



Framework 8.1

T-Server for NEC NEAX/APEX

Deployment Guide

The information contained herein is proprietary and confidential and cannot be disclosed or duplicated without the prior written consent of Genesys Telecommunications Laboratories, Inc.

Copyright © 1991–2014 Genesys Telecommunications Laboratories, Inc. All rights reserved.

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 Program Guide](#) for complete contact information and procedures.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the [Genesys Licensing Guide](#).

Released by

Genesys Telecommunications Laboratories, Inc. www.genesys.com

Document Version: 81fr_dep-ts_nec_07-2014_v8.1.002.00



Table of Contents

List of Procedures	9
Preface	11
	About T-Server for NEC NEAX/APEX	11
	Intended Audience.....	12
	Reading Prerequisites	13
	Making Comments on This Document	13
	Contacting Genesys Customer Care.....	13
	Document Change History	13
	New in Document Version 8.1.002.00	13
	New in Document Version 8.1.001.02	14
Part 1	T-Server Deployment	15
	New for All T-Servers in 8.1	15
Chapter 1	T-Server Fundamentals.....	17
	Learning About T-Server	18
	Framework and Media Layer Architecture	18
	T-Server Requests and Events	20
	Advanced Disconnect Detection Protocol	23
	Redundant T-Servers	24
	Multi-Site Support	28
	Agent Reservation	28
	Client Connections	29
	Next Steps	29
Chapter 2	T-Server General Deployment.....	31
	Prerequisites.....	31
	Software Requirements	31
	Hardware and Network Environment Requirements	32
	Licensing Requirements	33

	About Configuration Options.....	35
	Deployment Sequence	36
	Deployment of T-Server.....	36
	Configuration of Telephony Objects	36
	Configuration of T-Server	39
	Installation of T-Server	41
	Next Steps	43
Chapter 3	High-Availability Deployment.....	45
	Warm Standby Redundancy Type	46
	Hot Standby Redundancy Type	47
	Prerequisites.....	49
	Requirements.....	49
	Synchronization Between Redundant T-Servers	49
	Warm Standby Deployment.....	50
	General Order of Deployment.....	50
	Modification of T-Servers for Warm Standby	51
	Warm Standby Installation of Redundant T-Servers	52
	Hot Standby Deployment.....	52
	General Order of Deployment.....	52
	Modification of T-Servers for Hot Standby	53
	Hot Standby Installation of Redundant T-Servers	56
	Next Steps	56
Chapter 4	Multi-Site Support.....	57
	Multi-Site Fundamentals.....	58
	ISCC Call Data Transfer Service	59
	ISCC Call Flows.....	60
	ISCC Transaction Types	66
	T-Server Transaction Type Support.....	74
	Transfer Connect Service Feature.....	78
	ISCC/Call Overflow Feature	79
	Number Translation Feature.....	83
	Number Translation Rules	84
	Network Attended Transfer/Conference Feature.....	91
	Event Propagation Feature.....	93
	User Data Propagation	94
	Party Events Propagation	95
	Switch Partitioning	96
	Event Propagation Configuration	97
	ISCC Transaction Monitoring Feature	100
	Configuring Multi-Site Support.....	100

	Applications	101
	Switches and Access Codes	102
	DNs	108
	Configuration Examples	113
	Next Steps	114
Chapter 5	Starting and Stopping T-Server Components	115
	Command-Line Parameters	115
	Starting and Stopping with the Management Layer	117
	Starting with Startup Files	118
	Starting Manually	119
	HA Proxy	122
	T-Server	123
	Verifying Successful Startup	125
	Stopping Manually	125
	Starting and Stopping with Windows Services Manager	126
	Next Steps	126
Part 2	T-Server Configuration	127
	New in T-Server for NEC NEAX/APEX	127
Chapter 6	Switch-Specific Configuration	131
	Known Limitations	131
	Known Limitations for the Attendant Console	133
	Switch Platform General Information	133
	Large-System Switches	134
	Small-System Switches	136
	Setting DN Properties	138
	DN Type Cross-Reference Between NEC and Genesys	146
	Configuring Universal Routing Server	148
	Configuring URS with Redundancy	148
	Configuring Trunks	150
	Configuring External Trunks	150
	Configuring Announcement Trunks	150
	Support for NEC ACD Agent Break Modes	151
	Configuring Dedicated ACD Supervisors	151
	Configuring External ACD Connection	152
	Switch Configuration Requirements	153
	Switch-Based High Availability	154
	Warm and Hot Standby with Two CTI Links	154

	Configuring Unlicensed Devices.....	155
Chapter 7	Supported T-Server Features	159
	T-Library Functionality	159
	Agent Work Modes	168
	Call Parking	168
	Feature Configuration	169
	Call Pickup.....	169
	Force Function Modifier.....	170
	Multi-ACD Processor Support on a FCCS Network	171
	Multi-Line Operations	171
	Feature Configuration	172
	RTP Stream Monitoring API	172
	Switch Configuration Requirements.....	173
	Feature Configuration	173
	Using the RTP Stream Monitoring API.....	173
	SCF Calling Party Number Selection with OAI Requests.....	175
	Handling SCF OAI Requests	176
	Feature Specific Conditions	177
	Switch Configuration Requirements.....	178
	Feature Configuration	178
	T-Server Error Messages	178
Chapter 8	Common Configuration Options.....	183
	Setting Configuration Options.....	183
	Mandatory Options	184
	log Section.....	184
	Log Output Options.....	190
	Examples	194
	Debug Log Options	195
	log-extended Section.....	198
	log-filter Section.....	200
	log-filter-data Section.....	200
	security Section	201
	sml Section	201
	common Section.....	203
	Changes from 8.0 to 8.1	203
Chapter 9	T-Server Common Configuration Options	205
	Setting Configuration Options.....	205
	Mandatory Options	206

	TServer Section	206
	license Section	211
	agent-reservation Section.....	214
	extrouter Section	215
	ISCC Transaction Options	217
	Transfer Connect Service Options.....	221
	ISCC/COF Options	222
	Event Propagation Options.....	224
	Number Translation Option.....	225
	GVP Integration Option.....	226
	backup-sync Section	226
	call-cleanup Section	228
	Translation Rules Section.....	229
	security Section	230
	Timeout Value Format	230
	Changes from Release 8.0 to 8.1	231
Chapter 10	T-Server-Specific Configuration Options	233
	Mandatory Options	233
	TServer Section.....	235
	Obsolete Options	258
	PPN-Mapping Section	259
	CTI-Link Section	260
	Changes from Release 8.0 to 8.1	261
Supplements	Related Documentation Resources	263
	Document Conventions	265
Index	267



List of Procedures

Configuring T-Server	39
Configuring multiple ports	40
Installing T-Server on UNIX	41
Installing T-Server on Windows	42
Verifying the installation of T-Server.	43
Modifying the primary T-Server configuration for warm standby	51
Modifying the backup T-Server configuration for warm standby	52
Modifying the primary T-Server configuration for hot standby	53
Modifying the backup T-Server configuration for hot standby	55
Activating Transfer Connect Service	79
Configuring Number Translation.	91
Activating Event Propagation: basic configuration	98
Modifying Event Propagation: advanced configuration	98
Configuring T-Server Applications	101
Configuring Default Access Codes.	103
Configuring Access Codes	104
Configuring access resources for the route transaction type	108
Configuring access resources for the dnis-pool transaction type	110
Configuring access resources for direct-* transaction types	110
Configuring access resources for ISCC/COF.	111
Configuring access resources for non-unique ANI.	111
Modifying DNs for isolated switch partitioning	112
Configuring T-Server to start with the Management Layer.	117
Starting T-Server on UNIX with a startup file	118
Starting T-Server on Windows with a startup file	119
Starting HA Proxy on UNIX manually	123
Starting HA Proxy on Windows manually.	123
Starting T-Server on UNIX manually	124
Starting T-Server on Windows manually	124

Stopping T-Server on UNIX manually	125
Stopping T-Server on Windows manually	125
Configuring the External ACD Connection	152
Configuring unlicensed devices	156



Preface

Welcome to the *Framework 8.1 T-Server for NEC NEAX/APEX 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 NEC NEAX/APEX. 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.

This document is valid only for the 8.1 release of this product.

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

This preface contains the following sections:

- [About T-Server for NEC NEAX/APEX, page 11](#)
- [Intended Audience, page 12](#)
- [Making Comments on This Document, page 13](#)
- [Contacting Genesys Customer Care, page 13](#)
- [Document Change History, page 13](#)

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on [page 263](#).

About T-Server for NEC NEAX/APEX

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 5.1 for NEC NEAX/APEX
- T-Server for NEC

The current name is T-Server for NEC NEAX/APEX.

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, 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 complete information on the T-Server events, call models, and requests.

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*. The *Framework 8.1 Deployment Guide* 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 Program Guide](#) for complete contact information and procedures.

Document Change History

New in Document Version 8.1.002.00

This version of the *Framework 8.1 T-Server for NEC NEAX/APEX Deployment Guide* has been updated with the following topics:

New Configuration Options

Added descriptions of the following configuration options:

- alive-message-tolerance, [page 235](#)
- enable-answer-term-info, [page 237](#)
- multi-observing-enabled, [page 247](#)
- report-dterm-ip-addr, [page 250](#)

**Documentation
Updates to
Support New
Features and
Software
Modifications**

- rtp-info-password, [page 251](#)
- scf-calling-nbr-encoding, [page 252](#)
- “New in T-Server for NEC NEAX/APEX” on [page 127](#) summarizes the feature enhancements that NEC NEAX/APEX provides.
- Minor correction in Table 19, “Setting the DN Properties for the NEC NEAX/APEX Switch,” on [page 138](#) to indicate the addition of the SIP Phone to the Personal Handyphone extension.
- [Procedure: Configuring unlicensed devices](#), on [page 156](#) was revised to add new information.
- A new subsection, “Call Parking” on [page 168](#), was added to support the business features on the NEC UNIVERGE SV8500 switch.
- A new subsection, “Call Pickup” on [page 169](#), was added to support the business features on the NEC UNIVERGE SV8500 switch.
- A new subsection, “Multi-Line Operations” on [page 171](#), was added to support the business features on the NEC UNIVERGE SV8500 switch.
- A new subsection, “RTP Stream Monitoring API” on [page 172](#), was added to support the standard Genesys RTP stream monitoring API.
- A new subsection, “SCF Calling Party Number Selection with OAI Requests” on [page 175](#), was added to specify the calling party number.
- Added description of the epn key in the Extensions attribute in the [Procedure: Modifying DNs for isolated switch partitioning](#), on [page 112](#).
- Modified RequestRouteCall to TRouteCall in the Note section for [Procedure: Activating Transfer Connect Service](#), on [page 79](#).

New in Document Version 8.1.001.02

This version of the *Framework 8.1 T-Server for NEC NEAX/APEX Deployment Guide* has been updated with the following topic:

- The propagated-call-type configuration option is correctly documented in the TServer section.



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 17](#), 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 31](#), presents configuration and installation procedures for all T-Servers.
- Chapter 3, “High-Availability Deployment,” on [page 45](#), addresses high availability (HA).
- Chapter 4, “Multi-Site Support,” on [page 57](#), details the variations available for T-Server implementations across geographical locations.
- Chapter 5, “Starting and Stopping T-Server Components,” on [page 115](#), 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 206](#) for details.

- T-Server now supports the Unresponsive Process Detection feature. The following configuration options enable this feature:
 - “heartbeat-period” on [page 201](#)
 - “hangup-restart” on [page 202](#)

For more information, refer to the *Framework 8.0 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 Acrezzo 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 231](#).

- 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.



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 15](#), 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, page 18](#)
- [Advanced Disconnect Detection Protocol, page 23](#)
- [Redundant T-Servers, page 24](#)
- [Multi-Site Support, page 28](#)
- [Agent Reservation, page 28](#)
- [Client Connections, page 29](#)
- [Next Steps, page 29](#)

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.

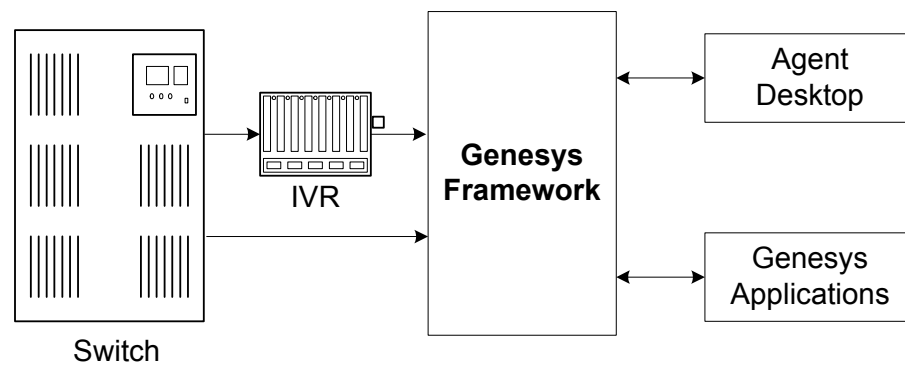


Figure 1: Framework in a Genesys Solution

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

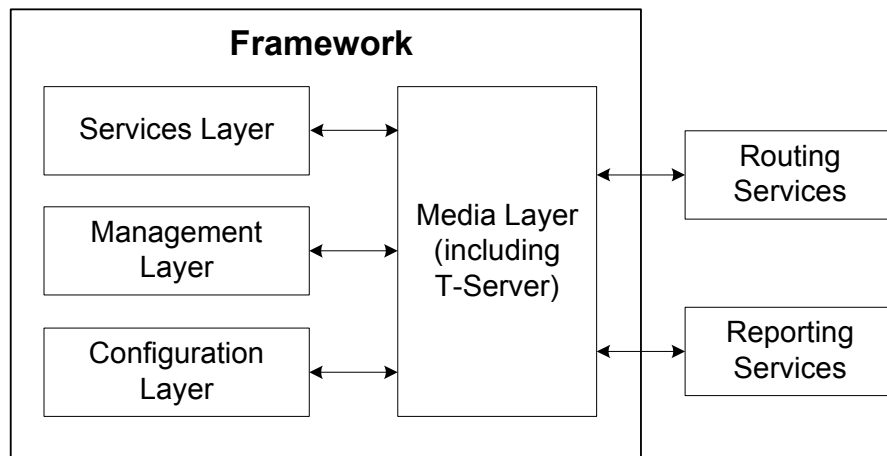


Figure 2: The Media Layer in the Framework Architecture

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.

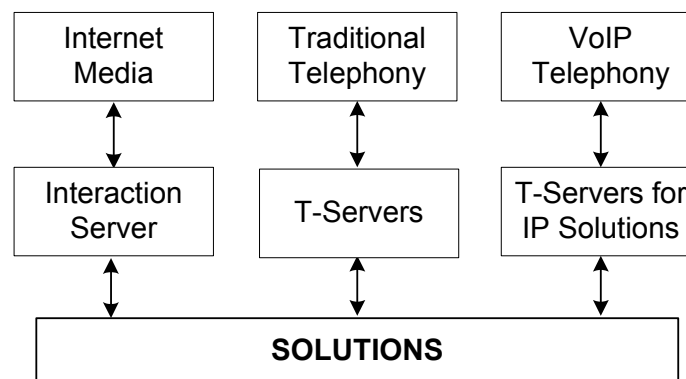


Figure 3: Media Layer Architecture

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 portion of the *Voice 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 19](#) (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.

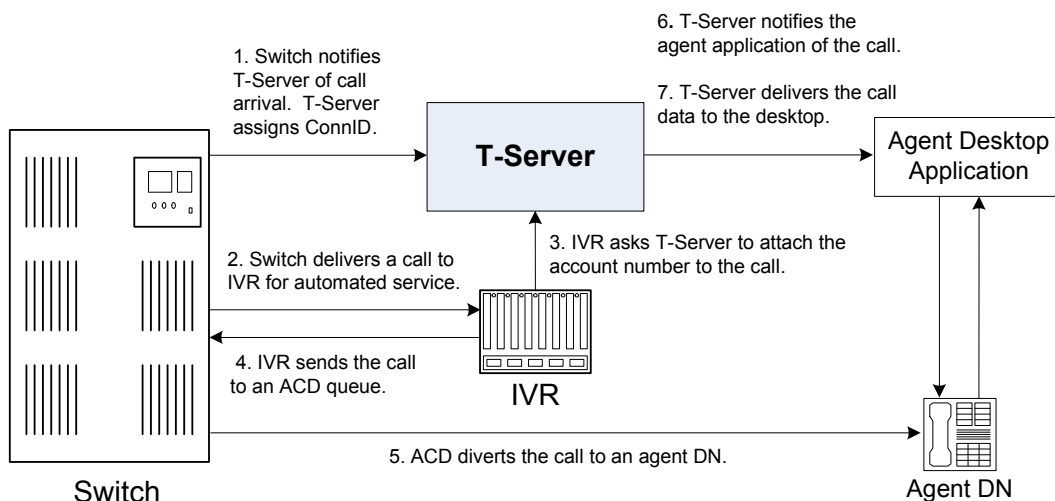


Figure 4: Functional T-Server Steps

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 Media Interfaces Reference Manual](#).

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	—

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
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	—
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	—	—

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
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 SCA114 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 57](#).

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 59](#)), 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 66](#) 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 *Voice 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 “agent-reservation Section” on [page 214](#) 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 214](#)).

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

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 31](#). Otherwise, you may want to jump to Part Two of this *Deployment Guide*, where you will find information about your specific T-Server.



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:

- [Prerequisites, page 31](#)
- [Deployment Sequence, page 36](#)
- [Deployment of T-Server, page 36](#)
- [Next Steps, page 43](#)

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 6.x, 7.x, and 8.x.

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.

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 9, “T-Server Common Configuration Options,” on [page 205](#). *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 <i>Framework 8.1 Deployment Guide</i> for details.
2. Deploy Network objects (such as Host objects).	See the <i>Framework 8.1 Deployment Guide</i> for details.
3. Deploy the Management Layer.	See the <i>Framework 8.1 Deployment Guide</i> for details.
4. Deploy T-Server.	See “Deployment of T-Server” on page 36 .
5. Test your configuration and installation.	See Chapter 5, “Starting and Stopping T-Server Components,” on page 115 .

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 57](#), 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 100](#), 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 57](#).

End of procedure

Next Steps

- See “Installation of T-Server” on [page 41](#).

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 39](#) 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 43](#).
- To test your configuration and installation, go to Chapter 5, “Starting and Stopping T-Server Components,” on [page 115](#), and try it out.
- To configure and install redundant T-Servers, see Chapter 3, “High-Availability Deployment,” on [page 45](#).
- To install T-Servers for a multi-site environment, proceed to Chapter 4, “Multi-Site Support,” on [page 57](#).

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 39](#) 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 43](#).
- To test your configuration and installation, go to Chapter 5, “Starting and Stopping T-Server Components,” on [page 115](#), and try it out.
- To configure and install redundant T-Servers, see Chapter 3, “High-Availability Deployment,” on [page 45](#).
- To install T-Servers for a multi-site environment, proceed to Chapter 4, “Multi-Site Support,” on [page 57](#).

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 41](#)
- [Procedure: Installing T-Server on Windows](#), on [page 42](#)

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 115](#), and try

it out. Otherwise, if you want to configure and install redundant T-Servers, see Chapter 3, “High-Availability Deployment,” on [page 45](#). If you want to install T-Servers for a multi-site environment, proceed to Chapter 4, “Multi-Site Support,” on [page 57](#).

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 25](#) 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, page 46](#)
- [Hot Standby Redundancy Type, page 47](#)
- [Prerequisites, page 49](#)
- [Warm Standby Deployment, page 50](#)
- [Hot Standby Deployment, page 52](#)
- [Next Steps, page 56](#)

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.

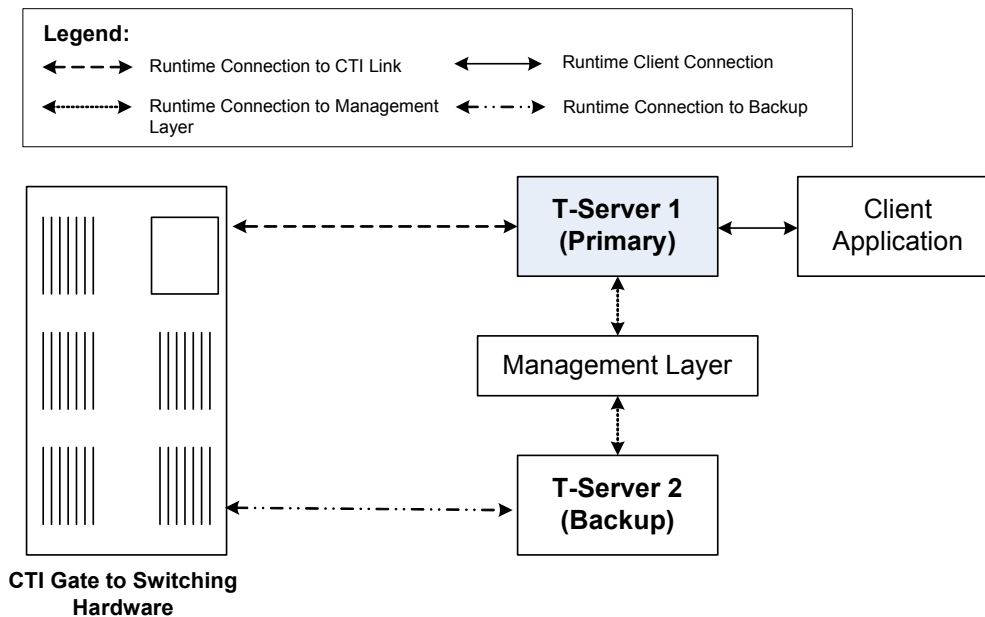


Figure 5: Warm Standby Redundancy Architecture

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 48](#).)

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 25](#)), refer to Part Two for detailed information on the available hot standby schema.

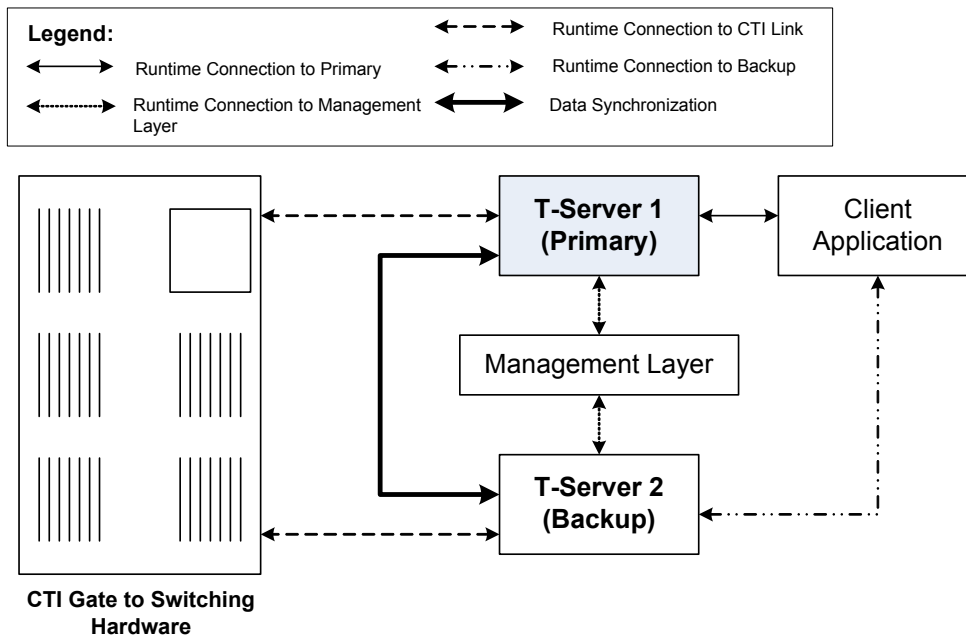


Figure 6: Hot Standby Redundancy Architecture

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 59](#) 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 39](#).
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 39](#).
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 52](#)).

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 52](#)

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 41](#) 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 39](#).

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 36](#).
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 56](#)).

Table 1 on [page 25](#) 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 documentation website at

http://docs.genesys.com/Special:Repository/g_ref_smi.pdf?id=73c96eb2-c7cb-4839-95e5-0c910861e615.

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 40](#).

- 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 55](#)

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 40](#).

- 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 41](#) 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 115](#), to test your configuration and installation, or continue with Chapter 4, “Multi-Site Support,” on [page 57](#), for more possibilities.



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:

- [Multi-Site Fundamentals, page 58](#)
- [ISCC Call Data Transfer Service, page 59](#)
- [ISCC/Call Overflow Feature, page 79](#)
- [Number Translation Feature, page 83](#)
- [Network Attended Transfer/Conference Feature, page 91](#)
- [Event Propagation Feature, page 93](#)
- [ISCC Transaction Monitoring Feature, page 100](#)
- [Configuring Multi-Site Support, page 100](#)
- [Next Steps, page 114](#)

Note: Each switch/T-Server combination offers different multi-site options. For details describing your specific switch/T-Server environment, refer to Chapter 9, “T-Server Common Configuration Options,” on [page 205](#).

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 75](#) and Table 4 on [page 80](#).

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 66](#) and “Transfer Connect Service Feature” on [page 78](#).
 - 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 79](#)).
 - Number Translation feature (see [page 83](#)).
 - Network Attended Transfer/Conference (NAT/C) feature (see [page 91](#)).

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 94](#))
 - Party Events propagation (see [page 95](#))

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*.

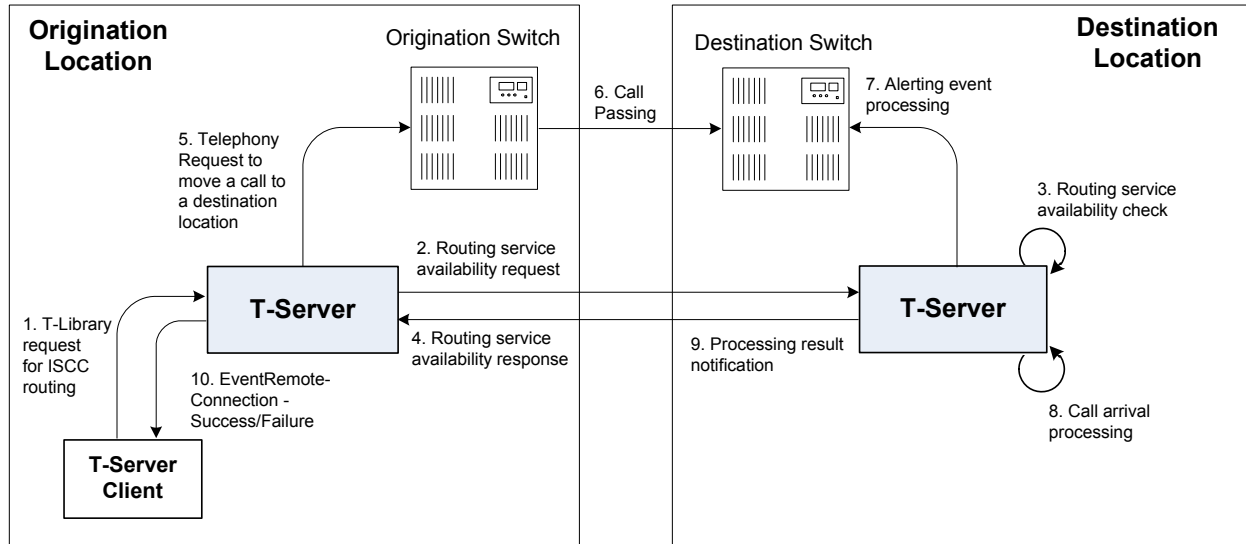


Figure 7: Steps in the ISCC Process

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 *Voice 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-uu`, `direct-digits`, `reroute`, `dnis-pool`, `pullback`, or `route-uu`.
- 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 102](#).

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 option descriptions, refer to Chapter 9, “T-Server Common Configuration Options,” on [page 205](#) for option descriptions.

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, 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, 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 67](#)).

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 67](#). Use Table 3 on [page 75](#) 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 9, “T-Server Common Configuration Options,” on [page 205](#) 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 67](#)
- `direct-notoken`, [page 69](#)
- `dnis-pool`, [page 70](#)
- `pullback`, [page 71](#)
- `reroute`, [page 72](#)
- `route` (aliased as `route-notoken`), the default transaction type, [page 73](#)

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

- `direct-callid` (aliased as `direct`), [page 68](#)
- `direct-digits` (reserved for Genesys Engineering)
- `direct-network-callid`, [page 68](#)
- `direct-uui`, [page 69](#)
- `route-uui`, [page 74](#)

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 111](#) 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](#).

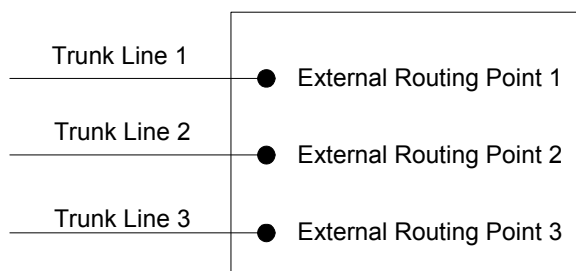


Figure 8: Point-to-Point Trunk Configuration

Note: Dedicated DNs of the External Routing Point type must be configured in a switch. See “Configuring Multi-Site Support” on [page 100](#).

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](#).

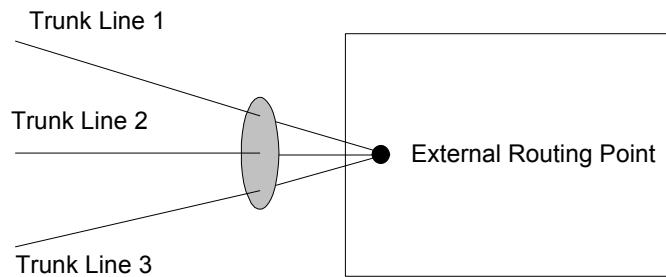


Figure 9: Multiple-to-Point Trunk Configuration

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 73) and the UUI matching feature of the `direct-uui` transaction type (page 69). 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 Communica- tion Manager	Yes				Yes	Yes	Yes				
Avaya INDeX	Yes					Yes	Yes ^b				
Avaya TSAPI	Yes				Yes	Yes	Yes				
Cisco UCCE	Yes					Yes	Yes				
Cisco Unified Communica- tions 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-uuui / route-uuui	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-uuui / route-uuui	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 `UUUI` 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-uuui` 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 `RequestRouteCall` 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 83](#).

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.

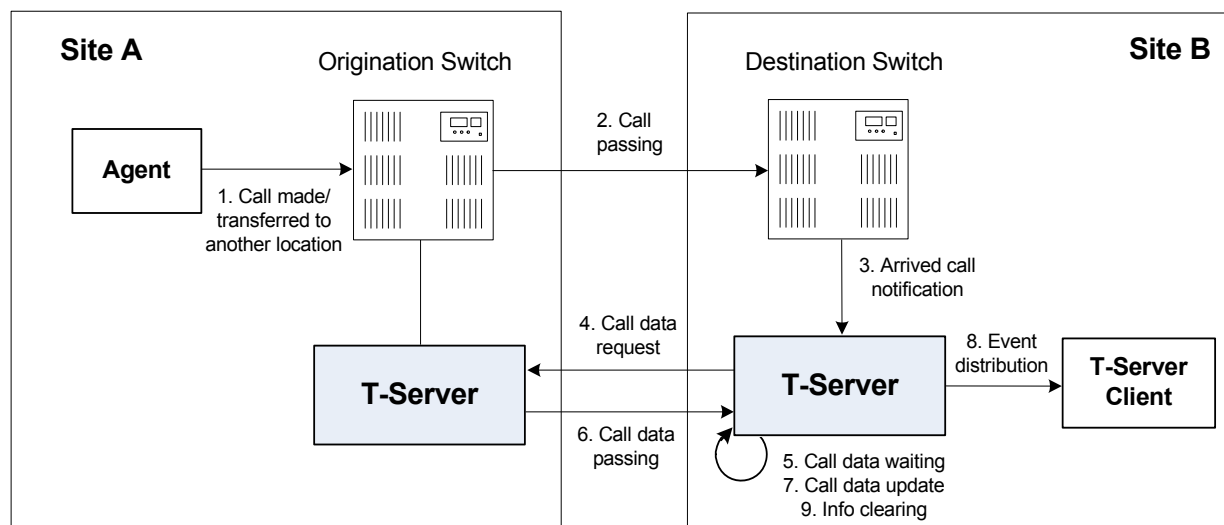


Figure 10: Steps in the ISCC/COF Process

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 84](#) 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 89](#) 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 91](#).

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 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)`

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 form (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 89](#)) 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=1AAAAAAAAA; 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]ABBBBBBBB; 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=91AAAAAAAAA; 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 9, “T-Server Common Configuration Options,” on [page 205](#).

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).

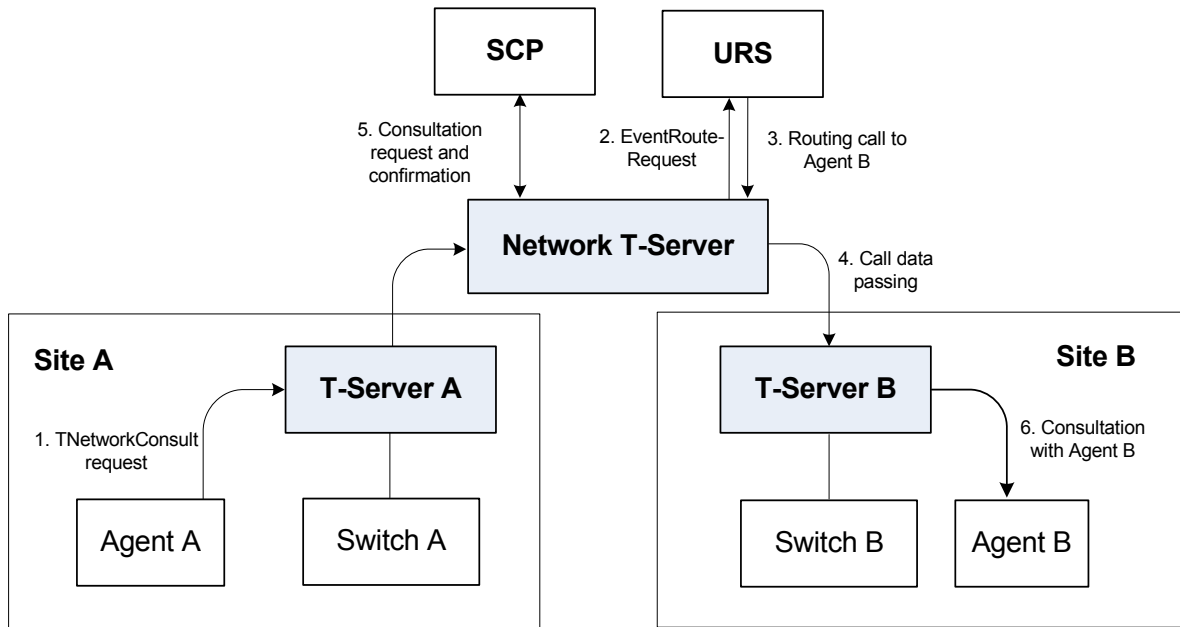


Figure 11: Steps in the NAT/C Process in URS-Controlled Mode

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 *Voice 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 59](#) 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.

Site A

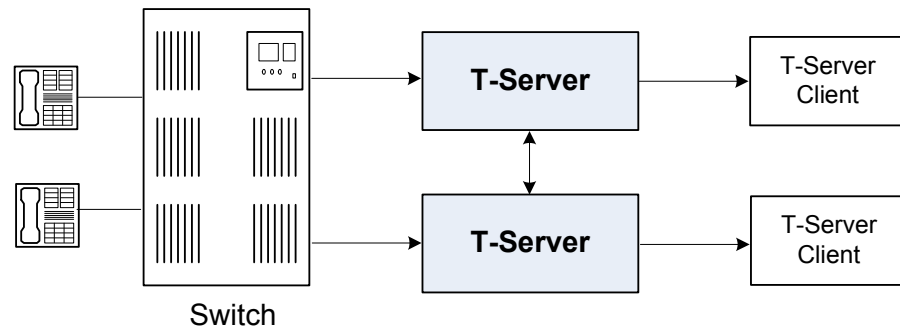


Figure 12: Switch Partitioning Architecture

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 210](#).

To control race conditions that may occur in the switch-partitioned environment, use the `epp-tout` configuration option (see [page 225](#)).

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 9, “T-Server Common Configuration Options,” on [page 205](#).
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 98](#)

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 98](#)

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 103](#) for instructions. If no Access Codes are configured, see [page 104](#) 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 events¹, 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 *Voice 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 33](#), as well as previous sections of this chapter on multi-site deployment. In particular, Table 3 on [page 75](#) shows which transaction types are supported by a specific T-Server, while Table 4 on [page 80](#) 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 215](#) 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 108](#) 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 215](#), 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 107](#).

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 98 .
direct-network-callid=<>	For configuration information, see Part Two of this document. (Use Table 4 on page 80 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 UII	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-uuu 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 80](#) 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 102](#).

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 108](#).
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 104](#).

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 104](#).

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 115](#) to test your configuration and installation.

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, page 115](#)
- [Starting and Stopping with the Management Layer, page 117](#)
- [Starting with Startup Files, page 118](#)
- [Starting Manually, page 119](#)
- [Verifying Successful Startup, page 125](#)
- [Stopping Manually, page 125](#)
- [Starting and Stopping with Windows Services Manager, page 126](#)
- [Next Steps, page 126](#)

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	<p>The license address. Use for the server applications that check out technical licenses. Can be either of the following:</p> <ul style="list-style-type: none"> • 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. <p>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.</p>
-V	<p>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.</p>
-nco X/Y	<p>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.</p>
-lmspath	<p>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.</p> <p>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.</p>
- transport-port <port number>	<p><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 <i>Genesys 8.x Security Deployment Guide</i> for more information.</p>
- transport-address <IP address>	<p><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 <i>Genesys 8.x Security Deployment Guide</i> for more information.</p>

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 115](#).
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 119](#) 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 115](#).

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_teltronics 2020	ha_proxy_teltronics 2020.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 123](#).

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 115](#).

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 120](#) 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 120](#) 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 115](#).

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 120](#) 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 120](#) 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 115](#) 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.



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 131](#), 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 T-Server Features,” on [page 159](#), describes the features that are supported by this T-Server, including T-Library functionality, and error messages.
- Chapter 8, “Common Configuration Options,” on [page 183](#), describes the log configuration options common to all Genesys server applications.
- Chapter 9, “T-Server Common Configuration Options,” on [page 205](#), describes the configuration options common to all T-Server types, including options for multi-site configuration.
- Chapter 10, “T-Server-Specific Configuration Options,” on [page 233](#), 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.

New in T-Server for NEC NEAX/APEX

The following new features are available in the 8.1.001.00 release of T-Server for NEC NEAX/APEX:

- **Support for Call Parking, Call Pickup, and Multi-line Operation.** T-Server now supports three new business features on the NEC UNIVERGE SV8500 switch. See [pages 168, 169, and 171](#) for details.

- **Support for RTP Monitoring API.** T-Server supports the standard Genesys Real-Time Protocol (RTP) stream monitoring API to allow integration with third-party recording solutions. See [page 172](#) for details.
- **Support for SCF Calling Party Number Selection.** T-Server supports the SCF Calling Party Number Selection to enable you to use the NEC UNIVERGE SV8500 switch to specify your calling party number according to your needs. See [page 175](#) for details.
- **Enhanced heartbeat functionality.** The new configuration option, [alive-message-tolerance](#), specifies the number of heartbeat time intervals (in milliseconds) that can expire before the link is reset. See the option description on [page 235](#) for details.
- **Enhanced OAI event functionality.** A new configuration option, [enable-answer-term-info](#), enables the TermNumber Extensions attribute in the EventEstablished and EventRetrieved events. See the option description on [page 237](#) for details.
- **Support for multi-observing.** A new configuration option, [multi-observing-enabled](#), enables multi-observing functionality. See the option description on [page 247](#) for details.
- **Enhanced D-Term functionality.** A new configuration option, [report-dterm-ip-addr](#), discovers D-Term IP addresses. See the option description on [page 250](#) for details.
- **Enhanced voice-monitoring application functionality.** A new configuration option, [rtp-info-password](#), specifies the password to allow voice-monitoring applications to monitor any call. See the option description on [page 251](#) for details.
- **Enhanced calling number party functionality.** A new configuration option, [scf-calling-nbr-encoding](#), specifies the encoding method of the calling party number information. See the option description on [page 252](#) for details.

The following new features are available in the initial 8.1 release of T-Server for NEC NEAX/APEX:

- **Infolink heartbeat functionality.** T-Server is able to simulate heartbeat polling of an ACD link. A new configuration option [enable-infolink-heartbeat](#), implements this functionality. See the option description on [page 238](#) for details.
- **Enhanced agent state notification.** The new configuration option [trigger-agent-state-response-by](#) modifies the distribution of agent state notifications. See the option description on [page 257](#) for details.
- **Support of the NEC dual redundancy feature.** In dual redundancy mode, in the event of a failover, the switch can now preserve T-Server subscription data. This functionality is controlled by the new configuration option, [aarq-logical-port](#). See the option description on [page 235](#) for details.

- **Multi-ACD processor support on an FCCS network.** T-Server fully supports a multi-ACD processor on a Fusion Call-Control System (FCCS) network. See [page 171](#) for details.
- T-Server is now supported on the following platforms:
 - IBM AIX 7 64-bit
 - Red Hat Enterprise Linux 5 64-bit
 - HP-UX 11i v3

Notes: • For a list of new features common to all T-Servers, see Part 1, “T-Server Deployment,” on [page 15](#).

- Configuration option changes that apply to your T-Server are described in “Changes from Release 8.0 to 8.1” on [page 261](#).

6

Switch-Specific Configuration

This chapter presents switch-specific reference information for configuring T-Server for the NEC NEAX/APEX switch. It includes these sections:

- [Known Limitations, page 131](#)
- [Switch Platform General Information, page 133](#)
- [Setting DN Properties, page 138](#)
- [Configuring Universal Routing Server, page 148](#)
- [Configuring Trunks, page 150](#)
- [Support for NEC ACD Agent Break Modes, page 151](#)
- [Configuring Dedicated ACD Supervisors, page 151](#)
- [Configuring External ACD Connection, page 152](#)
- [Switch Configuration Requirements, page 153](#)
- [Switch-Based High Availability, page 154](#)
- [Configuring Unlicensed Devices, page 155](#)

Known Limitations

Several known limitations result from the T-Server/NEC NEAX/APEX interface:

1. Prior to Revision 11 of the main software on large-system switches, only one T-Server can monitor each DN at any one time. This makes it impossible to have a hot standby T-Server monitoring the same set of DNs on the same switch. DNs already monitored by one T-Server are not monitored by a second T-Server, causing the second T-Server to improperly track call state. However, you can connect two T-Servers to one switch to monitor a distinct group of DNs (for example, separate contact center tenants) in the following ways:

- When configuring multiple T-Servers to connect to a switch with integrated OAI Ethernet (that is, Ethernet running in the main processor, not on a CC98 or PA-CC00 card), connect all T-Servers to TCP port 60030.
- When configuring multiple T-Servers to a switch without integrated OAI Ethernet (that is, a switch with a CC98 or PA-CC00 card installed), connect each T-Server to a different TCP port in the range of 1024 to 1048.

Notes: Large-system switches, with appropriate software and configuration, are capable of supporting up to eight applications that simultaneously monitor a single DN. Check with your switch vendor to confirm support capability of your release of switch software.

Small-system switches can support up to four T-Servers monitoring the same DNs simultaneously.

2. Specify all devices that are a part of the contact center configuration in the Configuration Manager DN section under the appropriate switch. This includes all trunks, extensions (agent and nonagent stations), splits, voice-mail ports, and any device that is involved during a call, with the exception of Personal Pilot Numbers (PPNs), which you should not configure as DNs.
3. T-Server ignores all messages received from the switch referencing an unconfigured device, and it generates no T-Library events for the calls associated with those messages (for example, if a call to an agent originates from a handset outside of the formal contact center). You must include the handsets of the administrative staff in the device configuration; otherwise events are not generated for those devices. Other Genesys applications, such as Universal Routing Server, do not correctly reflect the actual state of the agent as a call unless you have configured all related devices.
4. You can configure only a single CTI link for each T-Server. If you configure any extra links, T-Server ignores them.
5. You cannot suppress the ring tone for inbound calls via ISDN trunks by using the SCF FID=127 (Tone Control) OAI request. This limitation affects Genesys functionality that relies on SCF FID=127 requests, such as the processing of `TGiveSilenceTreatment` requests, or the suppression of the ring tone on devices configured as `SPLIT_SILENT`, `SPLIT_ROUTE_POINT_SILENT`, or `ACD_PILOT_SILENT`.
6. When T-Server is used for reporting only on a large-system switch with ACD integration, and if the ACD is reset, the Infolink will stop sending notification messages to T-Server, until T-Server re-subscribes to the Infolink service. Currently, the only way to make T-Server to re-subscribe to the Infolink service is to restart T-Server.

Known Limitations for the Attendant Console

The following are known limitations when enhanced support of call scenarios is added to the switch, and the destination of the call is an Attendant Console:

1. The destination of an Attendant Console initiated call cannot be controlled by CTI requests. This means that most of the requests (for example: `TAnswerCall`, `THoldCall`, `TInitiateTransfer`, and `TRouteCall`) will fail with an error.
2. The switch does not invoke the Call Control Vector (CCV) call processing for call connections with an Attendant Console. Therefore, an Attendant Console initiated call or transfer to a Split ACD Queue is not distributed to an agent until the transfer is completed. Likewise, no `EventRouteRequest` event is sent on a call from an operator to a Split Routing Point.
3. D-Term is unable to initiate a transfer of an established call between the Attendant Console and a D-Term.
4. T-Server is unable to handle a direct call between two Attendant Consoles.
5. T-Server is unable to handle a direct call between an Attendant Console and an ACD Queue or Routing Point.
6. T-Server is unable to differentiate between transfer-reconnect and transfer-complete scenarios when a consultation call between Attendant Console and a transfer destination is established. T-Server handles such reconnections as call transfers, causing a call state de-synchronization between T-Server and the actual state on the switch.

Switch Platform General Information

Note: The information in this section has been provided by your switch vendor. It is *not* T-Server support information; it is included in this *T-Server Deployment Guide* solely for your convenience. For information on T-Server support, refer to the [Genesys Supported Media Interfaces Reference Manual](#).

Genesys defines a *large-system switch* as an NEC switch that supports an internal ACD (Automatic Call Distribution) processor. This processor is either integrated with the main processor or resides on a CP53/CP54 ACD module installed in the switch. A *small-system switch*, on the other hand, supports only basic Uniform Call Distribution (UCD) internally.

Large-system and small-system switches differ in a number of ways with respect to DN configuration and ACD/UCD functionality. Both switch types

require a different configuration within the Configuration Layer, as noted in the documentation.

Note: T-Server supports OAI via TCP/IP Ethernet connection only.

The following tables present information from your vendor on large-system switches (Tables 10–13) and small-system switches (Tables 14–18) from different geographic regions, as well as the hardware and software releases that are required in order to operate these switches.

Large-System Switches

Table 10: Japanese Market

Switch Model	Switch Name	Processor	Maximum Number of Ports	PBX Software Release	ACD/IP Software Release	Peripheral Equipment
ICS	APEX 7400 IMS M140	V70	1536	J-Version (V4 or later)	F4 or later (SC-2243)	PA-CC98 (OAI Ethernet)
	APEX 7400 IMS M160		6144			
	APEX 7400 IMS M180		24576			
	APEX 7400 IMS M140	Intel 486	1536			
	APEX 7400 IMS M150		3072			
	APEX 7400 IMS M160SE	Pentium	6144			
	APEX 7400 IMS M180SE		24576			
IMX	APEX 7600 MMX M140	Pentium	1536	R6 Software (V2 or later)		
	APEX 7600 MMX M160		6144			
	APEX 7600 MMX M180		24576			

Table 11: North American/Brazilian Market

Switch Model	Switch Name	Processor	Maximum Number of Ports	PBX Software	ACD/IP Software Release	Peripheral Equipment
ICS	NEAX 2400 IMS IMG	V70	1536	J-Version (V4 or later)	F4 or later (SC-2243)	PA-CC98 (OAI Ethernet)
	NEAX 2400 IMS MMG		6144			
	NEAX 2400 IMS UMG		24576			
	NEAX 2400 IMS SIM	Intel 486	384			
	NEAX 2400 IMS IMGx	1536				
	NEAX 2400 IMS IMGxh	3072				
	NEAX 2400 IMS IMGdxh	Pentium	6144			
	NEAX 2400 IMS UMGx	24576				
IMX	NEAX 2400 IMS (IMX) IMG	Pentium	6144	R5 Software (V3 or later)		
	NEAX 2400 IMS (IMX) UMG		24576			
IPX	NEAX 2400 IPX	Pentium	24576	R16 or later	F4 or later (SC-2243)	None

Table 12: Australia/New Zealand Market

Switch Model	Switch Name	Processor	Maximum Number of Ports	PBX Software	ACD/IP Software Release	Peripheral Equipment
ICS	NEAX 7400 ICS M140	Intel 486	1536	J-Version (V4 or later)	F4 or later (SC-2243)	PA-CC98 (OAI Ethernet)
	NEAX 7400 ICS M160S		3072			
	NEAX 7400 ICS M160SE	Pentium	6144			
	NEAX 7400 ICS M180SE		24576			
IMX	NEAX 7600 IMX M240	Pentium	1536	R6 Software (V2 or later)		
	NEAX 7600 IMX M260		6144			
	NEAX 7600 IMX M280		24576			

Table 13: Other Markets

Switch Model	Switch Name	Processor	Maximum Number of Ports	PBX Software	ACD/IP Software Release	Peripheral Equipment
ICS	NEAX 7400 ICS M160	V70	6144	J-Version (V4 or later)	F4 or later (SC-2243)	PA-CC98 (OAI Ethernet)
	NEAX 7400 ICS M180		24576			
	NEAX 7400 ICS M140	Intel 486	1536			
	NEAX 7400 ICS M150		3072			
	NEAX 7400 ICS M160SE	Pentium	6144			
	NEAX 7400 ICS M180SE		24576			
IMX	NEAX 7400 ICS M140MX	Pentium	1536	R5 Software (V3 or later)		
	NEAX 7400 ICS M160MX		6144			
	NEAX 7400 ICS M180MX		24576			

Small-System Switches

Table 14: Japanese Market

Switch Model	Switch Name	Maximum Number of Ports	MP Version	AP Version	Peripheral Equipment
ICS-VS	APEX 1000	64	Version MP SC-2733 SHS BSC PROG-CM	SC-2162 SHS IP PROG-CB (V9 or later)	PN-CC00 (OAI Ethernet board)
	APEX 1000H	64			
	APEX 3000	512			
	APEX 3600	128			
CI	APEX2000 VS	72			
	APEX1000 VS	48			

Table 15: North American Market

Switch Model	Switch Name	Maximum Number of Ports	MP Version	AP Version	Peripheral Equipment
ICS-VS	NEAX 2000 IVS	512	1900R2 Series (TAPI 2.1) SC-2733 SHS BSC PROG-CM	SC-2688 SHS IP PROG-CC	PN-CC00 (OAI Ethernet board)
CI	NEAX 2000 IVS	72			
	NEAX 1000 IVS	48			
IPS	NEAX 2000 IPS	1020	3400 Series R9 (SC-3326 IPS BSC PROG-J1) or later		PZ-M606 (Ethernet board)

Table 16: Australia/New Zealand Market

Switch Model	Switch Name	Maximum Number of Ports	MP Version	AP Version	Peripheral Equipment
ICS-VS	NEAX 7400 ICS M120	512	4200 Series SC-2733 SHS BSC PROG-CM	SC-2162 SHS IP PROG-CB (V9 or later)	PN-CC00 (OAI Ethernet board)
CI	NEAX 7400 ICS M110	72			

Table 17: French Market

Switch Model	Switch Name	Maximum Number of Ports	MP Version	AP Version	Peripheral Equipment
ICS-VS	BCS Exchange 500	128	Release 11 (TAPI 2.1) SC-2733 SHS BSC PROG-CM	SC-2688 SHS IP PROG-CC	PN-CC00 (OAI Ethernet board)

Table 18: Other Markets

Switch Model	Switch Name	Maximum Number of Ports	MP Version	AP Version	Peripheral Equipment
ICS-VS	NEAX 7400 ICS M100ES/ED NEAX2400 IMS-SP (Brazil)	512	Version 7 or later SC-2681 SHS BSC PROG-CL	SC-2162 SHS IP PROG-CB (V9 or later)	PN-CC00 (OAI Ethernet board)
CI	NEAX 7400 ICS M80VS NEAX 2400 IMS-CI (Brazil)	72			

Setting DN Properties

Table 19 contains information on how to set DN types and properties, depending on the switch configuration. Use this information to set DN properties in the Configuration Layer.

Note: All DNs to be used in a customer environment must be registered in the Configuration Layer.

Table 19: Setting the DN Properties for the NEC NEAX/APEX Switch

DN Type	Switch DN Type	Switch-Specific Type	Comments
Extension	STATION_DTERM_Extension	1	D-Term Extension on the mainline (PBX line). On large-system switches, this DN type is for extensions that have been configured in the ACD using the ACDPSN command. If the station is not configured in the ACD, use DN type STATION_DTERM_BUSINESS_EXTENSION.
Extension	STATION_ANALOG	2	Analog phone lines such as IVR and S-Dialer ports. On large-system switches, this DN type is for extensions that have not been configured in the ACD using the ACDPSN command. If the station is configured in the ACD, use DN type STATION_ANALOG_ACD.

Table 19: Setting the DN Properties for the NEC NEAX/APEX Switch (Continued)

DN Type	Switch DN Type	Switch-Specific Type	Comments
Extension	STATION_DTERM_EXTENSION_SUBLINE	3	<p>D-Term Extension on the subline. Overrides device type STATION_DTERM_EXTENSION when you have set the <code>position-is-subline</code> option to false.</p> <p>This device type is used only in the case of a small-system switch that is not configured for use with Universal Routing Server.</p> <p>The UCD, by default, distributes calls only to the mainline (PBX line). Secondary (outbound and personal) calls should therefore be handled through the subline.</p> <p>Note: Use the <code>Association</code> property to set the corresponding handset mainline (PBX line).</p>
Extension	STATION_CORDLESS_VOICE	4	Cordless voice line.
Extension	STATION_DTERM_DATA	5	Digital DATA handsets.
Extension	STATION_FISDN_TERM	6	ISDN handsets.
Extension	ATTENDANT_CONSOLE	7	For Attendant Consoles on large-system and IMX switches, you must specify the correct port number in the <code>Association</code> field of the <code>General</code> tab of the DN properties when configuring the Attendant Console in Configuration Manager. Contact your switch technician to determine which port number the Attendant Console is connected to. Depending on the switch model, the general limit of Attendant Consoles is 64.
Extension	ATTENDANT_CONSOLE_2	8	Attendant console for the small-system switches (between 0 and 4).
Extension	DICTATION_TRUNK	9	Dictation Trunk.
Extension	STATION_DTERM_BUSINESS_EXTENSION	11	<p>D-Term Extension on the mainline (PBX line).</p> <p>For stations not configured in the <code>ACDMAT ACDPSN</code> command.</p>

Table 19: Setting the DN Properties for the NEC NEAX/APEX Switch (Continued)

DN Type	Switch DN Type	Switch-Specific Type	Comments
Extension	STATION_DTERM_ BUSINESS_EXTENSION_ SUBLINE	12	D-Term Extension on the subline. For stations not configured in the ACDMAT ACDPSN command. Note: Use the <i>Association</i> property to set the corresponding handset mainline (PBX line).
Extension	STATION_HOTLINE	13	This device is used to configure Hotlines, which correspond to OAI CPI=0x00, TEC=0x32 on a large-system switch. Note: For additional information on Brokerage Hotline configurations on the switch, see the <i>Feature Programming</i> manual of your PBX.
Extension	STATION_HOTLINE_ SUBLINE	14	This device is used to configure Hotlines, which correspond to OAI CPI=0x00, TEC=0x32 on a large-system switch. The primary terminal number (mainline) of the hotline must be specified. Note: Use the <i>Association</i> property to set the corresponding handset mainline.
Extension	SIP_OR_HANDYPHONE_ EXTENSION	15	SIP Phone or Personal Handyphone extension. This device is designated in OAI protocol as Terminal type (CPI): 0x00 and Terminal sub-type (TEC): 0x29.
Extension	STATION_VIRTUAL	99	Virtual device that T-Server creates and maintains in its internal DN table.
ACD Position	STATION_DTERM_ POSITION_SUBLINE	1	D-Term ACD position on the subline. On large-system switches the ACD Position corresponds to the ACD Line (subline) of the handset. On small-system switches utilizing Universal Routing Server, the ACD Position corresponds to the non-UCD line (the subline). Note: Use the <i>Association</i> property to set the corresponding handset mainline (PBX line).

Table 19: Setting the DN Properties for the NEC NEAX/APEX Switch (Continued)

DN Type	Switch DN Type	Switch-Specific Type	Comments
ACD Position	STATION_DTERM_POSITION	2	D-Term ACD position on the mainline (PBX line). Overrides device type STATION_DTERM_POSITION_SUBLINE when you have set the <code>position-is-subline</code> option to <code>false</code> . This device type is used only in small-system switches that are not configured for use with Universal Routing Server. The UCD by default distributes calls only to the mainline (PBX line).
ACD Position	STATION_ANALOG_ACD	3	Analog phone lines such as IVR and S-Dialer ports that are configured in the ACD as ACD agent positions under the ACDSPL command. Note: This device type is not supported on a non-ACD switch.
ACD Position	DTERM_SUPERVISOR_SUBLINE	4	D-Term subline device. This device must be dedicated solely to ACD call supervision functions. All calls coming into a DTERM_SUPERVISOR_SUBLINE device are processed as part of the Service Observing call model only and, as such, these devices must be excluded from all other scenarios. Note: Use the <code>Association</code> property to set the corresponding handset mainline (PBX line).
ACD Position	SIP_OR_HANDYPHONE_POSITION	5	SIP Phone or Personal Handyphone ACD position. This device is designated in OAI protocol as Terminal type (CPI): <code>0x00</code> and Terminal sub-type (TEC): <code>0x29</code> .
Routing Point	MONITOR_NUMBER	1	On ACD switches, use this device type only for External Routing Points and configure it in the PBXMAT under the AMN0 command. Do not configure this device type as an ACD Pilot object in the ACDMAT. On non-ACD switches (set the <code>infolink</code> option to <code>false</code>), this device type is used for Universal Routing Server Routing Points as well as External Routing Points, and you should configure it within a UCD group as a pilot type 3, using the CAT 17 command.

Table 19: Setting the DN Properties for the NEC NEAX/APEX Switch (Continued)

DN Type	Switch DN Type	Switch-Specific Type	Comments
Routing Point	MONITOR_NUMBER_ACD	2	<p>Deprecated. Do not use unless directed to do so by Genesys Customer Care.</p> <p>Note: This type is the same as a MONITOR_NUMBER, but it is shared by an ACD Pilot, and T-Server uses ACD to route the call.</p>
Routing Point	SPLIT_ROUTE_POINT	3	<p>A Routing Point that corresponds to a Queue-to-Split step contained within an ACD Call Control Vector.</p> <p>On ACD switches, use this device type as the Routing Point for Universal Routing Server-controlled calls and configure it on the large-system ACDMAT using the ACDSPL and ACDCCV commands, with no agents configured against the split. Do not use this type of Routing Point for External Routing Points.</p> <p>The device type is derived from the three-digit split (zero padded) in the Queue-to-1 step of the ACD Call Control Vector when the Queue-to-1 step corresponds to DN 001.</p> <p>This device is not supported on a non-ACD switch.</p> <p>Note: Use the Association property to set the pilot number associated with the ACD Call Control Vector as defined in the ACDPLT command.</p>
Routing Point	SPLIT_ROUTE_POINT_SILENT	4	<p>Same as a SPLIT_ROUTE_POINT, but when the call arrives at the Routing Point, T-Server applies tone control request to stop the ringback.</p>

Table 19: Setting the DN Properties for the NEC NEAX/APEX Switch (Continued)

DN Type	Switch DN Type	Switch-Specific Type	Comments
Routing Point	MONITOR_NUMBER_NO_TONE	5	<p>This DN type is only for a small-system switch and is similar to the MONITOR_NUMBER, but it is registered on the switch with the SMFR function number, 125 (0x7D) monitor quiet. A caller does not hear a ringback tone while connected to the monitored number registered with SMFR (FN=125). Typically, when someone connects to a MONITOR_NUMBER_NO_TONE, an application uses tone control request to give the caller the desired tone.</p> <p>Note: Client request RequestRouteCall with AttributeRouteType=RouteTypeReject for object MONITOR_NUMBER_NO_TONE is translated into a release request for a caller OAI message SCF (FN=2, TI=1, ReleaseCall with BusyTone to the caller).</p>
ACD Queue	ACD_PILOT	1	Deprecated. Configure ACD Queues as splits rather than as pilot numbers. Do not use unless directed to do so by Genesys Customer Care.
ACD Queue	UCD_PILOT	2	<p>The pilot number of a UCD Hunt Group configured on a small-system switch. The equivalent of an ACD Queue pilot on a large-system switch.</p> <p>The small-system switch does not support an internal ACD. Instead, internal call distribution is performed using UCD, which implements basic station hunting.</p> <p>Configure this device on the small-system switch as a pilot type 1, using the CAT 17 command.</p>
ACD Queue	ACD_PILOT_SILENT	3	Deprecated. Configure ACD Queues as splits rather than as pilot numbers. Do not use unless directed to do so by Genesys Customer Care.

Table 19: Setting the DN Properties for the NEC NEAX/APEX Switch (Continued)

DN Type	Switch DN Type	Switch-Specific Type	Comments
ACD Queue	SPLIT	4	<p>An ACD Queue that corresponds to a Queue-to-Split step contained within an ACD Call Control Vector. Multisplit Call Control Vectors are not supported. In this situation, only the first split is reported on.</p> <p>The device type is derived from the three-digit split (zero padded) in the Queue-to-Split step of the ACD Call Control Vector. The Queue-to-1 step corresponds to DN 001.</p> <p>This device is not supported on a non-ACD switch.</p> <p>Note: Use the Association property to set the pilot number associated with the ACD Call Control Vector as defined in the ACDPLT command.</p>
ACD Queue	SPLIT_SILENT	5	Same as a SPLIT, but when the call arrives at the ACD Queue, T-Server applies tone control request to stop ringback.
ACD Queue	UCD_VIRTUAL	99	<p>Use when you are configuring UCD on a large-system switch. The large system does not support the concept of a UCD pilot number, unlike a small-system switch. As a result, you must configure a virtual pilot.</p> <p>You must create and configure a virtual station (MAT command ASDT, station TEC=18) to call-forward all calls to the head member of the UCD group. You must configure this virtual station as a UCD_PILOT_VIRTUAL, and then treat it as the pilot number. It must be the destination of all calls arriving at the UCD group.</p>
Trunk	EXTERNAL_TRUNK_CO_DDD	1	<p>External Trunk (CO DDD).</p> <p>Most trunks are configured as this default Trunk type in the Configuration Layer.</p>
Trunk	EXTERNAL_TRUNK_CO_FX	2	External Trunk (CO FX).

Table 19: Setting the DN Properties for the NEC NEAX/APEX Switch (Continued)

DN Type	Switch DN Type	Switch-Specific Type	Comments
Trunk	EXTERNAL_TRUNK_CO_WATS	3	External Trunk (CO WATS).
Trunk	EXTERNAL_TRUNK_TIE_TIE	4	External Trunk (TIE TIE).
Trunk	EXTERNAL_TRUNK_TIE_CCSA	5	External Trunk (TIE CCSA).
Trunk	EXTERNAL_TRUNK_TOLL	6	External Trunk (TOLL).
Trunk	EXTERNAL_TRUNK_CAS	7	External Trunk (CAS).
Trunk	EXTERNAL_TRUNK_PAGING	8	External Trunk (PAGING).
Trunk	EXTERNAL_TRUNK_RADIO_PAGING	9	External Trunk (RADIO PAGING).
Trunk	EXTERNAL_TRUNK_UNKNOWN	10	Deprecated. Use when configuring a trunk with an unknown type. You should know all device types before configuring in the Configuration Layer. Do not use unless directed to do so by Genesys Customer Care.
Music Port	ANNOUNCEMENT_TRUNK	1	Announcement Trunk.
Virtual Queue	VIRTUAL_QUEUE	1	Virtual device T-Server creates and maintains in its internal DN table.
Virtual Route Point	VIRTUAL_ROUTE_POINT	1	Virtual device T-Server creates and maintains in its internal DN table.
E-mail Address	EMAIL_POSITION	1	Virtual device T-Server creates and maintains in its internal DN table.
Any DN type not in the table	ADDRESS_OTHER	1	Virtual device T-Server creates and maintains in its internal DN table. Captures all other types not defined in the table but configured in the Configuration Layer by mistake.

DN Type Cross-Reference Between NEC and Genesys

Table 20 contains cross-reference information on DN types that Genesys and NEC use.

Table 20: DN Type Cross-Reference between NEC and Genesys

Genesys			NEC		
DN Type	SST	NEC T-Server Type	Term Type	Sub-Type	NEC Type
Extension	1	STATION_DTERM_EXTENSION	0x00	0x0a	D-Term (Voice)
	2	STATION_ANALOG	0x00	0x01	Ordinary Phone Set
	3	STATION_DTERM_EXTENSION_SUBLINE	0x00	0x0a	D-Term (Voice)
	4	STATION_CORDLESS_VOICE	0x00	0x0b	Cordless Phone Set
	5	STATION_DTERM_DATA	0x00	0x14	Dterm (Data)
	6	STATION_FISD_TERM	0x00	0x28	ISDN Terminal
	7	ATTENDANT_CONSOLE	0x01	1–64	Attendant Console (large systems)
	8	ATTENDANT_CONSOLE_2	0x01	0–7	Attendant Console (small systems)
	9	DICTATION_TRUNK	0x04	0x01	Dictation Trunk
	10	Reserved			Reserved
	11	STATION_DTERM_BUSINESS_EXTENSION	0x00	0x0a	D-Term (Voice)
	12	STATION_DTERM_BUSINESS_EXTENSION_SUBLINE	0x00	0x0a	D-Term (Voice)
	13	STATION_HOTLINE	0x00	0x32 (50)	Hot Line or House Phone
	14	STATION_HOTLINE_SUBLINE	0x00	0x32 (50)	Hot Line or House Phone
	15	HANDYPHONE_EXTENSION	0x00	0x29	Phone Set

Table 20: DN Type Cross-Reference between NEC and Genesys (Continued)

Genesys			NEC		
DN Type	SST	NEC T-Server Type	Term Type	Sub-Type	NEC Type
ACD Position	1	STATION_DTERM_POSITION_SUBLINE	0x00	0x0a	D-Term (Voice)
	2	STATION_DTERM_POSITION		0x0a	D-Term (Voice)
	3	STATION_ANALOG_ACD	0x00	0x01	Ordinary Phone Set
	4	DTERM_SUPERVISOR_SUBLINE			Not available
ACD Queue	1	ACD_PILOT	0x00	0x1e	Monitored Number
	2	UCD_PILOT	0x00	0x01	Ordinary Phone Set
	3	ACD_PILOT_SILENT	0x00	0x1e	Monitored Number
	4	SPLIT			Not available
	5	SPLIT_SILENT			Not available
Routing Point	1	MONITOR_NUMBER	0x00	0x1e	Monitored Number
	2	MONITOR_NUMBER_ACD	0x00	0x1e	Monitored Number
	3	SPLIT_ROUTE_POINT			Not available
	4	SPLIT_ROUTE_POINT_SILENT			Not available
	5	MONITOR_NUMBER_NO_TONE	0x00	0x1e	Monitored Number
Trunk	1	EXTERNAL_TRUNK_CO_DDD	0x02	0x01	CO Line Trunk (DDD)
	2	EXTERNAL_TRUNK_CO_FX	0x02	0x02	CO Line Trunk (FX)
	3	EXTERNAL_TRUNK_CO_WATS	0x02	0x03	CO Line Trunk (WATS)
	4	EXTERNAL_TRUNK_TIE_TIE	0x02	0x04	TieLine Trunk (TIE)

Table 20: DN Type Cross-Reference between NEC and Genesys (Continued)

Genesys			NEC		
DN Type	SST	NEC T-Server Type	Term Type	Sub-Type	NEC Type
	5	EXTERNAL_TRUNK_TIE_CCSA	0x02	0x05	TieLine Trunk (CCSA)
	6	EXTERNAL_TRUNK_TOLL	0x02	0x06	Toll Terminal
	7	EXTERNAL_TRUNK_CAS	0x02	0x07	CAS Trunk
	8	EXTERNAL_TRUNK_PAGING	0x02	0x08	Paging Trunk
	9	EXTERNAL_TRUNK_RADIO_PAGING	0x02	0x0c	Radio-Paging Trunk
Music Port	1	ANNOUNCEMENT_TRUNK	0x03	0x01	Announcement Trunk
		Not supported	0x05	0x00	Fixed (kind of tone)
		Not supported	0x05	0x01	Disconnected Tone

Configuring Universal Routing Server

When configuring Universal Routing Server (URS) on NEC large-system switches, you can configure the internal switch ACD Processor to operate as a backup mechanism (redundant backup configuration) for distributing calls in the situation when a system failure disables URS. This configuration works because the ACD Processor and URS can share the same call-distribution point, known as a monitored number (known as a Routing Point to URS, or as a Pilot Number to the ACD).

Configuring URS with Redundancy

1. Define a monitored number when configuring either a Routing Point object or an ACD Pilot object using the MAT command `AMNO`.
2. Once you have defined the monitored number, configure it as a Pilot Number in the ACD using the `ACDMAT` command `ACDPLT`.
3. Associate the ACD Pilot with a Call Control Vector (CCV) in the following form:

```
Queue-to-Split 50
Pause 5
Queue-to-Split 1
```

The `Pause` command specifies a time period, in seconds, that ACD waits for URS to route a call to the second Routing Point. If URS does not route the call within the period specified, ACD takes control of the call and routes it to the next available agent using simple ACD algorithms. The smaller the pause value is, the smaller the additional delay for all calls in a failure scenario.

However, with a value of less than 5 seconds, processing a call might take longer during peak times because the ACD takes control of a call while URS is running. Configure the split to route calls to agents in a Semi-Intelligent mode.

4. Once you have configured the monitored number as an ACD Pilot, add a Routing Point object to Configuration Manager. This Routing Point should correspond to the first `Queue-to-Split` instruction in the Call Control Vector (in this example, the DN would be 050), should be configured as type Routing Point, and have a switch-specific Type property set to 3. Set the Association property to the pilot number associated with the ACD Call Control Vector as defined in the `ACDPLT` command.

In Configuration Manager, configure the second Split defined in the Call Control Vector as type ACD Queue, with a switch-specific Type property set to 4 (in this example the DN would be 001). Set the Association property to the pilot number associated with the ACD Call Control Vector as defined in the `ACDPLT` command.

5. Next, develop a strategy to allow URS to route the call to a correct destination.

Because the Call Control Vector in the ACD pilot configuration allows URS only 5 seconds to perform routing, any form of agent routing might not function, in turn causing the ACD timeout to be exceeded until an agent becomes available. Consequently, two Routing Points and strategies are required.

With the first strategy, the call is routed from the first Routing Point to a second Routing Point (one not shared by an ACD pilot number), using a Force route. The second Routing Point is normally configured in Configuration Manager as a Routing Point, with its switch-specific type set to 3 (`SPLIT_ROUTE_POINT`). The CCV should typically have a single step of `Queue-to-Split`, and the split should not be configured for delivery to any agent. This second Routing Point has the second strategy loaded, which directs agent routing (group routing, agent routing, skills-based routing, and so on).

If the system fails and disables URS, calls coming to the first Routing Point wait for 5 seconds and are then sent to the next available agent by the ACD. ACD call distribution cannot intelligently segment the inbound calls; however, it can distribute the calls to available agents. Agent state is synchronized between the two mechanisms because both URS and the ACD monitor the act of logging in, via either the soft phone or hard phone.

Under normal circumstances, URS routes calls immediately from the first Routing Point to the second one. At the second Routing Point, URS executes a second strategy that identifies the correct agent to route to. The call remains on this second Routing Point until the agent becomes available.

Configuring Trunks

You must specify all external and announcement trunks in the device configuration.

Configuring External Trunks

The NEC switch defines all external trunks by two values:

- **Trunk Route**—Number specifying the trunk group individually within the switch.
- **Trunk ID**—Sequence number, starting at 1 and identifying each trunk channel within the group.

The two numbers (Trunk Route and Trunk ID) are encoded into one integer as follows:

<XXXXYY>

where:

XXX is the Trunk Route and YYY is the Trunk ID.

The device specification must be six digits, with two fields of three zeros (0) padded to the left. For example, a tieline trunk group with a route number of 8 and four trunk lines (from 1 to 4) is defined as four devices:

```
008001
008002
008003
008004
```

Configuring Announcement Trunks

The `TGiveMusicTreatment` request uses an announcement trunk to specify which music or announcements are to be played for a call. You must specify the music trunk either as the `MusicPath` attribute in the request (which must be enclosed in quotes) or as a default value in the `default-announcement-trunk` option. Before an announcement trunk can be used in the `TGiveMusicTreatment` request, you must specify it in the device configuration. The format of the announcement trunk device is similar to that for external trunks:

0<XXXXYY>

You must prefix the announcement trunk configuration with a 0; XXX is the Message Kind, and YYY is the Equipment Number.

On the large-system and IMX switches, Message Kind is between 1–58, and Equipment Number is usually 1.

On the small-system switches, Message Kind is always 1, and Equipment Number is between 0–127.

Providing Busy Signal Treatment Through Music Treatments

To provide a busy signal for a call waiting on the Routing Point, you must specify the MusicPath attribute Busy in the TGiveMusicTreatment request.

Support for NEC ACD Agent Break Modes

On the NEC large-system switches, an agent can typically enter a number from 1–9 when setting a break mode (for example, NotReady mode) via the handset. The number selected indicates what type of break an agent is on.

The corresponding AgentNotReady event contains a key-value pair in the Extensions attribute that reflects the break type the agent selected. The key is ReasonCode and the string value is the digit ranging from 1 to 9 that the agent has selected.

Additionally, when a TAgentSetNotReady request is made from a desktop client, a key-value pair of type integer can be passed in the Extensions attribute of the request, in the form break_type, <value>, which enables the specific break mode.

Thus, in CCPulse+, you can define the statistics for the amount of time an agent spends in the NotReady mode, by defining a filter that divides the overall not-ready time into the specific break types (see Table 21 on [page 160](#)). This breakdown is available only for some CCPulse+ reports (see the CCPulse+ and Stat Server documentation for further details).

Configuring Dedicated ACD Supervisors

T-Server cannot properly track Service Observing when one party of the call is an ACD Queue, an agent PPN, or a Routing Point. This is due to CTI messaging limitations. However, this issue can be resolved by configuring dedicated ACD Supervisor devices as a DTERM_SUPERVISOR_SUBLINE in Configuration Manager (see Table 19 on [page 138](#)). This allows T-Server to overcome CTI limitations by relying on the device type to correctly process the call scenarios mentioned above. However, all calls coming into a DTERM_SUPERVISOR_SUBLINE device are processed as part of the Service Observing call model only, and these devices must therefore be excluded from all other scenarios.

Configuring External ACD Connection

For the NEC PBX configurations, where the NEC ACD is an external module outside of the main PBX, T-Server for NEC NEAX/APEX must be configured to connect to a dedicated CTI-link to the ACD module in addition to the connection to the CTI-link of the main PBX. Complete the following steps to properly configure the T-Server Application object in Configuration Manager:

Procedure: Configuring the External ACD Connection

Purpose: To properly configure the T-Server Application object in Configuration Manager.

Start of procedure

1. In the T-Server section, set the `number-of-links` option to 2. (See the option description on [page 247](#).)
2. Configure the `link-n-name` option to point to a new CTI-Link section for the main PBX connection—for example, `link-1-name = pbx-link`.
3. Configure another instance of the `link-n-name` option to point to a new CTI-Link section for the additional external ACD connection—for example, `link-2-name = acd-link`.
4. In the Options tab, add a CTI-Link section for the main PBX connection—for example, `pbx-link`.

Follow these configuration steps for this section:

- a. Configure `hostname`, `port`, and `protocol` options according to the option descriptions. (See “CTI-Link Section” on [page 260](#).)
 - b. Set the `link-type` option to `oai` to indicate that this link is dedicated for a connection to the main PBX.
5. In the Options tab, add a CTI-Link section for the external ACD connection—for example, `acd-link`.

Follow these configuration steps for this section:

- a. Configure the `hostname`, `port`, and `protocol` options according to the option descriptions. (See “CTI-Link Section” on [page 260](#).)
- b. Set the `link-type` option to `infoLink` to indicate that this link is dedicated for a connection to the external ACD module.

End of procedure

Switch Configuration Requirements

You must configure the following switch parameters for T-Server to work properly with the NEC NEAX/APEX switches.

The following options apply to large-system switches with an integrated ACD capability:

1. Enable Answer (Connection) Notification for external trunks.

In the MAT configuration:

- Command: ASYD
- Sys: 1
- Index: 32
- Set Bit 3 (counting from 0) to 1. (Do not change any other bits.)

In the Route Class configuration for all dialing trunks, set the SMDR option to 1.

2. Enable Extended Error Detail.

In the MAT configuration:

- Command: ASYD
- Sys: 1
- Index: 241
- Set Bit 3 (counting from 0) to 1. (Do not change any other bits.)

3. Disable Immediate Ringback on ACD Entry in Campaign Manager, for silent outbound connection.

Note: If you change this configuration, outbound calls from Campaign Manager receive a “short burst” ring before being connected to the outbound agent.

In the MAT configuration:

- Command: ASYD
- Sys: 1
- Index: 27
- Set Bit 7 (highest-order bit) to 0. (Do not change any other bits.)

Configure all required Routing Points as Monitored Numbers.

In the MAT configuration:

- Command: AMNO

4. After performing these modifications, save them to the storage on the switch, using the MAT BKUPI command. Failure to do so means that if the switch loses power, these changes are lost, and T-Server may not function correctly.

5. Under ACD MAT configuration in the ACDUD1 command (ACD User Data 1):
 - Set Incoming Call Message to Iq (q must be in lowercase).
 - Set Infolink with Call ID to no.
6. Configure outbound trunks with the AFRS-LCR access method.
7. On large-system switches, you must configure—using the ACDPLT command—a separate pilot number in the ACD, based on the following criteria. This pilot number:
 - Should *not* have any splits configured and should *not* be the destination for any call.
 - Should have a number smaller than all real pilot numbers in the switch.
 - Should *not* be configured in the Configuration Layer.
 - Is used as a location for the ACD to temporarily hold some calls (the ACD selects the lowest-configured pilot number for this role).
 - Should *not* be shared with a pilot number for a real queue that T-Server is to report on; otherwise, statistics will be incorrect for that queue.
8. When configuring monitored numbers using the AMN0 command for large-system switches, confirm that the MNI (Monitor Number Index) configured for each monitored number is unique.

Switch-Based High Availability

In conjunction with the high-availability (HA) functionality that the NEC switch provides, T-Server for NEC NEAX/APEX now supports a NEC-specific high-availability implementation option: dual CTI-links that link directly to two T-Servers. The following section outlines this option:

Warm and Hot Standby with Two CTI Links

Figure 13 on [page 155](#) shows the NEC CTI configuration with two links and two connected T-Servers in either warm standby or hot standby mode. The warm and hot standby configurations are almost identical. Both T-Servers and their corresponding switch links carry the same information. For hot standby mode only, the primary T-Server synchronizes with the backup through a direct connection.

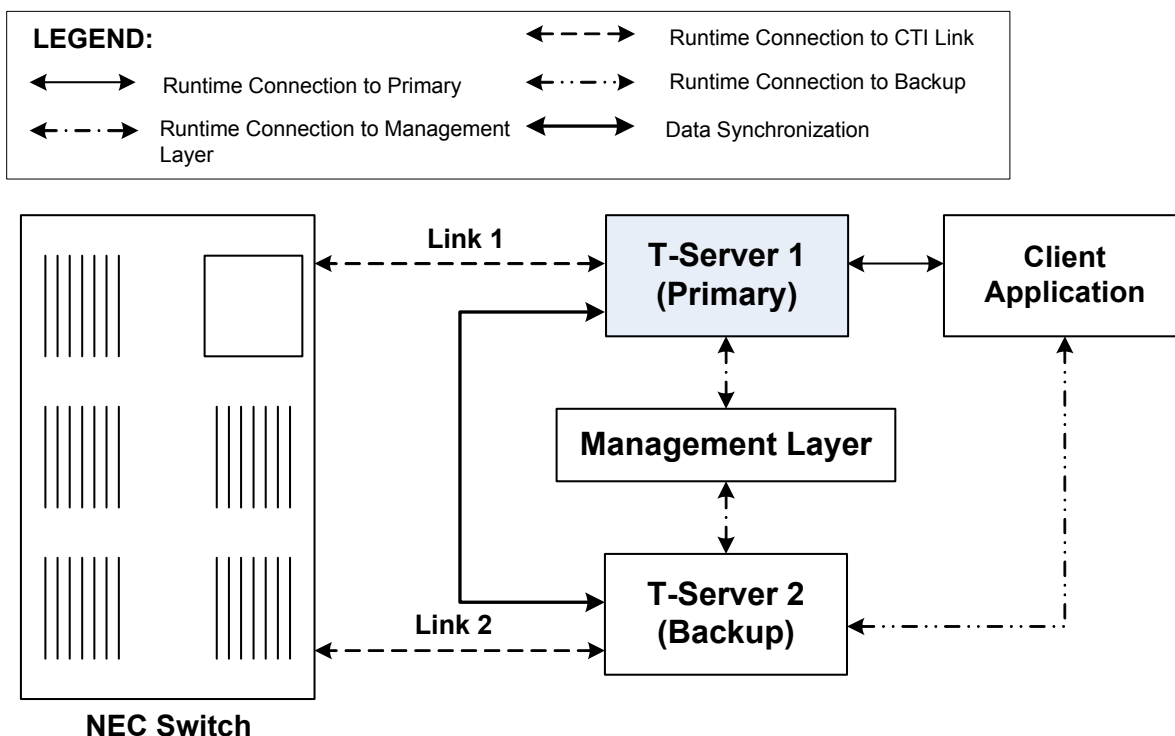


Figure 13: Warm and Hot Standby T-Server Configuration

By default, T-Servers start in Backup mode. The Management Layer is responsible for switching the appropriate T-Server into Primary mode. In most instances, the T-Server that starts first becomes the primary T-Server.

The NEC switch connects directly to both T-Server 1 and T-Server 2 on separate links. After a successful connection, both T-Servers send `CTILinkConnected` and `EventLinkConnected` messages.

Note: In both warm standby and hot standby modes, both primary and backup T-Servers register for, and monitor, all DN's that are configured in the Configuration Layer. Therefore, your switch software release must enable monitoring of a DN by more than one external OAI application. If you are unsure whether your switch is capable of such functionality, check with your switch vendor.

Configuring Unlicensed Devices

In Configuration Manager, you must create configuration objects for the agent DN's, trunks, and IVR ports within a contact center, and you must allot a T-Server license to each device. However, there are some devices on the same switch that are involved with contact center functions, but to which no license

was allocated. These devices must also be configured in Configuration Manager.

Note: Any calls involving unlicensed DNs are tracked incorrectly unless the DNs are configured in Configuration Manager.

Procedure: Configuring unlicensed devices

Purpose: To configure unlicensed DNs in Configuration Manager.

Start of procedure

1. Create a new user account:
 - a. Under the Persons folder, create a new Person—for example, T-ServerNEC.
 - b. Make sure the Is Agent check box is cleared.
 - c. Click OK to save configuration changes.
2. Open the applicable T-Server Application object:
 - a. Click the Security tab.
 - b. In the Log On As section, select This Account and use the Browse button to search for the user account, T-ServerNEC, that you created in [Step 1](#).
 - c. Click Add.
 - d. Click OK to save configuration changes.
 - e. Click Permissions.
 - f. Click Add.
 - g. In the Add dialog box, click Show Users, then select the T-ServerNEC user account.
 - h. In the Type of Access drop-down menu, select Special Access.
 - i. In the newly opened Check Read & Execute dialog box, click OK.
 - j. Click Add.
 - k. Click OK to close the Add window.
 - l. Click OK to close the Object Permission window.
 - m. Click OK to close the T-Server Application Properties window.
3. Grant Read permissions for the T-ServerNEC user account to access the primary (and if applicable, the backup) Configuration Server application.
For each applicable Configuration Server application:
 - a. Click the Security tab.
 - b. Click Permissions.
 - c. Click Add.

- d. In the Add dialog window, click **Show Users**, then select the T-ServerNEC user account.
 - e. In the **Type of Access** drop-down menu, select **Read**.
 - f. Click **Add**.
 - g. Click **OK** to save the configuration changes.
4. If the T-Server user account is configured under any Tenant object, grant Read permissions for the T-ServerNEC user account to access the Tenant by following these steps:
 - a. Open the applicable Tenant.
 - b. Click the **Security** tab.
 - c. Click **Permissions**.
 - d. Click **Add**.
 - e. In the Add dialog window, click **Show Users**, then select the T-ServerNEC user account.
 - f. In the **Type of Access** drop-down menu, select **Read**.
 - g. Click **Add**.
 - h. Click **OK** to save the configuration changes.
5. Under **Switches**, select the **Switch** to which the configured T-Server relates, and select the **DNs** folder.
 - a. Create a new **DNs** folder that will contain the DN's to be monitored by T-Server and its clients—for example, the **LicensedDNs** folder.
 - b. Create another **DNs** folder that will contain the unlicensed DN's to be monitored only by T-Server—for example, the **UnlicensedDNs** folder.
 - c. Under the **LicensedDNs** folder, add all the DN's that will be monitored by T-Server and its clients.
 - d. Under the **UnlicensedDNs** folder, add all the DN's that will be monitored only by T-Server.
 - e. Open the **UnlicensedDNs Properties** dialog box.
 - f. Click the **Security** tab. Click **Permissions**.
 - g. In the **Object Permission** dialog box, remove the **SYSTEM** account, and select **Replace Permissions Recursively**. Click **Add**.
 - h. In the Add dialog box, click **Show Users** and select the user account you added in [Step 1](#). Click **Add**.
 - i. Click **OK** to save configuration changes.
6. Restart the corresponding T-Server.

End of procedure

Next Steps

- There are no next steps.

7

Supported T-Server Features

This chapter describes the telephony functionality T-Server for NEC NEAX/APEX supports and includes these sections:

- [T-Library Functionality, page 159](#)
- [Agent Work Modes, page 168](#)
- [Call Parking, page 168](#)
- [Call Pickup, page 169](#)
- [Force Function Modifier, page 170](#)
- [Multi-ACD Processor Support on a FCCS Network, page 171](#)
- [Multi-Line Operations, page 171](#)
- [RTP Stream Monitoring API, page 172](#)
- [SCF Calling Party Number Selection with OAI Requests, page 175](#)
- [T-Server Error Messages, page 178](#)

T-Library Functionality

The table in this section presents T-Library functionality supported in T-Server for NEC NEAX/APEX switch. The table entries use these notations:

N—Not supported

Y—Supported

I—Supported, but reserved for Genesys Engineering

E—Event only is supported

In [Table 21](#), 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 complete information on the T-Server events, call models, and requests.

Table 21 reflects only the switch functionality used by Genesys software and might not include the complete set of events offered by the switch.

Certain requests listed in Table 21 are reserved for Genesys Engineering and are listed here merely for completeness of information.

Notes describing any specific functionality appear at the end of this table.

Table 21: 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
Registration Requests			
TRegisterAddress ^a		EventRegistered	Y
TUnregisterAddress ^a		EventUnregistered	Y
Call-Handling Requests			
TMakeCall ^b	MakeCallRegular	EventDialing	Y
	MakeCallDirectAgent		N
	MakeCallSupervisorAssist		N
	MakeCallPriority		N
	MakeCallDirectPriority		N
TAnswerCall		EventEstablished	Y
TReleaseCall		EventReleased	Y
TClearCall		EventReleased	N

Table 21: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
THoldCall		EventHeld	Y
TRetrieveCall		EventRetrieved	Y
TRedirectCall ^c		EventReleased	Y
TMakePredictiveCall		EventDialing*, EventQueued	Y
Transfer/Conference Requests			
TInitiateTransfer ^{b,d}		EventHeld, EventDialing*	Y
TCompleteTransfer		EventReleased*, EventPartyChanged	Y
TInitiateConference ^d		EventHeld, EventDialing*	Y
TCompleteConference		EventReleased*, EventRetrieved, EventPartyChanged, EventPartyAdded	Y
TDeleteFromConference		EventPartyDeleted*, EventReleased	Y
TReconnectCall		EventReleased, EventRetrieved*	Y
TAlternateCall		EventHeld*, EventRetrieved	Y
TMergeCalls	MergeForTransfer	EventReleased*, EventPartyChanged	N
	MergeForConference	EventReleased*, EventRetrieved, EventPartyChanged, EventPartyAdded	N
TMuteTransfer ^{b,d}		EventHeld, EventDialing*, EventReleased, EventPartyChanged	Y
TSingleStepTransfer ^{b,e}		EventReleased*, EventPartyChanged	Y
TSingleStepConference		EventRinging*, EventEstablished	Y

Table 21: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
Call-Routing Requests			
TRouteCall ^b	RouteTypeUnknown	EventRouteUsed	I
	RouteTypeDefault		Y
TRouteCall ^b	RouteTypeLabel		N
	RouteTypeOverwriteDNIS		N
	RouteTypeDDD		N
	RouteTypeIDDD		N
	RouteTypeDirect		N
	RouteTypeReject		Y
	RouteTypeAnnouncement		N
	RouteTypePostFeature		N
	RouteTypeDirectAgent		N
	RouteTypePriority		N
	RouteTypeDirectPriority		N
	RouteTypeAgentID		N
	RouteTypeCallDisconnect		N
Call-Treatment Requests			
TApplyTreatment	TreatmentUnknown	(EventTreatmentApplied + EventTreatmentEnd) / EventTreatmentNotApplied	N
	TreatmentIVR		N
	TreatmentMusic		Y
	TreatmentRingBack		Y
	TreatmentSilence		Y
	TreatmentBusy		Y
	TreatmentCollectDigits		N

Table 21: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TApplyTreatment (cont)	TreatmentPlay-Announcement		N
	TreatmentPlay-AnnouncementAndDigits		N
	TreatmentVerifyDigits		N
	TreatmentRecordUser-Announcement		N
	TreatmentDeleteUser-Announcement		N
	TreatmentCancelCall		Y
	TreatmentPlayApplication		N
	TreatmentSetDefaultRoute		N
	TreatmentTextToSpeech		N
	TreatmentTextToSpeech-AndDigits		N
	TreatmentFastBusy		N
	TreatmentRAN		N
TGiveMusicTreatment		EventTreatmentApplied	I
TGiveRingBackTreatment		EventTreatmentApplied	I
TGiveSilenceTreatment		EventTreatmentApplied	I
DTMF (Dual-Tone Multifrequency) Requests			
TCollectDigits		EventDigitsCollected	N
TSendDTMF ^f		EventDTMFSent	Y
Voice-Mail Requests			
TOpenVoiceFile		EventVoiceFileOpened	N
TCloseVoiceFile		EventVoiceFileClosed	N
TLoginMailBox		EventMailBoxLogin	N
TLogoutMailBox		EventMailBoxLogout	N

Table 21: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TPlayVoice		EventVoiceFileEndPlay	N
Agent and DN Feature Requests			
TAgentLogin	See Table 22	EventAgentLogin	Y
TAgentLogout		EventAgentLogout	Y
TAgentSetReady		EventAgentReady	Y
TAgentSetNotReady		EventAgentNotReady	Y
TMonitorNextCall	MonitorOneCall	EventMonitoringNextCall	N
	MonitorAllCalls		Y
TCancelMonitoring		EventMonitoringCanceled	Y
TCallSetForward	ForwardModeNone	EventForwardSet	Y
	ForwardModeUnconditional		Y
	ForwardModeOnBusy		Y
	ForwardModeOnNoAnswer		Y
	ForwardModeOnBusyAnd-NoAnswer		N
	ForwardModeSendAllCalls		Y
TCallCancelForward		EventForwardCancel	Y
TSetMuteOff		EventMuteOff	Y
TSetMuteOn		EventMuteOn	Y
TListenDisconnect		EventListenDisconnected	N
TListenReconnect		EventListenReconnected	N
TSetDNDOOn		EventDNDOOn	Y
TSetDNDOff		EventDNDOff	Y
TSetMessageWaitingOn		EventMessageWaitingOn	Y
TSetMessageWaitingOff		EventMessageWaitingOff	Y

Table 21: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TSetMessageWaitingOff		EventOffHook	Y
		EventOnHook	Y
		EventDNBackInService	N
		EventDNOutOfService	N
Query Requests			
TQuerySwitch ^{a,d}	SwitchInfoDateTime	EventSwitchInfo	Y
	SwitchInfoClassifierStat		N
TQueryCall ^a	CallInfoPartiesQuery	EventPartyInfo	N
	CallInfoStatusQuery		Y
TQueryAddress ^a	AddressInfoAddressStatus	EventAddressInfo	Y
	AddressInfoMsgWaiting-Status		N
	AddressInfoAssociation-Status		N
	AddressInfoCallForwarding-Status		N
	AddressInfoAgentStatus		N
	AddressInfoNumberOf-AgentsInQueue		N
	AddressInfoNumberOf-AvailableAgentsInQueue		N
	AddressInfoNumberOfCalls-InQueue		N
	AddressInfoAddressType		N
	AddressInfoCallsQuery		N
	AddressInfoSendAllCalls-Status		N
	AddressInfoQueueLogin-Audit		N

Table 21: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TQueryAddress ^a	AddressInfoNumberOfIdle-Trunks		N
	AddressInfoNumberOf-TrunksInUse		N
	AddressInfoDatabaseValue		N
	AddressInfoDNStatus		Y
	AddressInfoQueueStatus		Y
TQueryLocation ^a	LocationInfoAllLocations	EventLocationInfo ^g	I
	LocationInfoLocationData		I
	LocationInfoMonitor-Location		I
	LocationInfoCancelMonitor-Location		I
	LocationInfoMonitorAll-Locations		I
	LocationInfoCancelMonitor-AllLocations		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
TCancelReqGetAccess-Number		EventReqGetAccess-NumberCanceled	I
TReserveAgent		EventAgentReserved	Y

Table 21: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
Special Requests			
TSendEvent		EventACK	I
TSendEventEx		EventACK	I
TSetCallAttributes		EventCallInfoChanged	I
TSendUserEvent		EventACK	Y
TPrivateService		EventPrivateInfo	N
Network Attended Transfer/Conference Requests^h			
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

- Only the requester receives a notification of the event associated with this request.
- Since this feature request can be made across locations in a multi-site environment, 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 is a second event response that contains the same `ReferenceId` as the first event. This second event is either `EventRemoteConnect ionSuccess` or `EventRemoteConnect ionFai led`.
- The `TRedirectCall` option is supported for redirecting a call to a Routing Point or ACD pilot; however, it is not possible to redirect a call to another station.
- The `ConsultUserData` key in the `Extensions` attribute for this request is not supported.
- T-Server supports this request only when the mute and single-step transfer functionalities are available on the switch.

- f. The `TSendDTMF` function is supported only on special firmware versions for ICS and IMX release PBXs. Verify with local NEC technicians that SCF 20 is supported by the OAI software release before using this function.
- g. Two subtypes are supported by `EventLocationInfo`: `LocationInfoLocationMonitorCanceled` and `LocationInfoAllLocationsMonitorCanceled`.
- h. All T-Servers support NAT/C requests with the `AttributeHomeLocation` attribute, 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.

Agent Work Modes

Table 22 indicates the types of agent work modes that T-Server for NEC NEAX/APEX supports.

Table 22: Supported Agent Work Modes

Agent Work Mode Type	Feature Request	Supported
AgentWorkModeUnknown	TAgentLogin, TAgentSetReady, TAgentSetNotReady	Y
AgentAfterCallWork	TAgentSetNotReady	Y

Note: The level of T-Server support for each agent work mode depends on the related capabilities of the switch.

Call Parking

T-Server supports the Call Parking feature available with the NEC UNIVERGE SV8500 switch. This feature allows an agent to park an existing call on an internal parking DN within T-Server for NEC NEAX/APEX. Another agent may unpark the call by dialing the appropriate park DN number. The attached data is maintained for calls that are parked or unparked.

Figure 14 on page 169 illustrates how T-Server reports call parking by making a *blind two-step transfer* to a virtual Extension-B. The blind two-step transfer conforms to the Genesys Call Model (GCM) as described in the *Genesys Events and Models Reference Manual* and the *Platform SDK 8.x .NET (or Java) API Reference*. A parked call stays on the virtual extension in the ringing state. When the parked call is answered through the 1PCC operation on Terminal-C, T-Server generates an `EventEstablished` event on the parking line “B” and indicates that the parked call was answered on Terminal-C in the `TermNumber` key-value pair in the `Extensions` attribute.

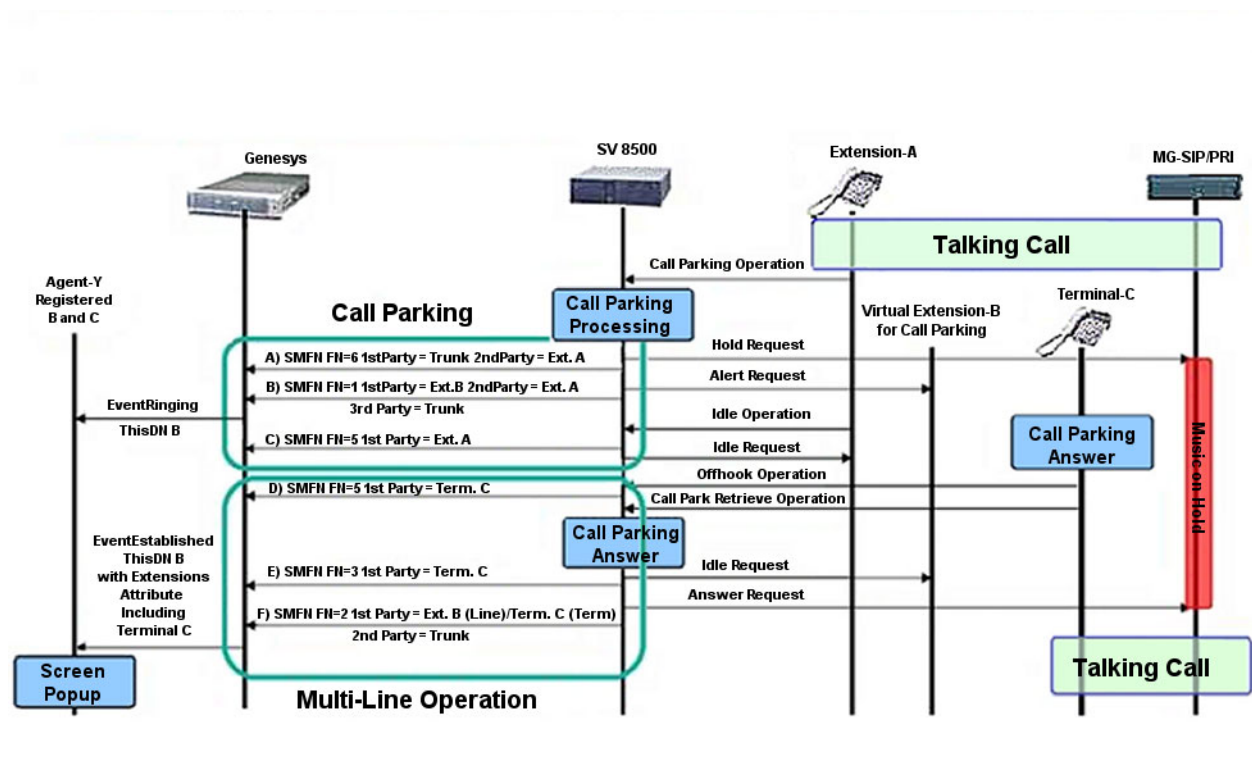


Figure 14: Call Parking

Feature Configuration

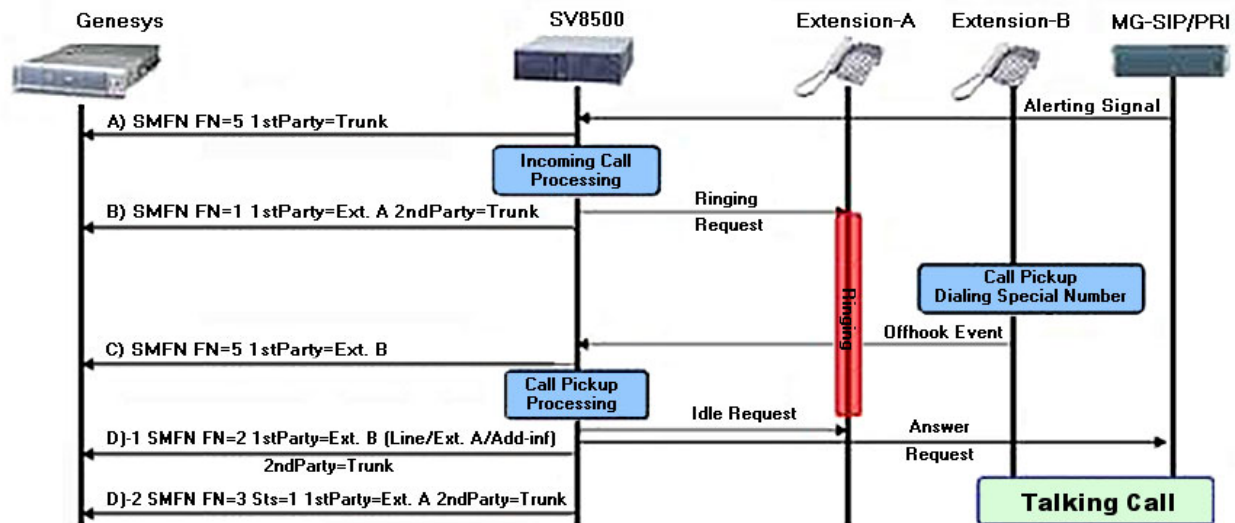
The `enable-answer-term-info` configuration option supports the Call Parking feature.

Call Pickup

T-Server supports the Call Pickup feature available with the NEC UNIVERGE SV8500 switch. This feature allows an agent to pick up calls within their own group. T-Server automatically dials the appropriate Call Pickup group number when the user activates this feature from a NEC phone. The attached data is maintained for calls that are picked up. In call-pickup scenarios, T-Server indicates answered terminal information in the `EventEstablished` event in the `TermNumber` key-value pair of the `Extensions` attribute on the answered line.

Figure 15 on [page 170](#) describes how T-Server reports the call transition from Extension-A to Extension-B as a *call-forward* transfer. The call-forward transfer conforms to the Genesys Call Model (GCM) as described in the

Genesys Events and Models Reference Manual and the Platform SDK 8.x .NET (or Java) API Reference.



Legend:

- A) An inbound call originates from the trunk:
SMFN (FN=5) 1stParty=Trunk
- B) This call rings on Extension-A:
SMFN (FN=1) 1stParty=Extension-A 2nd Party=Trunk
- C) The agent picks up a call from Extension-B by entering a special call pickup sequence on the phone:
SMFN (FN=5) 1stParty=Extension-B
- D) An inbound call is established on Extension-B and Extension-A is released:
SMFN (FN=2) 1stParty=Extension- B (answer line)/Extension-A (called term) 2ndParty=Trunk

Figure 15: Call Pickup Scenario

Force Function Modifier

T-Server clients are able to modify the `TRetrieveCall` function behavior by forcing the execution of OAI requests using the `ForceFunc` modifier. To do this,

you must specify a string value pair in the `Extensions` attribute using the case-sensitive format described in [Table 23](#).

Table 23: Force Function Modifier

Key	Valid Values	Description
ForceFunc	scf10	The scf10 value forces the SCF Ring Stop (FID=10) OAI request invocation.
	scf14	The scf14 value forces the SCF Hooking (FID=14) OAI request invocation.

If the ForceFunc modifier is not specified, or is specified incorrectly, T-Server evaluates the call state and, if acceptable, translates the `TRetrieveCall` request into the `SCF Answer Call` (FID=11) OAI request.

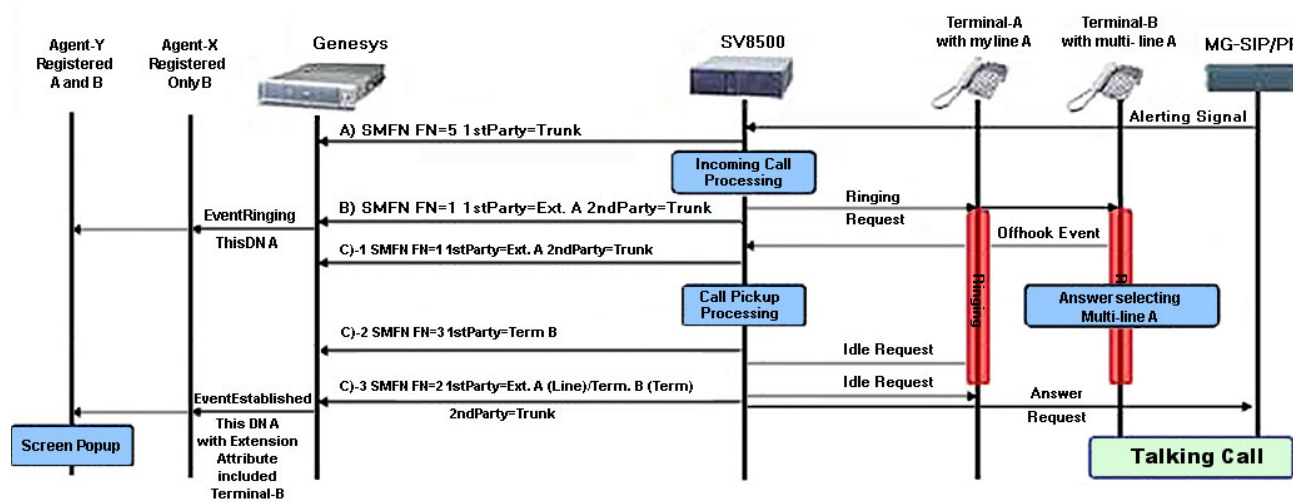
Multi-ACD Processor Support on a FCCS Network

T-Server for NEC NEAX/APEX fully supports a multi-ACD processor on a Fusion Call-Control System (FCCS) network configured as a single switch in Configuration Manager that is handled by one T-Server or a high-availability (HA) pair. Configurations with separate T-Servers connected to the separate nodes and controlling the overlapping subsets of telephony objects are only partially supported and might cause inconsistent reporting in Interaction Concentrator (ICON) or other reporting clients connected to the T-Servers.

Multi-Line Operations

T-Server supports the Multi-Line Operations feature available with the NEC UNIVERGE SV8500 switch. This feature allows incoming calls that are not answered to be sent to the next available line. In multi-line scenarios, T-Server indicates answered terminal information in the `EventEstablished` or `EventRetrieved` event in the `TermNumber` key-value pair of the `Extensions` attribute on the answered line.

Figure 16 on [page 172](#) describes a multi-line operation call scenario.



Legend:

A) An inbound call originates from the trunk:
SMFN (FN=5) 1stParty=Trunk

B) This call rings on Extension-A:
SMFN (FN=1) 1stParty=Extension-A 2ndParty=Trunk

C) The call is answered on the multi-line A of Terminal B:
SMFN (FN=5) 1stParty=Terminal B
SMFN (FN=2) 1stParty=Extension-A(answer line / Terminal-B (answer terminal) 2ndParty=Trunk

D) The call is released on my-line A of Terminal A

Figure 16: Multi-Line Operation Call Scenario

Feature Configuration

The `enable-answer-term-info` configuration option supports the Multi-Line Operations feature.

RTP Stream Monitoring API

T-Server supports the standard Genesys Real-Time Protocol (RTP) stream monitoring API. This API allows integration with third-party recording solutions—for example, Verint. The RTP stream monitoring API is only available on NEC NEAX/APEX switch releases that support SSFM/SSFN (M0ID=16) (IP Registration) OAI messages and are only provided for IP terminals.

Switch Configuration Requirements

You must configure the following switch parameters to enable the SSFM/SSFN (MOID=16) OAI message exchange:

In the MAT configuration:

- Command: ASYD
- Sys: 1
- Index: 79 (OAI)
- Set Service Bit 6 to 0 (zero)

- Command: ASYD
- Sys: 1
- Index: 241 (OAI SMFN Call Event)
- Set Service Bit 2 to 1

- Command: ASYD
- Sys: 1
- Index: 241 (OAI SSFN)
- Set Service Bit 7 to 1

- Command: ASYDL
- Sys: 1
- Index: 877 (OAI SSFN/SSFN IP Registration output)
- Set Service Bit 4 to 1

Feature Configuration

The following configuration options support this feature:

- `report-dterm-ip-addr`
- `rtp-info-password`

Using the RTP Stream Monitoring API

Registering with T-Server

T-Servers communicate to the clients by using events. To get events from T-Server, clients may use the following requests, as described below. For more information on these requests, refer to the *T-Library SDK 7.2 C Developer's Guide*.

TRegisterAll

This request registers all call-related events from all DNs that are configured for T-Server. This request does not result in any voice-monitoring events; however, unless another client was registered for voice-monitoring events, these voice-monitoring events will also go to the client that registers with a TRegisterAll request.

Note: This request was deprecated and is no longer described in the *T-Library SDK 7.2 C Developer's Guide*.

TRegisterAddress

This request registers all call-events from a *specified* DN only. This request also registers the client to receive voice-monitoring events *if* the following Extensions attribute is specified as part of the request:

Specify RTP-PASSWORD as the key-value pair in the Extensions attribute of the request. The key-value pair must have a key equal to the `rtp-info-password` option configured in T-Server as a string value.

If this attribute is missing, or the password value is incorrect, then the application is registered to only receive call-events from the DN.

Note: Under any circumstances, the client application receives an EventRegistered event.

TPrivateService

This request registers all voice-monitoring events from all agent-related DNs that are configured in T-Server. This request does not register any call-related events. The following parameters are required for this request:

- AttributePrivateMsgID (service ID): 2
- Extensions attribute key-value pair: RTP-PASSWORD
- Extensions attribute value for the key: The `rtp-info-password` option configured in T-Server as a string value.

If any of these attributes is missing, or if the password is incorrect, then T-Server is not registered for voice-monitoring events (and receives an EventError message from T-Server). If all parameters are correct then the client application receives an EventACK message from T-Server.

Voice-Monitoring Event Format

Once the client application are registered to T-Server for voice-monitoring events, they will receive events of the following type:

- Event Name: EventUserEvent

- **ThisDN**: T-Server controlled agent-DN that now sends and receives the RTP voice.
- **ConnID**: The Connection ID for the call that sends and receives the RTP voice.
- **UserData**: The user data for this call.
- **Extensions** attributes:
 KEY: VALUE
 OPER: “Started”/”Stopped”
 X-RTP: host:port
 X-PI-RTP (optional): @host:port

where:

- **OPER**: Starts when the RTP streaming has begun for the specified DN. Stops when the RTP stops.
- **PAYLOAD_TYPE**: Not currently used and is reserved for future implementations.
- **X-RTP**: Is the receiving host:port for RTP data for the **OtherDN** attribute, or the customer IP-phone/gateway/stream-manager leg. If X-RTP describes a stream-manager leg for a bi-directional RTP, then X-PI-RTP may be omitted.
- **X-PI-RTP**: Is the receiving host:port for RTP data for the **ThisDN** attribute, or the agent IP-phone. It may be omitted for in cases where X-RTP describes a stream-manager leg for a bi-directional RTP.

The voice-monitoring application receives an **EventUserEvent** (started) message when the RTP stream begins to monitor the DN. The **EventUserEvent** (started) event might be repeated, if the codec or ports are re-negotiated. The **EventUserEvent** (stopped) message is sent when the RTP stream stops, but it might not be at the end of the call as the **EventUserEvent** (started) event might be sent again for the call. When the call ends, an **EventUserEvent** (stopped) event is sent. If the application has registered for call-related events, then these events are also sent.

SCF Calling Party Number Selection with OAI Requests

The SCF Calling Party Number Selection feature, available when a call originates from a trunk by using a SCF OAI (FN=1, 4, 7, or 128) request, enables you to use the NEC UNIVERGE SV8500 switch to specify your calling party number according to your needs in the following two methods:

1. Adding the desired ID following the **@** (0x40) delimiter in the 2nd **Called Number** field of the SCF OAI request.
2. Entering the desired ID in the provided **Calling Number** field (0x89) of the SCF OAI request.

Handling SCF OAI Requests

Figure 17 and Figure 18 on page 176 illustrate the two scenarios where a SCF OAI (FN=1) request is handled by using the above methods:

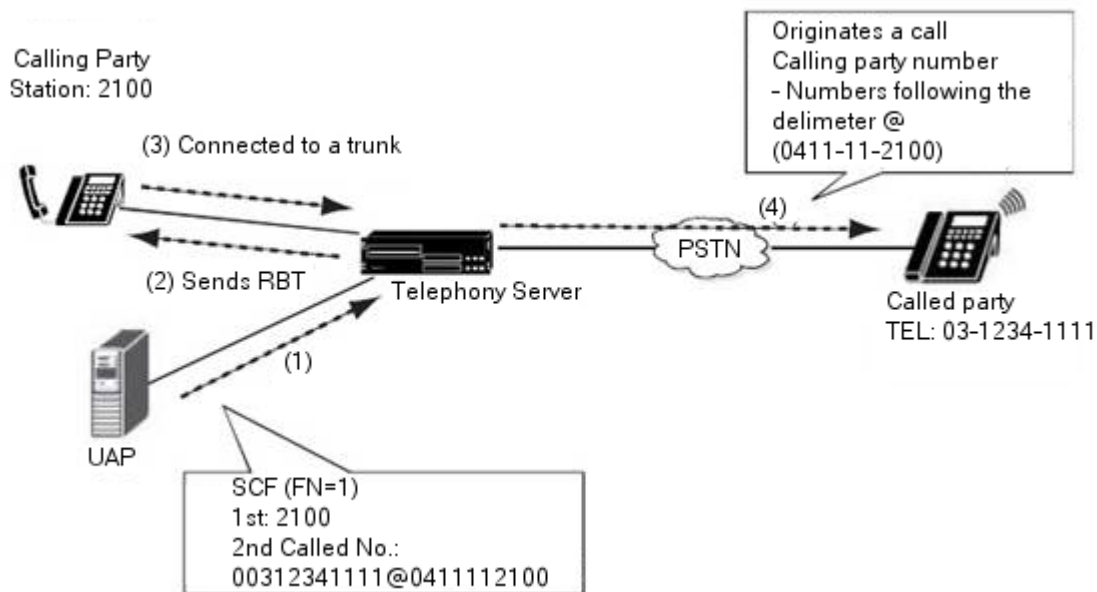


Figure 17: Using the Called Number Parameter in the SCF Request

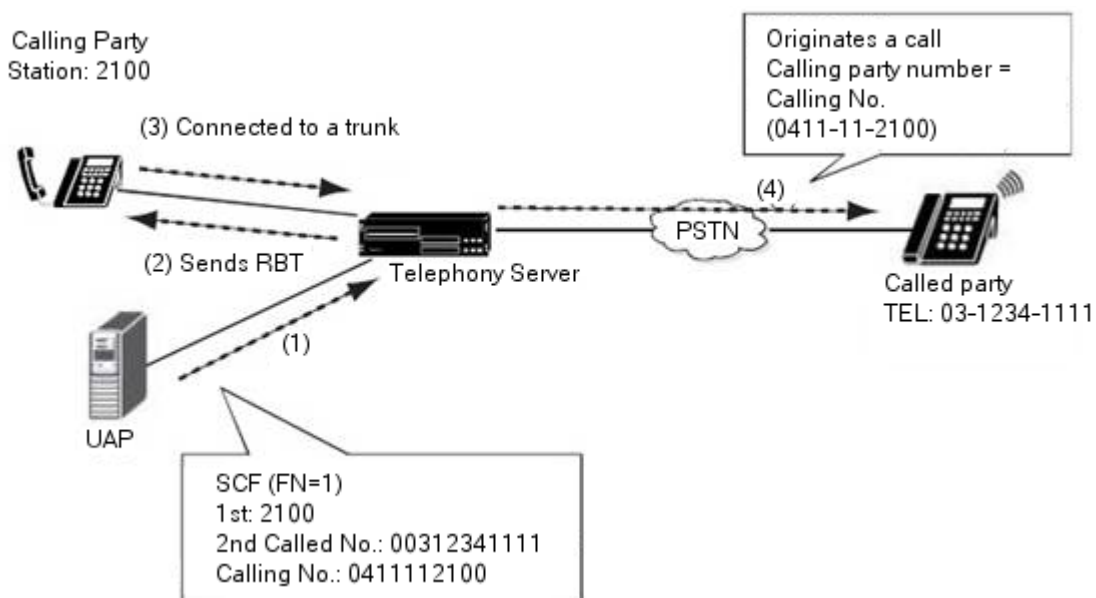


Figure 18: Using the Called Number Information in the SCF OAI Request

This feature is available only for the FP85-107 S4 PABX release, or later.

The following terminals are supported as originating terminals:

- Digital terminal
- IP terminal
- Soft Phone
- SIP Multiple Line terminal
- WLAN Handset (MH Series) (See note below)
- PS (See note below)
- Analog MC (See note below)

Notes:

- This feature can be used on terminals supported by SCF (FN=1, 4, or 7).
- SCF (FN=1) does not support the following terminals:
 - WLAN Handset (MH Series)
 - PS
 - Analog MC

Feature Specific Conditions

The following conditions are specific to this feature:

- The station call must be originated by the SCF (FN=128).
- The destination terminal must be a PS/WLAN Handset (MH Series).
- PRT, BRT, MG(PRI), MG(BRI) and MG(SIP) are supported as outgoing trunks. When a call is originated from an unsupported trunk, the calling party number specified in OAI is disabled. In this scenario, the calling party number that the system normally uses is enabled.
- This feature cannot be used when a call is originated manually from a telephone.
- This feature may be restricted when an invalid calling party number is used. In this case, contact your system administrator.
- When a calling party number is newly defined by this feature on a terminal to which the office data assignment for calling party number has already been made, the number for this feature is sent as a notification to the network side.

Note: The calling party numbers specified by this feature override the caller information of the office data.

- Even when this feature is in use, you can block calling party number from being transmitted to the network

- by setting ASFC/ASFCN SF194=1 (notified as <not displayable>). However, since no Service Feature Restriction Classes (SFC) are provided for the monitored numbers, this blocking function is not available when a call is originated by the SCF (FN=128) request.

Switch Configuration Requirements

You must configure the following switch parameters for T-Server to work properly with the NEC NEAX/APEX switches.

In the MAT configuration:

- Command: ASYDL/ASYDN
- SYS: 1
- INDEX: 1188
- Set Bit 0 to 1 and Bit 1 to 1: This enables the calling party number selection function with SCF and an extended function of SCF (FN=128).

Note: Specifying 0 (disable) in either Bit 0 or Bit 1 disables this feature. If you want to enable this feature, set 1 (enable) to both Bit 0 and Bit 1.

Feature Configuration

The [scf-calling-nbr-encoding](#) configuration option supports the SCF Calling Party Number Selection feature.

T-Server Error Messages

[Table 24](#) presents the complete set of error messages T-Server distributes in the EventError event.

Table 24: Error Messages for T-Server for NEC NEAX/APEX

Code	Error	Description
T-Server Common Part (TSCP) Error Messages		
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	Unrecognized error

Table 24: Error Messages for T-Server for NEC NEAX/APEX (Continued)

Code	Error	Description
51	TERR_UNSUP_OPER	Unsupported operation
52	TERR_INTERNAL	Internal error
53	TERR_INVALID_ATTR	Invalid attribute value
54	TERR_NO_SWITCH	No connection to the switch
55	TERR_PROTO_VERS	Incorrect protocol version
56	TERR_INV_CONNID	Invalid ConnectionID
57	TERR_TIMEOUT	Timeout expired
58	TERR_OUT_OF_SERVICE	Out of service
59	TERR_NOT_CONFIGURED	DN is not configured in the Configuration Database
T-Server-Specific Error Messages		
91	TERR_TERM_PTY_BUSY	Terminating Party Busy
182	TERR_TARG_NOT_AGENT	Target set is not an ACD agent
185	TERR_SET_WRONG_STATE	A set is in a wrong state for invocation
218	TERR_INTERNAL_ERR	Internal processing error within the switch
269	TERR_API_NOT_SUP	An API function is not supported
1101	TERR_CSTA_RO_GEN_UNREC_ADPU	Unrecognized APDU
1102	TERR_CSTA_RO_GEN_MISTYP_ADPU	Mistyped APDU
1103	TERR_CSTA_RO_GEN_BAD_STR_ADPU	Badly structured APDU
1110	TERR_CSTA_RO_INV_DUPLICATE	Duplicate invocation
1111	TERR_CSTA_RO_INV_UNREC_OP	Unrecognized operation
1112	TERR_CSTA_RO_INV_MISTYP_ARG	Mistyped argument
1113	TERR_CSTA_RO_INV_RES_LIMIT	Resource limitation
1120	TERR_CSTA_RO_RES_UNREC_INVOK	Unrecognized invocation
1121	TERR_CSTA_RO_RES_RESP_UNEXP	Result response unexpected
1122	TERR_CSTA_RO_RES_MISTYP_RES	Mistyped result

Table 24: Error Messages for T-Server for NEC NEAX/APEX (Continued)

Code	Error	Description
1130	TERR_CSTA_RO_ERR_UNREC_INVOK	Unrecognized invocation
1131	TERR_CSTA_RO_ERR_RESP_UNEXP	Error response unexpected
1132	TERR_CSTA_RO_ERR_UNREC_ERROR	Unrecognized error
1133	TERR_CSTA_RO_ERR_UNEXP_ERROR	Unexpected error
1134	TERR_CSTA_RO_ERR_MISTYP_PARAM	Mistyped parameter
1140	TERR_CSTA_OPER_GENERIC	Generic operation error
1141	TERR_CSTA_OPER_REQ_INCOMPAT	Request incompatible with object
1144	TERR_CSTA_OPER_INV_CALLING	Invalid calling device
1145	TERR_CSTA_OPER_INV_CALLED	Invalid called device
1150	TERR_CSTA_OPER_INV_CALL_ID	Invalid call identifier specified
1151	TERR_CSTA_OPER_INV_DEV_ID	Invalid device identifier
1152	TERR_CSTA_OPER_INV_CONN_ID	Invalid Connection ID is specified
1154	TERR_CSTA_OPER_INV_FEATURE	Invalid feature
1160	TERR_CSTA_INCOMP_GENERIC	Generic state incompatibility error ^a
1161	TERR_CSTA_INCOMP_INCORR_STATE	Incorrect object state
1172	TERR_CSTA_SYSRES_RES_BUSY	Resource busy
1191	TERR_CSTA_SECUR_SEQ_NUM_VIOL	Sequence number violated
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

Table 24: Error Messages for T-Server for NEC NEAX/APEX (Continued)

Code	Error	Description
1907	TERR_NATC_UNEXP_NPS	Unexpected request TNetworkPrivateService
1908	TERR_NATC_UNEXP_MSG	Unexpected message

- a. This error is only raised when a station is in an unmatched state after a dual redundancy restoration.

8

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:

- [Setting Configuration Options, page 183](#)
- [Mandatory Options, page 184](#)
- [log Section, page 184](#)
- [log-extended Section, page 198](#)
- [log-filter Section, page 200](#)
- [log-filter-data Section, page 200](#)
- [security Section, page 201](#)
- [sml Section, page 201](#)
- [common Section, page 203](#)
- [Changes from 8.0 to 8.1, page 203](#)

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:

<code>all</code>	All log events (that is, log events of the Standard, Trace, Interaction, and Debug levels) are generated.
<code>debug</code>	The same as <code>all</code> .
<code>trace</code>	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.
<code>interaction</code>	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.
<code>standard</code>	Log events of the Standard level are generated, but log events of the Interaction, Trace, and Debug levels are not generated.
<code>none</code>	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 190](#).

Note: For definitions of the Standard, Interaction, Trace, and Debug log levels, refer to the *Framework 8.0 Management Layer User’s Guide*, *Framework 8.0 Genesys Administrator Help*, or to *Framework 8.0 Solution Control Interface Help*.

buffering

Default Value: `true`

Valid Values:

<code>true</code>	Enables buffering.
<code>false</code>	Disables buffering.

Changes Take Effect: Immediately

Turns on/off operating system file buffering. The option is applicable only to the `stderr` and `stdout` output (see [page 190](#)). Setting 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:

<code>false</code>	No segmentation is allowed.
<code><number> KB</code> or <code><number></code>	Sets the maximum segment size, in kilobytes. The minimum segment size is <code>100 KB</code> .
<code><number> MB</code>	Sets the maximum segment size, in megabytes.
<code><number> hr</code>	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:

<code>false</code>	No expiration; all generated segments are stored.
<code><number> file</code> or <code><number></code>	Sets the maximum number of log files to store. Specify a number from <code>1–1000</code> .
<code><number> day</code>	Sets the maximum number of days before log files are deleted. Specify a number from <code>1–100</code> .

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 will be automatically reset to `10`.

keep-startup-fileDefault Value: `false`

Valid Values:

<code>false</code>	No startup segment of the log is kept.
<code>true</code>	A startup segment of the log is kept. The size of the segment equals the value of the <code>segment</code> option.
<code><number> KB</code>	Sets the maximum size, in kilobytes, for a startup segment of the log.
<code><number> MB</code>	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 will be equal to the size of the regular log segment defined by the `segment` option. The value of this option will be 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_formatDefault Value: `short`

Valid Values:

<code>short</code>	An application uses compressed headers when writing log records in its log file.
<code>full</code>	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 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 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. 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 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 8.0 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 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 190](#)). The new snapshot overwrites the previously written data. If the application terminates abnormally, this file will contain 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 190](#).

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 network log output. If you change the 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 194](#).

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):

<code>stdout</code>	Log events are sent to the Standard output (<code>stdout</code>).
<code>stderr</code>	Log events are sent to the Standard error output (<code>stderr</code>).
<code>network</code>	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 <code>all</code> log level option to the network output enables an application to send log events of the <code>Standard</code> , <code>Interaction</code> , and <code>Trace</code> levels to Message Server. Debug-level log events are neither sent to Message Server nor stored in the Log Database.
<code>memory</code>	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
<code>[filename]</code>	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):

<code>stdout</code>	Log events are sent to the Standard output (<code>stdout</code>).
<code>stderr</code>	Log events are sent to the Standard error output (<code>stderr</code>).
<code>network</code>	Log events are sent to Message Server, which resides anywhere on the network, and Message Server stores the log events in the Log Database.
<code>memory</code>	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
<code>[filename]</code>	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):

<code>stdout</code>	Log events are sent to the Standard output (<code>stdout</code>).
<code>stderr</code>	Log events are sent to the Standard error output (<code>stderr</code>).
<code>network</code>	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
<code>memory</code>	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
<code>[filename]</code>	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):

<code>stdout</code>	Log events are sent to the Standard output (<code>stdout</code>).
<code>stderr</code>	Log events are sent to the Standard error output (<code>stderr</code>).
<code>network</code>	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
<code>memory</code>	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
<code>[filename]</code>	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):

<code>stdout</code>	Log events are sent to the Standard output (<code>stdout</code>).
<code>stderr</code>	Log events are sent to the Standard error output (<code>stderr</code>).
<code>network</code>	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
<code>memory</code>	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
<code>[filename]</code>	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):

<code>stdout</code>	Log events are sent to the Standard output (<code>stdout</code>).
<code>stderr</code>	Log events are sent to the Standard error output (<code>stderr</code>).
<code>memory</code>	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
<code>[filename]</code>	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 Technical Support 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 Technical Support 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 Technical Support.

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 8.0 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:

- | | |
|-----------------------|---|
| <code>0</code> | This method of detecting an unresponsive application is not used by this application. |
| <code>3-604800</code> | 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 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 Technical Support.
• 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 Technical Support.

Changes from 8.0 to 8.1

There are no changes in common configuration options between 8.0 and 8.1 releases.

9

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:

- [Setting Configuration Options, page 205](#)
- [Mandatory Options, page 206](#)
- [TServer Section, page 206](#)
- [license Section, page 211](#)
- [agent-reservation Section, page 214](#)
- [extrouter Section, page 215](#)
- [backup-sync Section, page 226](#)
- [call-cleanup Section, page 228](#)
- [Translation Rules Section, page 229](#)
- [security Section, page 230](#)
- [Timeout Value Format, page 230](#)
- [Changes from Release 8.0 to 8.1, page 231](#)

T-Server also supports common log options described in Chapter 8, “Common Configuration Options,” on [page 183](#).

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 will be reported for all calls for which it is available. When this option is set to suppressed, the ANI attribute will not be reported for any calls. When this option is set to inbound-calls-only, the ANI attribute will be 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 230](#).

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:

<code>separate</code>	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.
<code>inherited</code>	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.
<code>joint</code>	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 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 96](#)

Specifies whether DNs associated with the `Switch`, `Switching Office`, or `Tenant` objects will be 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` will be in the T-Server monitoring scope. With a value of `office`, all DNs associated with the switches that are within the `Switching Office` will be in the T-Server monitoring scope. With a value of `switch`, all DNs associated with the `Switch` will be 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):

<code>+/-iscc</code>	Turns on/off the writing of information about Inter Server Call Control (ISCC) transactions.
<code>+/-cfg\$dn</code>	Turns on/off the writing of information about DN configuration.
<code>+/-cfgserv</code>	Turns on/off the writing of messages from Configuration Server.
<code>+/-passwd</code>	Turns on/off the writing of <code>AttributePassword</code> in <code>TEvents</code> .
<code>+/-udata</code>	Turns on/off the writing of attached data.
<code>+/-devlink</code>	Turns on/off the writing of information about the link used to send CTI messages to the switch (for multilink environments).
<code>+/-sw</code>	Reserved by Genesys Engineering.
<code>+/-req</code>	Reserved by Genesys Engineering.
<code>+/-callops</code>	Reserved by Genesys Engineering.
<code>+/-conn</code>	Reserved by Genesys Engineering.
<code>+/-client</code>	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:

<code>main-only</code>	T-Server attaches user data from the remaining call only.
<code>merged-only</code>	T-Server attaches user data from the merging call.
<code>merged-over-main</code>	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.
<code>main-over-merged</code>	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 207](#).)

propagated-call-type

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Switch Partitioning” on [page 96](#)

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 212](#).

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 Technical Support.

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.

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.
 - 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 7, “DNs and Agent Logins,” [page 38](#).
-

License Checkout

[Table 25](#) shows how to determine the number of seat-related DN licenses that T-Server attempts to check out. See the examples on [page 213](#).

Table 25: 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
max (or 0)	0	0
x	max	x

Table 25: License Checkout Rules (Continued)

Options Settings ^a		License Checkout ^b
num-of-licenses	num-sdn-licenses	
x	y	min (y, x)
x	0	0

- 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.
- 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 DN's
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 DN's
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 DN's
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 DN's
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 28](#) 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:

- | | |
|--------------------|---|
| <code>true</code> | T-Server rejects subsequent requests. |
| <code>false</code> | 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 230](#).

Changes Take Effect: Immediately

Specifies the interval that agent reservation requests are collected before a reservation is granted. During this 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 230](#).

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 217](#)
- [Transfer Connect Service Options, page 221](#)
- [ISCC/COF Options, page 222](#)
- [Event Propagation Options, page 224](#)
- [Number Translation Option, page 225](#)
- [GVP Integration Option, page 226](#)

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:

- | | |
|--------------------|--|
| <code>true</code> | ISCC does not process (match) an inbound call that has already been processed (matched). |
| <code>false</code> | 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 230](#).

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:

- | | |
|--------------------|--|
| <code>true</code> | <code>EventPartyChanged</code> is generated. |
| <code>false</code> | <code>EventPartyChanged</code> 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:

<code>active</code>	The values of <code>UserData</code> and <code>ConnID</code> attributes are taken from the consultation call.
<code>original</code>	The values of <code>UserData</code> and <code>ConnID</code> attributes are taken from the original call.
<code>active-data-original-call</code>	The value of the <code>UserData</code> attribute is taken from the consultation call and the value of <code>ConnID</code> attribute is taken from the original call.
<code>current</code>	<p>If the value of <code>current</code> is specified, the following occurs:</p> <ul style="list-style-type: none"> • Before the transfer or conference is completed, the <code>UserData</code> and <code>ConnID</code> attributes are taken from the consultation call. • After the transfer or conference is completed, <code>EventPartyChanged</code> is generated, and the <code>UserData</code> and <code>ConnID</code> 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-uu`, `reroute`, `direct-callid`, `direct-uu`, `direct-network-callid`, `direct-notoken`, `direct-digits`, `direct-ani`, `dnis-pool`, `pullback`

Valid Values: `route`, `route-uu`, `reroute`, `direct-callid`, `direct-uu`, `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 75](#) 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 230](#).

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 230](#).

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 230](#).

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-modeDefault 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-maximumDefault Value: `0`

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 230](#).

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 230](#).

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

Valid Values: See “Timeout Value Format” on [page 230](#).

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 230](#).

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 will be

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 230](#).

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`

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 230](#).

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 230](#).

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 8.0 Management Layer User’s Guide*.

This section must be called `call-cleanup`.

cleanup-idle-tout

Default Value: 0

Valid Values: See “Timeout Value Format” on [page 230](#).

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 230](#).

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 230](#).

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 84](#). See “Configuring Number Translation” on [page 91](#) 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 26](#) 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 26: 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 206 .

10

T-Server-Specific Configuration Options

This chapter describes the configuration options that are unique to the T-Server for NEC NEAX/APEX. It includes these sections:

- [Mandatory Options, page 233](#)
- [TServer Section, page 235](#)
- [PPN-Mapping Section, page 259](#)
- [CTI-Link Section, page 260](#)
- [Changes from Release 8.0 to 8.1, page 261](#)

To establish a link connection, configure the link options that are applicable to the connection protocol used in your environment (for example, TCP/IP).

The options common to all T-Servers are described in Chapter 8, “Common Configuration Options,” on [page 183](#) and Chapter 9, “T-Server Common Configuration Options,” on [page 205](#).

You set configuration options in Configuration Manager in the corresponding sections on the `Options` tab for the T-Server `Application` object.

Mandatory Options

[Table 27](#) lists the options that you must configure for basic T-Server operation. All other options in this chapter are configured to enable T-Server to support various features.

To establish a link connection, simply configure the link options (TCP/IP) that are applicable to the connection protocol used in your environment.

Table 27: Mandatory Options

Option Name	Default Value	Details
T-Server Section		
link- <i>n</i> -name	No default value	Specifies the section name containing the configuration options assigned to that link, where <i>n</i> is a consecutive number for a CTI-link. See the description on page 244 .
default-announcement-trunk	No default value	Specifies the announcement trunk when no music port is selected. See the description on page 236 .
CTI-Link Section		
protocol	No default value	Specifies the connection protocol T-Server uses in communicating with the switch. See the description on page 261 .
hostname	No default value	Specifies the host of the link according to the switch configuration. See the description on page 260 .
port	No default value	Specifies the TCP/IP port of the link according to the switch configuration. See the description on page 260 .
PPN-Mapping Section		
<n>	No default value	Specifies a unique number between 0 (zero) and 1999. See the description on page 259 .

TServer Section

This section must be called TServer.

The configuration options are presented in the alphabetical order within the section.

aarq-logical-port

Default Value: 0 (zero)

Valid Values: 0-16

Changes Take Effect: After the link is restarted

Specifies logical port in the AARQ open association request. When set to a value of 0, the logical port is not used. A value must be set for this option if dual redundancy mode (AARE Result=11) is to be used to preserve the T-Server subscription data.

Note: During the failover process, devices may enter into an unmatched state. A device in an unmatched state causes the agent to be logged out by sending a forced logoff message. T-Server then distributes an EventAgentLogout event and places ReasonCode forced_logout within AttributeExtensions. Calls from unmatched DNs may also be released if a T-Server client invokes a CTI request on a device in an unmatched state. In this scenario, T-Server distributes an EventError message with an error code of 1160.

alive-message-timer

Default Value: 60000

Valid Values: Any integer from 0-100000

Changes Take Effect: Immediately

Specifies the time interval (in milliseconds) that T-Server waits before sending an alive message (heartbeat) to the switch. T-Server sends the alive message only when there is no request/response activity within the alive-message-timer interval. A corresponding heartbeat response is expected from the switch within the alive-message-timer interval. If the heartbeat response is not received for three consecutive heartbeat requests, T-Server resets the link. If set to 0 (zero), T-Server stops sending alive messages to the switch.

alive-message-tolerance

Default Value: 3

Valid Values: Any integer from 0-65535

Changes Take Effect: Immediately

Specifies the number of heartbeat time intervals (in milliseconds) that can expire before the link is reset.

answer-mode

Default Value: 1

Valid Values: Any integer from 1–4

Changes Take Effect: Immediately

Determines the answer behavior of all T-Server call origination functions (TMakeCall, TInitiateTransfer, TInitiateConference, TRouteCall, and so on) when the destination is another internal handset.

If you set this option to either 1 or 2, the destination handset rings, waiting for the destination party to answer (Manual Answer Mode). The 1 and 2 settings each provide different ring tones. If you set the option to either 3 or 4, the destination handset automatically answers, going off-hook without manual assistance (Auto Answer Mode). The 3 and 4 settings each provide different zip tones.

backup-reset-threshold-interval

Default Value: 0 (zero)

Valid Values: 0, or any positive integer

Changes Take Effect: Immediately

Specifies the time interval (in seconds) that T-Server, when operating in Backup mode, waits before releasing its internal state after it has lost a synchronization link with the primary T-Server. When set to 0, the backup T-Server reset is disabled.

default-announcement-trunk

Default Value: Mandatory option. No default value.

Valid Values: Any valid announcement trunk

Changes Take Effect: Immediately

Specifies the announcement trunk to which T-Server sends a call that is being routed, if a T-Server client requests that the call be provided with music treatment, but no music port is specified in Universal Routing Server. When the announcement trunk is used for music, configure it to loop in the switch configuration in the DAT card. You must configure the announcement trunk in the Configuration Layer.

Note: You must specify a value for this option.

dnd-mode

Default Value: 2

Valid Values:

- 1 Use an SSFR OAI Facility request to set DND (Do-Not-Disturb).
- 2 Use an RCF OAI Facility request to set DND.

Changes Take Effect: Immediately

Specifies which switch mechanism is used to set DND on or off for an agent handset. Which option you select depends on the type of switch.

enable-alive-message

Default Value: `true`

Valid Values:

- `true` Heartbeats are sent periodically between the switch and T-Server.
- `false` No heartbeats are sent between the switch and T-Server.

Changes Take Effect: After T-Server or the link is restarted

Specifies whether T-Server opens a connection to the switch that supports heartbeats. Setting this option to `true` enables T-Server and the switch to detect hardware and network failures. However, in some environments this can cause T-Server to periodically disconnect from the switch and send `EventLinkDisconnected` messages to the clients. If this happens frequently during normal operation, reset this option to `false`.

enable-answer-term-info

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: After the link is restarted

Related Features: “Call Parking” on [page 168](#), “Call Pickup” on [page 169](#), and “Multi-Line Operations” on [page 171](#)

If this option is set to `true`, T-Server enables the `TermNumber Extensions` attribute in the `EventEstablished` or `EventRetrieved` event that corresponds to the `1st Party Term Info` of the `OAI SMFN (FN=2)` event. If this option is set to `false`, this attribute is not enabled.

In multi-line and call-parking scenarios, `TermNumber` indicates the answered terminal information.

enable-dnd-monitor

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: After the link is restarted

If the `enable-dnd-monitor` option is set to `true`, T-Server requests DND monitoring for digital terminals when a link is restarted by sending a `SSFM FID+2 (DND)` monitoring request to the switch during DN registration. If the DND monitoring request was successful, the PBX sends out a `SSFN FID=3 OAI` event notification when the `DTerm`’s DND status is changed. This request is then translated by T-Server into the corresponding `EventDNDOn` and `EventDNDOff` events. If the `enable-dnd-monitor` option is set to `false`, then DND monitoring is disabled.

enable-infolink-heartbeatDefault Value: `false`

Valid Values:

`true` Enables Infolink polling.
`false` Disables Infolink polling.

Changes Take Effect: Immediately

If this option is set to a value of `true`, Infolink polling is controlled by the following T-Server options: `enable-alive-message`, and `alive-message-timer`. T-Server sends Infolink query agent status (`lc`) requests to the link to poll the Infolink status. Any response to the request indicates Infolink is alive. When set to a value of `false`, Infolink polling is disabled.

enable-hotline-term-infoDefault Value: `false`Valid Values: `true`, `false`

Changes Take Effect: After T-Server is restarted

If this option is set to `true`, when a call is originated from the hotline, T-Server propagates the primary terminal info (mainline) of the hotline with an `EventDialing` event in the `Extensions` attribute as a TKVList key-value pair in the following format: `TermNumber '<hotline's primary terminal number>'`.

- If a hotline is configured as `STATION_HOTLINE_SUBLINE`, T-Server uses the mainline number specified in the `Configuration Layer Association` field.
- If a hotline is configured as `STATION_HOTLINE`, T-Server extracts the mainline number from the OAI message.

enable-retrieve-on-releaseDefault Value: `true`

Valid Values:

`true` T-Server automatically generates an `EventRetrieved` message when a consultation call is released.
`false` T-Server waits for event notification from the PBX that an original call is retrieved before generating an `EventRetrieved` message.

Changes Take Effect: Immediately

Defines how T-Server generates an `EventRetrieved` message when a consultation call is released.

enable-query-addr-statusDefault Value: `false`Valid Values: `true`, `false`

Changes Take Effect: After the link is restarted

When set to `true`, this option enables T-Server to query `DTerm` device status by sending a `SMFR status check OAI` request during device registration on the switch. This discovered device status becomes available to the T-Server clients

in the `Extensions` attribute, via the status key-value pair of the `EventRegistered` and `EventAddressInfo` events.

enable-query-dnd-status

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: After the link is restarted

When set to `true`, this option enables T-Server to query the Do-Not-Disturb device status by sending an SSFM FID=3 `Status inquiry` OAI request during the device registration on the switch. This identified device status is communicated to the T-Server clients in the `Extensions` attribute, via the status key-value pair of the `EventRegistered` and `EventAddressInfo` events.

enable-query-fwd-status

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: After the link is restarted

When set to `true`, this option enables T-Server to query the call-forward device status by sending an SSFM FID=1, 6, 7 `Status inquiry` OAI request during the device registration on the switch. This identified device status is communicated to the T-Server clients in the `Extensions` attribute, via the status key-value pair of the `EventRegistered` and/or `EventAddressInfo` events.

enable-query-mwl-status

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: After the link is restarted

When set to `true`, this option enables T-Server to query the message-waiting-lamp device status by sending an SSFM FID=5 `Status inquiry` OAI request during device registration on the switch. This identified device status is communicated to the T-Server clients in the `Extensions` attribute, via the status key-value pair of the `EventRegistered` and `EventAddressInfo` events.

enable-split-prefix

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: After T-Server is restarted.

Enables split prefixing with the tenant number when set to `true`. (See “Setting DN Properties” on [page 138](#) for more details on the Split configuration.)

Note: When you enable this option, you must configure Split ACD Queues and Routing Points using the format `<YXXX>`, where `Y` is a switch tenant number and `XXX` is a Split number within the Tenant.

Warning! DN numbers must be unique in the Configuration Layer. Therefore, before using split prefixing, ensure that the derived Split numbers do not overlap with the numbering of other telephony objects, such as Extensions, ACD Positions, and so on.

enable-ssc

Default Value: `false`

Valid Values:

`true` Enables use of `TSingleStepConference` for silent call monitoring

`false` Disables use of `TSingleStepConference` for silent call monitoring

Changes Take Effect: Immediately

Specifies whether the `TSingleStepConference` function is to be used for initiating silent call monitoring.

Note: This option is provided for backward compatibility only. Setting this option to `true` may affect T-Server processing of silent call monitoring scenarios initiated by a `TSingleStepConference` request. Under regular circumstances, the `TMonitorNextCall` function must be used for initiating silent call monitoring.

extrouter-default-dn

Default Value: No default value

Valid Values: Any valid dialable destination

Changes Take Effect: Immediately

Specifies the destination to which T-Server automatically default-routes a call when, for example, an ISCC-generated `TRouteCall` request for an external Routing Point fails to deliver a call to the requested destination because the destination device is invalid or busy.

failed-registration-timeout

Default Value: `5000`

Valid Values: Any positive integer

Changes Take Effect: After T-Server is restarted

Specifies the expected interval (in milliseconds) in which to receive responses from the switch if T-Server receives no response. After the interval defined in this option, T-Server resends a registration request. In rare circumstances, the switch ignores DN registration requests sent from the T-Server—that is, the switch sends no positive or negative acknowledgment to T-Server.

Note: Always set this option to `5000`, unless Genesys Customer Care directs you to do otherwise.

fail-no-ppn-routeDefault Value: `true`

Valid Values:

- `true` T-Server does not route or transfer a call to an agent's ACD Position, if a personal pilot number (PPN) has not been configured.
- `false` T-Server routes or transfers the call directly to the ACD Position, if a PPN has not been configured.

Changes Take Effect: Immediately

Applies only to large-system switches when you have set the [use-infolink](#) option to `true`, with either of the [make-call-agent-ppn](#), [transfer-agent-ppn](#), or [route-agent-ppn](#) options set to `true`.

Note: Routing or transferring calls directly to the ACD Position may cause problems within the NEC ACD.

fast-transferDefault Value: `false`

Valid Values:

- `true` T-Server completes a transfer without waiting for a response from the switch about the transfer initiation completion.
- `false` T-Server completes a transfer after it receives a response from the switch about the successful transfer initiation completion.

Changes Take Effect: Immediately

Specifies whether T-Server uses the fast transfer method to execute `TMuteTransfer` and `TSingleStepTransfer` requests.

Note: Always set this option to `false`, unless Genesys Customer Care directs you to do otherwise.

fast-transfer-ppnDefault Value: `false`

Valid Values:

- `true` T-Server uses the fast transfer method to execute `TMuteTransfer`, `TSingleStepTransfer`, and `TRedirectCall` requests, which is done in one step by using the SCF Ring Stop (FID=10) request.
- `false` T-Server uses a two-step operation.

Changes Take Effect: Immediately

Specifies what method T-Server uses to perform `TMuteTransfer`, `TSingleStepTransfer`, or `TRedirectCall` request, if the selected destination is a PPN (Personal Pilot Number). Refer to the following configuration options

related to PPN functionality: [transfer-agent-ppn](#), [query-agent-ppn](#), and [fail-no-ppn-route](#).

Note: Always set this option to `false`, unless otherwise directed by Genesys Customer Care.

fix-double-hold

Default Value: `false`

Valid Values: `true`, `false`

Change Take Effect: Immediately

With a value of `true`, T-Server fixes the OAI inconsistency in scenarios where two agents put a call on hold and retrieve the call, but only one SMFN2 (STS=2) OAI request is propagated by the switch. When T-Server receives a SMFN2 (STS=2) OAI request, it determines whether the other party is in the Hold state and generates an `EventRetrieved` message when needed. With a value of `false`, T-Server does no extra checking and generates events based solely on switch messages.

fix-supervisor-release

Default Value: `false`

Valid Values:

`true` T-Server sends two SCF Release (FID=2) requests to the switch. See “`snd-scf2-delay`” on [page 253](#).

`false` T-Server sends only one SCF Release (FID=2) request to the switch.

Changes Takes Effect: Immediately

Specifies whether T-Server, when executing a `TReleaseCall` request for an ACD Supervisor party, sends one or two SCF Release (FID=2) requests to the switch.

There is known switch-specific behavior, where, in order to hang up (via CTI) an ACD Supervisor monitoring call, two SCF Release (FID=2) requests have to be sent for the ACD Supervisor line. The event notification indicating ACD Supervisor release is sent only after the second SCF Release (FID=2) request (when the ACD Supervisor line actually goes on hook). Only one SCF Release (FID=2) request is required in order to release a call and obtain notification for a regular D-Term (Agent).

flow-control-rate-runtime

Default Value: `0`

Valid Values: `0-10000`

Changes Take Effect: Immediately

Sets the maximum number of requests per second that T-Server sends to the switch for processing during the regular runtime after the initialization phase is complete. A value of `0` (zero) sets no restriction to this number.

flow-control-rate-startup

Default Value: 20

Valid Values: 0-10000

Changes Take Effect: Immediately

Sets the maximum number of requests per second that T-Server sends to the switch for processing during the link start-up when DNs are registered. A value of 0 (zero) sets no restriction to this number.

flow-control-link-alarm-gap

Default Value: 10

Valid Values: Any unsigned integer

Changes Take Effect: Immediately

Sets the percentage of the currently active `flow-control-rate` option value which must be subtracted from the percentage of the currently active `flow-control-rate` option value given by the `link-alarm-high` option to obtain the message rate, falling below this rate causes T-Server to generate the 24513 (GCTM_FC_LINK_ALARM_LOW) LMS message. This alarm is raised to reset the 24512 (GCTM_FC_LINK_ALARM_HIGH) LMS message and indicates that the link is no longer overloaded. If this value is greater than the `link-alarm-high` option value, then the rate must be 0 (zero) to raise the 24513 alarm. The currently active `flow-control-rate` option is `flow-control-rate-startup` or `flow-control-rate-runtime` depending on whether initialization is occurring or is complete.

Note: Always set this option to 10, unless Genesys Customer Care directs you to do otherwise.

high-water-mark

Default Value: 1000

Valid Values: 10-10000

Changes Take Effect: Immediately

Specifies the maximum allowable number of outstanding messages buffered in order to maintain flow control before T-Server warns of a backlog. When the number of buffered messages increases above the high-water-mark value, T-Server generates the following LMS message:

24510 (GCTM_FC_ABOVE_HIGH_WATER_MARK) Flow Control: Above high water mark (x messages buffered).

When the number of buffered messages drops below the high-water-mark value, T-Server generates the following LMS message:

24511 (GCTM_FC_BELOW_HIGH_WATER_MARK) Flow Control: Below high water mark (x messages buffered).

Note: The high-water-mark option must be used in conjunction with the `flow-control-rate-startup` and `flow-control-rate-runtime` options.

link-alarm-high

Default Value: 0

Valid Values: 0–100

Changes Take Effect: Immediately

Specifies the percentage of the `flow-control-rate-runtime` option value that, when the switch message flow rate exceeds it, will cause T-Server to generate the following LMS message: 24512 (GCTM_FC_LINK_ALARM_HIGH) is reported. A value of 0 (zero) disables this LMS message.

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 consecutive number for a CTI link, and *n* cannot be 0 (zero). You must specify a value for this option.

Warning! Do not update the link configuration while T-Server is running. Doing so causes a temporary link disconnection. If that happens, you must validate each configuration option in the `link` section before the connection is reestablished.

link-reconstitution

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

A NEC-specific option that enables T-Server to handle trunk optimization. Set this option to `true` if T-Server is operating within a network of NEC switches that are connected using #7 signaling links where redundant tromboning call paths are automatically reduced by the switch. Set this option to `false` if T-Server is operating with a stand-alone switch or a network of switches that only use PSTN trunks.

link-retry-timer

Default Value: 60000

Valid Values: Any positive integer

Changes Take Effect: Immediately

Specifies the time interval (in milliseconds) after which T-Server attempts to reestablish the connection to the switch, if a link connection fails.

login-mode

Default Value: 2

Valid Values:

- 0 No change—When an agent logs in through a desktop soft phone, T-Server generates an AgentLoggedIn event but does not perform switch operation.
- 1 Auto Ready—When an agent logs in through a desktop soft phone, T-Server clears UCD Busy-out (that is, it makes the agent Ready) on the physical handset and generates an AgentLoggedIn event.
- 2 Auto Not-Ready—When an agent logs in through the software, T-Server sets UCD Busy-out (that is, it makes the agent NotReady) on the physical handset and generates an AgentLoggedIn event.

Changes Take Effect: Immediately

Allows T-Server to change the agent's UCD Busy-out state when you have set [use-infoLink](#) to false.

Note: When the agent logs out through the soft phone, T-Server automatically sets the handset to UCD Busy-out (that is, it makes the agent NotReady, no matter what value you specify for login-mode).

make-call-agent-ppn

Default Value: true

Valid Values: true, false

Changes Take Effect: Immediately

Applies only to large-system switches. With a value of true, T-Server intercepts a TMakeCall T-Library request to call an agent's ACD Position directly, and redirects the request to that agent's personal pilot number (PPN). This redirection is necessary because the NEC ACD does not allow calls to be made directly to an agent's ACD line. Instead, these calls must be made through a PPN configured in the ACDMAT for each agent under the ACDL06 command.

Depending on the setting of option [query-agent-ppn](#), T-Server can either query the ACD dynamically for the agent PPN or statically retrieve the information configured in the PPN section of T-Server. This translation is transparent to T-Library clients, because the call appears to be transferred directly to the ACD line.

manual-answer-mode

Default Value: 0 (zero)

Valid Values: Any integer from 0–4

Changes Take Effect: Immediately

Determines the answer behavior using a TAnswerCall request.

If you set this option to a value ranging from 0–2, the destination handset answers without any zip tone in the headset. If you set this option to 3, the destination handset answers providing a single zip tone to the headset. If you set this option to 4, the destination handset answers providing a double zip tone to the headset.

max-calls

Default Value: 2003

Valid Values: Any positive integer

Changes Take Effect: After T-Server is restarted

Controls the size of the hash table, maintained by T-Server, that contains all currently active calls. The size of the table is defined as the maximum number of simultaneous calls handled by the contact center, multiplied by 2:

$(2 * \text{number of inbound trunks} + \text{number of agents}) * 2$

If the table fills with current calls, T-Server does not fail, but performance is degraded.

Note: Set the value of this option to a prime number to assist in the hashing process.

max-dns

Default Value: 2003

Valid Values: Any positive integer

Changes Take Effect: After T-Server is restarted

Controls the size of the hash table, maintained by T-Server, that contains all of the switch DNSs and trunks. The size of the table is defined as the maximum number of devices in the contact center, multiplied by 2.

If the table fills, T-Server does not fail, but the performance is degraded.

Note: Set the value of this option to a prime number to assist in the hashing process.

message-wait-off-mode

Default Value: 17

Valid Values: Any integer from 1–255

Changes Take Effect: Immediately

Determines which indicator on the handset is extinguished when a `TSetMessageWaitingOff` request is called. In most cases (when the `message-wait-on-mode` value turns an indicator ON), the value of this option corresponds to the value of the `message-wait-on-mode` option plus 1.

Note: See the NEC switch documentation for supported values. Values specified in this option should be in the decimal format.

message-wait-on-mode

Default Value: 16

Valid Values: Any integer from 1–255

Changes Take Effect: Immediately

Specifies which indicator on the handset is illuminated when a `TSetMessageWaitingOn` request is called.

Note: See the NEC switch documentation for supported values. The values specified in this option should be in the decimal format.

multi-observing-enabled

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether multi-observing is enabled, or not, on a switch. When multi-observing is enabled, T-Server executes a `TMonitorNextCall` request by invoking an Infolink multi-observe request (IH with operation type U), otherwise it uses an Infolink observe request (IH with operation type O).

monitor-call-using

Default Value: `infolink`

Valid Values: `scf`, `infolink`

Changes Take Effect: Immediately

Specifies which switch capability T-Server uses while executing `TMonitorNextCall` and `TCancelCallMonitoring` client requests.

no-response-timeout

Default Value: 5000

Valid Values: 1–600000

Changes Take Effect: Immediately

Specifies the timeout value (in milliseconds) that T-Server waits for a switch response. If a response is not received during the specified period, T-Server processes the corresponding request as `failed`.

number-of-links

Default Value: 1

Valid Values:

- 1 Set this value for a PBX with integrated ACD capabilities.
- 2 Set this value for a PBX that uses an external ACD module outside of the main PBX.

Changes Take Effect: After T-Server is restarted

Specifies the number of active links configured in the system.

oai-hold

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: After T-Server or Link is restarted

Supported only on switches with software levels that support SMFR FN=0x85 (hold monitor)/SMFN FN=0x06 (hold notification) OAI requests.

With a value of `true`, T-Server can detect and generate appropriate messages when calls are held and retrieved from the handset, as well as from a T-Server client. With a value of `false` (for all large-system switches), T-Server generates only the `EventHeld` and `EventRetrieved` events when initiated through a T-Server client.

Note: When this option is set to `true`, T-Server must re-register a DN on the switch to work correctly.

oai-reconnect-request

Default Value: `auto`

Valid Values:

- `auto` T-Server chooses a SCF OAI request based on the state of the consultation destination party. If the consultation destination party is connected to the call, T-Server uses SCF Ring Stop (FID=10); otherwise it uses SCF Hooking (FID=14).
- `scf10` T-Server always uses SCF Ring Stop (FID=10).
- `scf14` T-Server always uses SCF Hooking (FID=14).

Changes Take Effect: Immediately

Allows T-Server to correctly perform a `TReconnectCall` request on a small-system switch, by defining which SCF OAI request is used to perform a `TReconnectCall` request.

position-is-subline

Default Value: `true`

Valid Values:

- `true` Set this value for small-system switches utilizing Universal Routing Server, and for all large-system switches. By default, the ACD Position on these switches is the subline.
- `false` Set this value for small-system switches not utilizing Universal Routing Server. By default, the UCD line on these switches is the mainline (PBX line).

Changes Take Effect: Immediately

Specifies whether an ACD Position is configured as the subline.

postpone-dn-registration-msec

Default Value: 0

Valid Values: 0–180000

Changes Take Effect: Immediately

Specifies the timeout (in milliseconds) that T-Server waits before initiating DN registration on the switch at link startup. In Fusion (FCCS) configuration, the switch propagates all of the Facility Subscription requests from T-Server to all of the Fusion nodes. T-Server must wait until the subscription propagation in the Fusion network is complete before initiating the DN registration. The recommended option value for Fusion configuration is 90000 milliseconds. If, at link re-start, the Facility Subscription response indicates "Fast restart" ReturnCode = 10, T-Server does not delay DN registration. If this option is set to 0 (zero), T-Server does not postpone DN registration.

post-transfer-recall

Default Value: false

Valid Values:

- true A and B parties are immediately connected after a TReconnectCall request.
- false A and B parties are not immediately connected after a TReconnectCall request.

Changes Take Effect: Immediately

Specifies whether parties A and B are immediately reconnected after a TReconnectCall request.

Note: Certain small-system switches will not cleanly reconnect the A and B parties in the event of a TReconnectCall request. In this case, the A party rings and must be manually answered, because the call is returned to the A party after the reconnect.

query-agent-ppn

Default Value: true

Valid Values:

- true T-Server automatically queries the ACD for an agent's personal pilot number (PPN) when the agent logs into the switch.
- false T-Server receives the Agent-to-PPN mapping information from the PPN that is configured in T-Server.

Changes Take Effect: Immediately

Applies only to large-system switches when the [use-infolink](#) option is set to true. T-Server can either query the ACD dynamically for the agent's PPN or statically retrieve the information configured in the PPN section of T-Server. Refer to the following configuration options related to the PPN functionality:

`make-call-agent-ppn`, `transfer-agent-ppn`, `route-agent-ppn`,
`fast-transfer-ppn`, and `fail-no-ppn-route`.

Note: Configure the PPN for each agent in the ACDMAT under the ACDLOG command.

recall-clear-delay

Default Value: 5000

Valid Values: Any positive integer

Changes Take Effect: Immediately

Automatically clears a call, after the interval specified (in milliseconds), when the call has been abandoned from a ringing station and not recalled to another station.

If the destination to which a call is blind-transferred or redirected does not answer within a switch-defined time, then the call is released from that destination station and recalled to the transferring station. T-Server ensures that the call is correctly tracked, by maintaining the call in the T-Server call tables until the call arrives at the recall destination. If the call is recalled to an unmonitored device, T-Server never clears this call information from the tables.

Note: Always set this option to 5000, unless Genesys Customer Care directs you to do otherwise.

release-not-ready

Default Value: false

Valid Values: true, false

Changes Take Effect: Immediately

Specifies whether T-Server, after receiving a call-released message from the switch, generates an EventAgentNotReady event for a call on an ACD Position before generating an EventReleased event.

Note: Always set this option to false, unless Genesys Customer Care directs you to do otherwise.

report-dterm-ip-addr

Default Value: false

Valid Values: true, false

Changes Take Effect: After link initialization

Related Feature: “RTP Stream Monitoring API” on [page 172](#)

Discovers D-Term IP addresses. If the value of this option is set to true, T-Server invokes the OAI SSFN/SSFM facilities to discover the IP address of a monitored D-Term IP. A discovered IP address is reported in the RTP

monitoring events by the X-PI-RTP parameter. If the value of this option is set to `false`, it disables the IP address discovery.

route-agent-ppn

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Applies only to large-system switches when you have set the `use-infoLink` option to `true`. With a value of `true`, T-Server intercepts a T-Library request to route a call directly to an agent's ACD Position, and redirects the routing request to that agent's personal pilot number (PPN). This redirection is necessary because the NEC ACD does not allow calls to be directly transferred to an agent's ACD line. Instead, these transfers must be made through a PPN configured in the ACDMAT for each agent under the ACDLOG command. This translation is transparent to T-Library clients, because the call appears to be transferred directly to the ACD line.

router-default-dn

Default Value: No default value

Valid Values: Any valid dialable destination

Changes Take Effect: Immediately

Specifies the destination to which T-Server automatically default-routes a call when, for example, a URS-generated `TRouteCall` request for a regular Routing Point fails to deliver a call to the requested destination, because the destination device is invalid or busy.

rtp-info-password

Default Value: An empty string

Valid Values: Any valid string

Changes Take Effect: Immediately

Related Feature: "RTP Stream Monitoring API" on [page 172](#)

Specifies the password to allow voice-monitoring applications to monitor any call. If the password is not provided, no applications are able to perform the monitoring.

Warning! This password is not secure. It is sent as text to the voice-monitoring application.

scf10-answer-type

Default Value: 1

Valid Values: Any integer from 1–255

Changes Take Effect: Immediately

Defines the value used by T-Server to specify the Answer type argument of the SCF Ring Stop (FID=10) OAI request, when executing a call transfer via

TSingleStepTransfer or TMuteTransfer requests to an ACD Queue or Routing Point. If the value is set to 1 (hexadecimal 0x01), the ring-back tone is applied to the call after the call is connected to the transfer destination. If the value is set to 65 (hexadecimal 0x41), a music-on-hold tone is applied to the call after the call is connected to the transfer destination.

Note: Refer to the NEC OAI Protocol documentation for supported values for this new feature.

scf-calling-nbr-encoding

Default Value: called-number

Valid Values: called-number, calling-number-info

Changes Take Effect: Immediately

Related Feature: SCF Calling Party Number Selection

Specifies the encoding method of the calling party number information.

If the value is set to called-number, T-Server encodes the calling party number in the called number element of the 2nd Party Info parameter of the SCF OAI request.

If set to calling-number-info, T-Server uses the calling number info element (element tag 0x89) of the 2nd Party Info parameter of the SCF OAI request.

The calling party number information can be specified in the otherDN attribute as destination@calling_number_info. The calling party number information feature availability depends on the switch release level and configuration.

This feature is available for the FP85-107 S4 release, or later, of the NEC UNIVERGE SV8500 switch.

scf-notify-identifier

Default Value: 3

Valid Values: Any integer from 0–255 (0x0–0xFF hexadecimal)

Changes Take Effect: Immediately

If this option is set to any value other than 0, T-Server specifies the Notify Identifier (NI) attribute in the following SCF OAI requests:

- Make Call (FID=1)
- Make Ring (FID=3)
- Queue Connect Call (FID=4)
- Call Transfer (FID=7)

This enables the modification of a toll restriction check for the CTI-initiated telephony operations, namely the initiation of regular and consultation calls, and call routing.

If this option is set to 3, T-Server specifies the NI attribute in SCF OAI requests, which enables a toll restriction check for CTI-initiated telephony operations on the switch side.

If this option is set to 0 (zero), T-Server omits the NI attribute in SCF OAI requests, which disables a toll restriction check for CTI-initiated telephony operations on the switch side.

Note: Refer to the NEC OAI documentation for more information on other possible applications of the NI attribute in SCF OAI requests.

snd-scf2-delay

Default Value: 500

Valid Values: Any integer from 0–5000

Changes Take Effect: Immediately

Specifies the time interval (in milliseconds) that T-Server waits, after receiving an acknowledgment from the switch of the first SCF Release (FID=2) OAI request for an ACD Supervisor party release, before sending the second SCF Release (FID=2) request.

software-agent-state

Default Value: false

Valid Values: true, false

Changes Take Effect: Immediately

Modifies the behavior of agent status/work mode requests.

With a value of true, agent-state requests do not map to real switch requests, and AgentState events do not reflect the agent's actual state on the switch.

When T-Server is not communicating with the ACD or UCD, the value of the software-agent-state option is forced automatically to true.

Note: Do not use this option unless Genesys Customer Care directs you to do so.

subline-makecall-ring

Default Value: true

Valid Values: true, false

Changes Take Effect: Immediately

When set to true, the option instructs T-Server to use the SCF Make Ring (FID=3) OAI message instead of the SCF Make Call (FID=1) message when it receives a TMakeCall request for a D-Term device subline. This option does not affect the processing of a TMakeCall request for a main line.

suppress-trunk-events

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether trunk events are suppressed.

Note: Always set this option to `false`, unless Genesys Customer Care directs you to do otherwise.

tlib-response-timeout

Default Value: `120000`

Valid Values: `5000–120000`

Changes Take Effect: Immediately

Specifies the timeout value (in milliseconds) that T-Server waits for a response to the active T-Library request. All subsequent requests on a given DN are queued until processing of the first request is complete. If a response is not received during the specified period, T-Server generates an `EventError` message and initiates the processing of the next `TRequest`.

tenant-number

Default Value: `1`

Valid Values: Any valid tenant number

Changes Take Effect: After T-Server is restarted

Specifies the Tenant number used for communicating with the switch. If there are multiple Tenants, you must specify a value of `0` (zero) (all Tenants) to correspond with the switch configuration. If you specify a value of `0`, all DNs require a valid association with the appropriate Tenant number. Specify the Tenant number in Configuration Manager using the `.x` notation in the `Association` field of the `DN Properties` dialog box. That is, when you are configuring a Tenant object, enter `.x` as the value for the `Association` field, where `.x` is an integer that corresponds to the tenant's number. For example, enter `.1`.

This option supports the configuration of multi-tenant environments for the NEC NEAX/APEX range of switches. When processing Infolink messages from the global Tenant number `0` (all Tenants), T-Server associates the messages that involve a DN action (such as `AgentLogin`) with the appropriate Tenant number.

Note: If you set this option to `0` (zero), you must also configure the `tenant-number-list` option.

tenant-number-list

Default Value: No default value

Valid Values: Any comma-separated list of valid Tenant numbers

Changes Take Effect: After T-Server is restarted

Specifies the list of Tenants configured for a multi-tenant environment. Because the Infolink does not accept the global Tenant number of 0 (zero) (all Tenants), T-Server subscribes to the Infolink facilities for each Tenant number specified at startup. When T-Server processes Infolink messages that involve a DN action (such as AgentLogin), an association is made between the DN and the Tenant number. Specify the Tenant number in Configuration Manager using .x notation in the Association field of the DN Properties dialog box.

Note: You must set the [tenant-number](#) configuration option to 0 (zero) (all Tenants) to enable this option.

trace-level

Default Value: 2

Valid Values:

0	No additional tracing.
1	Report critical errors.
2	Report critical and minor errors.
3	Report all errors and detailed internal state information.
4	Report all errors and very detailed information.
5	Report all errors, detailed information, and very detailed information, such as function entry and exit.

Changes Take Effect: Immediately

Specifies the level of T-Server message log tracing, if you have set the verbose option to true. The higher the value, the more tracing T-Server does.

Note: Setting this option to 2, or higher, may slow T-Server processing on Windows NT. If so, minimize the T-Server window.

trace-modules

Default Value: +all, -hash

Valid Values:

+/- all	Turns on/off writing of all debugging information for all messages.
+/- device	Reserved for Genesys Engineering.
+/- hash	Reserved for Genesys Engineering.
+/- msgrec	Reserved for Genesys Engineering.
+/- callid	Reserved for Genesys Engineering.
+/- callmodel	Reserved for Genesys Engineering.

+/- encode	Turns on/off writing of encoded messages.
+/- decode	Turns on/off writing of decoded messages.
+/- handler	Reserved for Genesys Engineering.
+/- messages	Reserved for Genesys Engineering.
+/- utils	Reserved for Genesys Engineering.
+/- client	Reserved for Genesys Engineering.
+/- linkdump	Reserved for Genesys Engineering.

Note: Genesys recommends keeping this option set to +linkdump (on).

Changes Take Effect: After T-Server is restarted

Enables or disables internal tracing for all or selected T-Server modules. The following option format is applied: [+/-] module, [+/-] module.

transfer-agent-ppn

Default Value: true

Valid Values: true, false

Changes Take Effect: Immediately

Applies only to large-system switches. With a value of true, T-Server intercepts TInitiateTransfer, TMuteTransfer, TInitiateConference, TSingleStepTransfer, and TRedirectCall T-Library requests to transfer a call directly to an agent's ACD Position, and redirects the transfer request to that agent's PPN instead. This redirection is necessary because the NEC ACD does not allow calls to be transferred directly to an agent's ACD line. Instead, these transfers must be made through a PPN configured in the ACDMAT for each agent under the ACDLOG command.

Depending on the value setting of the [query-agent-ppn](#) option, T-Server can either query the ACD dynamically for the agent PPN, or statically retrieve the information configured in the PPN section of T-Server. You should also configure the PPN in the ACDMAT under the ACDLOG command. This translation is transparent to T-Library clients, because the call appears to be transferred directly to the ACD line.

transfer-allow

Default Value: true

Valid Values:

true	The call may be forwarded to another destination.
false	If the destination station has call forwarding set, and the station is busy, the SCF Make Call fails. If the destination station has call forwarding set, and the station is not busy, the call goes to the destination station. (The call is not forwarded.)

Changes Take Effect: Immediately

Modifies the behavior of `TMakeCall` and `TRouteCall` requests (when routing off of a monitored number).

Specifies the outcome of the call, if the destination party has `CallForwardingBusy` or `CallForwardingAll` set on the station. The `transfer-allow` option does not affect the behavior of `CallForwardingNoAnswer` set on a station.

transfer-release-delay

Default Value: 1000

Valid Values: Any positive integer

Changes Take Effect: Immediately

Specifies the time interval (in milliseconds) that T-Server waits after a transfer request was initiated before requesting the switch to release a transferring party. This option is used in single-step transfer scenarios (`TMuteTransfer` and `TSingleStepTransfer`). A value of 0 (zero) indicates no delay; however, on a heavily loaded switch, this setting might cause problems in completing the transfer.

trigger-agent-state-response-by

Default Value: event

Valid Values:

- | | |
|-------|---|
| event | T-Server distributes responses only after event notifications about agent state changes are received. |
| ack | T-Server distributes responses upon receiving a positive acknowledgement from the PBX. |

Changes Take Effect: Immediately

When set to `ack`, T-Server distributes responses to `TAgentReady` and `TAgentNotReady` requests upon receiving positive acknowledgement from the PBX. The pending work mode (`EventReady\EventNotReady`) is indicated in the `Extensions` attribute by the key-value pair, `ReasonCode pending`. When set to `event`, T-Server distributes responses only after the event notifications about the agent state changes are received.

use-infolink

Default Value: true

Valid Values: true, false

Changes Take Effect: After T-Server is restarted

Enables T-Server to send agent-state change requests to, and receive agent-state change notifications from, the ACD. On large-system switches utilizing an ACD, you must set this option to `true`. On small-system switches, you must set it to `false`, because no ACD is available for communication. Use the `use-ucd` option, in this case, to simulate agent-state functionality through UCD Busy-out functions on the agent handset.

use-link-bandwidth

Default Value: 100

Valid Values: 0-3000

Changes Take Effect: Immediately

Controls the number of messages that can be sent to the link within a one second period. A value of 0 (zero) disables flow control.

use-ucd

Default Value: false

Valid Values: true, false

Changes Take Effect: Immediately

Enables T-Server to send agent-state change requests to, and receive agent-state change notifications from, the UCD mechanism in the switch. This option controls the simulation of agent-state requests (Login, Ready, Not Ready, Logout) using UCD Busy-out. The options `use-infolink` and `use-ucd` are mutually exclusive; if you set one of them to true, you must set the other to false. If you set both to false, T-Server does not send any request to the switch to simulate agent state. When you set `use-ucd` to true, set the `login-mode` option based on customer requirements.

Obsolete Options

Warning! Unless advised by Genesys Customer Care or Professional Services, do *not* use these options. Doing so prevents T-Server from conforming to the Genesys Call Model.

simulate-established (obsolete)

Default Value: false

Valid Values: true, false

Changes Take Effect: Immediately

Enables T-Server to simulate an `EventEstablished` message after a configurable timeout from trunk seizure for an outbound call. This is required for situations where, because of the switch or network configuration, T-Server does not receive an acknowledgment from the switch that an external party has answered a call. This setting allows some applications (such as S-Dialer) to receive the `EventEstablished` they require to enter a `Talking` state.

Note: Genesys recommends that you *not* set this option to true when the network or switch generates an answer notification on the trunk; otherwise, a duplicated `EventEstablished` event is sent.

simulate-established-delay (obsolete)

Default Value: 5000

Valid Values: Any positive integer

Changes Take Effect: Immediately

Specifies the time interval (in milliseconds) that T-Server waits, after trunk seizure for an outbound call, before generating a simulated EventEstablished. This option is ignored if you have set `simulate-established` to false.

PPN-Mapping Section

This section must be called PPN. This configuration section allows static mappings to be applied between specific agent IDs and personal pilot numbers (PPNs). This section is significant only if you have set the type option `query-agent-ppn` to false.

<n>

Default Value: Mandatory field. No default value.

Valid Values: Any string in the ID, PPN format

Changes Take Effect: After agent login

As an option name, specifies a unique number between 1 and 1999. The value consists of two numbers separated by a comma. The first number is the AgentID (as entered in the Configuration Database AgentID object configuration), and the second number is the PPN configured against that agent in the ADDLOG command of the ACDMAT. You must specify a value for this option. [Table 28](#) lists some of the possible values for this option.

Table 28: PPN-Mapping Options

Option Name	Option Value
1	45054, 45010
2	45055, 45011

Note: You do not need to complete this section if the T-Server is configured to dynamically query PPN information from the ACD (that is, when you have set type option `query-agent-ppn` to true).

CTI-Link Section

The section name is specified by the `link-n-name` option.

Note: `link-n-name` refers to the link number and the section name (for example, `link-1-name`).

hostname

Default Value: Mandatory field. No default value.

Valid Values: Any valid host name

Changes Take Effect: Immediately

Specifies the host of the link, according to the switch configuration. You must specify a value for this option.

link-type

Default Value: `oai+infolink`

Valid Values:

<code>oai</code>	The link that only supports the OAI capabilities. Specify this link type when configuring a connection to the main PBX without an integrated ACD module.
<code>infolink</code>	The link that only supports the Infolink capabilities. Specify this link type when configuring a connection to an external ACD module.
<code>oai+infolink</code>	The link that supports both OAI and Infolink capabilities. Specify this link type when configuring a connection to the PBX with integrated ACD capabilities (usually a large-switch system).

Changes Take Effect: After T-Server is restarted

Specifies the type of the PBX link.

port

Default Value: Mandatory field. No default value.

Valid Values: Any valid port address

Changes Take Effect: Immediately

Specifies the TCP/IP port of the link, according to the switch configuration. You must specify a value for this option.

Note: For NEAX IMX or APEX 7600 switches with internal OAI (not CC98) only, you must set the port option to `60030` for all T-Servers.

protocol

Default Value: Mandatory field. No default value.

Valid Values: tcp

Changes Take Effect: Immediately

Specifies the connection protocol T-Server uses in communicating with the switch. You must specify a value for this option.

Changes from Release 8.0 to 8.1

[Table 29](#) lists configuration options that changed between the 8.0 and 8.1 releases of T-Server.

Table 29: T-Server Option Changes from 8.0 to 8.1

Option Name	Type of Change	Details
TServer Section		
aarq-logical-port	New	See page 235 for details.
alive-message-tolerance	New	See page 235 for details.
enable-answer-term-info	New	See page 237 for details.
enable-infolink-heartbeat	New	See page 238 for details.
multi-observing-enabled	New	See page 247 for details.
report-dterm-ip-addr	New	See page 250 for details.
rtp-info-password	New	See page 251 for details.
scf-calling-nbr-encoding	New	See page 252 for details.
trigger-agent-state-response-by	New	See page 257 for details.

Related Documentation Resources

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

T-Server for NEC NEAX-APEX

- 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 30](#) describes and illustrates the type conventions that are used in this document.

Table 30: Type Styles

Type Style	Used For	Examples
Italic	<ul style="list-style-type: none"> Document titles Emphasis Definitions of (or first references to) unfamiliar terms Mathematical variables <p>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 266).</p>	<p>See the <i>Genesys Migration Guide</i> for more information.</p> <p>Do <i>not</i> use this value for this option.</p> <p>A <i>customary and usual</i> practice is one that is widely accepted and used within a particular industry or profession.</p> <p>The formula, $x + 1 = 7$ where x stands for . . .</p>
Monospace font (Looks like teletype or typewriter text)	<p>All programming identifiers and GUI elements. This convention includes:</p> <ul style="list-style-type: none"> The <i>names</i> 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. <p>Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.</p>	<p>Select the Show variables on screen check box.</p> <p>In the Operand text box, enter your formula.</p> <p>Click OK to exit the Properties dialog box.</p> <p>T-Server distributes the error messages in EventError events.</p> <p>If you select true for the inbound-bsns-calls option, all established inbound calls on a local agent are considered business calls.</p> <p>Enter exit on the command line.</p>
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.	<code>smcp_server -host [/flags]</code>
Angle brackets (< >)	<p>A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise.</p> <p>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.</p>	<code>smcp_server -host <confighost></code>



Index

Symbols

[] (square brackets)	266
< > (angle brackets)	266
<key name>	
common log option	200

A

aaq-logical-port	
configuration option	235
Access Code	
configuration	104
defined	37, 102
ADDP	54
addp-remote-timeout	
configuration option	226
addp-timeout	
configuration option	227
addp-trace	
configuration option	227
Advanced Disconnect Detection Protocol	23
Agent Login objects	38
agent reservation	
defined	28
agent-reservation section	
configuration options	214–215
alarm	
common log option	191
alive-message-timer	
configuration option	235
alive-message-tolerance	
configuration option	235
all	
common log option	190
American/Brazilian market	
supported platforms	135
angle brackets	266
ANI	67
ani-distribution	
configuration option	206

answer-mode	
configuration option	236
app	
command line parameter	115
Application objects	
multi-site operation	101
audience, for document	12
Australian/New Zealand market	
supported platforms	135

B

background-processing	
configuration option	206
background-timeout	
configuration option	207
backup servers	45
backup-reset-threshold-interval	
configuration option	236
backup-sync section	
configuration options	226–227
configuring hot standby	54
brackets	
angle	266
square	266
buffering	
common log option	184

C

call-cleanup section	
configuration options	228–229
cast-type	
configuration option	66, 217
CDN	73
changes from 8.0 to 8.1	261
common configuration options	203
T-Server common configuration options	231
check-point	
common log option	188

- check-tenant-profile
 - configuration option 207
- cleanup-idle-tout
 - configuration option 228
- Code property 104, 105
- cof-ci-defer-create
 - configuration option 222
- cof-ci-defer-delete
 - configuration option 222
- cof-ci-req-tout
 - configuration option 82, 222
- cof-ci-wait-all
 - configuration option 223
- cof-feature
 - configuration option 223
- cof-rci-tout
 - configuration option 223
- collect-lower-priority-requests
 - configuration option 214
- command line parameters 115
 - app 115
 - host 115
 - I 116
 - lmspath 116
 - nco X/Y 116
 - port 115
 - V 116
- commenting on this document 13
- common configuration options 184–204
 - changes from 8.0 to 8.1 203
 - common section 203
 - disable-rbac 201
 - enable-async-dns 203
 - hangup-restart 202
 - heartbeat-period 201
 - heartbeat-period-thread-class-<n> 202
 - log section 184–198
 - log-extended section 198–200
 - log-filter section 200
 - log-filter-data section 200–201
 - mandatory 184
 - rebind-delay 203
 - security section 201
 - setting 183
 - sml section 201–203
 - suspending-wait-timeout 202
- common log options 184–200
 - <key name> 200
 - alarm 191
 - all 190
 - buffering 184
 - check-point 188
 - compatible-output-priority 189
 - debug 193
 - default-filter-type 200
 - expire 185
 - interaction 192
 - keep-startup-file 186
 - level-reassign-<eventID> 198
 - level-reassign-disable 200
 - log section 184–198
 - log-extended section 198–200
 - log-filter section 200
 - log-filter-data section 200–201
 - mandatory options 184
 - memory 188
 - memory-storage-size 189
 - message_format 186
 - messagefile 186
 - print-attributes 188
 - segment 185
 - setting 183
 - spool 189
 - standard 192
 - time_convert 187
 - time_format 187
 - trace 192
 - verbose 184
 - x-conn-debug-all 198
 - x-conn-debug-api 197
 - x-conn-debug-dns 197
 - x-conn-debug-open 196
 - x-conn-debug-security 197
 - x-conn-debug-select 196
 - x-conn-debug-timers 196
 - x-conn-debug-write 196
- common options
 - common log options 184–200
 - common section 203
 - mandatory options 184
 - sml section 201–203
- common section
 - common options 203
- compatible-output-priority
 - common log option 189
- compound-dn-representation
 - configuration option 224
- Configuration Manager
 - configuring T-Server 39
 - multiple ports 40
- configuration options. 233
 - aarq-logical-port 235
 - addp-remote-timeout 226
 - addp-timeout 227
 - addp-trace 227
 - agent-reservation section 214–215
 - alive-message-timer 235
 - alive-message-tolerance 235
 - ani-distribution 206
 - answer-mode 236
 - background-processing 206
 - background-timeout 207

backup-reset-threshold-interval	236	license section	211–214
backup-sync section	226–227	link-alarm-high	244
call-cleanup section	228–229	link- <i>n</i> -name	244
cast-type	217	link-reconstitution	244
changes from 8.0 to 8.1	231	link-retry-timer	244
check-tenant-profile	207	link-type	260
cleanup-idle-tout	228	local-node-id	223
cof-ci-defer-create	222	login-mode	245
cof-ci-defer-delete	222	log-trace-flags	209
cof-ci-req-tout	222	make-call-agent-ppn	245
cof-ci-wait-all	223	management-port	209
cof-feature	223	mandatory options	184, 233
cof-rci-tout	223	manual-answer-mode	245
collect-lower-priority-requests	214	match-call-once	216
common log options	184–200	max-calls	246
common options	184–204	max-dns	246
compound-dn-representation	224	merged-user-data	209
consult-user-data	207	message-wait-off-mode	246
CTI–Link section	260	message-wait-on-mode	247
customer-id	208	monitor-call-using	247
default-announcement-trunk	236	multi-observing-enabled	247
default-dn	218	<i>n</i>	259, 219, 247, 228, 247, 211, 212
default-network-call-id-matching	224	oai-hold	248
direct-digits-key	218	oai-reconnect-request	248
dnd-mode	236	periodic-check-tout	228
dn-for-unexpected-calls	219	port	260
dn-scope	96, 208	position-is-subline	248
enable-alive-message	237	postpone-dn-registration-msec	249
enable-answer-term-info	237	post-transfer-recall	249
enable-dnd-monitor	237	PPN-Mapping section	259
enable-hotline-term-info	238	propagated-call-type	96, 210
enable-infolink-heartbeat	238	protocol	227, 261
enable-query-addr-status	238	query-agent-ppn	249
enable-query-dnd-status	239	recall-clear-delay	250
enable-query-fwd-status	239	reconnect-tout	216
enable-query-mwl-status	239	register-attempts	219
enable-retrieve-on-release	238	register-tout	219
enable-split-prefix	239	reject-subsequent-request	215
enable-ssc	240	release-not-ready	250
epp-tout	97, 225	report-connid-changes	216
event-propagation	225	report-dterm-ip-addr	250
extrouter section	215–226	request-collection-time	215
extrouter-default-dn	240	request-tout	219
failed-registration-timeout	240	reservation-time	215
fail-no-ppn-route	241	resource-allocation-mode	220
fast-transfer	241	resource-load-maximum	220
fast-transfer-ppn	241	route-agent-ppn	251
fix-double-hold	242	route-dn	220
fix-supervisor-release	242	router-default-dn	251
flow-control-link-alarm-gap	243	rtp-info-password	251
flow-control-rate-runtime	242	rule-< <i>n</i> >	229
flow-control-rate-startup	243	scf10-answer-type	251
handle-vsp	226	scf-calling-nbr-encoding	252
high-water-mark	243	scf-notify-identifier	252
hostname	260	security section	230
inbound-translator-< <i>n</i> >	225	server-id	210

setting	205
common	183
simulate-established (obsolete)	258
simulate-established-delay (obsolete)	259
snd-scf2-delay	253
software-agent-state	253
subline-makecall-ring	253
suppress-trunk-events	254
sync-reconnect-tout	227
tcs-queue	221
tcs-use	222
tenant-number	254
tenant-number-list	255
timeout	221
timeout value format	230
tlib-response-timeout	254
trace-level	255
trace-modules	255
transfer-agent-ppn	256
transfer-allow	256
transfer-release-delay	257
Translation Rules section	229
trigger-agent-state-response-by	257
TServer section	206–211, 235–258
use-data-from	217
use-implicit-access-numbers	221
use-infolink	257
use-link-bandwidth	258
user-data-limit	211
use-ucd	258
configuring	
high availability	
T-Server	53–55
multi-site operation	101–114
steps	101
T-Server	39
multiple ports	40
consult-user-data	
configuration option	207
conventions	
in document	265
type styles	266
CTI-Link section	
configuration options	260
customer-id	
configuration option	208

D

debug	
common log option	193
Default Access Code	
configuration	103
defined	102

default-announcement-trunk	
configuration option	236
default-dn	
configuration option	218
default-filter-type	
common log option	200
default-network-call-id-matching	
configuration option	224
destination location	60
destination T-Server	66
direct-ani	
ISCC transaction type	67, 75
direct-callid	
ISCC transaction type	68, 75
direct-digits	
transaction type	75
direct-digits-key	
configuration option	218
direct-network-callid	
ISCC transaction type	68, 75
direct-notoken	
ISCC transaction type	69, 75
direct-uui	
ISCC transaction type	69, 75
disable-rbac	
common configuration option	201
DN objects	38
DN types between NEC and Genesys	
switch configuration	146
dnd-mode	
configuration option	236
dn-for-unexpected-calls	
configuration option	219
dnis-pool	
in load-balancing mode	71
ISCC transaction type	62, 70, 75
DNs	
configuring for multi-sites	108
dn-scope	
configuration option	96, 208
document	
audience	12
change history	13
conventions	265
errors, commenting on	13
version number	265

E

enable-alive-message	
configuration option	237
enable-answer-term-info	
configuration option	237
enable-async-dns	
common configuration option	203

enable-dnd-monitor
 configuration option 237

enable-hotline-term-info
 configuration option 238

enable-infolink-heartbeat
 configuration option 238

enable-query-addr-status
 configuration option 238

enable-query-dnd-status
 configuration option 239

enable-query-fwd-status
 configuration option 239

enable-query-mwl-status
 configuration option 239

enable-retrieve-on-release
 configuration option 238

enable-split-prefix
 configuration option 239

enable-ssc
 configuration option 240

epp-tout
 configuration option 97, 225

error messages
 supported functionality 178

Event Propagation
 defined 93

EventAttachedDataChanged 94

event-propagation
 configuration option 225

expire
 common log option 185

extrouter section
 configuration options 215–226
 configuring for multi-site operation 102
 configuring party events propagation 98
 configuring the Number Translation feature . 91

extrouter-default-dn
 configuration option 240

F

failed-registration-timeout
 configuration option 240

fail-no-ppn-route
 configuration option 241

fast-transfer
 configuration option 241

fast-transfer-ppn
 configuration option 241

figures
 hot standby redundancy 48
 Multiple-to-Point mode 74
 Point-to-Point mode 73
 steps in ISCC/Call Overflow 81

fix-double-hold
 configuration option 242

fix-supervisor-release
 configuration option 242

flow-control-link-alarm-gap
 configuration option 243

flow-control-rate-runtime
 configuration option 242

flow-control-rate-startup
 configuration option 243

font styles
 italic 266
 monospace 266

Force Function Modifier
 supported functionality 170

H

HA
 See also high availability
 See hot standby

HA configuration 45–55

HA Proxy
 starting 122, 123

handle-vsp
 configuration option 226

hangup-restart
 common configuration option 202

heartbeat-period
 common configuration option 201

heartbeat-period-thread-class-<n>
 common configuration option 202

high-availability configuration 45–55

high-water-mark
 configuration option 243

host
 command line parameter 115

hostname
 configuration option 260

hot standby 24, 45
 defined 25
 figure 48
 T-Server configuration 52

I

inbound-translator-<n>
 configuration option 225

intended audience 12

Inter Server Call Control 60–79

Inter Server Call Control/Call Overflow . . 79–83

interaction
 common log option 192

ISCC
 destination T-Server 66
 origination T-Server 66

ISCC transaction types 61, 66

- direct-ani 67, 75
- direct-callid 68, 75
- direct-digits 75
- direct-network-callid 68, 75
- direct-notoken 69, 75
- direct-uuui 69, 75
- dnis-pool 70, 75
 - in load-balancing mode 71
- pullback 71, 75
- reroute 72, 75
- route 73, 75
- route-uuui 74
- supported 75
- ISCC/COF
 - supported 80
- iscc-xaction-type 61
- italics 266

J

- Japanese market
 - supported platforms 134

K

- keep-startup-file
 - common log option 186

L

- l
 - command line parameter 116
- level-reassign-<eventID>
 - common log option 198
- level-reassign-disable
 - common log option 200
- license section
 - configuration options 211–214
- link-alarm-high
 - configuration option 244
- link-*n*-name
 - configuration option 244
- link-reconstitution
 - configuration option 244
- link-retry-timer
 - configuration option 244
- link-type
 - configuration option 260
- lmspath
 - command line parameter 116
- local-node-id
 - configuration option 223
- location parameter 60
- log configuration options 184–190
- log section
 - common log options 184–198

- log-extended section
 - common log options 198–200
- log-filter section
 - common log options 200
- log-filter-data section
 - common log options 200–201
- login-mode
 - configuration option 245
- log-trace-flags
 - configuration option 209

M

- make-call-agent-ppn
 - configuration option 245
- Management Layer 36
- management-port
 - configuration option 209
- mandatory options
 - configuration options 206, 233
- manual-answer-mode
 - configuration option 245
- match-call-once
 - configuration option 216
- max-calls
 - configuration option 246
- max-dns
 - configuration option 246
- memory
 - common log option 188
- memory-storage-size
 - common log option 189
- merged-user-data
 - configuration option 209
- message_format
 - common log option 186
- messagefile
 - common log option 186
- message-wait-off-mode
 - configuration option 246
- message-wait-on-mode
 - configuration option 247
- monitor-call-using
 - configuration option 247
- monospace font 266
- multi-observing-enabled
 - configuration option 247
- Multiple-to-One mode 73
- Multiple-to-Point mode 73, 74

N

- n*
 - configuration option 259
- NAT/C feature 91
- error messages 180
- supported requests 167

nco X/Y
 command line parameter 116
network attended transfer/conference 91
network objects. 36
network-request-timeout
 configuration option 219
no-response-timeout
 configuration option 247
notify-idle-tout
 configuration option 228
Number Translation feature 83–91
number translation rules 84
number-of-links
 configuration option 247
num-of-licenses
 configuration option 211
num-sdn-licenses
 configuration option 212

O

oai-hold
 configuration option 248
oai-reconnect-request
 configuration option 248
objects
 Agent Logins 38
 DNSs 38
 network 36
 Switches 37
 Switching Offices 37
One-to-One mode 73
origination location 60
origination T-Server 66
other markets
 supported platforms 136

P

periodic-check-tout
 configuration option 228
Point-to-Point mode 73
port
 command line parameter 115
 configuration option 260
position-is-subline
 configuration option 248
postpone-dn-registration-msec
 configuration option 249
post-transfer-recall
 configuration option 249
PPN-Mapping section
 configuration option 259
primary servers 45
print-attributes
 common log option 188

propagated-call-type
 configuration option 96, 210
protocol
 configuration option 227, 261
pullback
 ISCC transaction type. 71, 75

Q

query-agent-ppn
 configuration option 249

R

rebind-delay
 common configuration option 203
recall-clear-delay
 configuration option 250
reconnect-tout
 configuration option 216
redundancy
 hot standby 24, 45
 warm standby 24, 45
redundancy types 49, 50, 52
 hot standby 25
register-attempts
 configuration option 219
register-tout
 configuration option 219
reject-subsequent-request
 configuration option 215
release-not-ready
 configuration option 250
report-connid-changes
 configuration option 216
report-dterm-ip-addr
 configuration option 250
request-collection-time
 configuration option 215
request-tout
 configuration option 62, 219
reroute
 ISCC transaction type. 72, 75
reservation-time
 configuration option 215
resource-allocation-mode
 configuration option 220
resource-load-maximum
 configuration option 220
route
 ISCC transaction type. 62, 73, 75, 108
route-agent-ppn
 configuration option 251
route-dn
 configuration option 220

- router-default-dn
 - configuration option 251
- route-uui
 - ISCC transaction type 74
- routing
 - Inter Server Call Control 66–79
- rtp-info-password
 - configuration option 251
- rule-<n>
 - configuration option 229
- run.bat 119
- run.sh. 118

S

- scf10-answer-type
 - configuration option 251
- scf-calling-nbr-encoding
 - configuration option 252
- scf-notify-identifier
 - configuration option 252
- security section
 - common configuration options 201, 230
- segment
 - common log option 185
- server-id
 - configuration option 210
- setting configuration options
 - common 183
- setting the DN Types
 - switch configuration 138
- simulate-established (obsolete)
 - configuration option 258
- simulate-established-delay (obsolete)
 - configuration option 259
- sml section
 - common options 201–203
- snd-scf2-delay
 - configuration option 253
- software-agent-state
 - configuration option 253
- spool
 - common log option 189
- square brackets 266
- standard
 - common log option 192
- starting
 - HA Proxy 122
 - T-Server 123
- subline-makecall-ring
 - configuration option 253
- supported agent work modes
 - supported functionality 168
- supported functionality
 - error messages 178
 - Force Function Modifier 170

- supported agent work modes 168
- T-Library functionality 159
- supported functionality table 160
- supported platforms
 - American/Brazilian market 135
 - Australian/New Zealand market 135
 - Japanese market 134
 - other markets 136
- suppress-trunk-events
 - configuration option 254
- suspending-wait-timeout
 - common configuration option 202
- switch configuration
 - DN types between NEC and Genesys . . 146
 - setting the DN Types 138
 - Universal Routing Server configuration . . 148
- switch configuration requirements 153
- Switch objects 37
 - multi-site operation 101
- switch partitioning
 - defined 96
 - T-Server support 97
- Switching Office objects 37
 - multi-site operation 102, 103, 104, 108
- sync-reconnect-tout
 - configuration option 227

T

- Target ISCC
 - Access Code configuration 105
 - Default Access Code configuration 104
- tcs-queue
 - configuration option 221
- tcs-use
 - configuration option 222
- tenant-number
 - configuration option 254
- tenant-number-list
 - configuration option 255
- time_convert
 - common log option 187
- time_format
 - common log option 187
- timeout
 - configuration option 62, 221
- timeout value format
 - configuration options 230
- TInitiateConference 60
- TInitiateTransfer 60
- T-Library functionality
 - supported functionality 159
- tlib-response-timeout
 - configuration option 254
- TMakeCall 60
- TMuteTransfer 60

trace
 common log option 192
 trace-level
 configuration option 255
 trace-modules
 configuration option 255
 transaction types (ISCC) 61, 66
 supported 75
 transfer connect service 78
 transfer-agent-ppn
 configuration option 256
 transfer-allow
 configuration option 256
 transfer-release-delay
 configuration option 257
 Translation Rules section
 configuration option 229
 trigger-agent-state-response-by
 configuration option 257
 TRouteCall 60
 trunk configuration
 switch configuration 150
 trunk lines 73
 T-Server
 configuring Application objects 39
 for multi-sites 101
 configuring redundancy 50
 HA 52
 high availability 52
 hot standby 52
 multi-site operation 101–114
 redundancy 49, 50, 52
 starting 123, 124
 using Configuration Manager 39
 multiple ports 40
 warm standby 50
 TServer section
 configuration options 206–211, 235–258
 TSingleStepTransfer 60
 TXRouteType 61
 type styles
 conventions 266
 italic 266
 monospace 266
 typographical styles 265, 266

U

Universal Routing Server configuration
 switch configuration 148
 UNIX
 installing T-Server 41
 starting applications 119
 starting HA Proxy 123
 starting T-Server 124

 starting with run.sh 118
 use-data-from
 configuration option 217
 use-implicit-access-numbers
 configuration option 221
 use-infolink
 configuration option 257
 use-link-bandwidth
 configuration option 258
 user data propagation 94
 user-data-limit
 configuration option 211
 use-ucd
 configuration option 258

V

V
 command line parameters 116
 VDN 73
 verbose
 common log option 184
 version numbering, document 265

W

warm standby 24, 45
 figure 46
 T-Server configuration 50
 Windows
 installing T-Server 42
 starting applications 119
 starting HA Proxy 123
 starting T-Server 124
 starting with run.bat 119

X

x-conn-debug-all
 common log option 198
 x-conn-debug-api
 common log option 197
 x-conn-debug-dns
 common log option 197
 x-conn-debug-open
 common log option 196
 x-conn-debug-security
 common log option 197
 x-conn-debug-select
 common log option 196
 x-conn-debug-timers
 common log option 196
 x-conn-debug-write
 common log option 196

