`EventReleased` with a `CTI_MANAGER_FAILURE`) and will send out-of-service messages for all DNs registered on this link and the LMS message `JTAPILinkFailure` is generated.

An event link disconnect is only triggered when all JTAPI links to an active T-Server have failed (for failure of all primary CTI Managers, for example). A link disconnect event will cause Genesys SCS to initiate a switchover to the backup T-Server.

Chapter

9 Integration with Genesys Media Server

This chapter describes the T-Server-CUCM to Media Server Connector that handles treatment requests from T-Server for CUCM and adapt to Session Initiation protocol (SIP) supported by Genesys Media Server.

This chapter contains the following section:

When integrated with T-Server for CUCM, Genesys Media Server provides Real-Time Protocol (RTP) streaming for treatments using the Media Server Markup Language (MSML). Cisco UCM directly sends SIP requests to Media Server through GVP Resource Manager to initiate call recording. Media Server persist call recording into storage and also allows the recording to be played back by other Media Server instances. Figure 24 demonstrates the overall deployment model using Media Server.

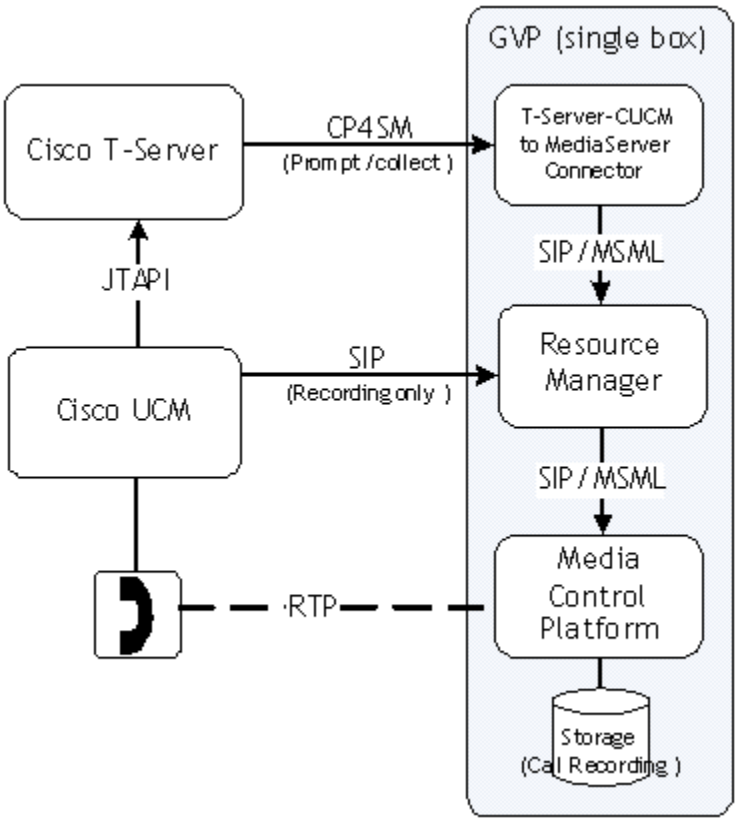

Figure 24: Genesys Media Server Deployment

For Genesys Media Server configuration with Connector, and deployment, reference the *Genesys Voice Platform 8.1 Deployment Guide.*

To configure Media Server to work with T-Server for Cisco Unified Communications Manager, `TServerAddress`, located in the `TServer` section on the `Options` tab of the `Connector Application` object in Configuration Manager, must be set. Reference the *Genesys Media Server 8.1 Deployment Guide* for more details on how to set this option.

For a small deployments, Connector, Resource Manager, and Media Control Platform can be deployed on the same machine. All three components Connector, Resource Manager, and Genesys Media Server are required for small deployments.

High Availability

Redundant instances of GVP components can be deployed to provide high availability.

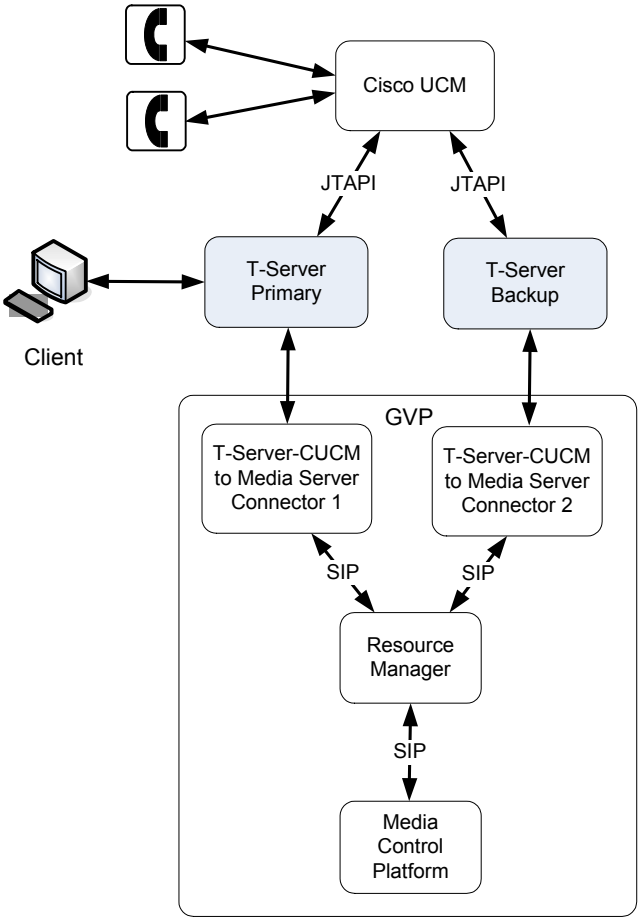

Figure 25: Genesys Media Server High Availability Deployment

To enable the connections between T-Server and Connector, configure the T-Server option `sm-port` in the T-Server application.

Note: A second Connector is needed to point to the backup T-Server.

For high availability (HA) configuration with Connector, refer to the *Genesys Voice Platform 8.1 Deployment Guide.*

Chapter

10 SIP Server In Front Deployment

This chapter describes the deployment configuration for the Genesys Voice Platform in conjunction with Cisco Unified Communications Manager (CUCM) that involves the Genesys SIP Server and Media Server deployed in front of the Cisco switch. This chapter contains the following section:

SIP Server In Front Configuration

This architecture provides a higher scalability and more flexibility compared with other standard configurations, and is indicated in cases of large deployments or special media requirements in the initial caller qualification phase.

Background

When a customer deploys the Cisco UCM as (IP-)PBX, the Genesys module that is used to integrate this switch with the Genesys environment is T-Server for CUCM. The simplest configuration is shown in Figure 26.

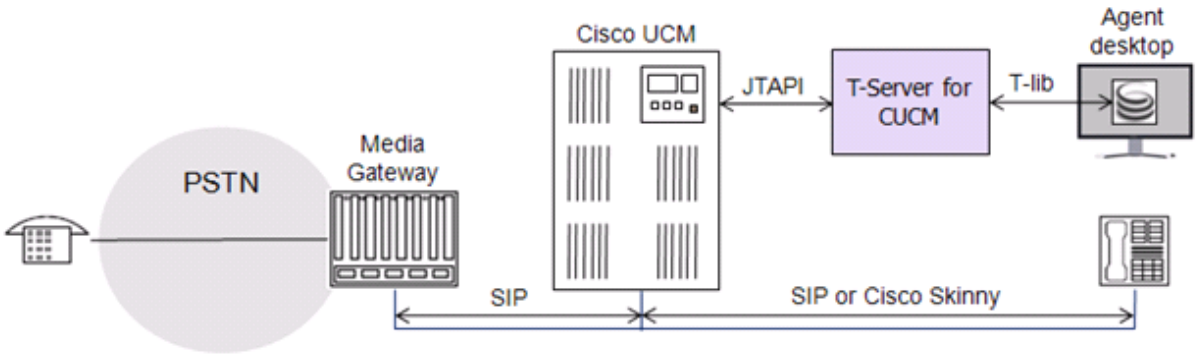

Figure 26: Simple Cisco UCM Integration Architecture

Note: No Genesys media service nodes are shown in this figure; typically either Genesys Media Server or Stream Manager would be used to provide media services, but this is not relevant for the purpose of this discussion.

Here, incoming calls hit the Cisco UCM, which notifies T-Server for CUCM over the CTI link (implemented using JTAPI – Java Telephony API). T-Server checks about agent's availability with the rest of the Genesys infrastructures (according to the configured strategies), then:

- If an agent is available, instructs CUCM to connect the call to the corresponding DN.
- If no agent is available, the call is queued internally; when an agent becomes available, T-Server will command CUCM to transfer the call to the corresponding DN.

This configuration is simple, but there are some drawbacks:

- With Cisco UCM Release 8.5, the maximum rating is 500 agents/JTAPI link due to call queuing constraints. Since there is a maximum of 4 JTAPI links per CUCM cluster, the maximum number of supported agents is 2000 per cluster.
- Treatment possibilities before the call is forwarded to an agent are limited to the functionality that JTAPI makes available to the T-Server.
- There is also the possibility of delivering treatment to calls in queue using a Stream Manager or Media Server installation, but in this case, the call must be connected to the SM DN before media can be applied. This is illegal in some situations and countries (especially for pay services before they are connected to an agent). It is possible for T-Server for CUCM to command Cisco UCM to play a file as ring back tone to callers and this satisfies the requirement; however in this case queuing is done on CUCM, with constraints on performance.

To address these limitations, Genesys designed and tested a different deployment configuration, which uses the power of the Genesys SIP Server and Media Server and the flexibility of Genesys routing to expand the number of agents supported by a Cisco UCM cluster and the flexibility of call treatment.

SIP Server in Front Configuration

In this configuration, an instance of SIP Server is the first landing point for calls coming into the system. The Media Gateway is configured to send all calls coming in from the PSTN to SIP Server.

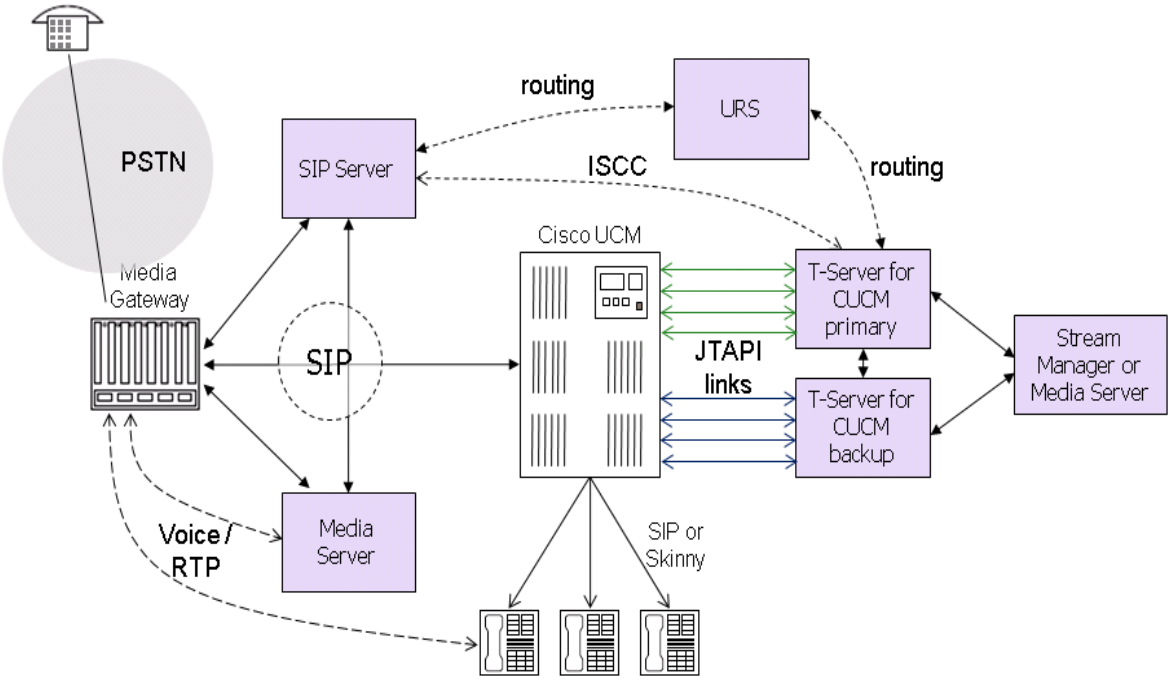

Figure 27: SIP Server in Front Integration Architecture

SIP Server queues an incoming call on the Routing Point, then the Universal Routing Server (URS) processes the call. The routing strategy selects the agent who will handle the call. If/when an agent is available, SIP Server uses the `REFER` method towards the Media Gateway to transfer the call to Cisco UCM, while also informing of the call the T-Server for CUCM over ISCC. After sending the `REFER` message, the SIP Server is out of the signaling path and the call is under the control of Cisco UCM.

Cisco UCM receives the call, and alerts T-Server, which routes the call to the available agent over the JTAPI link. This way, no calls are queued onto Cisco UCM, and the capacity of the system increases to 1000 agents per JTAPI link, or a maximum of 4000 agents with Cisco UCM releases up to 8.5.

Figure 27 above also shows the high-availability architecture of the JTAPI connection between T-Server for CUCM and the Cisco UCM cluster, with active links in green and hot-standby backup links in blue. Refer to Chapter 8, "HA Configuration and Operation with CUCM JTAPI," on page 193 for a description of the switchover process when an active JTAPI link goes down.

Note that with this configuration, the Media Gateway handles each call twice – once when it first comes in from the PSTN, and once when it is transferred to the Cisco UCM cluster through the SIP `REFER`. So, traffic on the Media Gateway doubles compared with the simple configuration of Figure 26 on page 202, and the gateway needs to be sized accordingly. The Media Gateway must also be of a type that supports call transfer via SIP `REFER`.

Configuration Details

For the SIP Server in front deployment, you must configure the Media Gateway and the SIP Server.

Media Gateway Dial Peers

The following example is for a Cisco AS5400XM Media Gateway, together with SIP Server and Cisco UCM 8.6.

```
dial-peer voice 5100 voip
 tone ringback alert-no-PI
 description CUCM86
 destination-pattern 5[1-3]..
 session target ipv4:135.225.59.86
 incoming called-number .
 voice-class codec 1
 dtmf-relay h245-signal h245-alphanumeric
 no vad
!
dial-peer voice 152 voip
 tone ringback alert-no-PI
 description NinaZSIPS
 destination-pattern 152.
 session protocol sipv2
 session target ipv4:135.225.54.221:31770
 incoming called-number .
 dtmf-relay rtp-nte
 codec g711ulaw
 no vad
!
voice service voip
 allow-connections h323 to h323
```

```
 allow-connections h323 to sip
 allow-connections sip to h323
 allow-connections sip to sip
 h323
  emptycapability
 sip
!
voice class codec 1
 codec preference 1 g711ulaw
 codec preference 3 g723r63
 codec preference 4 g729r8
 codec preference 5 g729br8
!
voice class h323 5
  call preserve
!
```

SIP Server Configuration

SIP Server must be configured with a few specific option values to enable the SIP Server in front configuration. The following is a list of these options on Application level and DN level. For details on SIP Server configuration, see the *SIP Server 8.1 Deployment Guide*.

SIP Server Application:

- `default-route-point`=<Route point number where calls will be queued>

Switch > DNs:

- DN of type `Trunk`:
  - `refer-enabled = true`
    When set to `true`, SIP Server will route calls to an external destination using the `REFER` method.
  - `oosp-transfer-enabled = true`
    When set to `true`, SIP Server puts itself in the Out Of Signaling Path (OOSP) after the single-step transfer or routing to the external destination has been completed.
  - `contact` = <...>
    This must point to the IP address of the Media Gateway.
- DN of type `Routing Point`

  The Routing Point where incoming calls will be parked.
- DN of type `Voice over IP Service`:
  - `service-type` = `treatment`
    This specifies the service type to be provided by Stream Manager or Media Server.
  - `contact` = <IP address of Stream Manager or Media Server : port>

Chapter

11 Common Configuration Options

Unless otherwise noted, the common configuration options that this chapter describes are common to all Genesys server applications and applicable to any Framework server component. This chapter includes the following sections:

Note: Some server applications also support log options that are unique to them. For descriptions of a particular application's unique log options, refer to the chapter/document about that application.

Setting Configuration Options

Unless specified otherwise, set common configuration options in the `Options` of the `Application` object, using one of the following navigation paths:

- In Genesys Administrator—`Application` object > `Options` tab > `Advanced View (Options)`
- In Configuration Manager—`Application` object > `Properties` dialog box > `Options` tab

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

Mandatory Options

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

log Section

This section must be called `log`.

verbose

Default Value: `all`

Valid Values:

`all`	All log events (that is, log events of the Standard, Trace, Interaction, and Debug levels) are generated.
`debug`	The same as `all`.
`trace`	Log events of the Trace level and higher (that is, log events of the Standard, Interaction, and Trace levels) are generated, but log events of the Debug level are not generated.
`interaction`	Log events of the Interaction level and higher (that is, log events of the Standard and Interaction levels) are generated, but log events of the Trace and Debug levels are not generated.
`standard`	Log events of the Standard level are generated, but log events of the Interaction, Trace, and Debug levels are not generated.
`none`	No output is produced.

Changes Take Effect: Immediately

Determines whether a log output is created. If it is, specifies the minimum level of log events generated. The log events levels, starting with the highest priority level, are Standard, Interaction, Trace, and Debug. See also "Log Output Options" on page 214.

Note: For definitions of the Standard, Interaction, Trace, and Debug log levels, refer to the *Framework 8.x Management Layer User's Guide*, *Framework 8.x Genesys Administrator Help,* or to *Framework 8.x Solution Control Interface Help.*

buffering

Default Value: `true`
Valid Values:

`true`	Enables buffering.
`false`	Disables buffering.

Changes Take Effect: Immediately

Turns on/off the operating system file buffering. The option is applicable only to the `stderr` and `stdout` output (see page 214). Setting the value of this option to `true` increases the output performance.

Note: When buffering is enabled, there might be a delay before log messages appear at the console.

segment

Default Value: `false`
Valid Values:

`false`	No segmentation is allowed.
`<number> KB` or `<number>`	Sets the maximum segment size, in kilobytes. The minimum segment size is `100 KB`.
`<number> MB`	Sets the maximum segment size, in megabytes.
`<number> hr`	Sets the number of hours for the segment to stay open. The minimum number is `1` hour.

Changes Take Effect: Immediately

Specifies whether there is a segmentation limit for a log file. If there is, sets the mode of measurement, along with the maximum size. If the current log segment exceeds the size set by this option, the file is closed and a new one is created. This option is ignored if log output is not configured to be sent to a log file.

expire

Default Value: `false`
Valid Values:

`false`	No expiration; all generated segments are stored.
`<number> file` or `<number>`	Sets the maximum number of log files to store. Specify a number from `1`–`1000`.
`<number> day`	Sets the maximum number of days before log files are deleted. Specify a number from `1`–`100`.

Changes Take Effect: Immediately

Determines whether log files expire. If they do, sets the measurement for determining when they expire, along with the maximum number of files (segments) or days before the files are removed. This option is ignored if log output is not configured to be sent to a log file.

Note: If an option's value is set incorrectly—out of the range of valid values— it is automatically reset to `10`.

keep-startup-file

Default Value: `false`
Valid Values:

`false`	No startup segment of the log is kept.
`true`	A startup segment of the log is kept. The size of the segment equals the value of the `segment` option.
`<number> KB`	Sets the maximum size, in kilobytes, for a startup segment of the log.
`<number> MB`	Sets the maximum size, in megabytes, for a startup segment of the log.

Changes Take Effect: After restart

Specifies whether a startup segment of the log, containing the initial T-Server configuration, is to be kept. If it is, this option can be set to `true` or to a specific size. If set to `true`, the size of the initial segment is equal to the size of the regular log segment defined by the `segment` option. The value of this option is ignored if segmentation is turned off (that is, if the `segment` option set to `false`).

Note: This option applies only to T-Servers.

messagefile

Default Value: As specified by a particular application
Valid Values: `<string>.lms` (message file name)
Changes Take Effect: Immediately, if an application cannot find its `*.lms` file at startup

Specifies the file name for application-specific log events. The name must be valid for the operating system on which the application is running. The option value can also contain the absolute path to the application-specific `*.lms` file. Otherwise, an application looks for the file in its working directory.

Warning! An application that does not find its `*.lms` file at startup cannot generate application-specific log events and send them to Message Server.

message_format

Default Value: `short`
Valid Values:

`short`	An application uses compressed headers when writing log records in its log file.
`full`	An application uses complete headers when writing log records in its log file.

Changes Take Effect: Immediately

Specifies the format of log record headers that an application uses when writing logs in the log file. Using compressed log record headers improves the application performance and reduces the log file's size.

With the value set to `short`:

- A header of the log file or the log file segment contains the information about the application (such as the application name, application type, host type, and time zone), whereas single log records within the file or segment omit this information.
- A log message priority is abbreviated to `Std`, `Int`, `Trc`, or `Dbg`, for Standard, Interaction, Trace, or Debug messages, respectively.
- The message ID does not contain the prefix `GCTI` or the application type `ID`.

A log record in the full format looks like this:

```
2002-05-07T18:11:38.196 Standard localhost cfg_dbserver GCTI-00-05060
Application started
```

A log record in the short format looks like this:

```
2002-05-07T18:15:33.952 Std 05060 Application started
```

Note: Whether the full or short format is used, time is printed in the format specified by the `time_format` option.

time_convert

Default Value: `Local`
Valid Values:

`local`	The time of log record generation is expressed as a local time, based on the time zone and any seasonal adjustments. The time zone information of the application's host computer is used.
`utc`	The time of log record generation is expressed as Coordinated Universal Time (UTC).

Changes Take Effect: Immediately

Specifies the system in which an application calculates the log record time when generating a log file. The time is converted from the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970).

time_format

Default Value: `time`
Valid Values:

`time`	The time string is formatted according to the `HH:MM:SS.sss` (hours, minutes, seconds, and milliseconds) format.
`locale`	The time string is formatted according to the system's locale.
`ISO8601`	The date in the time string is formatted according to the ISO 8601 format. Fractional seconds are given in milliseconds.

Changes Take Effect: Immediately

Specifies how to represent, in a log file, the time when an application generates log records.

A log record's time field in the ISO 8601 format looks like this:

```
2001-07-24T04:58:10.123
```

print-attributes

Default Value: `false`
Valid Values:

`true`	Attaches extended attributes, if any exist, to a log event sent to log output.
`false`	Does not attach extended attributes to a log event sent to log output.

Changes Take Effect: Immediately

Specifies whether the application attaches extended attributes, if any exist, to a log event that it sends to log output. Typically, log events of the Interaction log level and Audit-related log events contain extended attributes. Setting this option to `true` enables the audit capabilities, but negatively affects performance. Genesys recommends enabling this option for Solution Control Server and Configuration Server when using audit tracking. For other applications, refer to *Genesys 8.x Combined Log Events Help* to find out whether an application generates Interaction-level and Audit-related log events; if it does, enable the option only when testing new interaction scenarios.

check-point

Default Value: `1`
Valid Values: `0`–`24`
Changes Take Effect: Immediately

Specifies, in hours, how often the application generates a `check point` log event, to divide the log into sections of equal time. By default, the application generates this log event every hour. Setting the option to `0` (zero) prevents the generation of `check-point` events.

memory

Default Value: No default value
Valid Values: `<string>` (memory file name)
Changes Take Effect: Immediately

Specifies the name of the file to which the application regularly prints a snapshot of the memory output, if it is configured to do this (see "Log Output Options" on page 214). The new snapshot overwrites the previously written data. If the application terminates abnormally, this file contains the latest log messages. Memory output is not recommended for processors with a CPU frequency lower than 600 MHz.

Note: If the file specified as the `memory` file is located on a network drive, an application does not create a snapshot file (with the extension `*.memory.log`).

---

memory-storage-size

Default Value: `2 MB`
Valid Values:

`<number> KB` or `<number>`	The size of the memory output, in kilobytes. The minimum value is `128 KB`.
`<number> MB`	The size of the memory output, in megabytes. The maximum value is `64 MB`.

Changes Take Effect: When memory output is created

Specifies the buffer size for log output to the memory, if configured. See also "Log Output Options" on page 214.

spool

Default Value: The application's working directory
Valid Values: `<path>` (the folder, with the full path to it)
Changes Take Effect: Immediately

Specifies the folder, including full path to it, in which an application creates temporary files related to the network log output. If you change this option value while the application is running, the change does not affect the currently open network output.

compatible-output-priority

Default Value: `false`
Valid Values:

`true`	The log of the level specified by "Log Output Options" is sent to the specified output.
`false`	The log of the level specified by "Log Output Options" and higher levels is sent to the specified output.

Changes Take Effect: Immediately

Specifies whether the application uses 6.x output logic. For example, you configure the following options in the `log` section for a 6.x application and for a 7.x application:

```
[log]
verbose = all
debug = file1
standard = file2
```

The log file content of a 6.x application is as follows:

- `file1` contains Debug messages only.
- `file2` contains Standard messages only.

The log file content of a 7.x application is as follows:

- `file1` contains Debug, Trace, Interaction, and Standard messages.
- `file2` contains Standard messages only.

If you set `compatible-output-priority` to `true` in the 7.x application, its log file content will be the same as for the 6.x application.

Warning! Genesys does not recommend changing the default value of this option unless you have specific reasons to use the 6.x log output logic—that is, to mimic the output priority as implemented in releases 6.x. Setting this option to `true` affects log consistency.

Log Output Options

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

You can use:

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

You must separate the log output types by a comma when you are configuring more than one output for the same log level. See “Examples” on page 218.

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

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

all

Default Value: No default value
Valid Values (log output types):

`stdout`	Log events are sent to the Standard output (`stdout`).
`stderr`	Log events are sent to the Standard error output (`stderr`).
`network`	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. Setting the `all` log level option to the `network` output enables an application to send log events of the `Standard`, `Interaction`, and `Trace` levels to Message Server. `Debug`-level log events are neither sent to Message Server nor stored in the Log Database.
`memory`	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends all log events. The log output types must be separated by a comma when more than one output is configured. For example:

```
all = stdout, logfile
```

Note: To ease the troubleshooting process, consider using unique names for log files that different applications generate.

alarm

Default Value: No default value
Valid Values (log output types):

`stdout`	Log events are sent to the Standard output (`stdout`).
`stderr`	Log events are sent to the Standard error output (`stderr`).
`network`	Log events are sent to Message Server, which resides anywhere on the network, and Message Server stores the log events in the Log Database.
`memory`	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the Alarm level. The log output types must be separated by a comma when more than one output is configured. For example:

```
standard = stderr, network
```

standard

Default Value: No default value
Valid Values (log output types):

`stdout`	Log events are sent to the Standard output (`stdout`).
`stderr`	Log events are sent to the Standard error output (`stderr`).
`network`	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
`memory`	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Standard` level. The log output types must be separated by a comma when more than one output is configured. For example:

```
standard = stderr, network
```

interaction

Default Value: No default value
Valid Values (log output types):

`stdout`	Log events are sent to the Standard output (`stdout`).
`stderr`	Log events are sent to the Standard error output (`stderr`).
`network`	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
`memory`	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Interaction` level and higher (that is, log events of the `Standard` and `Interaction` levels). The log outputs must be separated by a comma when more than one output is configured. For example:

```
interaction = stderr, network
```

trace

Default Value: No default value
Valid Values (log output types):

`stdout`	Log events are sent to the Standard output (`stdout`).
`stderr`	Log events are sent to the Standard error output (`stderr`).
`network`	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
`memory`	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Trace` level and higher (that is, log events of the `Standard`, `Interaction`, and `Trace` levels). The log outputs must be separated by a comma when more than one output is configured. For example:

```
trace = stderr, network
```

debug

Default Value: No default value
Valid Values (log output types):

`stdout`	Log events are sent to the Standard output (`stdout`).
`stderr`	Log events are sent to the Standard error output (`stderr`).
`memory`	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Debug` level and higher (that is, log events of the `Standard`, `Interaction`, `Trace`, and `Debug` levels). The log output types must be separated by a comma when more than one output is configured—for example:

```
debug = stderr, /usr/local/genesys/logfile
```

Note: `Debug`-level log events are never sent to Message Server or stored in the Log Database.

Log File Extensions

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

- `*.log`—Assigned to log files when you configure output to a log file. For example, if you set `standard = confservlog` for Configuration Server, it prints log messages into a text file called `confservlog.<time_stamp>.log`.
- `*.qsp`—Assigned to temporary (spool) files when you configure output to the network but the network is temporarily unavailable. For example, if you set `standard = network` for Configuration Server, it prints log messages into a file called `confserv.<time_stamp>.qsp` during the time the network is not available.
- `*.snapshot.log`—Assigned to files that contain the output snapshot when you configure output to a log file. The file contains the last log messages that an application generates before it terminates abnormally. For example, if you set `standard = confservlog` for Configuration Server, it prints the last log message into a file called `confserv.<time_stamp>.snapshot.log` in case of failure.

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

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

Examples

This section presents examples of a `log` section that you might configure for an application when that application is operating in production mode and in two lab modes, debugging and troubleshooting.

Production Mode Log Section

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

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

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

Lab Mode Log Section

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

With this configuration, an application generates log events of the `Standard`, `Interaction`, `Trace`, and `Debug` levels, and sends them to the standard output and to a file named `logfile`, which the application creates in the `/usr/local/genesys/` directory. In addition, the application sends log events of the `Standard`, `Interaction`, and `Trace` levels to Message Server. Use this configuration to test new interaction scenarios in a lab environment.

Failure-Troubleshooting Log Section

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

With this configuration, an application generates log events of the `Standard` level and sends them to Message Server. It also generates log events of the `Standard`, `Interaction`, `Trace`, and `Debug` levels, and sends them to the memory output. The most current log is stored to a file named `logfile`, which the application creates in its working directory. Increased memory storage allows an application to save more of the log information generated before a failure.

Note: If you are running an application on UNIX, and you do not specify any files in which to store the memory output snapshot, a core file that the application produces before terminating contains the most current application log. Provide the application's core file to Genesys Customer Care when reporting a problem.

Debug Log Options

The options in this section enable you to generate `Debug` logs containing information about specific operations of an application.

x-conn-debug-open

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about "open connection" operations of the application.

Warning! Use this option only when requested by Genesys Customer Care.

x-conn-debug-select

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about "socket select" operations of the application.

Warning! Use this option only when requested by Genesys Customer Care.

x-conn-debug-timers

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about the timer creation and deletion operations of the application.

Warning! Use this option only when requested by Genesys Customer Care.

x-conn-debug-write

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about "write" operations of the application.

Warning! Use this option only when requested by Genesys Customer Care.

x-conn-debug-security

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about security-related operations, such as Transport Layer Security and security certificates.

Warning! Use this option only when requested by Genesys Customer Care.

x-conn-debug-api

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about connection library function calls.

Warning! Use this option only when requested by Genesys Customer Care.

x-conn-debug-dns

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about DNS operations.

Warning! Use this option only when requested by Genesys Customer Care.

x-conn-debug-all

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about open connection, socket select, timer creation and deletion, write, security-related, and DNS operations, and connection library function calls. This option is the same as enabling or disabling all of the previous `x-conn-debug-<op type>` options.

Warning! Use this option only when requested by Genesys Customer Care.

log-extended Section

This section must be called `log-extended`.

level-reassign-<eventID>

Default Value: Default value of log event `<eventID>`
Valid Values:

`alarm`	The log level of log event `<eventID>` is set to `Alarm`.
`standard`	The log level of log event `<eventID>` is set to `Standard`.
`interaction`	The log level of log event `<eventID>` is set to `Interaction`.
`trace`	The log level of log event `<eventID>` is set to `Trace`.
`debug`	The log level of log event `<eventID>` is set to `Debug`.
`none`	Log event `<eventID>` is not recorded in a log.

Changes Take Effect: Immediately

Specifies a log level for log event `<eventID>` that is different than its default level, or disables log event `<eventID>` completely. If no value is specified, the log event retains its default level. This option is useful when you want to customize the log level for selected log events.

These options can be deactivated with the option `level-reassign-disable`.

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

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

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

In addition to the preceding warning, take note of the following:

- Logs can be customized only by release 7.6 or later applications.

- When the log level of a log event is changed to any level except `none`, it is subject to the other settings in the `[log]` section at its new level. If set to `none`, it is not logged and is therefore not subject to any log configuration.
- Using this feature to change the log level of a log changes only its priority; it does not change how that log is treated by the system. For example, increasing the priority of a log to `Alarm` level does not mean that an alarm will be associated with it.
- Each application in a High Availability (HA) pair can define its own unique set of log customizations, but the two sets are not synchronized with each other. This can result in different log behavior depending on which application is currently in primary mode.
- This feature is not the same as a similar feature in Universal Routing Server (URS) release 7.2 or later. In this Framework feature, the priority of log events are customized. In the URS feature, the priority of debug messages only are customized. Refer to the *Universal Routing Reference Manual* for more information about the URS feature.
- You cannot customize any log event that is not in the unified log record format. Log events of the `Alarm`, `Standard`, `Interaction`, and `Trace` levels feature the same unified log record format.

Example

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

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

Before the log levels of the log are changed:

- Log event 1020, with default level `standard`, is output to `stderr` and `log_file`, and sent to Message Server.
- Log event 2020, with default level `standard`, is output to `stderr` and `log_file`, and sent to Message Server.
- Log event 3020, with default level `trace`, is output to `stderr`.
- Log event 4020, with default level `debug`, is output to `stderr`.

Extended log configuration section:

```
[log-extended]
level-reassign-1020=none
level-reassign-2020=interaction
level-reassign-3020=interaction
level-reassign-4020=standard
```

After the log levels are changed:

- Log event 1020 is disabled and not logged.

- Log event 2020 is output to `stderr` and `log_file`.
- Log event 3020 is output to `stderr` and `log_file`.
- Log event 4020 is output to `stderr` and `log_file`, and sent to Message Server.

level-reassign-disable

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When this option is set to `true`, the original (default) log level of all log events in the `[log-extended]` section are restored. This option is useful when you want to use the default levels, but not delete the customization statements.

log-filter Section

The `log-filter` section contains configuration options used to define the default treatment of filtering data in log output. This section contains one configuration option, `default-filter-type`. Refer to the chapter "Hide Selected Data in Logs" in the *Genesys 8.x Security Deployment Guide* for complete information about this option.

log-filter-data Section

The `log-filter-data` section contains configuration options used to define the treatment of filtering data in log output on a key-by-key basis. This section contains one configuration option in the form of `<key name>`. Refer to the chapter "Hide Selected Data in Logs" in the *Genesys 8.x Security Deployment Guide* for complete information about this option.

security Section

The `security` section contains configuration options used to specify security elements for your system. In addition to other options that may be required by your application, this section contains the configuration option `disable-rbac`, which is used to enable or disable Role-Based Access Control for an application. Refer to the chapter "Role-Based Access Control" in the *Genesys 8.x Security Deployment Guide* for complete information about this option.

sml Section

This section must be called `sml`.

Options in this section are defined in the `Annex` of the `Application` object, as follows:

- in Genesys Administrator—`Application` object > `Options` tab > `Advanced View (Annex)`
- in Configuration Manager— `Application` object > `Properties` dialog box > `Annex` tab

Warning! Use the first three options in this section (`heartbeat-period`, `heartbeat-period-thread-class-<n>`, and `hangup-restart`) with great care, and only with those applications of which support for this functionality has been announced. Failure to use these options properly could result in unexpected behavior, from ignoring the options to an unexpected restart of the application.

heartbeat-period

Default Value: None
Valid Values:

`0`	This method of detecting an unresponsive application is not used by this application.
`3-604800`	Length of timeout, in seconds; equivalent to 3 seconds–7 days.

Changes Take Effect: Immediately

Specifies the maximum amount of time, in seconds, in which heartbeat messages are expected from an application. If Local Control Agent (LCA) does not receive a heartbeat message from the application within this period, it assumes the application is not responding and carries out corrective action.

This option can also be used to specify the maximum heartbeat interval for threads registered with class zero (`0`). This thread class is reserved for use by the Management Layer only.

If this option is not configured, or it is set to zero (`0`), heartbeat detection is not used by this application.

heartbeat-period-thread-class-<n>

Default Value: None
Valid Values:

`0`	Value specified by `heartbeat-period` in application is used.
`3-604800`	Length of timeout, in seconds; equivalent to 3 seconds–7 days.

Changes Take Effect: Immediately

Specifies the maximum amount of time, in seconds, in which heartbeat messages are expected from a thread of class <n> registered by an application.

If a heartbeat message from the thread is not received within this period, the thread is assumed to be not responding, and therefore, the application is unable to provide service.

If this option is not configured or is set to zero (`0`), but the application has registered one or more threads of class <n>, the value specified by the value of `heartbeat-period` for the application will also be applied to these threads.

Refer to application-specific documentation to determine what thread classes, if any, are used.

hangup-restart

Default Value: `true`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

If set to `true` (the default), specifies that LCA is to restart the unresponsive application immediately, without any further interaction from Solution Control Server.

If set to `false`, specifies that LCA is only to generate a notification that the application has stopped responding.

suspending-wait-timeout

Default Value: `10`
Valid Values: `5-600`
Changes Take Effect: Immediately

Specifies a timeout (in seconds) after the Stop Graceful command is issued to an application during which the status of the application should change to `Suspending` if the application supports graceful shutdown. If the status of the application does not change to `Suspending` before the timeout expires, it is assumed that the application does not support graceful shutdown, and it is stopped ungracefully.

Use this option if you are unsure whether the Application supports graceful shutdown.

Note: Genesys recommends that you do not set this option for any Management Layer component (Configuration Server, Message Server, Solution Control Server, or SNMP Master Agent) or any DB Server. These components by definition do not support graceful shutdown, so this option is not required.

common Section

This section must be called `common`.

enable-async-dns

Default Value: `off`
Valid Values:

`off`	Disables asynchronous processing of DNS requests.
`on`	Enables asynchronous processing of DNS requests.

Changes Take Effect: Immediately

Enables the asynchronous processing of DNS requests such as, for example, host-name resolution.

Warnings!
- Use this option only when requested by Genesys Customer Care.
- Use this option only with T-Servers.

rebind-delay

Default Value: `10`
Valid Values: `0`–`600`
Changes Take Effect: After restart

Specifies the delay, in seconds, between socket-bind operations that are being executed by the server. Use this option if the server has not been able to successfully occupy a configured port.

Warning! Use this option only when requested by Genesys Customer Care.

Changes from 8.0 to 8.1

There are no changes in common configuration options between 8.0 and 8.1 releases.

Chapter

12 T-Server Common Configuration Options

This chapter describes the configuration options that are generally common to all T-Server types, with some exceptions noted. It contains the following sections:

T-Server also supports common log options described in Chapter 11, "Common Configuration Options," on page 207.

Setting Configuration Options

Unless specified otherwise, set T-Server common configuration options in the `Options` of the `Application` object, using one of the following navigation paths:

- In Genesys Administrator—`Application` object > `Options` tab > `Advanced View (Options)`
- In Configuration Manager—`Application` object > `Properties` dialog box > `Options` tab

Mandatory Options

Except as noted for certain environments, the configuration of common options is not required for basic T-Server operation.

TServer Section

The `TServer` section contains the configuration options that are used to support the core features common to all T-Servers.

This section must be called `TServer`.

ani-distribution

Default Value: `inbound-calls-only`
Valid Values: `inbound-calls-only, all-calls, suppressed`
Changes Take Effect: Immediately

Controls the distribution of the ANI information in `TEvent` messages.

- When this option is set to `all-calls`, the `ANI` attribute is reported for all calls for which it is available.
- When this option is set to `suppressed`, the `ANI` attribute is not reported for any calls.
- When this option is set to `inbound-calls-only`, the `ANI` attribute is reported for inbound calls only.

background-processing

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

When set to `true`, T-Server processes all client requests in the background, giving higher priority to the rest of the messages. This ensures that it processes these messages without any significant delay.

With Background Processing functionality enabled, T-Server processes all switch messages immediately and waits until there are no switch messages before processing the message queue associated with T-Server client requests. T-Server reads all connection sockets immediately and places client requests in the input buffer, which prevents T-Server clients from disconnecting because of configured timeouts.

When T-Server processes client requests from the message queue, requests are processed in the order in which T-Server received them.

When set to `false`, T-Server processes multiple requests from one T-Server client before proceeding to the requests from another T-Server client, and so on.

background-timeout

Default Value: `60 msec`
Valid Values: See "Timeout Value Format" on page 254.
Changes Take Effect: Immediately

Specifies the time interval that T-Server waits before processing client requests in background mode. You must set the `background-processing` option to `true` in order for this option to take effect.

check-tenant-profile

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: For the next connected client

When set to `true`, T-Server only allows a client to register if the client provides the correct name and password of a T-Server Tenant. If the client provides the Tenant name concatenated with a slash (/) and the Tenant password for the Tenant to which T-Server belongs as the value of `AttributeApplicationPassword` in the `TRegisterClient` request, T-Server allows that client to register DNs that are included in the switch configuration in the Configuration Database, but it does not allow the client to register DNs that are *not* included in the switch configuration.

consult-user-data

Default Value: `separate`
Valid Values:

`separate`	Stores user data for original and consultation calls in separate structures. The data attached to the original call is available for review or changes only to the parties of that call. The data attached to the consultation call is available only to the parties of the consultation call.
`inherited`	Copies user data from an original call to a consultation call when the consultation call is created; thereafter, stores user data separately for the original and the consultation call. Changes to the original call's user data are not available to the parties of the consultation call, and vice versa.
`joint`	Stores user data for an original call and a consultation call in one structure. The user data structure is associated with the original call, but the parties of both the original and consultation calls can see and make changes to the common user data.

Changes Take Effect: For the next consultation call created

Specifies the method for handling user data in a consultation call.

Note: A T-Server client can also specify the `consult-user-data` mode in the `Extensions` attribute, `ConsultUserData` key, for a conference or a transfer request. If it is specified, the method of handling user data is based on the value of the `ConsultUserData` key-value pair of the request and takes precedence over the T-Server `consult-user-data` option. If it is not specified in the client request, the value specified in the `consult-user-data` option applies.

customer-id

Default Value: No default value. (A value must be specified for a multi-tenant environment.)
Valid Values: Any character string
Changes Take Effect: Immediately

Identifies the T-Server customer. You must set this option to the name of the tenant that is using this T-Server. You must specify a value for this option if you are working in a multi-tenant environment.

Note: Do not configure the `customer-id` option for single-tenant environments.

dn-scope

Default Value: `undefined`
Valid Values: `undefined, switch, office, tenant`
Changes Take Effect: Immediately
Related Feature: “Switch Partitioning” on page 98

Specifies whether DNs associated with the `Switch, Switching Office,` or `Tenant` objects is considered in the T-Server monitoring scope, enabling T-Server to report calls to or from those DNs as internal.

With a value of `tenant,` all DNs associated with the switches that are within the `Tenant` is in the T-Server monitoring scope. With a value of `office,` all DNs associated with the switches that are within the `Switching Office` are in the T-Server monitoring scope. With a value of `switch,` all DNs associated with the `Switch` are in the T-Server monitoring scope.

With a value of `undefined` (the default), pre-8.x T-Server behavior applies and the switch partitioning is not turned on.

Note: Setting the option to a value of `office` or `tenant,` which requires T-Server to monitor a large set of configuration data, may negatively affect T-Server performance.

log-trace-flags

Default Value: `+iscc, +cfg$dn, -cfgserv, +passwd, +udata, -devlink, -sw, -req, -callops, -conn, -client`

Valid Values (in any combination):

`+/-iscc`	Turns on/off the writing of information about Inter Server Call Control (ISCC) transactions.
`+/-cfg$dn`	Turns on/off the writing of information about DN configuration.
`+/-cfgserv`	Turns on/off the writing of messages from Configuration Server.
`+/-passwd`	Turns on/off the writing of `AttributePassword` in TEvents.
`+/-udata`	Turns on/off the writing of attached data.
`+/-devlink`	Turns on/off the writing of information about the link used to send CTI messages to the switch (for multilink environments).
`+/-sw`	Reserved by Genesys Engineering.
`+/-req`	Reserved by Genesys Engineering.
`+/-callops`	Reserved by Genesys Engineering.
`+/-conn`	Reserved by Genesys Engineering.
`+/-client`	Turns on/off the writing of additional information about the client's connection.

Changes Take Effect: Immediately

Specifies—using a space-, comma- or semicolon-separated list—the types of information that are written to the log files.

management-port

Default Value: `0`

Valid Values: `0` or any valid TCP/IP port

Changes Take Effect: After T-Server is restarted

Specifies the TCP/IP port that management agents use to communicate with T-Server. If set to `0` (zero), this port is not used.

merged-user-data

Default Value: `main-only`

Valid Values:

`main-only`	T-Server attaches user data from the remaining call only.
`merged-only`	T-Server attaches user data from the merging call.
`merged-over-main`	T-Server attaches user data from the remaining and the merging call. In the event of equal keys, T-Server uses data from the merging call.
`main-over-merged`	T-Server attaches data from the remaining and the merging call. In the event of equal keys, T-Server uses data from the remaining call.

Changes Take Effect: Immediately

Specifies the data that is attached to the resulting call after a call transfer, conference, or merge completion.

Note: The option setting does not affect the resulting data for merging calls if the `consult-user-data` option is set to `joint`. (See "consult-user-data" on page 231.)

propagated-call-type

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Switch Partitioning" on page 98

Determines what T-Server reports as the value of the `CallType` attribute in events related to calls that have been synchronized with another site via ISCC, as follows:

- When set to `false`, T-Server reports in events related to calls that have been synchronized with another site via ISCC the same value for the `CallType` attribute as it did in pre-8.0 releases and adds the new `PropagatedCallType` attribute with the value of the `CallType` attribute at the origination site. This provides backward compatibility with existing T-Server clients.
- When set to `true`, T-Server reports in events related to calls that have been synchronized with another site via ISCC the same value for the `CallType` attribute as at the origination site, and adds the new `LocalCallType` attribute with the same value as `CallType` in pre-8.0 releases.

server-id

Default Value: An integer equal to the value `ApplicationDBID` as reported by Configuration Server
Valid Values: Any integer from `0–16383`
Changes Take Effect: Immediately

Specifies the `Server ID` that T-Server uses to generate `Connection IDs` and other unique identifiers. In a multi-site environment, you must assign each T-Server a unique `Server ID`, in order to avoid confusion in reporting applications and T-Server behavior.

Configuration of this option is necessary for Framework environments in which there are two or more instances of the Configuration Database.

Note: If you do not specify a value for this option, T-Server populates it with the `ApplicationDBID` as reported by Configuration Server. Each data object in the Configuration Database is assigned a separate DBID that maintains a unique Server ID for each T-Server configured in the database.

Warning! Genesys does not recommend using multiple instances of the Configuration Database.

user-data-limit

Default Value: `16000`
Valid Values: `0`–`65535`
Changes Take Effect: Immediately

Specifies the maximum size (in bytes) of user data in a packed format.

Note: When T-Server works in mixed 8.x/7.x/6.x environment, the value of this option must not exceed the default value of `16000` bytes; otherwise, 6.x T-Server clients might fail.

license Section

The License section contains the configuration options that are used to configure T-Server licenses. They set the upper limit of the seat-related DN licenses (`tserver_sdn`) that T-Server tries to check out from a license file. See "License Checkout" on page 236.

This section must be called `license`.

Notes:
- T-Server also supports the `license-file` option described in the *Genesys Licensing Guide*.
- The `license` section is not applicable to Network T-Server for DTAG.
- On selected platforms, the limitation of 9999 licenses may no longer apply. Use values greater than 9999 only when instructed by Genesys Customer Care.

If you use two or more T-Servers, and they share licenses, you must configure the following options in the `license` section of the T-Servers.

num-of-licenses

Default Value: `0` or `max` (all available licenses)
Valid Values: String `max` or any integer from `0`–`9999`
Changes Take Effect: Immediately

Specifies how many DN licenses T-Server checks out. T-Server treats a value of `0` (zero) the same as it treats `max`—that is, it checks out all available licenses.

The sum of all `num-of-licenses` values for all concurrently deployed T-Servers must not exceed the number of seat-related DN licenses (`tserver_sdn`) in the corresponding license file. The primary and backup

T-Servers share the same licenses, and therefore they need to be counted only once. T-Server checks out the number of licenses indicated by the value for this option, regardless of the number actually in use.

num-sdn-licenses

Default Value: `0` or `max` (all DN licenses are seat-related)
Valid Values: String `max` (equal to the value of `num-of-licenses`), or any integer from `0`–`9999`
Changes Take Effect: Immediately

Specifies how many seat-related licenses T-Server checks out. A value of `0` (zero) means that T-Server does not grant control of seat-related DNs to any client, and it does not look for seat-related DN licenses at all. A value of `0` (zero) does not allow a T-Server client to register seat-related DNs, which causes issues in a client's attempt to acquire seat-related licenses from T-Server. In this case, T-Server prints the following log message:
`All 0 seat licenses are in use already, registration rejected.`

The sum of all `num-sdn-licenses` values for all concurrently deployed T-Servers must not exceed the number of seat-related DN licenses (`tserver_sdn`) in the corresponding license file. The primary and backup T-Servers share the same licenses, and therefore they need to be counted only once. T-Server checks out the number of licenses indicated by the value for this option, regardless of the number actually in use.

Notes:
- For Network T-Servers, Genesys recommends setting this option to `0` (zero).
- Be sure to configure in the Configuration Database all the DNs that agents use (Extensions and ACD Positions) and that T-Server should control. For further information, see Chapter 2, "DNs and Agent Logins," page 40.

License Checkout

Table 16 shows how to determine the number of seat-related DN licenses that T-Server attempts to check out. See the examples on page 237.

Table 16: License Checkout Rules

Options Settings[a]			License Checkout[b]
num-of-licenses	num-sdn-licenses		Seat-related DN licenses
max (or 0)	max		all available
max (or 0)	x		x

Table 16: License Checkout Rules (Continued)

Options Settings[a]			License Checkout[b]
num-of-licenses	num-sdn-licenses		Seat-related DN licenses
max (or 0)	0		0
x	max		x
x	y		min (y, x)
x	0		0

a. In this table, the following conventions are used: `x` and `y` - are positive integers; `max` is the maximum number of licenses that T-Server can check out; `min (y, x)` is the lesser of the two values defined by `y` and `x`, respectively.

b. The License Checkout column shows the number of licenses that T-Server attempts to check out. The actual number of licenses will depend on the licenses' availability at the time of checkout, and it is limited to 9999.

Examples

This section presents examples of option settings in the `license` section.

Example 1

If...		Then...
Options Settings	License File Settings	License Checkout
num-of-licenses = max	tserver_sdn = 500	500 seat-related DNs
num-sdn-licenses = max		

Example 2

If...		Then...
Options Settings	License File Settings	License Checkout
num-of-licenses = 1000	tserver_sdn = 500	500 seat-related DNs
num-sdn-licenses = max		

Example 3

If...		Then...
Options Settings	**License File Settings**	**License Checkout**
num-of-licenses = 1000	tserver_sdn = 600	400 seat-related DNs
num-sdn-licenses = 400		

Example 4

If...		Then...
Options Settings	**License File Settings**	**License Checkout**
num-of-licenses = max	tserver_sdn = 5000	1000 seat-related DNs
num-sdn-licenses = 1000		

agent-reservation Section

The `agent-reservation` section contains the configuration options that are used to customize the T-Server Agent Reservation feature. See "Agent Reservation" on page 30 section for details on this feature.

This section must be called `agent-reservation`.

Note: The Agent Reservation functionality is currently a software-only feature that is used to coordinate multiple client applications. This feature does not apply to multiple direct or ACD-distributed calls.

collect-lower-priority-requests

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether an agent reservation request is collected, depending on its priority during the time interval specified by the `request-collection-time` configuration option. When set to `false`, during the `request-collection-time` interval, T-Server collects reservation requests of the highest priority only, rejecting newly submitted requests that have a lower priority or rejecting all previously submitted requests if a request with a higher priority arrives. When set to `true` (the default), agent reservation requests are collected as they were in pre-8.x releases.

reject-subsequent-request

Default Value: `true`
Valid Values:

`true`	T-Server rejects subsequent requests.
`false`	A subsequent request prolongs the current reservation made by the same client application for the same agent.

Changes Take Effect: Immediately

Specifies whether T-Server rejects subsequent requests from the same client application, for an agent reservation for the same `Agent` object that is currently reserved.

Note: Genesys does not recommend setting this option to `false` in a multi-site environment in which remote locations use the Agent-Reservation feature.

request-collection-time

Default Value: `100 msec`
Valid Values: See "Timeout Value Format" on page 254.
Changes Take Effect: Immediately

Specifies the time interval that agent reservation requests are collected before a reservation is granted. During this time interval, agent reservation requests are delayed, in order to balance successful reservations between client applications (for example, Universal Routing Servers).

reservation-time

Default Value: `10000 msec`
Valid Values: See "Timeout Value Format" on page 254.
Changes Take Effect: Immediately

Specifies the default interval for which a an Agent DN is reserved. During this interval, the agent cannot be reserved again.

extrouter Section

The `extrouter` section contains the configuration options that are used to support multi-site environments with the Inter Server Call Control (ISCC) feature. The configuration options in this section of the document are grouped with related options that support the same functionality, as follows:

- ISCC Transaction Options, page 241
- Transfer Connect Service Options, page 245
- ISCC/COF Options, page 246
- Event Propagation Options, page 248
- Number Translation Option, page 249
- GVP Integration Option, page 250

This configuration section must be called `extrouter`.

For a description of the ways in which T-Server supports multi-site configurations and for an explanation of the configuration possibilities for a multi-site operation, see the "Multi-Site Support" chapter.

Note: In a multi-site environment, you must configure the `timeout`, `cast-type`, and `default-dn` options with the same value for both the primary and backup T-Servers. If you do not do this, the value specified for the backup T-Server overrides the value specified for the primary T-Server.

match-call-once

Default Value: `true`

Valid Values:

`true`	ISCC does not process (match) an inbound call that has already been processed (matched).
`false`	ISCC processes (attempts to match) a call as many times as it arrives at an ISCC resource or multi-site-transfer target.

Changes Take Effect: Immediately

Specifies how many times ISCC processes an inbound call when it arrives at an ISCC resource. When set to `false`, ISCC processes (attempts to match) the call even if it has already been processed.

Note: Genesys does not recommend changing the default value of the `match-call-once` option to `false` unless you have specific reasons. Setting this option to `false` may lead to excessive or inconsistent call data updates.

reconnect-tout

Default Value: `5 sec`

Valid Values: See "Timeout Value Format" on page 254.

Changes Take Effect: At the next reconnection attempt

Specifies the time interval after which a remote T-Server attempts to connect to this T-Server after an unsuccessful attempt or a lost connection. The number of attempts is unlimited. At startup, T-Server immediately attempts the first connection, without this timeout.

report-connid-changes

Default Value: `false`

Valid Values:

`true`	`EventPartyChanged` is generated.
`false`	`EventPartyChanged` is not generated.

Changes Take Effect: Immediately

Specifies whether the destination T-Server generates `EventPartyChanged` for the incoming call when the resulting `ConnID` attribute is different from the `ConnID` attribute of an instance of the same call at the origination location.

use-data-from

Default Value: `current`

Valid Values:

`active`	The values of `UserData` and `ConnID` attributes are taken from the consultation call.
`original`	The values of `UserData` and `ConnID` attributes are taken from the original call.
`active-data-original-call`	The value of the `UserData` attribute is taken from the consultation call and the value of `ConnID` attribute is taken from the original call.
`current`	If the value of `current` is specified, the following occurs: • Before the transfer or conference is completed, the `UserData` and `ConnID` attributes are taken from the consultation call. • After the transfer or conference is completed, `EventPartyChanged` is generated, and the `UserData` and `ConnID` are taken from the original call.

Changes Take Effect: Immediately

Specifies the call from which the values for the `UserData` and `ConnID` attributes are taken for a consultation call that is routed or transferred to a remote location.

Note: For compatibility with the previous T-Server releases, you can use the values `consult`, `main`, and `consult-user-data` for this option. These are aliases for `active`, `original`, and `current`, respectively.

ISCC Transaction Options

cast-type

Default Value: `route, route-uui, reroute, direct-callid, direct-uui, direct-network-callid, direct-notoken, direct-digits, direct-ani, dnis-pool, pullback`

Valid Values: `route, route-uui, reroute, direct-callid, direct-uui, direct-network-callid, direct-notoken, direct-digits, direct-ani, dnis-pool, pullback`

Changes Take Effect: For the next request for the remote service

Specifies—using a space-, comma- or semicolon-separated list—the routing types that can be performed for this T-Server.

The valid values provide for a range of mechanisms that the ISCC feature can support with various T-Servers, in order to pass call data along with calls between locations.

Because switches of different types provide calls with different sets of information parameters, some values might not work with your T-Server. See Table 3 on page 77 for information about supported transaction types by a specific T-Server. The "Multi-Site Support" chapter also provides detailed descriptions of all transaction types.

Notes: For compatibility with the previous T-Server releases, you can use the `direct` value for this option. This is an alias for `direct-callid`.

An alias, `route-notoken`, has been added to the `route` value.

default-dn

Default Value: No default value
Valid Values: Any DN
Changes Take Effect: For the next request for the remote service

Specifies the DN to which a call is routed when a Destination DN (`AttributeOtherDN`) is not specified in the client's request for routing. If neither this option nor the client's request contains the destination DN, the client receives `EventError`.

Note: This option is used only for requests with route types `route`, `route-uui`, `direct-callid`, `direct-network-callid`, `direct-uui`, `direct-notoken`, `direct-digits`, and `direct-ani`.

direct-digits-key

Default Value: `CDT_Track_Num`
Valid Values: Any valid key name of a key-value pair from the `UserData` attribute
Changes Take Effect: For the next request for the remote service

Specifies the name of a key from the `UserData` attribute that contains a string of digits that are used as matching criteria for remote service requests with the `direct-digits` routing type.

Note: For compatibility with the previous T-Server releases, this configuration option has an alias value of `cdt-udata-key`.

dn-for-unexpected-calls

Default Value: No default value
Valid Values: Any DN
Changes Take Effect: Immediately

Specifies a default DN for unexpected calls arriving on an External Routing Point.

network-request-timeout

Default Value: `20 sec`
Valid Values: See “Timeout Value Format” on page 254.
Changes Take Effect: For the next network request

For a premise T-Server, this option specifies the time interval that the premise T-Server waits for a response, after relaying a `TNetwork<...>` request to the Network T-Server. For a Network T-Server, this option specifies the time interval that the Network T-Server waits for a response from an SCP (Service Control Point), after initiating the processing of the request by the SCP.

When the allowed time expires, the T-Server cancels further processing of the request and generates `EventError`.

register-attempts

Default Value: `5`
Valid Values: Any positive integer
Changes Take Effect: For the next registration

Specifies the number of attempts that T-Server makes to register a dedicated External Routing Point.

register-tout

Default Value: `2 sec`
Valid Values: See “Timeout Value Format” on page 254.
Changes Take Effect: For the next registration

Specifies the time interval after which T-Server attempts to register a dedicated External Routing Point. Counting starts when the attempt to register a Routing Point fails.

request-tout

Default Value: `20 sec`
Valid Values: See “Timeout Value Format” on page 254.
Changes Take Effect: For the next request for remote service

Specifies the time interval that a T-Server at the origination location waits for a notification of routing service availability from the destination location. Counting starts when the T-Server sends a request for remote service to the destination site.

resource-allocation-mode

Default Value: `circular`
Valid Values:

`home` T-Server takes an alphabetized (or numerically sequential) list of configured DNs and reserves the first available DN from the top of the list for each new request. For example, if the first DN is not available, the second DN is allocated for a new request. If the first DN is freed by the time the next request comes, the first DN is allocated for this next request.

`circular` T-Server takes the same list of configured DNs, but reserves a subsequent DN for each subsequent request. For example, when the first request comes, T-Server allocates the first DN; when the second request comes, T-Server allocates the second DN; and so on. T-Server does not reuse the first DN until reaching the end of the DN list.

Changes Take Effect: Immediately

Specifies the manner in which T-Server allocates resources (that is, DNs of the `External Routing Point` type and Access Resources with the `Resource Type` set to `dnis`) for multi-site transaction requests.

resource-load-maximum

Default Value: `0` (zero)
Valid Values: Any positive integer
Changes Take Effect: Immediately

Specifies the maximum number of ISCC routing transactions that can be concurrently processed at a single DN of the `External Routing Point` route type. After a number of outstanding transactions at a particular DN of the `External Routing Point` type reaches the specified number, T-Server considers the DN not available. Any subsequent request for this DN is queued until the number of outstanding transactions decreases. A value of `0` (zero) means that no limitation is set to the number of concurrent transactions at a single External Routing Point. In addition, the `0` value enables T-Server to perform load balancing of all incoming requests among all available External Routing Points, in order to minimize the load on each DN.

route-dn

Default Value: No default value
Valid Values: Any DN
Changes Take Effect: Immediately

Specifies the DN that serves as a Routing Point for the `route` transaction type in the multiple-to-one access mode.

timeout

Default Value: `60 sec`
Valid Values: See "Timeout Value Format" on page 254.
Changes Take Effect: For the next request for remote service

Specifies the time interval that the destination T-Server waits for a call routed from the origination location. Counting starts when this T-Server notifies the requesting T-Server about routing service availability. The timeout must be long enough to account for possible network delays in call arrival.

use-implicit-access-numbers

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

Determines whether an External Routing Point in which at least one access number is specified is eligible for use as a resource for calls coming from switches for which an access number is not specified in the External Routing Point. If this option is set to `false`, the External Routing Point is not eligible for use as a resource for calls coming from such switches. If this option is set to `true`, an implicit access number for the External Routing Point, composed of the switch access code and the DN number of the External Routing Point, will be used.

Note: If an External Routing Point does not have an access number specified, this option will not affect its use.

Transfer Connect Service Options

tcs-queue

Default Value: No default value
Valid Values: Any valid DN number
Changes Take Effect: For the next request for the remote service

Specifies the TCS DN number to which a call, processed by the TCS feature, is dialed after the originating external router obtains an access number. This option applies only if the `tcs-use` option is activated.

tcs-use

Default Value: `never`
Valid Values:

`never`	The TCS feature is not used.
`always`	The TCS feature is used for every call.
`app-defined`	In order to use the TCS feature for a multi-site call transfer request, a client application must add a key-value pair with a TC-type key and a nonempty string value to the `UserData` attribute of the request.

Changes Take Effect: Immediately

Specifies whether the Transfer Connect Service (TCS) feature is used.

Note: For compatibility with the previous T-Server releases, you can use the value `up-app-depended` for this option. This is an alias for `app-defined`.

ISCC/COF Options

cof-ci-defer-create

Default Value: `0`
Valid Values: See "Timeout Value Format" on page 254.
Changes Take Effect: Immediately

Specifies the time interval that T-Server waits for call data from the switch before generating a negative response for a call data request from a remote T-Server. If T-Server detects the matching call before this timeout expires, it sends the requested data. This option applies only if the `cof-feature` option is set to `true`.

cof-ci-defer-delete

Default Value: `0` (zero)
Valid Values: See "Timeout Value Format" on page 254.
Changes Take Effect: Immediately

Specifies the time interval that T-Server waits before deleting call data that might be overflowed. If set to `0`, deletion deferring is disabled. This option applies only if the `cof-feature` option is set to `true`.

cof-ci-req-tout

Default Value: `500 msec`
Valid Values: See "Timeout Value Format" on page 254.
Changes Take Effect: For the next COF operation

Specifies the time interval during which T-Server will wait for call data requested with respect to a call originated at another site. After T-Server sends the call data request to remote T-Servers, all events related to this call are

suspended until either the requested call data is received or the specified timeout expires. This option applies only if the cof-feature option is set to true.

cof-ci-wait-all

Default Value: false
Valid Values:

true	T-Server waits for responses from all T-Servers that might have the requested call data before updating the call data with the latest information.
false	T-Server updates the call data with the information received from the first positive response.

Changes Take Effect: Immediately

Specifies whether T-Server, after sending a request for matching call data, waits for responses from other T-Servers before updating the call data (such as CallHistory, ConnID, and UserData) for a potentially overflowed call. The waiting period is specified by the cof-ci-req-tout and cof-rci-tout options. This option applies only if the cof-feature option is set to true.

cof-feature

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

Enables or disables the Inter Server Call Control/Call Overflow (ISCC/COF) feature.

cof-rci-tout

Default Value: 10 sec
Valid Values: See “Timeout Value Format” on page 254.
Changes Take Effect: For the next COF operation

Specifies the time interval that T-Server waits for call data from other T-Servers’ transactions. Counting starts when cof-ci-req-tout expires. This option applies only if the cof-feature option is set to true.

local-node-id

Default Value: 0 (zero)
Valid Values: 0 or any positive integer
Changes Take Effect: Immediately

This option, if enabled, checks all networked calls against the specified NetworkNodeID (the identity of the switch to which the call initially arrived). If the NetworkNodeID is the same as the value of this option, the request for call information is *not* sent. The default value of 0 disables the functionality of this

option. To establish an appropriate NetworkNodeID, specify a value other than the default. This option applies only if the cof-feature option is set to true.

Note: This option applies only to T-Server for Nortel Communication Server 2000/2100.

default-network-call-id-matching

Default Value: No default value
Valid Values: See the "T-Server-Specific Configuration Options" chapter for an option description for your T-Server
Changes Take Effect: Immediately

When a value for this option is specified, T-Server uses the NetworkCallID attribute for the ISCC/COF call matching.

To activate this feature, the cof-feature option must be set to true.

Note: SIP Server and several T-Servers support the NetworkCallID attribute for the ISCC/COF call matching in a way that requires setting this option to a specific value. For information about the option value that is specific for your T-Server, see the "T-Server-Specific Configuration Options" chapter of your *T-Server Deployment Guide*.

Event Propagation Options

compound-dn-representation

Default Value: true
Valid Values: true, false
Changes Take Effect: Immediately

Specifies which format T-Server uses to represent a DN when reporting an OtherDN or ThirdPartyDN attribute in event propagation messages.

When set to true, the <switch>::DN (compound) format is used. This option value supports backward compatibility for pre-8.x T-Server ISCC/EPP functionality and is provided for multi-site deployments where the same DNs are configured under several switches.

When set to false, the DN (non-compound) format is used. This option value ensures more transparent reporting of OtherDN or ThirdPartyDN attributes and is recommended for all single-site deployments, as well as for multi-site deployments that do not have the same DNs configured under several switches. This option applies only if the event-propagation option is set to list.

Note: Local DNs are always represented in the non-compound (DN) form.

epp-tout

Default Value: `0`
Valid Values: See "Timeout Value Format" on page 254.
Changes Take Effect: Immediately

Specifies the time interval during which T-Server attempts to resolve race conditions that may occur in deployments that use switch partitioning or intelligent trunks. This option applies only if the `event-propagation` option is set to `list`.

Note: If the time interval is not long enough to account for possible network switching delays, T-Server may produce duplicated events, such as events that are propagated by the ISCC and generated locally.

event-propagation

Default Value: `list`
Valid Values:

`list`	Changes in user data and party events are propagated to remote locations through call distribution topology.
`off`	The feature is disabled. Changes in user data and party events are not propagated to remote locations.

Changes Take Effect: Immediately

Specifies whether the Event Propagation feature is enabled.

Number Translation Option

inbound-translator-<*n*>

Default Value: No default value
Valid Value: Any valid name
Changes Take Effect: Immediately

Specifies the name of another configuration section as the value for the `inbound-translator` option.

For example,

```
inbound-translator-1 = ani-translator
```

where:

`ani-translator` is the name of the configuration that describes the translation rules for inbound numbers.

GVP Integration Option

handle-vsp

Default Value: `no`
Valid Values:

`requests`	ISCC will process and adjust requests related to this DN and containing a `Location` attribute before submitting them to the service provider.
`events`	ISCC will process and adjust each event received from the service provider in response to a request containing a `Location` attribute before distributing the event to T-Server clients.
`all`	ISCC will process and adjust both events and requests.
`no`	No ISCC processing of such requests and events takes place.

Changes Take Effect: Immediately

Specifies if multi-site Call Data synchronization of virtual calls or simulated call flows is performed by T-Server or is left to an external application (Service Provider) that has registered for a DN with the `AddressType` attribute set to `VSP` (Virtual Service Provider).

backup-sync Section

The backup-synchronization section contains the configuration options that are used to support a high-availability (`hot standby` redundancy type) configuration.

This section must be called `backup-sync`.

Note: These options apply only to T-Servers that support the `hot standby` redundancy type.

addp-remote-timeout

Default Value: `0`
Valid Values: Any integer from `0`–`3600`
Changes Take Effect: Immediately

Specifies the time interval that the redundant T-Server waits for a response from this T-Server after sending a polling signal. The default value of `0` (zero) disables the functionality of this option. To establish an appropriate timeout, specify a value other than the default. This option applies only if the `protocol` option is set to `addp`.

addp-timeout

Default Value: `0`
Valid Values: Any integer from `0`–`3600`
Changes Take Effect: Immediately

Specifies the time interval that this T-Server waits for a response from another T-Server after sending a polling signal. The default value of `0` (zero) disables the functionality of this option. To establish an appropriate timeout, specify a value other than the default. This option applies only if the `protocol` option is set to `addp`.

addp-trace

Default Value: `off`
Valid Values:

`off, false, no`	No trace (default).
`local, on, true, yes`	Trace on this T-Server side only.
`remote`	Trace on the redundant T-Server side only.
`full, both`	Full trace (on both sides).

Changes Take Effect: Immediately

Specifies whether `addp` messages are traced in a log file, to what level the trace is performed, and in which direction. This option applies only if the `protocol` option is set to `addp`.

protocol

Default Value: `default`
Valid Values:

`default`	The feature is not active.
`addp`	Activates the Advanced Disconnect Detection Protocol.

Changes Take Effect: When the next connection is established

Specifies the name of the method used to detect connection failures. If you specify the `addp` value, you must also specify a value for the `addp-timeout`, `addp-remote-timeout`, and `addp-trace` options.

sync-reconnect-tout

Default Value: `20 sec`
Valid Values: See “Timeout Value Format” on page 254.
Changes Take Effect: Immediately

Specifies the time interval after which the backup T-Server attempts to reconnect to the primary server (for a synchronized link).

call-cleanup Section

The call-cleanup section contains the configuration options that are used to control detection and cleanup of stuck calls in T-Server. For more information on stuck call handling, refer to the "Stuck Call Management" chapter in the *Framework 8.x Management Layer User's Guide*.

This section must be called `call-cleanup`.

cleanup-idle-tout

Default Value: `0` (zero)
Valid Values: See "Timeout Value Format" on page 254.
Changes Take Effect: Immediately

Specifies the time interval that T-Server waits for a call to be updated from its last update. After this time elapses, if no new events about the call are received, T-Server clears this call as a stuck call, either by querying the switch (if a CTI link provides such capabilities) or by deleting the call information from memory unconditionally. The default value of `0` disables the stuck calls cleanup.

Note: If the call-cleanup functionality is enabled in T-Server for Avaya Communication Manager, the UCID (Universal Call ID) feature must be enabled on the switch as well. This allows the UCID to be generated and passed to T-Server.

notify-idle-tout

Default Value: `0`
Valid Values: See "Timeout Value Format" on page 254.
Changes Take Effect: Immediately

Specifies the time interval that T-Server waits for a call to be updated from its last update. After this time elapses, if no new events about the call are received, T-Server reports this call as a stuck call. The default value of `0` disables the stuck calls notification.

periodic-check-tout

Default Value: `10 min`
Valid Values: See "Timeout Value Format" on page 254.
Changes Take Effect: Immediately

Specifies the time interval for periodic checks for stuck calls. These checks affect both notification and cleanup functionality, and are made by checking the T-Server's own call information with call information available in the switch. For performance reasons, T-Server does not verify whether the

notify-idle-tout or cleanup-idle-tout option has expired before performing this check.

Note: Setting this option to a value of less than a few seconds can affect T-Server performance.

Examples

This section presents examples of option settings in the call-cleanup section.

Example 1

```
cleanup-idle-tout = 0
notify-idle-tout = 0
periodic-check-tout = 10
```

With these settings, T-Server will not perform any checks for stuck calls.

Example 2

```
cleanup-idle-tout = 0
notify-idle-tout = 5 min
periodic-check-tout = 10 min
```

With these settings, T-Server performs checks every 10 minutes and sends notifications about all calls that have been idle for at least 5 minutes.

Example 3

```
cleanup-idle-tout = 20 min
notify-idle-tout = 5 min
periodic-check-tout = 10 min
```

With these settings, T-Server performs checks every 10 minutes, sends notifications about all calls that have been idle for at least 5 minutes, and attempts to clean up all calls that have been idle for more than 20 minutes.

Translation Rules Section

The section name is specified by the inbound-translator-<*n*> option. It contains options that define translation rules for inbound numbers.

You can choose any name for this section, provided that it matches the value of the section. Every option in this section corresponds to a rule and must conform to the format described below. You can configure as many rules as necessary to accommodate your business needs.

rule-<*n*>

Default Value: No default value
Valid Value: Any valid string in the following format:
in-pattern=<input pattern value>;out-pattern=<output pattern value>
Changes Take Effect: Immediately

Defines a rule to be applied to an inbound number. The two parts of the option value describe the input and output patterns in the rule. When configuring the

pattern values, follow the syntax defined in "Using ABNF for Rules" on page 86. See "Configuring Number Translation" on page 93 for examples of these rules as well as detailed instructions for creating rules for your installation. For example, a value for this configuration option might look like this:

```
rule-01 = in-pattern=0111#CABBB*ccD;out-pattern=ABD
```

security Section

The `security` section contains the configuration options that are used to configure secure data exchange between T-Servers and other Genesys components. Refer to the *Genesys 8.x Security Deployment Guide* for complete information on the security configuration.

Timeout Value Format

This section of the document describes the values to use for those T-Server common options that set various timeouts. The current format allows you to use fractional values and various time units for timeout settings.

For timeout-related options, you can specify any value that represents a time interval, provided that it is specified in one of the following formats:

`[[[`*hours*`:]`*minutes*`:]`*seconds*`][`*milliseconds*`]`

or

`[`*hours* `hr][`*minutes* `min][`*seconds* `sec][`*milliseconds* `msec]`

Where a time unit name in italic (such as *hours*) is to be replaced by an integer value for this time unit.

Integer values with no measuring units are still supported, for compatibility with previous releases of T-Server. When you do not specify any measuring units, the units of the default value apply. For example, if the default value equals `60 sec`, specifying the value of `30` sets the option to 30 seconds.

Example 1

The following settings result in a value of 1 second, 250 milliseconds:

```
sync-reconnect-tout = 1.25
```

```
sync-reconnect-tout = 1 sec 250 msec
```

Example 2

The following settings result in a value of 1 minute, 30 seconds:

```
timeout = 1:30
```

```
timeout = 1 min 30 sec
```

Changes from Release 8.0 to 8.1

Table 17 lists the configuration options that:

- Are new or changed in the 8.1 release of T-Server
- Have been added or changed since the most recent 8.0 release of this document

If a configuration option has been replaced with another that enables the same functionality, the new option name and its location in this chapter are noted.

Table 17: Option Changes from Release 8.0 to 8.1

Option Name	Option Values	Type of Change	Details
TServer Section			
background-processing	`true, false`	See Details	Default value changed to `true`. See the option description on page 230.

Chapter

13 T-Server-Specific Configuration Options

This chapter describes the configuration options that are unique to T-Server for Cisco Unified Communications Manager. It includes the following sections:

To establish a link connection, configure the link options that are applicable to the connection protocol used in your environment (TCP/IP).

Application-Level Options

Configuration options specific to T-Server functionality are set in Configuration Manager or Genesys Administrator, in the corresponding sections on the `Options` tab of the T-Server `Application` object.

For ease of reference, the options have been arranged in alphabetical order within their corresponding sections:

Mandatory Options

Table 18 contains the mandatory options when T-Server is operating in Socket mode. All other options in this chapter are configured to enable T-Server to support other features.

To establish a link connection, simply configure the link options that are applicable to the connection protocol used in your environment.

Table 18: Mandatory Options in Socket Mode

Option Name	Default Value	Details
TServer Section		
link-*n*-name	No default value	Specifies the section name containing the configuration options assigned to that link, where *n* is a consecutive number for a Link. See the description on page 266
Link Section		
ccm-host	No default value	Specifies the host name that Cisco Unified Communications Manager uses. See description on page 284.
hostname	localhost	Specifies the host of the link according to the switch configuration. This should always be `localhost`. See the description on page 284.
password	No default value	Specifies the `password` field for the user's `login ID`. See description on page 284.
port	No default value	Specifies the TCP/IP port of the link. See the description on page 284.

Table 18: Mandatory Options in Socket Mode (Continued)

Option Name	Default Value	Details
protocol	No default value	Specifies the `protocol` field. It should always be set to `tcp`. See description on page 284.
user-login	No default value	Specifies the user login ID configured in Cisco Unified Communications Manager that has permission to control all the DNs that T-Server will control. See description on page 285.

TServer Section

This section describes the configuration options that are unique to T-Server for Cisco Unified Communications Manager. Configure these options in the `TServer` section on the `Options` tab for the T-Server `Application` object in the Configuration Layer.

You must call this section `TServer`.

agent-no-answer-action

Default Value: `none`
Valid Values: `none, notready, walkaway`
Changes Take Effect: Immediately
Related Feature: "Redirect On No Answer" on page 162

Specifies the agent state to which T-Server will be set after the time period for the `agent-no-answer-timeout` option has expired. A value of `none` means that the agent will remain in its current state.

Note: The `walkaway` value is identical to the `notready` value unless a non-ACD (soft agents) setup is utilized. If a non-ACD (soft agents) T-Server setup is not utilized, the `agent-no-answer-action` option may be set to `notready`. This will enable the agent to change to a `NotReady` state while the call is ringing, and before sending the redirect.

The option can be set in the following places in order of precedence (highest to lowest):

1. The `AttributeExtensions` key `NO_ANSWER_ACTION` of `TRouteCall`.
2. The `TServer` section in the `Annex` tab of an `Agent Login` object.
3. The `TServer` section in the `OPtions` tab of a T-Server `Application` object.

agent-no-answer-overflow

Default Value: `none`
Valid Values:

`none`	The call will remain ringing on the agent phone.
`recall`	The call will be redirected back to the Routing Point or the ACD Queue that delivered the call to the agent.
Any destination digits	A valid destination DN must be provided (a Queue or Routing Point on the local switch is recommended).

Changes Take Effect: Immediately
Related Feature: “Redirect On No Answer” on page 162

After the time period for the `agent-no-answer-timeout` option has expired, T-Server will redirect the ringing call to the destination described by this option.

The option can be set in the following places in order of precedence (highest to lowest):

1. The `AttributeExtensions` key `NO_ANSWER_OVERFLOW` of `TRouteCall`.
2. The `TServer` section in the `Annex` tab of an `Agent Login` object.
3. The `TServer` section in the `OPtions` tab of a T-Server `Application` object.

agent-no-answer-timeout

Default Value: `0`
Valid Values: `0 to 600`
Changes Take Effect: Immediately
Related Feature: “Redirect On No Answer” on page 162

Calls ringing on an agent's phone that were distributed from an ACD Queue or a Routing Point will wait for the telephone to ring for this timeout period (in seconds) before performing the actions described by the `agent-no-answer-action` option and redirect the call to the destination described in the `agent-no-answer-overflow` option. The default value of `0` (zero) disables the functionality of this option.

The option can be set in the following places in order of precedence (highest to lowest):

1. The `AttributeExtensions` key `NO_ANSWER_TIMEOUT` of `TRouteCall`.
2. The `TServer` section in the `Annex` tab of an `Agent Login` object.
3. The `TServer` section in the `OPtions` tab of a T-Server `Application` object.

application

Default Value: `T-Server`
Valid Values: Any character string
Changes Take Effect: After T-Server is restarted

Specifies the name of the T-Server application shown in the Cisco Unified Communications Manager switch. It is used to identify an application to the Cisco switch.

audio-codec

Default Value: `1,3`
Valid Values: Comma-separated list of any of the following:

`1`	`G.711 mu-Law`
`2`	`G.711 a-Law`
`3`	`G.723`
`4`	`G.729A`
`8`	`MS-GSM and GSM Full Rate`

Changes Take Effect: After T-Server is restarted

Specifies the audio codec(s) to be used by this T-Server. G.711 operates at a higher bit rate than G.723 and G.729A, providing good quality but consuming more network resources. MS-GSM is intermediate, providing moderate quality and low network resource consumption. For G.711, mu-Law is used in North America and Japan and a-Law is used elsewhere, including international routes.

busy-tone

Default Value: `music/busy_5sec`
Valid Values: Name and path of any valid audio file
Changes Take Effect: Immediately for all new calls

Specifies an audio file to be played for the `Busy` treatment.

call-forward-action

Default Value: `logout`
Valid Values:

`none`	No agent state change is performed.
`logout`	Agent is automatically logged out and prevented from logging back in while the DN is forwarded.

notready	Agent is automatically put into the not ready state (workmode 0) and prevented from going ready (or potentially logging back in) while the DN is forwarded.
acw	Agent is automatically put into the not ready state (workmode AgentAfterCallWork) and prevented from going ready (or potentially logging back in) while the DN is forwarded.
walkaway	Agent is automatically put into the not ready state (workmode AgentWalkAway) and prevented from going ready (or potentially logging back in) while the DN is forwarded.

Changes Take Effect: Immediately for all new calls

Provides a new range of agent state change possibilities that can be performed by T-Server when forwarding is detected. This option can be used to replace the existing option logout-on-fwd. If there is a conflict between logout-on-fwd and call-forward-action within logout on forward, the logout-on-fwd option takes precedence. The values none, notready, acw, and walkaway take precedence over any setting of the logout-on-fwd option.

clean-calls-on-all-links-up

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

When set to true, T-Server queries calls, once all links become active in order to clean up any old calls that may have been released since links were disconnected (EventLinkDisconnected generated). This occurs for both primary and backup T-Servers to avoid stuck calls.

collect-tone

Default Value: music/collect
Valid Values: Name and path of any valid audio file
Changes Take Effect: Immediately for all new calls

Specifies that T-Server uses this non-completion tone to produce the sound played during DTMF digit collection. Basically, this option is a duplicate of the silence-tone option.

create-addr-on-register

Default Value: true
Valid Values: true, false
Changes Take Effect: Immediately

When set to true, clients can register and send requests for DNs that do not have an entry in the Configuration Layer. If set to false, clients registering for DNs not found in the Configuration Layer will see the following error message: DN is not configured in CME.

debug

Default Value: `+all`
Valid Values: `+/-all, +/-jtapi, +/-toop, +/-sm`
Changes Take Effect: Immediately

Specifies which submodules in Cisco Unified Communications Manager produce debug output. This output is ultimately controlled by the `Log` section. The format of the string is `+module1 -module2,` which means that `module1` produces debug output and `module2` does not. The module name `all` represents all modules. Current modules are `all`, `toop` (internal call object manipulation), `sm` (Stream Manager), and `jtapi` (JTAPI events and requests). All T-Server debug output that is a part of a debug module is prefixed with `(module name)*>>`.

default-dn

Default Value: No default value
Valid Values: Any character string. Do not use the period (`.`) character or the commercial "at" (`@`) sign in this string.
Changes Take Effect: Immediately

This option has the following functionality:

- Provides the default DN that calls are routed to if Universal Routing Server (URS) requests T-Server to route to a default target. If no default is provided, T-Server rejects the call if URS sends a default target.
- If the Agent Ring Redirect Timeout expires but there is no value in the user-data `RING_REDIRECT_DN` key, the call is redirected to the address specified in `default-dn` option. If the `default-dn` option is not set, the call is not redirected (it still rings) but the agent is still placed into the `NotReady` state.
- Provides the destination for calls that could not be queued on an ACD queue. (for example, if the ACD queue ran out of music ports).

default-monitor-mode

Default Value: `normal`
Valid Values:

`normal`	Silent monitoring (mute supervisor connection, possible warning beep).
`mute`	Silent monitoring (mute supervisor connection).
`coach`	Whisper coaching (only monitored agent can hear Supervisor).
`connect`	Open supervisor presence (Not supported).

Changes Take Effect: With the next received `TMonitorNextCall` request

Specifies the monitor mode for subscription. This option is used when the `MonitorMode` extension in the request `TMonitorNextCall` is not specified or specified incorrectly.

default-monitor-tone

Default Value: `local`
Valid Values:

`none`	No tone is generated on both legs of the call.
`local`	A periodic tone is generated on the agent's phone.
`remote`	A periodic tone is generated on the customer's phone.
`both`	A periodic tone is generated on both the customer's and the agent's phone.

Changes Take Effect: With the next monitored call

This option is set to generate a periodic tone to let people know that they are monitored.

default-record-tone

Default Value: `none`
Valid Values:

`none`	No tone is generated on both legs of the call.
`local`	A periodic tone is generated on the agent's phone.
`remote`	A periodic tone is generated on the customer's phone.
`both`	A periodic tone is generated on both the customer's and the agent's phone.

Changes Take Effect: With the next recorded call

This option is set to generate a periodic tone to let people know that they are being recorded.

delay-dialing

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Providing AttributeDNIS in EventDialing" on page 161

If set to `true`, T-Server includes `AttributeDNIS` in `EventDialing` for most call flows by delaying `EventDialing` until DNIS information is available.

Warning! Use this option only when requested by Genesys Customer Care.

enable-data-on-bridged

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

When set to `true`, the data on IP Phones will not be displayed if the phone is in a bridged state.

external-dn-length

Default Value: `0`
Valid Values: `0–60`
Changes Take Effect: Immediately.

Determines the length of the right-most digits to compare between two external DNs on the same conference call. This is to assist T-Server in determining whether or not the DNs are the same (differ only by prefix digits). A value of `0` (zero) turns this comparison off..

Warning! If this option is used, its value must be set high enough to avoid false matches between external DNs.

Use this option only when requested by Genesys Customer Care.

fast-busy-tone

Default Value: `music/atb_5sec`
Valid Values: Name and path of any valid audio file
Changes Take Effect: Immediately for all new calls

Specifies an audio file to be played for the `FastBusy` treatment.

force-moh-on-ms-down

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Disabling the Default MOH Treatment" on page 157

When set to `true`, T-Server uses Music-On-Hold (MOH) treatments provided by the CUCM MOH Server for client treatment requests resulting in `EventTreatmentApplied`. When set to `false`, T-Server responds to treatment requests with `EventTreatmentNotApplied` (no default treatment provided by the CUCM MOH Server is used). This option applies when there is no active Genesys Media Server connection.

free-form-terminal-id

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately for all new calls

When set to `true`, T-Server accepts any format for the terminal ID but limits its length to a maximum of 15 characters.

When set to `false`, T-Server accepts the terminal ID only in any of the following formats:

- SEP and 12 alpha-numeric symbols—for example, `SEP000F8FB7173C`.
- The terminal ID in the `xxx.xxx.xxx.xxx` format—for example, `123.123.123.123`.

intrusion-enabled

Default Value: `true`
Valid Values: `true, false`

`false`	Intrusion is disabled.
`true`	Intrusion is enabled.

Changes Take Effect: At the next request `TMonitorNextCall`

Controls whether intrusion is enabled. For example, request `TMonitorNextCall` will result in immediate single-step conference with Supervisor DN if there is an active call on Agent DN at the moment when `TMonitorNextCall` is processed.

jtapi-update-mode

Default Value: `startup`
Valid Values: `never`, `install`, `startup`
Changes Take Effect: After T-Server is restarted

Specifies when the `jtapi.jar` library is synchronized with the CTIManager host. T-Server needs `jtapi.jar` in order to communicate with Cisco CTIManager.

When this option is set to `never`, no synchronization takes place. In this case, the `jtapi.jar` must be manually copied from the CTI-Manager host to the T-Server working directory.

When this option is set to `install`, the `jtapi.jar` file is downloaded from the CTIManager host only when no existing `jtapi.jar` is available. The updated `jtapi.jar` file will be used upon the next T-Server restart.

When this option is set to `startup`, T-Server synchronizes `jtapi.jar` before startup. This adds several seconds to the startup time, during which T-Server verifies the CTIManager `jtapi.jar` version. If the version is the same as the one currently in use, no action is taken; otherwise the currently used version of `jtapi.jar` is replaced with the one downloaded from CTIManager and used for startup.

Unless the option is set to `never`, the user who started this T-Server must have "write" permission for the T-Server working directory for this feature to operate.

link-*n*-name

Default Value: Mandatory field. No default value.
Valid Values: Any valid name
Changes Take Effect: Immediately

Specifies the section name containing the configuration options assigned to that link, where *n* is a nonzero consecutive number for a link. You must specify a value for this option.

Notes: The `link-n-name` option name refers to the link number and the section name (for example, `link-1-name`).

See "Socket Mode of Communication" on page 168 for more details.

Warning! Do not update the link configuration while T-Server is running.

logout-on-agent-disconnect

Default Value: `false`
Valid Values: `false, true`
Changes Take Effect: Immediately

With value `true`, T-Server sends an `AgentLogout` message if an agent application disconnects.

logout-on-fwd

Default Value: `false`
Valid Values: `false, true`
Changes Take Effect: Immediately

With value `true`, T-Server send an `AgentLogout` message if a DN becomes `forwarded` to `OtherDN`.

logout-on-out-of-service

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether agents can log in to DNs that are in the `out-of-service` state. If the value is set to `true`, agents cannot log in.

DNs can enter the `out-of-service` state if their IP phone is unplugged or if the IP network associated with the phone is unreachable. T-Server raises `EventDnOutOfService` when a DN enters the `out-of-service` state, and `EventDnBackInService` when the DN returns to the `in-service` state. At T-Server startup all DNs are considered to be in the `out-of-service` state until their true state is known otherwise.

If the value is set to `true`, when a DN enters the `out-of-service` state while an agent is logged in there, T-Server logs out the agent and sends `EventAgentLogout`, and if an agent attempts to log in to the `out-of-service` DN, the T-Server responds with `EventError`.

If the value is set to `false`, agents can log in at any time, regardless of the DN state.

out-of-service-action

Default Value: `logout`
Valid Values: `none, logout, notready, acw, walkaway`
Changes Take Effect: Immediately
Related Option: `logout-on-out-of-service`
Related Feature: “Agent State for Out-Of-Service Agent DNs” on page 140

T-Server automatically changes the agent state to the specified state (and work mode) when the corresponding agent DN goes out of service (OOS). The agent state change may be delayed by specifying a delay in the related option `out-of-service-action-delay`. After T-Server changes the state, the agent cannot log in until the corresponding DN goes back in service. Valid option values are:

- `none`—No agent state change is performed.
- `logout`—The agent is automatically logged out. This is equivalent to setting the existing option `logout-on-out-of-service` to `true`.
- `notready`—The agent is automatically made not ready.
- `acw`—The agent is automatically made not ready with work mode `AgentAfterCallWork`.
- `walkaway`—The agent is automatically made not ready with work mode `AgentWalkAway`.

If both `out-of-service-action` and `logout-on-out-of-service` options are configured, the `logout-on-out-of-service` option takes precedence unless the `out-of-service-action` option is set to any value other than `logout`.

out-of-service-action-delay

Default Value: `0`
Valid Values: Integer from `0` to `3600000`
Changes Take Effect: Immediately
Related Option: `out-of-service-action`
Related Feature: “Agent State for Out-Of-Service Agent DNs” on page 140

Specifies the period of time, in milliseconds, that T-Server waits before changing the agent state specified by the `out-of-service-action` option. If the corresponding agent DN comes back in service before the delay is completed, T-Server does not change the agent state. This avoids the automatic agent state changes if a DN goes out of service (OOS) for only a short time. Note that this option will delay the effect of either `out-of-service-action` or `logout-on-out-of-service` option.

packet-size

Default value: `20`
Valid values: Comma-separated list of any of: `20,30,60` (see Table 19)
Changes Take Effect: After T-Server is restarted

Specifies the maximum packet size, as follows:

Table 19: Values of Packet-Size Option

Codec	Default Value	Valid Values	Unit
G.711 mu-Law or a-Law	`20`	`20, 30`	msec per packet
G.723, G.729A	`20`	`20, 30, 60`	msec per packet
MS-GSM	Not applicable		
GSM Full Rate	Not applicable		

If more than one audio-codec is specified, the values of `packet-size` correspond one-to-one, in the same order. For example, suppose that audio-codec has the values `1, 2, 4` and that `packet-size` has the values `20, 30, 2`. This means that this T-Server uses codecs G.711 mu-Law at 20 msec per packet, G.711 a-Law at 30 msec per packet, and G.729A at 2 frames per packet. You do not have to list the audio-codec values in ascending order (therefore `audio-codec = 4, 2, 1` and `packet-size = 2, 30, 20` would have the same effect as the example just described), although it is probably easier to do so. If a `packet-size` value is invalid for the codec it corresponds to, it is ignored, and that codec receives the default packet size.

queue-music

Default Value: `music/in_queue`
Valid Value: Name and path of any valid audio file
Changes Take Effect: Immediately for all new calls
Related Feature: "Music Treatment on ACD Queues" on page 156
Specifies the name of the file for music treatment of ACD Queues, when the ACD queue DN has no specific music file configured.

record-only-business-calls

Default Value: `false`
Valid Values: `false`, `true`
Changes Take Effect: Immediately

If set to `true`, only calls coming from a Routing Point or an ACD Queue to the target DN will be recorded.

recording-filename

Default Value: `CUCM/call-$REFCI$-at-$AGENTDN$-on-$DATE$`
Valid Values: Any valid file name using the variables specified below
Changes Take Effect: When the next call recording is initiated

Specifies the file name for call recording when call recording is initiated automatically, according to T-Server configuration. When this option contains a value, the generated file name is added as `UserData` to the call with the `GSIP_REC_FN` key.

The following variables are used when creating the file:

- `$AGENTDN$`—The DN where the call recording is initiated.
- `$REFCI$`—The CUCM `callId` of the call.
- `$DATE$`—The current date (GMT) in the Y-M-D format.
- `$TIME$`—The current time (GMT) in the H-M-S format.
- `$UUID$`—The call UUID.

recording-filename-pop

Default Value: `always`
Valid Values: `never, always, empty`
Changes Take Effect: Immediately

Specifies how T-Server populates and clears the `UserData` key `GSIP_REC_FN` with `recording-filename`:

- `never`—Never populates or clears the key.
- `always`—Always populates the key with `recording-filename` and clears it upon receiving stop recording from the switch.
- `empty`—Populates the key with `recording-filename` and clears it upon receiving stop recording from the switch only if the key is originally empty.

recording-filename-suffix

Default Value: NULL
Valid Values: String, representing part of filename
Changes Take Effect: Immediately

If specified, the value is appended to the generated filename. Can be set, for example, to `_pcma.wav` on Windows to achieve a constant filename match with Stream Manager.

reg-failed-delay

Default Value: 1000
Valid Values: 0·600000 (ms)
Changes Take Effect: Immediately
Related Option: reg-failed-retries

Specifies the time interval, in milliseconds, that T-Server waits after receiving TermRegistrationFailedEv before trying to reregister.

Warning! Use this option only when requested by Genesys Customer Care.

reg-failed-retries

Default Value: 10
Valid Values: 0–1000
Changes Take Effect: Immediately
Related Option: reg-failed-delay

Specifies the number of times T-Server tries to reregister after receiving TermRegistrationFailedEv, after which it generates log event 51109 and distributes EventDNOutOfService. A value of 0 (zero) means that T-Server does not try to reregister.

Warning! Use this option only when requested by Genesys Customer Care.

ring-tone

Default Value: music/ring_back
Valid Values: Name and path of any valid audio file
Changes Take Effect: Immediately for all new calls

Specifies an audio file to be played for the RingBack treatment.

rtp-info-password

Default Value: Empty string
Valid Values: Any string
Changes Take Effect: Immediately

Specifies the password to allow voice-monitoring applications to monitor any call. If the password is not provided, no applications will be able to perform monitoring.

Warning! This password is not secure. It is passed as text to the voice-monitoring application.

send-backinservice-after-switchover

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether the new primary T-Server sends `EventDNBackInService` messages after a switchover that is caused by a JTAPI link failure that results in `EventDNOutOfService` messages distributed to clients. When the value of this option is set to `true`, the new primary T-Server sends `EventDNBackInService` messages to keep client DNs in the correct states.

Warning! Use this option only when requested by Genesys Customer Care.

set-preferred-original-called-party

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When routing a call, T-Server sets `AttributeDNIS` in subsequent events based on the `AttributeDNIS` specified in the corresponding `TRouteCall` request if the `AttributeRouteType` is `RouteTypeOverwriteDNIS`. Setting this option to `true` causes T-Server to set `AttributeDNIS` regardless of the `AttributeRouteType`. This option value provides backward compatibility for pre-8.0.1 T-Server..

Warning! Use this option only when requested by Genesys Customer Care.

silence-tone

Default Value: `music/silence`
Valid Values: Name and path of any valid audio file
Changes Take Place: Immediately for all new calls

Specifies an audio file to be played for the `Silence` treatment.

sm-port

Default Value: `0`
Valid Values: `0`–`65535`
Changes Take Effect: After restart

Specifies the `listening port` for VoIP Stream Manager or T-Server-CUCM to Media Server Connector connection. If this option is set to `0`, T-Server does not allow Stream Manager or T-Server-CUCM to Media Server Connector connections and ignores all Stream Manager treatments options.

switchover-on-first-link-failure

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

When set to `true`, T-Server reports `LinkConnected` only when all configured links are up. If any link fails, T-Server immediately reports `LinkDisconnected`, causing Solution Control Server to trigger a switchover.

When set to `false`, T-Server reports `LinkConnected` as soon as any one link is up. After startup completion, T-Server sets DNs for any link that failed to out-of-service state, and does not trigger a switchover as long as at least one JTAPI link remains active.

use-called-party-display-name

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: “Called and Calling Parties Display Name” on page 150

If set to `true`, the called party display name appears in the `CALLED_PARTY_DISPLAY_NAME` key-value pair in the `Extensions` attribute of the specific call events.

use-calling-party-display-name

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: “Called and Calling Parties Display Name” on page 150

If set to `true`, the calling party display name appears in the `CALLING_PARTY_DISPLAY_NAME` key-value pair in the `Extensions` attribute of the specific call events.

use-default-route

Default Value: `false`
Valid Values: `false`, `true`
Changes Take Effect: With the next route request
Related Feature: “ACD-like Default Routing” on page 138

Specifies whether to use the default DN (see `default-dn`) as a default route destination if URS (Universal Routing Server) is not connected.

use-external-establish-from-other-link

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When the value is `true`, T-Server generates `EventEstablished` for the originating party of an outbound consultation call if the consultation call

arrives back at an internal DN that is flagged as "external" by JTAPI. Normally, T-Server ignores such JTAPI events.

Warning! Use this option only when requested by Genesys Customer Care.

use-party-display-name

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: Immediately
Related Feature: "Telephone Display Name" on page 150

When set to a value of `true`, T-Server attempts to obtain the display name information for DNs involved in calls, and places it within `AttributeExtensions` of specific call events. A value of `false`, disables this option.

use-ringing-for-net-alerting

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When this option is set to true, on receiving the JTAPI event `CallCtlConnNetworkAlertingEv`, T-Server generates `EventRinging` for external parties. This helps T-Server maintain correct external party representation on outbound calls.

Warning! Use this option only when requested by Genesys Customer Care.

jtapi Section

This section describes configuration options that are unique to this T-Server. Configure these options in the jtapi section on the Options tab for the T-Server `Application` object in the Configuration Layer.

You must call this section `jtapi`.

Note: Any option defined in this section overrides the equivalent option in `jtapi.ini`. See "JTAPI and Configuring JTAPI Options" on page 136. Any option not defined in this section takes its value from `jtapi.ini`; or, if the option is also not defined in `jtapi.ini`, the value is taken from the predefined JTAPI library default value for the option.

AlarmServiceHostname

Default Value: No default value. See Note on page 274.
Valid Values: Any string
Changes Take Effect: After T-Server is restarted

Specifies the host name for the JTAPI alarm service.

AlarmServicePort

Default Value: None. See Note on page 274.
Valid Value: Any integer
Changes Take Effect: After T-Server is restarted

Specifies the port for the JTAPI alarm service.

CTI_DEBUGGING

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, Cisco Unified Communications Manager CTI events are written to the JTAPI log (if logging is enabled).

CTIIMPL_DEBUGGING

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, internal CTI implementation is written to the JTAPI log (if logging is enabled).

CtiManagers

Default Value: No default value. See Note on page 274.
Valid Value: Any string
Changes Take Effect: After T-Server is restarted

Not used by T-Server.

CtiRequestTimeout

Default Value: None. See Note on page 274.
Valid Value: Any integer
Changes Take Effect: After T-Server is restarted

Specified the time, in seconds, during which T-Server waits for a response to a CTI request.

DEBUG

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, debugging information is written to the JTAPI log (if logging is enabled).

Note: For detailed descriptions of all JTAPI options, see the *Cisco Unified Communications Manager Administration Guide.*

DesiredServerHeartbeatInterval

Default Value: No default value. See Note on page 274.
Valid Value: Any integer
Changes Take Effect: After T-Server is restarted

Specifies the time, in seconds, between verification heartbeat messages between T-Server and the Cisco Unified Communications Manager cluster. If T-Server fails to receive heartbeats, it attempts to connect to the backup CTIManager.

Directory

Default Value: No default value. See Note on page 274.
Valid Value: Any string
Changes Take Effect: After T-Server is restarted

Specifies the directory used to store JTAPI log files.

FileNameBase

Default Value: No default value. See Note on page 274.
Valid Value: Any string
Changes Take Effect: After T-Server is restarted

Species the file name prefix to store JTAPI logs.

FileNameExtension

Default Value: No default value. See Note on page 274.
Valid Value: Any string
Changes Take Effect: After T-Server is restarted

Specifies the filename extension to store JTAPI logs.

HUNTLIST_ENABLED

Default value: `true`
Valid value: `true`, `false`
Changes Take Effect: After restart
Related Feature: “Hunt Groups” on page 153

When set to `true`, JTAPI activates Hunt Group support.

INFORMATIONAL

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, status events are written to the JTAPI log (if logging is enabled).

JTAPI_DEBUGGING

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, JTAPI methods and events are written to the JTAPI log (if logging is enabled).

JTAPIIMPL_DEBUGGING

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, JTAPI internal implementation is written to the JTAPI log (if logging is enabled).

MISC_DEBUGGING

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, miscellaneous tracing is written to the JTAPI log (if logging is enabled).

NumTraceFiles

Default Value: No default value. See Note on page 274.
Valid Value: Any integer
Changes Take Effect: After T-Server is restarted

Specifies the number of JTAPI logs to produce before overwriting old logs.

PeriodicWakeupEnabled

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, each JTAPI thread is woken up after a specified sleep interval in order to write a debug message to the JTAPI log (if logging is enabled).

PeriodicWakeupInterval

Default Value: No default value. See Note on page 274.
Valid Value: Any integer
Changes Take Effect: After T-Server is restarted

Specifies the interval, in seconds, between debug thread wakeup.

PROTOCOL_DEBUGGING

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, full CTI protocol trace is written to the JTAPI log (if logging is enabled).

ProviderOpenRequestTimeout

Default Value: No default value. See Note on page 274.
Valid Value: Any integer
Changes Take Effect: After T-Server is restarted

Specifies the amount of time, in seconds, that T-Server waits for a response to a `ProviderOpen` message.

ProviderRetryInterval

Default Value: No default value. See Note on page 274.
Valid Value: Any integer
Changes Take Effect: After T-Server is restarted

Specifies the amount of time, in seconds, that T-Server tries to reopen a connection to a Cisco Unified Communications Manager cluster after a failure.

QueueSizeThreshold

Default Value: No default value. See Note on page 274.
Valid Value: Any integer.
Changes Take Effect: After T-Server is restarted

Specifies the size of the internal JTAPI Queue which, when exceeded, causes a debug message to be written to the JTAPI log (when logging is enabled).

QueueStatsEnabled

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, if the internal JTAPI is greater than the threshold, a debug message is written to the JTAPI log (when JTAPI logging enabled).

RouteSelectTimeout

Default Value: `5000`. See Note on page 274.
Valid Value: Any integer
Changes Take Effect: After T-Server is restarted

Specifies the amount of time, in milliseconds, that T-Server waits for URS to respond to a route request. This setting is also affected by other Cisco Unified Communications Manager switch settings.

SyslogCollector

Default Value: No default value. See Note on page 274.
Valid Value: Any string
Changes Take Effect: After T-Server is restarted

Specifies the host name for a syslog collector.

SyslogCollectorUDPPort

Default Value: No default value. See Note on page 274.
Valid Value: Any integer
Changes Take Effect: After T-Server is restarted

Specifies a UDP port for the syslog collector.

TraceFileSize

Default Value: No default value. See Note on page 274.
Valid Value: An integer divisible by 1024 * 1024
Changes Take Effect: After T-Server is restarted

Specifies the maximum size, in bytes, of each JTAPI log file. The value must be divisible by 1024*1024.
Examples:

- `TraceFileSize=1000`
  - It is the incorrect value. It is not divisible by 1024 * 1024. JTAPI will reset to default 1 MB.
- `TraceFileSize=10485760`
  - It is the correct value. It is divisible by 1024 * 1024. 10,485,760 bytes = 10 MB

TracePath

Default Value: No default value. See Note on page 274.
Valid Value: Any character string
Changes Take Effect: After T-Server is restarted

Specifies the root directory used to store JTAPI log files.

TServerTraceFileBase

Default Value: `ts_trace`
Valid Values: Any character string
Changes Take Effect: After T-Server is restarted
Related Feature: "Logging of Network Connection Failures Between JTAPI and T-Server" on page 153

Determines the prefix of the log file name in a situation where the network connection fails between JTAPI and T-Server.

TServerTraceFileExt

Default Value: `log`
Valid Values: Any character string
Changes Take Effect: After T-Server is restarted
Related Feature: "Logging of Network Connection Failures Between JTAPI and T-Server" on page 153

Determines the suffix of the log file name in a situation where the network connection fails between JTAPI and T-Server.

TServerTraceMaxFiles

Default Value: `10`
Valid Values: `0`-Maximum Integer
Changes Take Effect: After T-Server is restarted
Related Feature: "Logging of Network Connection Failures Between JTAPI and T-Server" on page 153

Determines how many files should be in a created before the first one is overwritten for the files created with the `TServerTraceFileBase`, and `TServerTraceFileExt` options.

TServerTraceMaxFileSize

Default Value: `100`
Valid Values: `0` - Maximum Integer
Changes Take Effect: After T-Server is restarted
Related Feature: "Logging of Network Connection Failures Between JTAPI and T-Server" on page 153

Determines the maximum file size in bytes for individual files created with the `TServerTraceFileBase`, and `TServerTraceFileExt` options.

UseAlarmService

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, JTAPI alarms go to an alarm service on the specified host and port.

UseTraceFile

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, JTAPI log files are written.

UseJavaConsoleTrace

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, JTAPI log messages are sent to the console (stdout).

UseSameDirectory

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, all instances of the same application write JTAPI logs to the same directory.

UseSyslog

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, JTAPI log traces go to the syslog collector service at the specified UDP port and host.

WARNING

Default Value: No default value. See Note on page 274.
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

When set to `true`, low-level warning events are written to the JTAPI log (if logging is enabled).

globalgroup Section

This section describes the configuration options that are unique to this T-Server. Configure these options in the `globalgroup` section on the `Options` tab for the T-Server `Application` object in the Configuration Layer.

You must call this section `globalgroup`.

address-query-delay

Default Value: `0`
Valid Values: `0 to 3600000`
Changes Take Effect: Immediately

Specifies the amount of time T-Server waits after receiving an `EventDNBackInService` message before sending a `RequestQueryAddress` query for the DN. This option is used by T-Server to cleanup up old calls after a link disconnection/reconnection. When set to a value of `0`, this cleanup functionality is disabled.

Note: When using this option, a sufficient delay is required to allow CUCM sufficient time to reconstruct any existing calls.

callmgr-autopickup-on

Default Value: `false`
Valid Values: `false, true`
Changes Take Effect: Immediately

This option should be set to match the Cisco Unified Communications Manager Pickup mode: Auto Pickup (`true`) or Regular Pickup (`false`).

enable-jtapi-keep-alive

Default Value: `false`
Valid Values: `false, true`
Changes Take Effect: Immediately

This option enables a heartbeat between T-Sever and JTAPI as a keep alive functionality to detect possible conditions when JTAPI becomes unstable, and the JTAPI event sequence is no longer reliable.

ignore-cisco-cause-500

Default value: `true`
Valid Values: `true`, `false`
Changes Take Effect: After restart
Related Feature: “Hunt Groups” on page 153

When set to `true`, T-Server removes extra messages that are generated during conference and transfer calls.

java-home

Default Value: No default value
Valid Values: Any string
Changes Takes Effect: After link restart

Specifies the Windows Java Runtime Environment location in the same format as the `JAVA_HOME` environment variable. For example: “`C:\Program Files\Java\jdk1.8.0_6`”. T-Server automatically appends “`\bin\java`”. This option takes precedence over the environment variable `JAVA_HOME`.

jtapi-keep-alive-retries

Default Value: 10
Valid Values: Any positive integer from 1 to 999
Changes Take Effect: Immediately

Specifies how many skipped heartbeat attempts are permitted before T-Server will shut itself down. T-Server sends the STANDARD message JTAPI keepalive timer exceeded %d allowed retries. Exiting now. before exiting.

jtapi-keep-alive-timeout

Default Value: 60000
Valid Values: Any positive integer from 1000 to 3600000
Changes Take Effect: Immediately

Specifies the time (in milliseconds) between keep alive attempts. Any event from JTAPI is also considered as a keep alive message. T-Server will send the STANDARD message JTAPI keepalive timer expired. Retry %d of %d. when there are no messages within the timeout interval.

party-changed-from-external-release

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

If set to true, T-Server uses the JTAPI CallCtlConnDisconnectedEv to trigger the change in external originator digits if the cause is REDIRECTED. T-Server distributes EventPartyChanged with the new originating digits in AttributeOtherDN.

Warning! Use this option only when requested by Genesys Customer Care.

link Section

Configure these options in the link section on the Options tab for the T-Server Application object in the Configuration Layer.

Starting with release 7.6, T-Server operating in Socket mode uses a standard link-*n*-name link configuration option. The TServer section will contain options link-1-name, link-2-name, and so on, while separate sections with names equal to the link-*n*-name values will contain link parameters.

Warning! Changes to any of the options in this section will cause T-Server to restart the corresponding link.

ccm-host

Default Value: Mandatory field. No default value.
Valid Values: Any valid IP address.
Changes Take Effect: Changing this option will cause the link to restart

Specifies the IP address for the CTI manager for this account.

hostname

Default Value: `localhost`
Valid Values: Any valid host name
Changes Take Effect: Read-only value. Do not change

Specifies the host of the link according to the switch configuration. This should always be `localhost`.

password

Default Value: Mandatory field. No default value.
Valid Values: Any valid password.
Changes Take Effect: Changing this option will cause the link to restart

Specifies the CUCM password for this link.

port

Default Value: Mandatory field. No default value.
Valid Values: Any valid port address
Changes Take Effect: Changing this option will cause the link to restart

Specifies the TCP/IP port on the `localhost` that the Java link is opening to listen to.

protocol

Default Value: `tcp`
Valid Values: `tcp`
Changes Take Effect: Changing this option will cause the link to restart

Specifies the connection protocol T-Server uses in communicating with the switch.

tls-auth-code

Default Value: NULL
Valid Values: Any valid characters
Changes Take Effect: Changing this option will cause the link to restart

Specifies the authorization string configured in Cisco UCM. This code is used only once for client certificate download.

tls-instance-id

Default Value: NULL
Valid Values: Any valid characters
Changes Take Effect: Changing this option will cause the link to restart

Specifies the application instance ID, as configured on the switch side (Cisco UCM). Each TLS link requires a unique ID.

user-login

Default Value: Mandatory field. No default value.
Valid Values: Any valid user login.
Changes Take Effect: Changing this option will cause the link to restart

Specifies the CUCM user for this link.

link-tls Section

This section describes configuration options that are unique to this T-Server. Configure these options in the `link-tls` section on the `Options` tab for the T-Server `Application` object in the Configuration Layer.

You must call this section `link-tls`.

password

Default Value: NULL
Valid Values: Any valid characters
Changes Take Effect: After restart

Specifies a passphrase used to encrypt the local key store for certificates.

tls-capf-host

Default Value: NULL
Valid Values: Any valid address
Changes Take Effect: After restart

Specifies the hostname or IP address of the Cisco UCM CAPF server. Defined by the switch configuration.

tls-capf-port

Default Value: NULL
Valid Values: Any valid port
Changes Take Effect: After restart

Specifies the port number on which the CAPF server is running. Defined by the switch configuration (typically defaults to 3804).

tls-cert-path

Default Value: NULL
Valid Values: Any valid local path
Changes Take Effect: After restart

Specifies the local directory path where certificates should be installed.

tls-tftp-host

Default Value: NULL
Valid Values: Any valid characters
Changes Take Effect: After restart

Specifies the hostname or IP address of the Cisco UCM TFTP server.

tls-tftp-port

Default Value: NULL
Valid Values: Any valid port
Changes Take Effect: After restart

Specifies the port number on which the TFTP server is running. Defined by the switch configuration (typically defaults to 69).

Agent Login-Level Options

You set configuration options described in this section in the `TServer` section on the `Options/Annex` tab of the relevant `Agent Login` object in the Configuration Layer.

agent-no-answer-action

Default Value: `none`
Valid Values: `none, notready, walkaway, logout`
Changes Take Effect: Immediately
Related Feature: “Redirect On No Answer” on page 162

Specifies the agent state to which T-Server will be set after the time period for the `agent-no-answer-timeout` option has expired. A value of `none` means that the agent will remain in its current state.

Note: This option can also be configured at the Application level. The option setting at the Agent Login level takes precedence.

agent-no-answer-overflow

Default Value: `none`
Valid Values:

`none`	The call will remain ringing on the agent phone.
`recall`	The call will be redirected back to the Routing Point or the ACD Queue that delivered the call to the agent.
Any destination digits	A valid destination DN must be provided (a Queue or Routing Point on the local switch is recommended).

Changes Take Effect: Immediately
Related Feature: “Redirect On No Answer” on page 162

After the time period for the `agent-no-answer-timeout` option has expired, T-Server will redirect the ringing call to the destination described by this option.

Note: This option can also be configured at the Application level. The option setting at the Agent Login level takes precedence.

agent-no-answer-timeout

Default Value: `0`
Valid Values: `0 to 600`
Changes Take Effect: Immediately
Related Feature: “Redirect On No Answer” on page 162

Calls ringing on an agent’s phone that were distributed from an ACD Queue or a Routing Point will wait for the telephone to ring for this timeout period (in seconds) before performing the actions described by the `agent-no-answer-action` option and redirect the call to the destination described in the `agent-no-answer-overflow` option. The default value of `0` (zero) disables the functionality of this option.

Note: This option can also be configured at the Application level. The option setting at the Agent Login level takes precedence.

DN-Level Options

You set configuration options described in this section in the `TServer` section on the `Annex` tab of the relevant `DN` object You cannot define them in the T-Server `Application` object.

intercomDN

Default Value: No default value
Valid Values: Any valid DN name
Changes Take Effect: Immediately
Related Feature: “Whisper Coaching and Extra Instance of Intercom Call” on page 173

Specifies the name for the Intercom DN.

moh-server-music

Default Value: No default value
Valid Values: A string
Changes Take Effect: For the next call
Related Feature: “Music and Announcements” on page 154

When specified, T-Server uses Cisco Unified Communications Manager Music On Hold Server instead of Stream Manager or Media Server to play music treatment if the name of the music file, as defined in the strategy, is equal to the value of this option. For all other music files Stream Manager or Media Server will be used. This option is set on Routing Point DNs.

queue-music

Default Value: No default value
Valid Value: Name and path of any valid audio file in the following format:

`<directory/music file name>`

where:

- `<directory>` is a sub-directory of the `Stream Manager` or Media Server root directory
- `<music file name>` is the name of the file, without the codec extension

Changes Take Effect: Immediately
Related Feature: “Music Treatment on ACD Queues” on page 156
Specifies the name of the file for music treatment of ACD Queues.

record

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When this option is set to `true`, recording will be started automatically when the call is established on the corresponding DN. Call recording will be stopped when this DN leaves the call.

Changes from 8.0 to 8.1

Table 20 provides all configuration option changes for this T-Server between release 8.0 and the latest 8.1 release.

Table 20: T-Server Option Changes from 8.0 to 8.1

Option Name	Type of Change	Details
globalgroup Section		
address-query-delay	New in 8.1.0	See page 282 for details.
java-home	New in 8.1.2	See page 282 for details.
party-changed-from-external-release	New in 8.1.2	See page 283 for details.
TServer Section		
call-forward-action	New in 8.1.1	See page 261 for details.
clean-calls-on-all-links-up	New in 8.1.2	See page 262 for details.
delay-dialing	New in 8.1.2	See page 264 for details.
free-form-terminal-id	New in 8.1.2	See page 265 for details.
external-dn-length	New in 8.1.1	See page 265 for details.
force-moh-on-ms-down	New in 8.1.2	See page 265 for details.
out-of-service-action	New in 8.1.2	See page 268 for details.
out-of-service-action-delay	New in 8.1.2	See page 268 for details.
recording-filename-pop	New in 8.1.2	See page 270 for details.
reg-failed-delay	New in 8.1.2	See page 271 for details.
reg-failed-retries	New in 8.1.2	See page 271 for details.

Table 20: T-Server Option Changes from 8.0 to 8.1 (Continued)

Option Name	Type of Change	Details
set-preferred-original-called-party	New in 8.1.1	See page 272 for details.
use-party-display-name	New in 8.1.0	See page 274 for details.
use-called-party-display-name	New in 8.1.1	See page 273 for details.
use-calling-party-display-name	New in 8.1.1	See page 273 for details.
use-external-establish-from-other-link	New in 8.1.2	See page 273 for details.
use-ringing-for-net-alerting	New in 8.1.2	See page 274 for details.
link Section		
tls-auth-code	New in 8.1.2	See page 284 for details.
tls-instance-id	New in 8.1.2	See page 285 for details.
link-tls Section		
password	New in 8.1.2	See page 285 for details.
tls-cert-path	New in 8.1.2	See page 286 for details.
tls-capf-host	New in 8.1.2	See page 285 for details.
tls-capf-port	New in 8.1.2	See page 285 for details.
tls-tftp-host	New in 8.1.2	See page 286 for details.
tls-tftp-port	New in 8.1.2	See page 286 for details.

Chapter

14 Stream Manager Configuration

This chapter describes the configuration options that are used for configuring Stream Manager to work with T-Server for Cisco Unified Communications Manager. It contains the following sections:

Options that are common to all T-Servers are described in Chapter 11, "Common Configuration Options," on page 207 and in Chapter 12, "T-Server Common Configuration Options," on page 229.

Note: Starting with release 8.1.1, Media Server is supported for media service by the T-Server for CUCM, in addition to Stream Manager. See the *Genesys Media Server 8.1 Deployment Guide* for more detail on configuring Media Server.

Stream Manager Configuration Options with T-Server

Stream Manager is a Genesys client application that streams media files in order to provide announcements and music to callers queued on Routing Points and ACD Queues (please refer to "Music and Announcements" on page 154 of this document).

In order to enable the connections, configure the T-Server option `sm-port` in the T-Server application (refer to page 272 for more information).

To configure Stream Manager to work with T-Server for Cisco Unified Communications Manager, select the Stream Manager application object in the Configuration Layer and then add a connection to the T-Server application.

Multiple Stream Managers can connect to one T-Server. In this scenario, T-Server distributes calls to all connected Stream Managers in a load-balanced arrangement.

Multiple Stream Managers can be deployed in a load-balancing configuration, (though not in a primary/backup configuration). This configuration provides the benefit of *N+1* availability. In the event of Stream Manager failure, the call is relocated to another Stream Manager and the treatment is restarted there, ensuring that no calls are lost.

Configuring Stream Manager to Control a Routing Point

The following configuration enables a Routing Point to be controlled by a particular Stream Manager. It works with Cisco Unified Communications Manager 4.0 or later.

1. On the `Options` tab in your `Stream Manager` application, create a section named `TServer`, and in this section create an option called `smloc`. You can set any string value for this option.
2. On the `Annex` tab of your `Routing Point/ACD Queue`, also create a section named `TServer`, and in this section create an option called `smloc`. The value must be the same as the value specified previously in the Stream Manager application configured above.

For example, if a call arrives at a Routing Point or ACD Queue, and service from Stream Manager is required, the `smloc` value as configured above is retrieved.

However, note that:

- If there are multiple Stream Managers with the same `smloc` value, the call leg is created on the least busy of these Stream Managers.
- If there are no Stream Managers found in the Configuration Layer, but none of them share the same `smloc` value, the leg is created on the least busy of all the Stream Managers, regardless of the assigned `smloc` value.
- If the Routing Point has no `smloc` value, the call leg is created on the least busy of all Stream Managers, regardless of the smloc values assigned to the Stream Managers.

Audio File Formats for Stream Manager

Stream Manager must be able to access the audio files that T-Server requests to play. These files are located in subdirectories of the installed Stream Manager root directory. The files must be in the appropriate codec format, with a filename suffix corresponding to the codec type. Files for Stream Manager on a UNIX platform must have the extension `.au`. If you are using a Windows

platform, files for Stream Manager must have the extension `.wav`. The supported formats and file suffixes are:

- G.711 mu-law: `mulaw.au, mulaw.wav`
- G.711 a-law: `alaw.au, alaw.wav`
- G.723: `_g7231.au, _g7231.wav`
- G.729A: `_g729a.au, _g729a.wav`
- GSM: `_gsm.au, _gsm.wav`

To facilitate the use of the full range of supported codecs, Genesys has made Audio Transcoding Utility (ATU) available through Genesys Customer Care. ATU takes as input an audio file in any format supported by Stream Manager and produces versions of the file in all supported formats.

When these files are produced, ensure that all Stream Managers connected to a T-Server contains the same announcement/music files in the same directory structure. T-Server will instruct the Stream Manager to play a file, but if the file does not exist in the specified directory, the treatment will fail.

Stream Manager Configuration Options

Starting with T-Server 7.6, Stream Manager options are now grouped into the following sections: `codecs`, `contact`, `limits`, `log`, and `x-config`. The updated options descriptions for these sections are located in the *Framework 7.6 Stream Manager Deployment Guide*.

None of the options for Stream Manager configuration are mandatory.

Supplements

Related Documentation Resources

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

T-Server for Cisco Unified Communications Manager

- The Release Notes and Product Advisories for this product, which are available on the Genesys Documentation website.

Management Framework

Consult these additional resources as necessary:

- The *Framework 8.1 Deployment Guide*, which will help you configure, install, start, and stop Framework components.
- The *Framework 8.1 Configuration Manager Help*, which describes how to use Configuration Manager in either an enterprise or multi-tenant environment.
- The *Framework 8.1 Genesys Administrator Help,* which describes how to use Genesys Administrator in either an enterprise or multi-tenant environment.
- The *Framework 8.1 Configuration Options Reference Manual*, which will provide you with descriptions of configuration options for other Framework components.

Platform SDK

- The *Genesys Events and Models Reference Manual*, which contains an extensive collection of events and call models describing core interaction processing in Genesys environments.
- The *Platform SDK 8.x .NET (or Java) API Reference*, which contains technical details of T-Library functions.

Genesys

- The *Genesys Events and Models Reference Manual*, which contains the T-Library API, information on `TEvents`, and an extensive collection of call models.
- *Genesys Technical Publications Glossary,* which provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms used in this document.
- *Genesys Migration Guide*, which provides documented migration strategies for Genesys product releases. Contact Genesys Customer Care for more information.

Information about supported operating systems and third-party software is available on the Genesys Documentation website in the following documents:

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

Consult the following additional resources as necessary:

- *Genesys Hardware Sizing Guide,* which provides information about Genesys hardware sizing guidelines for the Genesys 8.x releases.
- *Genesys Interoperability Guide,* which provides information on the compatibility of Genesys products with various Configuration Layer Environments; Interoperability of Reporting Templates and Solutions; and Gplus Adapters Interoperability.
- *Genesys Licensing Guide,* which introduces you to the concepts, terminology, and procedures that are relevant to the Genesys licensing system.
- *Genesys Database Sizing Estimator 8.x Worksheets,* which provides a range of expected database sizes for various Genesys products.

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

Genesys product documentation is available on the:

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

Document Conventions

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

Document Version Number

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

80fr_ref_06-2008_v8.0.001.00

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

Screen Captures Used in This Document

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

Type Styles

Table 21 describes and illustrates the type conventions that are used in this document.

Table 21: Type Styles

Type Style	Used For	Examples
Italic	• Document titles • Emphasis • Definitions of (or first references to) unfamiliar terms • Mathematical variables Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on page 298).	Please consult the *Genesys Migration Guide* for more information. Do *not* use this value for this option. A *customary and usual* practice is one that is widely accepted and used within a particular industry or profession. The formula, $x + 1 = 7$ where x stands for . . .
Monospace font (Looks like `teletype` or `typewriter text`)	All programming identifiers and GUI elements. This convention includes: • The *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages. • The values of options. • Logical arguments and command syntax. • Code samples. Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.	Select the `Show variables on screen` check box. In the `Operand` text box, enter your formula. Click `OK` to exit the `Properties` dialog box. T-Server distributes the error messages in `EventError` events. If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls. Enter `exit` on the command line.
Square brackets ([ ])	A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.	`smcp_server -host [/flags]`
Angle brackets (< >)	A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise. **Note:** In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.	`smcp_server -host <confighost>`

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X