



Framework 7.6

T-Server for Ericsson MD110

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 © 2000–2008 Genesys Telecommunications Laboratories, Inc. All rights reserved.

About Genesys

Genesys Telecommunications Laboratories, Inc., a subsidiary of Alcatel-Lucent, is 100% focused on software for call centers. Genesys recognizes that better interactions drive better business and build company reputations. Customer service solutions from Genesys deliver on this promise for Global 2000 enterprises, government organizations, and telecommunications service providers across 80 countries, directing more than 100 million customer interactions every day. Sophisticated routing and reporting across voice, e-mail, and Web channels ensure that customers are quickly connected to the best available resource—the first time. Genesys offers solutions for customer service, help desks, order desks, collections, outbound telesales and service, and workforce management. Visit www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support 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, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. 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.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North and Latin America	+888-369-5555 or +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-118-974-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp

Prior to contacting technical support, please refer to the [Genesys Technical Support 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 7 Licensing Guide](#).

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 76fr_dep-ts_md110_02-2008_v7.6.001.00



Table of Contents

	List of Procedures	9
Preface	11
	Intended Audience.....	12
	Reading Prerequisites	12
	Chapter Summaries.....	12
	Document Conventions	14
	Related Resources	16
	Making Comments on This Document	17
Part 1	Part One: Common Functions and Procedures	19
	New for All T-Servers in 7.6.....	20
Chapter 1	T-Server Fundamentals.....	21
	Learning About T-Server	22
	Framework and Media Layer Architecture	22
	T-Server Requests and Events	24
	Advanced Disconnect Detection Protocol	28
	Redundant T-Servers	29
	Multi-Site Support	32
	Agent Reservation	32
	Client Connections	33
	Next Steps	33
Chapter 2	T-Server General Deployment.....	35
	Prerequisites.....	35
	Software Requirements	36
	Hardware and Network Environment Requirements	37
	Licensing Requirements	37
	About Configuration Options.....	39
	Deployment Sequence	40
	Wizard Deployment of T-Server	41
	Wizard Configuration of T-Server	41

	Wizard Installation of T-Server	42
	Manual Deployment of T-Server	43
	Manual Configuration of Telephony Objects	44
	Manual Configuration of T-Server	46
	Manual Installation of T-Server	48
	Next Steps	50
Chapter 3	High-Availability Deployment.....	51
	Warm Standby Redundancy Type	52
	Hot Standby Redundancy Type	53
	Prerequisites.....	55
	Requirements.....	55
	Synchronization Between Redundant T-Servers	55
	Warm Standby Deployment.....	56
	General Order of Deployment.....	56
	Manual Modification of T-Servers for Warm Standby.....	57
	Warm Standby Installation of Redundant T-Servers	58
	Hot Standby Deployment.....	58
	General Order of Deployment.....	58
	Manual Modification of T-Servers for Hot Standby.....	59
	Hot Standby Installation of Redundant T-Servers	62
	Next Steps	62
Chapter 4	Multi-Site Support.....	63
	Multi-Site Fundamentals	64
	ISCC Call Data Transfer Service	65
	ISCC Transaction Types	70
	T-Server Transaction Type Support	78
	Transfer Connect Service Feature	82
	ISCC/COF Feature	83
	Number Translation Feature	87
	Number Translation Rules	88
	Network Attended Transfer/Conference Feature	95
	Event Propagation Feature	97
	User Data Propagation	97
	Party Events Propagation	99
	Basic and Advanced Configuration	99
	ISCC Transaction Monitoring Feature	102
	Configuring Multi-Site Support.....	103
	Applications	103
	Switches and Access Codes	104
	DNs	110

	Configuration Examples	114
	Next Steps	116
Chapter 5	Start and Stop T-Server Components	117
	Command-Line Parameters	117
	Starting and Stopping with the Management Layer	119
	Starting with Startup Files	120
	Starting Manually	121
	HA Proxy	124
	T-Server	125
	Verifying Successful Startup	126
	Stopping Manually	127
	Starting and Stopping with Windows Services Manager	128
	Next Steps	128
Part 2	Part Two: Reference Information	129
	New in T-Server for Ericsson MD110	130
Chapter 6	Ericsson MD110 Switch-Specific Configuration	131
	Known Limitations	131
	Setting the DN Properties	134
	Support of Switch/CTI Environments	136
	Partitioned Application Link Configuration	137
	Configuring Devices in Configuration Layer	137
	VoIP Devices	138
	Routing Configuration	138
	Emulating Routing by Using Hunt Groups	138
	Using CTI Group as a Single Routing Point/ External Routing Point	140
	Supervised Routing	140
	Device-Specific Behavior	141
	Initiating a Consult Call from an ODN or ADN	141
	Completing Two-Step Conferences from ODNs or ADNs	141
	One-Step Transfer from Specific Devices	141
	Free Seating	141
	Support for PBX Boss/Secretary Feature	142
	Support for Maintain Position in Queue	142
Chapter 7	Supported Functionality in T-Server for Ericsson MD110	145
	Business-Call Handling	145

T-Server Call Classification.....	145
Support for Emulated Agents	147
Emulated Agent Login/Logout	148
Emulated Agent Ready/NotReady	149
Emulated After-Call Work (ACW).....	149
HA Synchronization	155
Support for Emulated Predictive Dialing.....	155
Limiting Distribution Time.....	156
Call Progress Detection	156
Unsolicited Calls on Predictive Dialing Devices.....	156
Support for No-Answer Supervision	157
Agent No-Answer Supervision	157
Extension No-Answer Supervision	157
Position No-Answer Supervision	158
Configuration Options for Device-Specific Overrides.....	158
Extension Attributes for Overrides for Individual Calls.....	158
Private Calls.....	158
Recall Scenarios	159
Reporting	159
Smart OtherDN Handling.....	160
Configuration Options and Extension	160
Supported Requests	161
Keep-Alive Feature.....	163
Examples	164
VoIP Absence Code Handling	168
Configuring Absence Codes	169
T-Library Functionality	170
Support for Agent Work Modes	180
Use of the Extensions Attribute	180
User Data Keys	186
Error Messages	186

Chapter 8

Common Configuration Options.....	195
Setting Configuration Options.....	195
Mandatory Options	196
Log Section.....	196
Log Output Options.....	202
Examples	206
Debug Log Options	207
Log-Extended Section	210
Log-Filter Section	212
Log-Filter-Data Section.....	213

	Common Section	213
	Changes from 7.5 to 7.6	214
Chapter 9	T-Server Common Configuration Options	217
	Setting Configuration Options.....	217
	Mandatory Options	218
	T-Server Section.....	218
	License Section	223
	Agent-Reservation Section	225
	Multi-Site Support Section	226
	ISCC Transaction Options	228
	Transfer Connect Service Options.....	232
	ISCC/COF Options	233
	Event Propagation Option.....	235
	Number Translation Option	235
	Translation Rules Section.....	235
	Backup-Synchronization Section	236
	Call-Cleanup Section.....	237
	Security Section.....	239
	Timeout Value Format	239
	Option Changes from Release 7.5 to 7.6	240
Chapter 10	Configuration Options in T-Server for Ericsson MD110	241
	Mandatory Options	241
	T-Server Section.....	242
	Reason-code-end Section	266
	Switch-Specific Type Section.....	266
	Annex Tab Options	266
	Application-Link Section	269
	Link-Control Section	270
	Changes from 7.5 to 7.6	274
Chapter 11	High Availability (HA)	277
	Genesys HA with Single Application Link.....	277
	Hot-Standby Mode	278
	Warm-Standby Mode	278
Chapter 12	Ericsson MD110 Call Flows	279
	ADN-ODN Conference Call Flow	279

Description.....	279
Service Completion After Alternating the Call.....	279
Service Completion Without Alternating the Call	282

Index	285
--------------------	------------



List of Procedures

Installing T-Server on UNIX using Wizard	42
Installing T-Server on Windows using Wizard	43
Configuring T-Server manually	46
Configuring multiple ports	47
Installing T-Server on UNIX manually	48
Installing T-Server on Windows manually	49
Verifying the manual installation of T-Server	50
Modifying the primary T-Server configuration for warm standby	57
Modifying the backup T-Server configuration for warm standby	58
Modifying the primary T-Server configuration for hot standby	59
Modifying the backup T-Server configuration for hot standby	61
Activating Transfer Connect Service	82
Configuring Number Translation	94
Activating Event Propagation: basic configuration	100
Modifying Event Propagation: advanced configuration	101
Configuring T-Server Applications	103
Configuring Default Access Codes	105
Configuring Access Codes	106
Configuring access resources for the route transaction type	110
Configuring access resources for the dnis-pool transaction type	111
Configuring access resources for direct-* transaction types	112
Configuring access resources for ISCC/COF	112
Configuring access resources for non-unique ANI	113
Modifying DNSs for isolated switch partitioning	113
Configuring T-Server to start with the Management Layer	119
Starting T-Server on UNIX with a startup file	120
Starting T-Server on Windows with a startup file	121
Starting HA Proxy on UNIX manually	124
Starting HA Proxy on Windows manually	125
Starting T-Server on UNIX manually	126

Starting T-Server on Windows manually	126
Stopping T-Server on UNIX manually	127
Stopping T-Server on Windows manually	127
Configuring Digital Devices	137
Using a Hunt Group as Genesys Routing Point/ external Routing Point	139



Preface

Welcome to the *Framework 7.6 T-Server for Ericsson MD110 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 Ericsson MD110. 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 7.6 Deployment Guide*, and the Release Note for your T-Server.

This document is valid only for the 7.6 release of this product.

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

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

- [Intended Audience, page 12](#)
- [Chapter Summaries, page 12](#)
- [Document Conventions, page 14](#)
- [Related Resources, page 16](#)
- [Making Comments on This Document, page 17](#)

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.

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 7.6 Deployment Guide* and the Release Note mentioned earlier, and then read all of the sections of this document that apply to your software and its accompanying components. Refer back to the *Framework 7.6 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 7.6. If you take that approach, please also read Release Notes and refer to other related resources, such as the *Genesys 7 Events and Models Reference Manual* and *Voice Platform SDK 7.6 .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 7.6 Deployment Guide* before using this *T-Server Deployment Guide*. The *Framework 7.6 Deployment Guide* contains information about the Genesys software you must deploy before deploying T-Server.

Chapter Summaries

This *T-Server Deployment Guide* encompasses all information—including conceptual, procedural, and reference information—about Genesys T-Servers in general, and T-Server for Ericsson MD110 in particular. Depending on the subject addressed in a particular section, the document style may move from

narration, to instructions, to technical reference. To distinguish between general T-Server sections and those chapters intended for your particular T-Server, this document is divided into two main parts.

Part One—Common Functions and Procedures

Part One of this T-Server document, “Common Functions and Procedures,” consists of Chapters 1 through 5. These chapters contain architectural, functional, and procedural information common to all T-Servers:

- Chapter 1, “T-Server Fundamentals,” on [page 21](#), describes T-Server, its place in the Framework 7 architecture, T-Server redundancy, and multi-site issues. It does not, however, provide configuration and installation information.
- Chapter 2, “T-Server General Deployment,” on [page 35](#), presents configuration and installation procedures for all T-Servers. It follows the same general format you became familiar with during the configuration and installation of other Framework components, such as the Management Layer.
- Chapter 3, “High-Availability Deployment,” on [page 51](#), describes how to set up a high-availability environment for your T-Server.
- Chapter 4, “Multi-Site Support,” on [page 63](#), describes the variations available for T-Server implementations across geographical locations.
- Chapter 5, “Start and Stop T-Server Components,” on [page 117](#), describes how, and in what order, to start up T-Server among other Framework components. It also provides possible stopping commands.

Although you certainly would refer to these chapters if you have never before configured or installed T-Server, you might also use them, even if you are already familiar with T-Server, to discover any changes to functionality, configuration, and installation since you last deployed this component.

Genesys recommends that you use wizards to deploy T-Server. If you do, first read [Chapter 1](#) to familiarize yourself with T-Server, and then proceed with the deployment process using Framework wizards.

Part Two—Reference Information

Part Two of this T-Server document, “Reference Information,” consists of Chapters 6 through 12. These chapters contain reference information specific to T-Server for T-Server for Ericsson MD110. However, they also contain information on all T-Server options, both those specific to your T-Server and those common to all T-Servers.

- Chapter 6, “Ericsson MD110 Switch-Specific Configuration,” on [page 131](#), describes compatibility and configuration information specific to this T-Server, including how to set the DN properties and recommendations for the switch configuration.

- Chapter 7, “Supported Functionality in T-Server for Ericsson MD110,” on [page 145](#), describes which features this T-Server supports, including T-Library functionality, use of the Extensions attribute, and error messages.
- Chapter 8, “Common Configuration Options,” on [page 195](#), describes log configuration options common to all Genesys server applications.
- Chapter 9, “T-Server Common Configuration Options,” on [page 217](#), describes configuration options that are common to all T-Server types including options for multi-site configuration.
- Chapter 10, “Configuration Options in T-Server for Ericsson MD110,” on [page 241](#), describes configuration options specific to this T-Server, including the link-related options—those that address the interface between T-Server and the switch.
- Chapter 11, “High Availability (HA),” on [page 277](#), describes High Availability as it applies to the T-Server for Ericsson MD110, where there are any applicable differences from the T-Server High Availability feature described in Part One.
- Chapter 12, “Ericsson MD110 Call Flows,” on [page 279](#) describes call flows that are specific to the Ericsson MD110/T-Server combination.

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:

72fr_ref_09-2005_v7.2.000.00

You will need this number when you are talking with Genesys Technical Support about this product.

Type Styles

Italic

In this document, italic is used for emphasis, for documents’ titles, for definitions of (or first references to) unfamiliar terms, and for mathematical variables.

- Examples:**
- Please consult the *Genesys 7 Migration Guide* for more information.
 - *A customary and usual practice* is one that is widely accepted and used within a particular industry or profession.
 - Do *not* use this value for this option.
 - The formula, $x + 1 = 7$ where x stands for . . .

Monospace Font

A monospace font, which looks like teletype or typewriter text, is used for all programming identifiers and GUI elements.

This convention includes the *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages; the values of options; logical arguments and command syntax; and code samples.

- Examples:**
- Select the Show variables on screen check box.
 - Click the Summation button.
 - In the Properties dialog box, enter the value for the host server in your environment.
 - In the Operand text box, enter your formula.
 - Click OK to exit the Properties dialog box.
 - The following table presents the complete set of error messages T-Server distributes in EventError events.
 - If you select true for the inbound-bsns-calls option, all established inbound calls on a local agent are considered business calls.

Monospace is also used for any text that users must manually enter during a configuration or installation procedure, or on a command line:

- Example:**
- Enter exit on the command line.

Screen Captures Used in This Document

Screen captures from the product GUI (graphical user interface), as used in this document, may sometimes contain a minor spelling, capitalization, or grammatical error. 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.

Square Brackets

Square brackets indicate that a particular parameter or value is optional within a logical argument, a command, or some programming syntax. That is, the parameter's or value's presence is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information. Here is a sample:

```
smcp_server -host [/flags]
```

Angle Brackets

Angle brackets indicate a placeholder for a value that the user must specify. This might be a DN or port number specific to your enterprise. Here is a sample:

```
smcp_server -host <confighost>
```

Related Resources

Consult these additional resources as necessary:

- The *Framework 7.6 Deployment Guide*, which will help you configure, install, start, and stop Framework components.
- The *Framework 7.6 Configuration Options Reference Manual*, which will provide you with descriptions of configuration options for other Framework components.
- The *Framework 7.6 Configuration Manager Help*, which will help you use Configuration Manager.
- The *Genesys 7 Migration Guide*, also on the Genesys Documentation Library CD, which contains a documented migration strategy from Genesys product releases 5.x and later to all Genesys 7.x releases. Contact Genesys Technical Support for additional information.
- The *Genesys 7 Events and Models Reference Manual*, which contains an extensive collection of events and call models describing core interaction processing in Genesys environments.
- The *Voice Platform SDK 7.6 .NET (or Java) API Reference*, which contains technical details of T-Library functions.
- The *Genesys Technical Publications Glossary*, which ships on the Genesys Documentation Library CD and which provides a comprehensive list of the Genesys and CTI terminology and acronyms used in this document.
- The Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at <http://genesyslab.com/support>.

Information on supported hardware and third-party software is available on the Genesys Technical Support website in the following documents:

- *[Genesys 7 Supported Operating Systems and Databases](#)*
- *[Genesys 7 Supported Media Interfaces](#)*

Genesys product documentation is available on the:

- Genesys Technical Support website at <http://genesyslab.com/support>.
- Genesys Documentation Library CD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.

Making Comments on This Document

If you especially like or dislike anything about this document, please feel free to e-mail your comments to Techpubs.webadmin@genesyslab.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 information in this document only and to the way in which the information is presented. Speak to Genesys Technical Support 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.



Part

1

Part One: Common Functions and Procedures

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 21](#), describes T-Server, its place in the Framework 7 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 35](#), presents configuration and installation procedures for all T-Servers.
- Chapter 3, “High-Availability Deployment,” on [page 51](#), addresses high availability (HA).
- Chapter 4, “Multi-Site Support,” on [page 63](#), details the variations available for T-Server implementations across geographical locations.
- Chapter 5, “Start and Stop T-Server Components,” on [page 117](#), describes how, and in what order, to start up T-Server among other Framework components. It also provides possible stopping commands.

New for All T-Servers in 7.6

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 7.6 release of T-Server:

- **ISCC Transaction Monitoring support.** This release of T-Server supports the ISCC Transaction Monitoring that allows T-Server clients to monitor ISCC transactions of the call data transfer between T-Servers in a multi-site environment. See “ISCC Transaction Monitoring Feature” on [page 102](#) for details.
- **ANI information distribution control.** This release introduces a new configuration option that controls the distribution of the ANI information in TEvent messages. See “ani-distribution” on [page 218](#) for details.
- **Enhancement of use-data-from configuration option .** This option now includes the new valid value `active-data-original-call`. See “use-data-from” on [page 228](#) for details.
- **Enhanced agent session ID reporting.** T-Server now generates and reports a session ID associated with each new agent login (key `AgentSessionID` in `AttributeExtensions`) in agent-state events (`EventAgentLogin`, `EventAgentLogout`, `EventAgentReady`, and `EventAgentNotReady`), and also in the `EventRegistered` and `EventAddressInfo` messages for resynchronization. The agent session IDs are not synchronized with a backup T-Server and new agent session IDs will be assigned to existing agent sessions after a T-Server switchover. See the T-Server client's documentation for agent session ID reporting. Refer to the *Genesys 7 Events and Models Reference Manual* and/or *Voice Platform SDK 7.6 .NET (or Java) API Reference* for details on the key `AgentSessionID` in `AttributeExtensions`.
- **Client-side port definition support.** This release of T-Server supports a new security feature that allows a client application to define its connection parameters before connecting to the server application. Refer to the *Genesys 7.6 Security Deployment Guide* for details.

Notes:

- Configuration option changes common to all T-Servers are described in “Option Changes from Release 7.5 to 7.6” on [page 240](#).
 - For information about the new features that are available in your T-Server in the initial 7.6 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 7.6” on [page 20](#), 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 22](#)
- [Advanced Disconnect Detection Protocol, page 28](#)
- [Redundant T-Servers, page 29](#)
- [Multi-Site Support, page 32](#)
- [Agent Reservation, page 32](#)
- [Client Connections, page 33](#)
- [Next Steps, page 33](#)

Learning About T-Server

The *Framework 7.6 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.

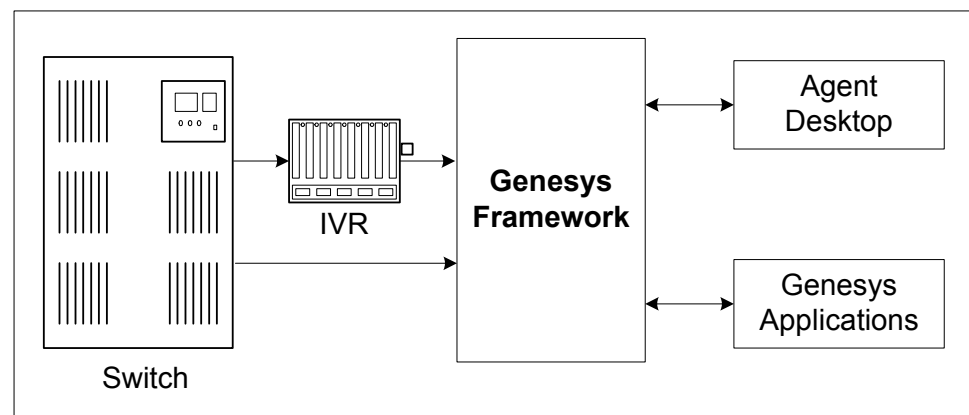


Figure 1: Framework in a Genesys Solution

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

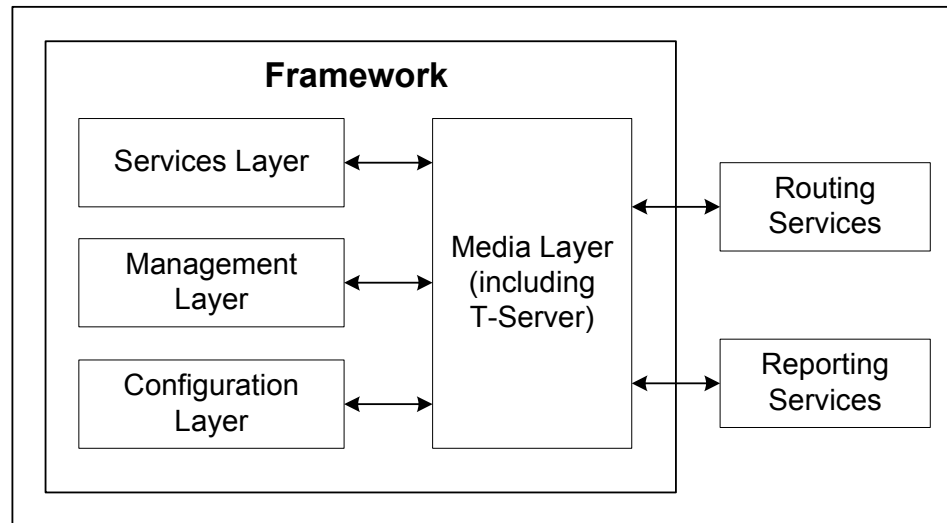


Figure 2: The Media Layer in the Framework Architecture

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.

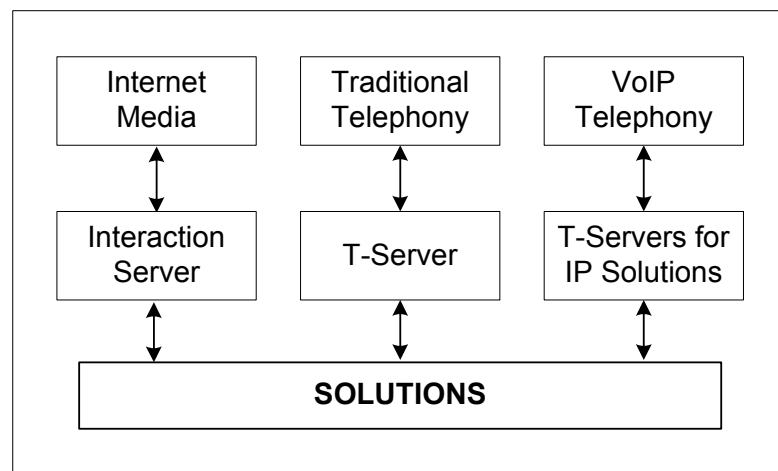


Figure 3: Media Layer Architecture

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, Call 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 7 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 7.6 .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 23](#) (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 T-Server you have 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.

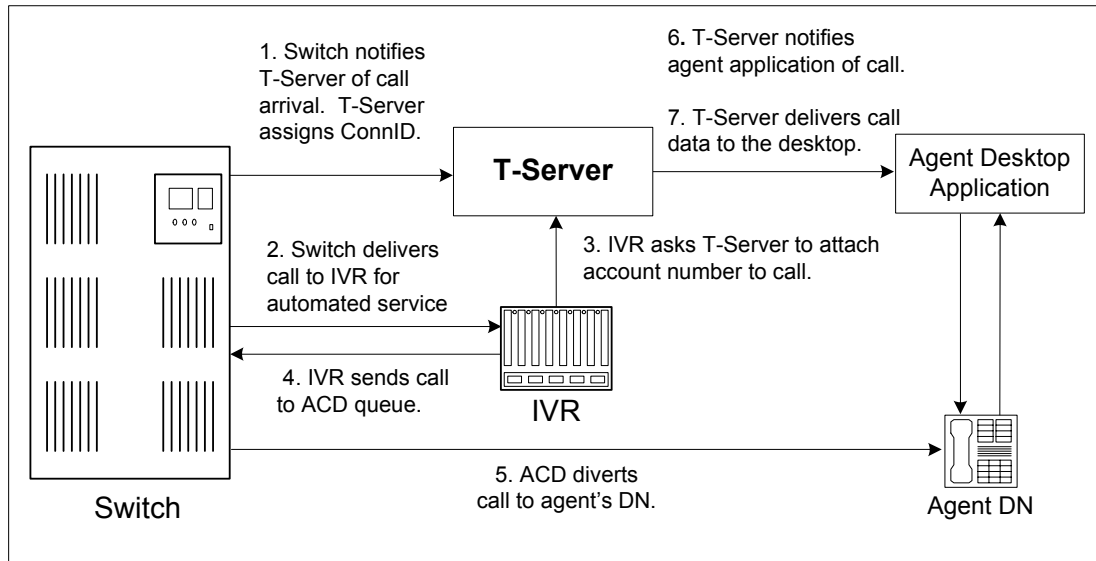


Figure 4: Functional T-Server Steps

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's 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 between the polling signal and the response to travel from one T-Server to another. If you don't 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.

Notes:

- Network T-Servers use a load-sharing redundancy schema instead of warm or hot standby. Specifics on your T-Server's HA capabilities are discussed in Part Two of this document.
 - IVR Server does not support simultaneous configuration of both Load Balancing functionality and warm standby. Only one of these is supported at a time.
-

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 7 Supported Media Interfaces* white paper located on the Technical Support website at <http://genesyslab.com/support/dl/retrieve/default.asp?item=A9CB309AF4DEB8127C5640A3C32445A7&view=item>.

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
Alcatel A4200/OXO	Yes	No	—
Alcatel A4400/OXE	Yes	No	—
Aspect ACD	Yes	No	1
Avaya Communication Manager	Yes	No ^a	—
Avaya INDeX	Yes	No	—
Cisco CallManager	Yes	No	—
DataVoice Dharma	Yes	No	—
Digitro AXS/20	Yes	No	—
EADS Intecom M6880	Yes	No	—
EADS Telecom M6500	Yes	No	—
eOn eQueue	Yes	No	—
Ericsson MD110	Yes	No	—
Fujitsu F9600	Yes	No	—
Huawei C&C08	Yes	No	—
Mitel SX-2000/MN-3300	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	—	—
Rockwell Spectrum	Yes	No	—
Samsung IP-PCX IAP	Yes	No	—
Siemens Hicom 300/HiPath 4000 CSTA I	Yes	No	—
Siemens HiPath 3000	Yes	No	—

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

T-Server Type	Hot Standby Supported	HA Proxy Required	Number of HA Proxy Components
Siemens HiPath 4000 CSTA III	Yes	No	—
Siemens HiPath DX	Yes	No	—
SIP Server	Yes	No	—
Tadiran Coral	Yes	No	—
Teltronics 20-20	Yes	Yes	1
Tenovis Integral 33/55	Yes	No	—
Network T-Servers^e			
AT&T	No	—	—
Concert	No	—	—
CRSP	No	—	—
DTAG	No	—	—
GenSpec	No	—	—
ISCP	No	—	—
IVR Server, using network configuration	No	—	—
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 (for which there is a Configuration Wizard) 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 63](#).

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 65](#)), 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 70](#) 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 7.6 .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 225](#) in the “T-Server Common Configuration Options” chapter in Part Two for more details.

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 and 64-bit modes (versions 5.1, 5.2, 5.3)	32767
HP-UX 32-bit and 64-bit modes (versions 11.0, 11.11, 11i v2)	2048
Linux 32-bit mode (versions RHEL 3.0, RHEL 4.0)	32768
Solaris 32-bit mode (versions 2.7, 8, 9)	4096
Solaris 64-bit mode (versions 2.7, 8, 9, 10)	65536
Tru64 UNIX (versions 4.0F, 5.1, 5.1B)	4096
Windows Server 2003	4096

Next Steps

Now that you have gained a general understanding of the roles and features available with T-Servers, you’re 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 35](#). 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 35](#)
- [Deployment Sequence, page 40](#)
- [Wizard Deployment of T-Server, page 41](#)
- [Manual Deployment of T-Server, page 43](#)
- [Next Steps, page 50](#)

Note: You *must* read the *Framework 7.6 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, Configuration Manager, and, at your option, Deployment Wizards. 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 7.6 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 7 Supported Operating Systems and Databases* white paper for the list of operating systems and database systems supported in Genesys releases 7.x. You can find this document on the Genesys Technical Support website at <http://genesyslab.com/support/dl/retrieve/default.asp?item=B6C52FB62DB42BB229B02755A3D92054&view=item>.

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 7.6 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 7.6 Deployment Guide* for recommendations on server locations.

Supported Platforms

Refer to the *Genesys Supported Media Interfaces* white paper for the list of supported switch and PABX versions. You can find this document on the Genesys Technical Support website at

<http://genesyslab.com/support/dl/retrieve/default.asp?item=A9CB309AF4DEB8127C5640A3C32445A7&view=item>.

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 7 Licensing Guide* available on the Genesys Documentation Library CD.

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 7 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 7 Licensing Guide* available on the Genesys Documentation Library CD.

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 in the relevant Wizard screens or 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 217](#). *Switch-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

Genesys recommends deploying T-Server by using the Media Configuration Wizard. However, if for some reason you must manually deploy T-Server, you will also find instructions for doing that in this chapter.

The recommended sequence to follow before deploying T-Server is described below. Steps 1 through 3 apply for both Wizard-based and manual deployment. For Wizard deployment, Steps 4 and 5 take place within the Wizard deployment process itself.

Wizard or Manual Deployment

1. Deploy Configuration Layer objects and ensure Configuration Manager is running (see the *Framework 7.6 Deployment Guide*).
2. Deploy Network objects (such as Host objects).
3. Deploy the Management Layer (see the *Framework 7.6 Deployment Guide*).

When manually deploying T-Server, you must continue with the next two steps. If you are deploying T-Server with the Configuration Wizard, the next two steps take place within the Wizard deployment process itself, where you can create and configure all the necessary objects for T-Server deployment.

Manual Deployment

4. Configure Telephony objects (see “Manual Configuration of Telephony Objects” on [page 44](#)):
 - Switching Offices
 - Switches
 - Agent Logins
 - DNs
5. Deploy the Media Layer:
 - T-Server (beginning with “Manual Configuration of T-Server” on [page 46](#)).
 - HA Proxy for a specific type of T-Server (applicable if you are using the hot standby redundancy type and your switch requires HA Proxy; see Table 1 on [page 30](#)).

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.

Verifying Starting Parameters

When installation is complete, verify the information on the **Start Info** tab to ensure that T-Server will run. See “Verifying the manual installation of T-Server” on [page 50](#).

Wizard Deployment of T-Server

Configuration Wizards facilitate component deployment. T-Server configuration and installation involves many steps, and Genesys strongly recommends that you set up T-Server using the Wizard rather than manually. T-Server Wizard guides you through a series of steps and options to customize your deployment of T-Server.

Wizard Configuration of T-Server

The first step to take for a Wizard-based configuration is to install and launch Genesys Wizard Manager. (Refer to the *Framework 7.6 Deployment Guide* for instructions.) When you first launch Genesys Wizard Manager, it suggests that you set up the Management Layer and then the Framework. The Framework setup begins with configuring and creating the objects related to T-Server, starting with the Switch and Switching Office objects, and the T-Server's Application object itself.

Note: With the Wizard, you create your T-Server Application object in the course of creating your Switch object.

During creation of the Switch object, you also have an opportunity to run the Log Wizard to set up T-Server logging. Then, you can specify values for the most important T-Server options. Finally, you can create contact center objects related to T-Server, such as DNSs, Agent Logins, and some others.

Note: During configuration of a Switch object, the Wizard prompts you to copy a T-Server installation package to an assigned computer. After that package is copied to the destination directory on the T-Server host, complete the last steps of the T-Server configuration. Then, install T-Server on its host.

After you complete the Framework configuration, the Genesys Wizard Manager screen no longer prompts you to set up the Framework. Instead, it suggests that you set up your solutions or add various contact center objects to the Framework configuration, including the Switch, DNSs and Places, Agent Logins, Agent Groups, Place Groups, and, in a multi-tenant environment, a Tenant. In each case, click the link for the object you wish to create. Again, you create a new T-Server Application object in the course of creating a new Switch object.

Wizard Installation of T-Server

After creating and configuring your T-Server and its related components with the Wizard, you proceed to T-Server installation. That installation process closely mimics that of previously installed components.

Note: Certain Wizard-related procedures are not described in this document. Refer to the *Framework 7.6 Deployment Guide* for general instructions.

Warning! Genesys does not recommend installation of its components via a Microsoft Remote Desktop connection. The installation should be performed locally.

Procedure: Installing T-Server on UNIX using Wizard

Start of procedure

1. In the directory to which the T-Server installation package was copied during Wizard configuration, 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, confirm the application name of the T-Server that is to be installed.
5. Specify the destination directory into which T-Server is to be installed, with the full path to it.
6. If asked which version of the product to install, the 32-bit or the 64-bit, choose the one appropriate to your environment.
7. Specify the license information that T-Server is to use.
8. 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 test your configuration and installation, go to Chapter 5, “Start and Stop T-Server Components,” on [page 117](#), and try it out.

- To configure and install redundant T-Servers, see Chapter 3, “High-Availability Deployment,” on [page 51](#).
- To install T-Servers for a multi-site environment, proceed to Chapter 4, “Multi-Site Support,” on [page 63](#).

Procedure: Installing T-Server on Windows using Wizard

Start of procedure

1. Open the directory to which the T-Server installation package was copied during Wizard configuration.
2. Locate and double-click `Setup.exe` to start the installation. The `Welcome` screen launches.
3. When prompted, specify the connection parameters to the Configuration Server associated with this T-Server.
4. Identify the T-Server `Application` object in the Configuration Layer to be used by this T-Server.
5. Specify the license information that T-Server is to use.
6. Specify the destination directory into which T-Server is to be installed.
7. Click `Install` to begin the installation.
8. 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 test your configuration and installation, go to Chapter 5, “Start and Stop T-Server Components,” on [page 117](#), and try it out.
- To configure and install redundant T-Servers, see Chapter 3, “High-Availability Deployment,” on [page 51](#).
- To install T-Servers for a multi-site environment, proceed to Chapter 4, “Multi-Site Support,” on [page 63](#).

Manual 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.

Manual Configuration of Telephony Objects

This section describes how to manually configure T-Server Telephony objects if you are using Configuration Manager.

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` 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 63](#), 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! DNs with the Register flag set to false may not be processed at T-Server startup; therefore, associations on the switch will be created only when T-Server client applications require DN registration.

Multi-Site Operations

See the section, “Configuring Multi-Site Support” on [page 103](#), for information on setting up DNs for multi-site operations.

Manual Configuration of T-Server

Note: Use the *Framework 7.6 Deployment Guide* to prepare accurate configuration information. You may also want to consult *Configuration Manager 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 manually

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 7.6 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. The configuration options common to all T-Servers are described in the “T-Server Common Configuration Options” chapter. The switch-specific configuration options are described in a separate chapter. T-Server also uses common Genesys log options, described in the “Common Configuration Options” chapter.

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 63](#).

End of procedure

Next Steps

- See “Manual Installation of T-Server” on [page 48](#).

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 7.6 Security Deployment Guide*.

e. Click OK.

5. Click OK to save the new configuration.

End of procedure

Manual Installation of T-Server

The following directories on the Genesys 7.6 Media product CD 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 manually

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 manually” on [page 46](#) 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 asked which version of the product to install, the 32-bit or the 64-bit, choose the one appropriate to your environment.
9. 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.
10. 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 manual installation of T-Server” on [page 50](#).
- To test your configuration and installation, go to Chapter 5, “Start and Stop T-Server Components,” on [page 117](#), and try it out.
- To configure and install redundant T-Servers, see Chapter 3, “High-Availability Deployment,” on [page 51](#).
- To install T-Servers for a multi-site environment, proceed to Chapter 4, “Multi-Site Support,” on [page 63](#).

Procedure: Installing T-Server on Windows manually

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 manually” on [page 46](#) 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 manual installation of T-Server” on [page 50](#).
- To test your configuration and installation, go to Chapter 5, “Start and Stop T-Server Components,” on [page 117](#), and try it out.
- To configure and install redundant T-Servers, see Chapter 3, “High-Availability Deployment,” on [page 51](#).
- To install T-Servers for a multi-site environment, proceed to Chapter 4, “Multi-Site Support,” on [page 63](#).

Procedure:

Verifying the manual installation of T-Server

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

Prerequisites

- [Installing T-Server on UNIX manually, page 48](#)
- [Installing T-Server on Windows manually, page 49](#)

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 either used the Wizard to configure and install T-Server, or you have done it manually, using Configuration Manager. In either case, if you want to test your configuration and installation, go to Chapter 5, “Start and Stop T-Server Components,” on [page 117](#), and try it out. Otherwise, if you want to configure and install redundant T-Servers, see Chapter 3, “High-Availability Deployment,” on [page 51](#). If you want to install T-Servers for a multi-site environment, proceed to Chapter 4, “Multi-Site Support,” on [page 63](#).



Chapter

3

High-Availability Deployment

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

Every switch/T-Server combination offers different high-availability options. The Framework Management Layer currently supports two types of redundant configurations: warm standby and hot standby. All T-Servers offer the warm standby redundancy type and, starting with release 7.1, the hot standby redundancy type is implemented in T-Servers for most types of switches. Some T-Servers support a switch's ability to provide two CTI links to two T-Servers or even one CTI link to two T-Servers. Other T-Servers require Genesys's HA Proxy in order to support the hot standby redundancy type. See Table 1 on [page 30](#) 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 52](#)
- [Hot Standby Redundancy Type, page 53](#)
- [Prerequisites, page 55](#)
- [Warm Standby Deployment, page 56](#)
- [Hot Standby Deployment, page 58](#)
- [Next Steps, page 62](#)

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 7.6 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.

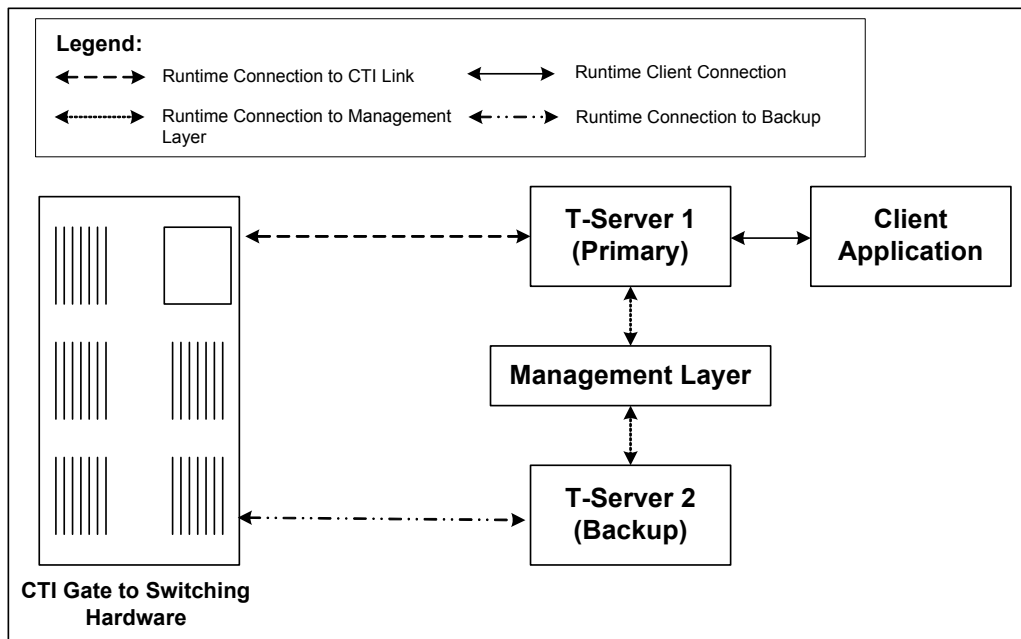


Figure 5: Warm Standby Redundancy Architecture

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 7.6 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 54](#).)

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*.

Note: Although most of T-Servers support hot standby (for which the documentation appears in this guide), IVR Server does not support this feature.

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

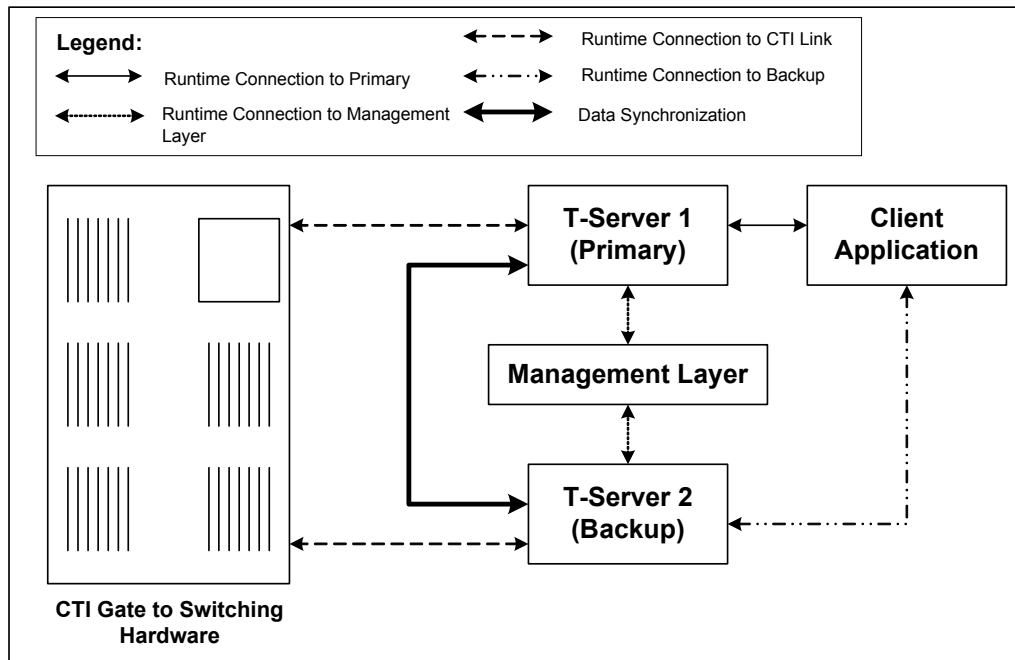


Figure 6: Hot Standby Redundancy Architecture

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 65](#) for ISCC feature descriptions.

- Allocation of ISCC-controlled resources.

- 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.
- 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 2, 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:

- | | |
|------------------------------|---|
| Wizard
Deployment | <ul style="list-style-type: none"> • If you used wizards to configure T-Servers and selected the warm standby redundancy type, no additional configuration is required for your T-Servers. |
| Manual
Deployment | <ul style="list-style-type: none"> • If you did not use wizards to configure T-Servers: <ol style="list-style-type: none"> a. Manually configure two T-Server <code>Application</code> objects as described in “Manual Configuration of T-Server” on page 46. b. Make sure the <code>Switch</code> object is configured for the switch these T-Servers should serve, as described in “Manual Configuration of T-Server” on page 46. c. Modify the configuration of the primary and backup T-Servers as instructed in the following sections. |

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

Manual 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

- [Modifying the backup T-Server configuration for warm standby, page 58](#)

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 “Manual Installation of T-Server” on [page 48](#) 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:

Wizard Deployment

- If you used wizards to configure T-Servers and selected the hot standby redundancy type, no additional configuration is required for your T-Servers.

**Manual
Deployment**

- If you did not use wizards to configure T-Servers:
 - a. Manually configure two T-Server Applications objects as described in “Configuring T-Server manually” on [page 46](#).
 - b. Make sure the Switch object is configured for the switch these T-Servers should serve, as described in “Manual Configuration of Telephony Objects” on [page 44](#).
 - c. 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 62](#)).

Table 1 on [page 30](#) summarizes hot standby redundancy support in various T-Servers. For detailed, up-to-date information on the subject, see the *Genesys 7 Supported Media Interfaces* white paper located on the Technical Support website at

<http://genesyslab.com/support/dl/retrieve/default.asp?item=A9CB309AF4DEB8127C5640A3C32445A7&view=item>.

Manual 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 47](#).

- 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

- [Modifying the backup T-Server configuration for hot standby, page 61](#)

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 47](#).

- 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 “Manual Installation of T-Server” on [page 48](#) for both installations.

Next Steps

At this point, you have learned how to configure and install redundant T-Servers. Go to Chapter 5, “Start and Stop T-Server Components,” on [page 117](#), to test your configuration and installation, or continue with Chapter 4, “Multi-Site Support,” on [page 63](#), 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 64](#)
- [ISCC Call Data Transfer Service, page 65](#)
- [ISCC/COF Feature, page 83](#)
- [Number Translation Feature, page 87](#)
- [Network Attended Transfer/Conference Feature, page 95](#)
- [Event Propagation Feature, page 97](#)
- [ISCC Transaction Monitoring Feature, page 102](#)
- [Configuring Multi-Site Support, page 103](#)
- [Next Steps, page 116](#)

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 217](#).

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 79](#) and Table 4 on [page 84](#).

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, call history). 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 70](#) and “Transfer Connect Service Feature” on [page 82](#).
 - 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 83](#)).
 - Number Translation feature (see [page 87](#)).
 - Network Attended Transfer/Conference (NAT/C) feature (see [page 95](#)).

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 97](#))
 - Party Events propagation (see [page 99](#))

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

Note: In distributed networks, Genesys recommends using call flows that prevent multiple reappearances of the same call instance, and call topology loops. This approach ensures that all T-Servers involved with the call report the same ConnID, and also optimizes telephony trunk allocation (that is, it prevents 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 ConnID of the call
- Updates to user data attached to the call at the previous site
- Call history

Note: Load-sharing IVR Servers and Network T-Servers cannot be designated as the destination location for ISCC.

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*.

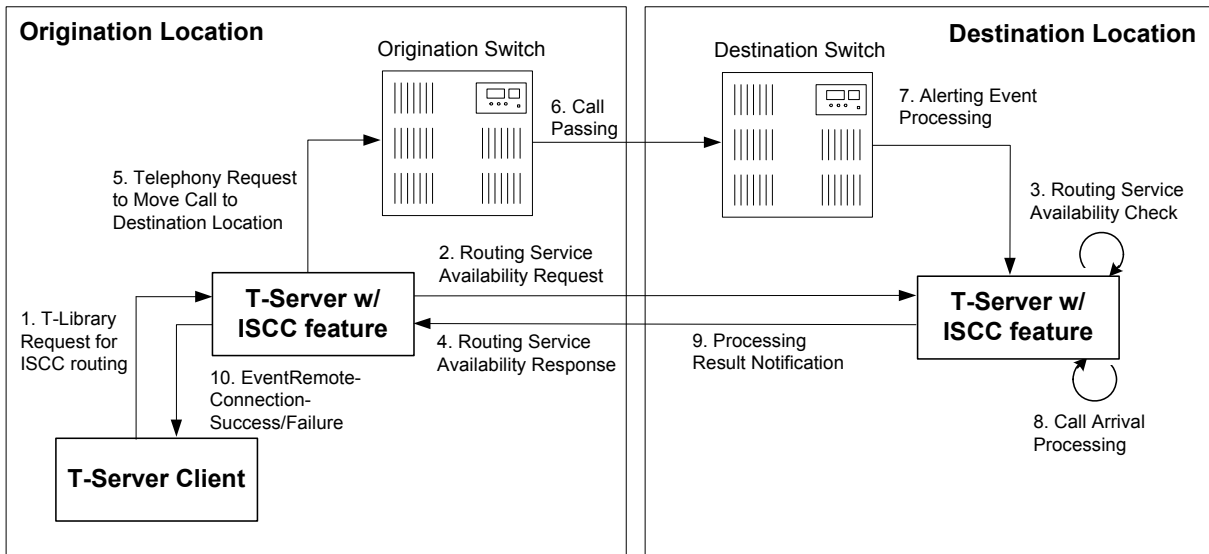


Figure 7: Steps in the ISCC Process

ISCC Call Flow

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 7.6 .NET (or Java) API Reference*) or a string value equal to one of the following: `default`, `route`, `direct` (or `direct-callid`), `direct-network-callid`, `direct-notoken`, `direct-ani`, `direct-uui`, `direct-digits`, `reroute`, `dnis-pool`, `pullback`, or `route-uui`.
- If the client does not specify the transaction type in the request or specifies the `default` transaction type, T-Server checks the Switch configuration for the transaction type configured in the `Access Code` (or `Default Access Code`) properties:
 - If the `Route Type` property of the `Access Code` is set to any value other than `default`, T-Server uses the specified value as the transaction type.
 - If the `Route Type` property of the `Access Code` is set to the `default` value, T-Server uses the first value from the list specified in the `cast-type` configuration option configured for the destination T-Server. If no value has been specified for the `cast-type` option, the default value of `route` is used as the transaction type.

Note: See “Switches and Access Codes” on [page 104](#) for more information on Access Codes and Default Access Codes.

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`, 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 a DNIS number 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. Refer to Chapter 9, “T-Server Common Configuration Options,” on [page 217](#) 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`, 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`, `uvi`, `ani`, or `digits` extension), the call reaches the destination DN directly.

For the transaction types `route` and `route-uvi`, 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`, 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 dn-is-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.

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 72](#)).

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 72](#). Use Table 3 on [page 79](#) 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 217](#) 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 72](#)
- `direct-notoken`, [page 74](#)
- `dnis-pool`, [page 74](#)
- `pullback`, [page 76](#)
- `reroute`, [page 76](#)
- `route` (aliased as `route-notoken`), the default transaction type, [page 77](#)

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

- `direct-callid` (aliased as `direct`), [page 72](#)
- `direct-digits` (reserved for Genesys Engineering)
- `direct-network-callid`, [page 73](#)
- `direct-uui`, [page 73](#)
- `route-uui`, [page 78](#)

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 network 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 is capable of using this network feature for call matching.

Warnings!

- Depending on the switch platform, it is possible to inherit the ANI attribute after routing a call to a remote destination, and after performing a Single-Step Transfer and other telephone actions. However, ISCC only works properly in scenarios where the ANI attribute on the destination T-Server is represented by exactly the same digit string as on the origination T-Server.
- Typically, the ANI attribute represents the original call identifier (customer phone number), which guarantees that the attribute remains unique. However, you can use the `non-unique-ani` resource type to block ISCC from matching calls based on an ANI that is known to be non unique. (See “Configuring access resources for non-unique ANI” on [page 113](#) for details.)

Notes:

- Some switches, such as Nortel Communication Server 2000/2100 (formerly DMS-100) and Avaya Communication Manager (formerly DEFINITY ECS (MV)), may omit the ANI attribute for internal calls—that is, for calls whose origination and destination DNs belong to the same switch. If this is the case, do not use the `direct-ani` transaction type when making, routing, or transferring internal calls with the ISCC feature.
- When the `direct-ani` transaction type is in use, the Number Translation feature becomes active. See “Number Translation Feature” on [page 87](#) for more information on the feature configuration.
- With respect to the `direct` transaction types, Network T-Servers and load-sharing IVR Servers are not meant to play the role of destination T-Servers for call routing. Using Network T-Server with these transaction types requires special architecture.

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. They are applied only to the call that is in progress, and do not apply to functions that involve in the creation of a new call (for example, `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. Refer to Part Two of this document for information about settings specific for your T-Server type.

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 as 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. Moreover, the trunks involved must 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 be reached from within the contact center only (for example, the second line of support, which customers cannot contact directly).
 - With respect to the `direct` transaction types, Network T-Servers and load-sharing IVR Servers are not meant to play the role of 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`, 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 employing a “behind the SCP” architecture—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`, `TSingleStepTransfer`, or `TGetAccessNumber` 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 be used only 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

Only Network T-Servers use the transaction type `reroute`, and only in the following scenario:

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—that is, it sends `EventRouteRequest` and attaches the call's user data.

Notes:

- The transaction type `reroute` can be used only 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](#).

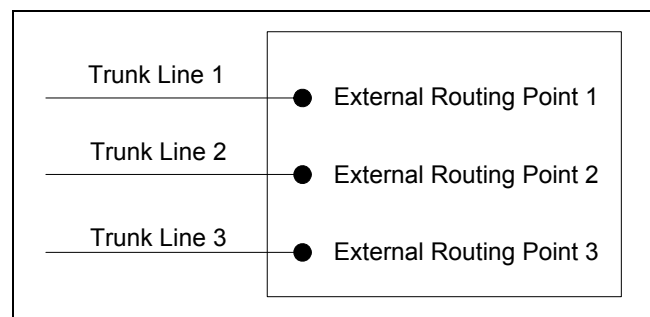


Figure 8: Point-to-Point Trunk Configuration

Note: Dedicated DNIs of the External Routing Point type must be configured in a switch. See “Configuring Multi-Site Support” on [page 103](#).

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](#).

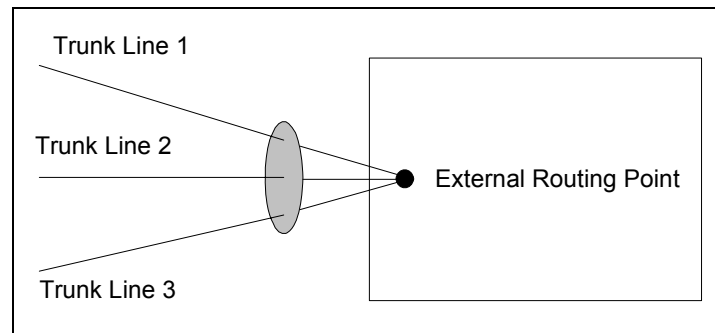


Figure 9: Multiple-to-Point Trunk Configuration

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 77](#)) and the UUI matching feature of the `direct-uui` transaction type ([page 73](#)). 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. Moreover, the trunks involved must 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-net-work-callid	dnis-pool	pull-back
	one-to-one	multiple-to-one									
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		Yes ^f	Yes ^f				
Avaya Communication Manager	Yes				Yes	Yes	Yes				
Avaya INDeX	Yes			Yes		Yes	Yes				
Cisco CallManager	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					
Ericsson MD110	Yes			Yes ^a		Yes	Yes ^a				
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-net-work-callid	dnis-pool	pull-back
	one-to-one	multiple-to-one									
Huawei C&C08	Yes			Yes							
Mitel SX-2000/MN3300	Yes			Yes		Yes	Yes				
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
Rockwell Spectrum	Yes	Yes		Yes		Yes ^f	Yes ^f				
Samsung IP-PCX IAP	Yes			Yes		Yes					
Siemens Hicom 300/HiPath 4000 CSTA I	Yes			Yes	Yes ^b	Yes	Yes				
Siemens HiPath 3000	Yes			Yes		Yes					
Siemens HiPath 4000 CSTA III	Yes				Yes ^b	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- net- work- callid	dnis- pool	pull- back
	one-to- one	multiple- to-one									
Siemens HiPath DX	Yes			Yes	Yes	Yes	Yes				
SIP Server	Yes				Yes	Yes					
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
Network SIP Server	Yes					Yes	Yes			Yes	
Sprint	Yes										

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

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

- Not supported in the case of function `TRequestRouteCall` 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.
- 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.
- Not supported if two T-Servers are connected to different nodes.
- There are some switch-specific limitations when assigning CSTA correlator data UUI to a call.
- Supported only on ABCF trunks (Alcatel internal network).
- To use this transaction type, you must select the `Use Override` check box on the Advanced tab of the DN Properties dialog box.

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

- Open the T-Server Application's Properties dialog box.
- Click the Options tab.
- 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 silence 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 `TRequestRouteCall` be played via the `ASAI-send-DTMF-single` procedure.

ISCC/COF 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 by means other than ISCC. 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 switches 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	Yes
Aspect ACD	Yes
Avaya Communication Manager	Yes
Nortel Communication Server 2000/2100	Yes
Nortel Communication Server 1000 with SCCS/MLS	Yes
Rockwell Spectrum	Yes

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, and CallHistory are received for the matched call from the call's previous location.

Warning! Depending on the switch platform, it is 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. See “Number Translation Feature” on [page 87](#) for more information on the feature configuration.

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.

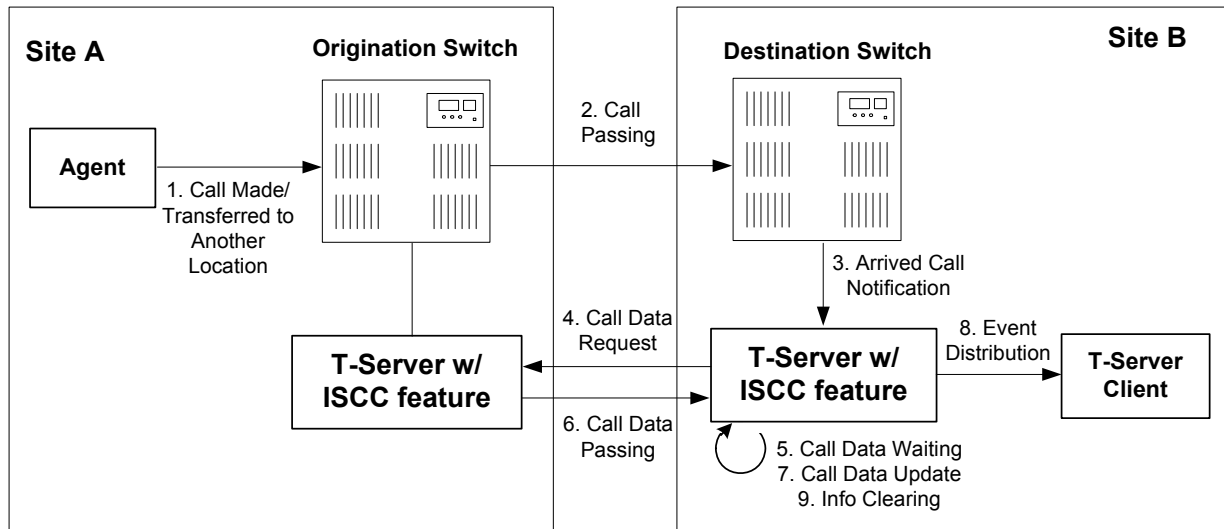


Figure 10: Steps in the ISCC/COF Process

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 was overflowed from any of them.

To determine which calls to check as possibly overflowed, T-Server relies on the Switch object configuration:

- If no COF DNs (that is, DNs of the Access Resources type with the Resource Type set to `cof-in` or `cof-not-in`) are configured for the destination switch, the ISCC/COF feature of the destination T-Server checks all arriving calls.
- If a number of COF DNs are configured for the destination switch, one of three scenarios occurs:

- If the COF DN's with the `cof-in` setting for the Resource Type property are configured, the ISCC/COF checks for overflow only those calls that arrive to those `cof-in` DN's that are Enabled.
- If no DN's with the `cof-in` setting for the Resource Type property are configured, but some DN's have the `cof-not-in` setting for the Resource Type property, the ISCC/COF checks for overflow only those calls that arrive to those `cof-not-in` DN's that are Disabled.
- If no DN's with the `cof-in` setting for the Resource Type property are configured, some DN's have the `cof-not-in` setting for the Resource Type property, and some other DN's do not have any setting for the Resource Type property, the ISCC/COF checks for overflow only those calls that arrive to the DN's without any setting for the Resource Type property.
- In all other cases, no calls are 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, 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, 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 88](#) 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 93](#) 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 94](#).

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 notations are explained starting at the highest level, with the name of a component notation and a basic definition of each component that comprises it. 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)`

where:

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

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

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

where:

- `symbol-part` represents digits, symbols, or a combination. Symbols are rarely used. They are not used in the United States.
- `group-identifier` are letters that represent groups of numbers. A letter in the out-pattern represents one or more digits, based on the number of times the letter is used in the in-pattern.
- `*param-part` represents an additional parameter, such as phone-context. Reminder: an asterisk means that 0 to an infinite number of these are acceptable.

For example, in `rule-04`; `in-pattern=1AAABBBCCC`; `out-pattern=91ABC`, `91` is the `symbol-part`; `A`, `B`, and `C` are `group-identifiers` in the out-pattern,

each representing three digits, since there are three instances of each in the in-pattern.

Note: Prefix an out-pattern value with a plus sign (+) for the inbound rule when the output must be in a global 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 93](#)) 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 = 322332.

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 217](#).

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).

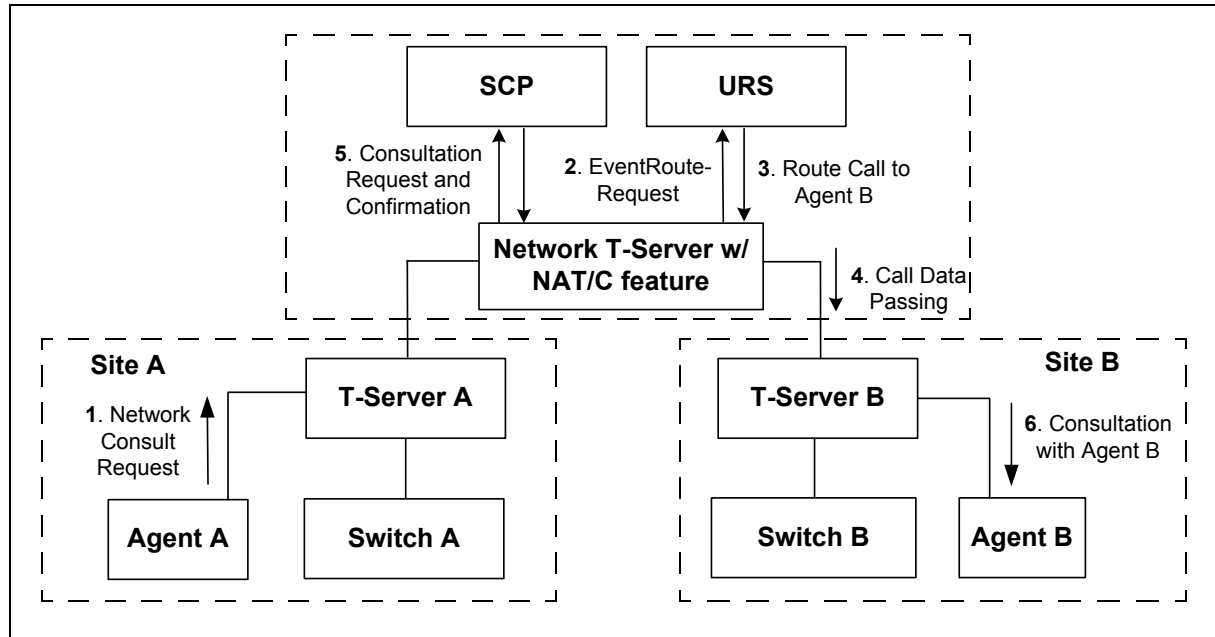


Figure 11: Steps in the NAT/C Process in URS-Controlled Mode

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 7.6 .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 65](#) 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.

Warnings!

- The `OtherDN` and `ThirdPartyDN` attributes might not be present in the events distributed via the Event Propagation feature.
 - The Event Propagation feature will not work properly with installations that use switch partitioning.
-

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 7 Events and Models Reference Manual*.

Basic and Advanced 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 217](#).
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:
[Modifying Event Propagation: advanced configuration, page 101](#)

Procedure: Modifying Event Propagation: advanced configuration

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

Prerequisites

- [Activating Event Propagation: basic configuration, page 100](#)

Overview

You can set Event Propagation parameters using:

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

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

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

Start of procedure

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

Note: If no Default Access Code is configured, see [page 105](#) for instructions. If no Access Codes are configured, see [page 106](#) for instructions.

4. In the Switch Access Code Properties dialog box that opens, specify a value for the ISCC Protocol Parameters field as follows:

- To enable distribution of both user data associated with the call and call-party-associated 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:
`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. See *Genesys 7 Events and Models Reference Manual* and *Voice Platform SDK 7.6 .NET (or Java) API Reference* for more information about support of this feature.

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

Configuring Multi-Site Support

Prior to configuring T-Server to support multi-site operation, you must read the “Licensing Requirements” on [page 37](#), as well as previous sections of this chapter on multi-site deployment. In particular, Table 3 on [page 79](#) shows which transaction types are supported by a specific T-Server, while Table 4 on [page 84](#) 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 names of each T-Server application, port assignments, switch names, and so on), and use Configuration Manager to create and partially configure each T-Server object. Review multi-site option values in the “Multi-Site Support Section” on [page 226](#) and determine what these values need to be, based on your network topology.

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

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

You must configure these objects for origination and destination locations. Multi-site support features activate automatically at T-Server startup. See “DNs” on [page 110](#) for details.

Applications

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

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

Procedure: Configuring T-Server Applications

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

Start of procedure

1. Open the T-Server Application's Properties dialog box.
2. Click the Connections tab and click Add to add a connection to the appropriate T-Server. The Connection Info Properties dialog box displays.
3. Use the Browse button to search for the T-Server you want to connect to, and fill in the following values:
 - Port ID
 - Connection Protocol
 - Local Timeout
 - Remote Timeout
 - Trace Mode
4. Click the Options tab. Create a new section called extrouter or open an existing section with this name.

Note: If you do not create the extrouter section, T-Server works according to 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 “Multi-Site Support Section” on [page 226](#), 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 7.6, or when using T-Servers of different releases (including 7.6) in the same environment, see “Compatibility Notes” on [page 109](#).

Procedure: Configuring Default Access Codes

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

Prerequisites

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

Start of procedure

1. Among configured Switches, select the Switch that the configured T-Server relates to.
2. Open the Switch Properties dialog box and click the Default Access Codes tab.
3. Click Add to open the Access Code Properties dialog box.
4. In the Code field, specify the access code used by remote switches to reach a DN at this switch. An access code is used as a prefix to the remote switch numbers.

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

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

End of procedure**Next Steps**

- See [“Configuring Access Codes.”](#)

Procedure: Configuring Access Codes

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

Prerequisites

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

Start of procedure

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

4. In the Switch field, specify the switch that this switch can reach using this access code. Use the Browse button to locate the remote switch.
5. In the Code field, specify the access code used to reach a DN at the remote switch from this switch. An access code is used as a prefix to the remote switch numbers.

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

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

When you select Target ISCC as your target type, the Properties dialog box changes its lower pane to the Source 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 Source 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 5](#):

Table 5: 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 101 .
direct-network-callid=<>	For configuration information, see Part Two of this document. (Use Table 3 on page 79 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 6](#):

Table 6: Target Type: ISCC Call Overflow Parameters

ISCC Call Overflow Parameters	Description
<code>match-callid</code>	Matches calls using network <code>CallID</code> .
<code>match-ani</code>	Matches calls using ANI.
<code>inbound-only=<boolean></code>	Default is <code>true</code> . Setting <code>inbound-only</code> to <code>true</code> 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 7](#) contains cross-reference information on transaction types that the Configuration Layer and T-Server use.

Table 7: 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 <code>cast-type</code> option for the T-Server at the destination site
Direct	<code>direct-callid</code>
Direct ANI	<code>direct-ani</code>
Direct Digits	<code>direct-digits</code>
Direct DNIS and ANI	Reserved
Direct Network Call ID	<code>direct-network-callid</code>
Direct No Token	<code>direct-notoken</code>
Direct UUI	<code>direct-uui</code>
DNIS Pooling	<code>dnis-pooling</code>
Forbidden	External routing to this destination is not allowed
ISCC defined protocol	Reserved
PullBack	<code>pullback</code>

Table 7: Route Type and ISCC Transaction Type Cross-Reference (Continued)

Route Type Field Value	ISCC Transaction Type
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 7.6, or when using T-Servers of different releases (including 7.6) 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 7.x, 6.5, and 6.1:
 - Use the Target ISCC access code for transactions with T-Servers of releases 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, 6.5, and 7.x.

- 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

Overview

You can use any configured DN as an access resource for the `direct-*` transaction types. (The `*` symbol stands for any of the following: `callid`, `vui`, `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—for example, Aspect ACD, Nortel Communication Server 2000/2100, and Spectrum.

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 84](#) 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.

Note: The number of the access resource must match the name of a DN configured on the switch (usually, an ACD Queue) so that T-Server can determine if the calls arriving to this DN are overflowed calls.

2. On the General tab of the DN Properties dialog box, specify the number of the configured DN as the value for the Number field.
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 DN's 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 DN's 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 DN's 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 DN's for isolated switch partitioning

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

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

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

Start of procedure

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

End of procedure

Configuration Examples

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

Multiple Transaction Types

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

In this example, you configure an origination and a destination switch for as described in “Switches and Access Codes” on [page 104](#).

1. Among configured Switches, select the origination Switch.
2. Open the Switch Properties dialog box and click the Default Access Codes tab.
3. Click Add to open the Access Code Properties dialog box.
4. Set the Access Code field to 9.
5. When you are finished, click Apply.
6. Among configured Switches, select the destination Switch.
7. Under the destination Switch, configure a DN as described in “Configuring access resources for the route transaction type” on [page 110](#).

8. Set the DN Number field to 5001234567.
9. Click the Advanced tab of this DN's Properties dialog box.
10. Select the Use Override check box and enter 1234567 in the Use Override field.
11. When you are finished, click Apply or Save.
12. Use a T-Server client application to register for this new DN with the destination T-Server and, therefore, with the switch.
13. Request to route a call from any DN at the origination switch to the destination DN you have just configured:
 - If you are using the route ISCC transaction type, the client requests that T-Server deliver a call to a destination location using the DN number 5001234567. ISCC requests that the switch dial one of the external routing points at the destination location, using the value either of the Access Number field or of the Access Code field, which is 9, concatenated with the external routing point at the destination location. The call is routed to the DN number 5001234567.
 - If you are using the direct-ani ISCC transaction type, the client requests that T-Server deliver a call to a destination location using the DN number 1234567, which is the Use Override value. ISCC requests that the switch dial 91234567, which is a combination of the Switch Access Code value and the Use Override value. The destination T-Server is waiting for the call to directly arrive at DN number 5001234567.

Call Overflow Methods

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

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

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

when configuring Switch Access Codes as described on [page 106](#).

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

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

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

when configuring Switch Access Codes as described on [page 106](#).

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

Next Steps

Continue with Chapter 5, “Start and Stop T-Server Components,” on [page 117](#) to test your configuration and installation.



Chapter

5

Start and Stop 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 117](#)
- [Starting and Stopping with the Management Layer, page 119](#)
- [Starting with Startup Files, page 120](#)
- [Starting Manually, page 121](#)
- [Verifying Successful Startup, page 126](#)
- [Stopping Manually, page 127](#)
- [Starting and Stopping with Windows Services Manager, page 128](#)
- [Next Steps, page 128](#)

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 7.6 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 7.6 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 117](#).
6. When you are finished, click Apply.
7. Click OK to save your changes and exit the Properties dialog box.

End of procedure

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

After its command-line parameters are correctly specified in the Properties dialog box, you can start and stop T-Server from Solution Control Interface (SCI), which is the graphical interface component of the Management Layer. (The starting procedure for SCI is described in the *Framework 7.6 Deployment Guide*.) *Framework 7.6 Solution Control Interface Help* provides complete instructions on starting and stopping applications.

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

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

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

Starting with Startup Files

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

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

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

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

Start of procedure

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

```
sh run.sh
```

End of procedure

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

Start of procedure

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

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

Or

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

```
startServer.bat
```

End of procedure

Starting Manually

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

The command-line parameters common to Framework server components are described on [page 117](#).

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

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

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

The following sections provide general instructions for starting HA Proxy and T-Server manually. Along with these instructions, refer to [Table 8](#), which lists T-Servers and HA Proxy executable file names for supported switches for Windows and UNIX operating systems.

Table 8: T-Server and HA Proxy Executable Names

Switch Type	T-Server Executable File Name		HA Proxy Executable File Name	
	UNIX	Windows	UNIX	Windows
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	
Cisco CallManager	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	
Ericsson MD110	md110_server	md110_server.exe	Not Applicable	
Fujitsu F9600	Not Applicable	F9600_server.exe	Not Applicable	
Huawei C&C08	cc08_server	cc08_server.exe	Not Applicable	
Mitel SX-2000/ MN 3300	SX2000_server	SX2000_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
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	

Table 8: T-Server and HA Proxy Executable Names (Continued)

Switch Type	T-Server Executable File Name		HA Proxy Executable File Name	
	UNIX	Windows	UNIX	Windows
Rockwell Spectrum	spectrum_server	spectrum_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 iCCL	RealitisDX-iCCL_server	RealitisDX-iCCL_server.exe	Not Applicable	
SIP Server	sip_server	sip_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	nts_server	nts_server.exe	Not Applicable	
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	

Table 8: T-Server and HA Proxy Executable Names (Continued)

Switch Type	T-Server Executable File Name		HA Proxy Executable File Name	
	UNIX	Windows	UNIX	Windows
NGSN	nts_server	nts_server.exe	Not Applicable	
Network SIP Server	tsip_server	tsip_server.exe	Not Applicable	
Sprint	sprint_server	sprint_server.exe	Not Applicable	
SR3511	sr3511_server	sr3511_server.exe	Not Applicable	
Stentor	stentor_server	stentor_server.exe	Not Applicable	

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

HA Proxy

If you do not use HA Proxy in your Genesys implementation, proceed to “T-Server” on [page 125](#).

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

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

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

The command-line parameters common to Framework server components are described on [page 117](#).

Procedure: Starting HA Proxy on UNIX manually

Start of procedure

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

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

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

Table 8 on [page 122](#) lists HA Proxy executable names for supported switches.

End of procedure

Procedure: Starting HA Proxy on Windows manually

Start of procedure

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

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

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

Table 8 on [page 122](#) lists HA Proxy executable names for supported switches.

End of procedure

T-Server

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

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

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

The command-line parameters common to Framework server components are described on [page 117](#).

Procedure: Starting T-Server on UNIX manually

Start of procedure

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

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

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

Table 8 on [page 122](#) lists T-Server executable names for supported switches.

End of procedure

Procedure: Starting T-Server on Windows manually

Start of procedure

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

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

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

Table 8 on [page 122](#) lists T-Server executable names for supported switches.

End of procedure

Verifying Successful Startup

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

If you used the Management Layer to start either T-Server or HA Proxy, check whether Solution Control Interface displays `Started` or `Service Unavailable`

status for the corresponding application. Refer to the “Troubleshooting” section of the *Framework 7.6 Management Layer User’s Guide* if the startup command does not result in either `Started` or `Service Unavailable` status for some period of time.

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

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

Stopping Manually

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

Procedure: Stopping T-Server on UNIX manually

Start of procedure

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

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

End of procedure

Procedure: Stopping T-Server on Windows manually

Start of procedure

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

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

End of procedure

Starting and Stopping with Windows Services Manager

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

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

where the command-line parameters common to Framework server components are described on [page 117](#) and

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

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

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

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

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

Next Steps

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



Part

2

Part Two: Reference Information

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, “Ericsson MD110 Switch-Specific Configuration,” on [page 131](#), describes compatibility and configuration information specific to this T-Server, including how to set the DN properties and recommendations for the switch configuration.
- Chapter 7, “Supported Functionality in T-Server for Ericsson MD110,” on [page 145](#), describes which features are supported by this T-Server, including T-Library functionality, use of the Extensions attribute, and error messages.
- Chapter 8, “Common Configuration Options,” on [page 195](#), describes log configuration options common to all Genesys server applications.
- Chapter 9, “T-Server Common Configuration Options,” on [page 217](#), describes configuration options that are common to all T-Server types, including options for multi-site configuration.
- Chapter 10, “Configuration Options in T-Server for Ericsson MD110,” on [page 241](#), describes configuration options specific to this T-Server, including the link-related options—those which address the interface between T-Server and the switch.
- Chapter 11, “High Availability (HA),” on [page 277](#), provides switch-specific information for high availability.
- Chapter 12, “Ericsson MD110 Call Flows,” on [page 279](#) provides details of call models that are specific to the Ericsson MD110/T-Server environment.

New in T-Server for Ericsson MD110

The following new features are now available in the initial 7.6 release of T-Server for Ericsson MD110:

- **Enhancement to Smart OtherDN Handling:** In release 7.6, new configuration option `clid-withheld-name` has been introduced. See “clid-withheld-name” on [page 264](#).
- **Enhancement to No-Answer Supervision:** In release 7.6, configuration option `nas-indication` has been introduced to enable T-Server to indicate in `EventReleased` whether an overflow has occurred as a result of No-Answer Supervision. See “nas-indication” on [page 259](#).
- **Enhancement to Keep-Alive:** In release 7.6, the Keep-Alive feature has been enhanced to include a configurable loss-rate counter. See “Keep-Alive Feature” on [page 163](#).
- **Enhancement to ACW:** In release 7.6, a wrap-up threshold can now be applied to ACW processing. See “wrap-up-threshold” on [page 254](#).
- **Enhancement to Maintain Position in Queue feature:** In release 7.6, a configurable delay has been introduced to control how quickly calls are returned to their queue position. See “queue-return-dly” on [page 265](#).

Note: Configuration option changes that apply to your T-Server are described in “Changes from 7.5 to 7.6” on [page 274](#).



Chapter

6

Ericsson MD110 Switch-Specific Configuration

This chapter presents switch-specific reference information for configuring T-Server for the Ericsson MD110 switch and includes these sections:

- [Known Limitations, page 131](#)
- [Setting the DN Properties, page 134](#)
- [Support of Switch/CTI Environments, page 136](#)
- [Partitioned Application Link Configuration, page 137](#)
- [Configuring Devices in Configuration Layer, page 137](#)
- [Routing Configuration, page 138](#)
- [Device-Specific Behavior, page 141](#)
- [Free Seating, page 141](#)
- [Support for PBX Boss/Secretary Feature, page 142](#)
- [Support for Maintain Position in Queue, page 142](#)

Known Limitations

1. The following is a statement from Ericsson:

“In order to maintain the same Call ID during a whole Call Deflect scenario, every involved device, including logical devices such as ACD groups, must be monitored. For example, if a PBX group is involved in the Call Deflect scenario, the Call ID may change because PBX groups cannot be monitored.”

In Genesys, this will cause problems in some emulated routing scenarios, because Hunt Groups cannot be monitored. Therefore, T-Server may not be able to maintain the Call ID, and consequently attached data will be lost.

2. Call Parking is not supported by the switch. The switch uses hold functionality instead.
3. It is not possible to camp a call when the destination is off hook, dialing or alerting.
4. The switch does not generate a `Transferred` event from the switch for transfers (including blind) invoked by an unmonitored DN. This prevents T-Server from distributing `EventPartyChanged` and, therefore, from maintaining connection IDs.
5. The Application Link does not report conference calls when queried after the switch link disconnects and then reconnects.
6. The switch releases the second party in a call after the first party performs a single-step transfer after trunk optimization over an ISDN trunk.
7. The switch does not provide CTI events for `Intrude`, so T-Server cannot support this feature.
8. When monitoring only ODNs in the Genesys environment, T-Server is unable to determine login state on T-Server startup. It may be necessary to manually lock or unlock the phone set to synchronize.
9. Stuck-calls management functions (see “Call-Cleanup Section” on [page 237](#)) rely on T-Server being able to query the switch. T-Server for Ericsson MD110 is not able to query the switch in the following circumstances:
 - A call staying on a queue or native Routing Point cannot be queried.
 - A call in which unmonitored (external) devices are involved cannot be queried.
 - Any combination of the above limitations.
10. Emulated agents are not supported where a device has to be unlocked on the switch to make calls, and therefore a login/logout event is sent via the Application Link; for example, ODNs with associated ADNs, or ODN-only configurations. Emulated agents can be used on VoIP devices after the login button is pressed on the phone set, because there are no login/logout events via the Application Link.
11. The Genesys call model requires that if a call is forwarded to a device, then the first event on that device should have call state `Forwarded`. The Ericsson MD110 does not always report the call as `Forwarded`, so T-Server cannot reliably report this. In such cases T-Server reports the forward state as `OK`.
12. Outbound Contact (OCS) does not have separate `Ready/NotReady` states for DNs on a place. It has `OffHook/OnHook` states for DNs only. After receiving `EventAgentReady`, OCS considers that the agent is ready to take an outbound call. OCS does not know which DN is responsible for receiving an outbound call. If the position DN works with outbound calls, use only this DN to change agent status (`Ready/NotReady`).

13. Application Link does not report conference completion correctly when all parties are internal and the conference controller is an unmonitored device.
14. Both Application Link and the switch report a nonexistent call arriving at a device that has been made out-of-service by pressing the program key. Physically the phone rings once, but no call is present. However, once the DN is made in-service by pressing the program key, the call is present on the DN and can be answered.
15. At T-Server start/restart, T-Server queries the switch for device status. If the switch reports the device status as logged in, T-Server by default uses the ODN value for that device as the agent `login ID` attribute. T-Server then distributes this ODN value to its clients.

On receiving a `RequestAgentLogin` from a client where the device is logged out, T-Server passes the ODN value to the switch in the login request, but communicates with the client using the agent `login ID` value from the client's `RequestAgentLogin`.

Therefore, any T-Server start/restart compromises the monitoring and reporting of agent status unless the agent login IDs in the Genesys environment are set up as identical to the ODN values.
16. T-Server should constantly monitor all extension numbers that might take part in any call activity visible to T-Server. If an agent telephone has one or more ADNs configured, then T-Server must monitor both ODNs and all ADNs. To achieve this, you must configure all extension numbers in the Framework Configuration Wizard or Configuration Manager.
17. It is not possible for primary and backup T-Servers to synchronize call information when the backup T-Server starts after a call was created. In this case, T-Server cannot maintain call control after switchover, and therefore ACW behavior cannot be predicted. Setting configuration option `ha-sync-dly-lnk-conn` to value `true` prevents call synchronization problems.

Setting the DN Properties

[Table 9](#) indicates how to set the DN properties for the Ericsson MD110 T-Server/switch combination.

Table 9: Setting the DN Properties

Switch Device Type	Genesys Device Type	Parameter Name	Parameter Value	Register
VoIP device Note: Emulated agents only	Extension	Switch-specific type	1	True
Free Seating DN	Extension	Switch-specific type	1	True
ODN which uses ACD authorization code to unlock Note: Real agents only	Extension	Switch-specific type	1 See “acd-login” on page 244 .	True
ODN which uses non-ACD authorization code to unlock Note: Real agents only	Extension	Switch-specific type	1 See “acd-login” on page 244 .	True
ODN which uses no authorization code to unlock ^a Note: Emulated agents only	Extension	Switch-specific type	1	True

Table 9: Setting the DN Properties (Continued)

Switch Device Type	Genesys Device Type	Parameter Name	Parameter Value	Register
<p>ODN with or without associated ADNs in switch configuration for Genesys Smart Monitoring.</p> <p>Note: ADNs behave in the same way as the ODN they are associated with—you do not need to define them as switch-specific type 10. Other devices requiring an Application Link license that are defined with switch-specific type 10 and do not support login are never registered against Application Link.</p> <p>Note: Real agents only</p>	Extension	Switch-specific type	<p>10</p> <p>T-Server will dynamically start/stop a device monitor (Smart Monitoring) on the Application Link only after a login/logout CTI request.</p> <p>Note: You cannot mix switch-specific types 1 and 9 for Smart Monitoring. See “acd-login” on page 244.</p> <p>When the agent is not logged in via CTI, T-Server does not report any activity on the device.</p> <p>See “Known Limitations” on page 131.</p>	True
<p>ADN</p> <p>Note: Real agents only</p>	ACD Position	Association	Valid ACD group number, followed by valid ODN number on the phone where ADN belongs, using a comma separator.	True
ACD	ACD Queue			True
CTI group not used for routing	ACD Queue			True

Table 9: Setting the DN Properties (Continued)

Switch Device Type	Genesys Device Type	Parameter Name	Parameter Value	Register
CTI group used for routing	Routing Point/ external Routing Point			True
Hunt Group, the members of which are used for routing	Routing Point/ external Routing Point			False
Hunt Group member (digital device) with ITYPE=0	Extension	Association	Valid virtual device number or Hunt Group number.	True
		Switch-specific type	2	True
Analog Port/CAS Extension	Extension Voice Treatment Port	Switch-specific type	8 Provides full support for caller hang-up scenarios when using VTO or CPD server	True
Emulated predictive dialing device. <ul style="list-style-type: none"> Digital extension, with or without handset Digital extension, without handset, with ITYPE=0 (recommended) 	Extension	Switch-specific type	4	True

- a. Any device of type `extension` that does not need an authorization code from the switch for unlocking can be used with Genesys emulated agents.

Support of Switch/CTI Environments

T-Server support of customer switch/CTI environments is dependent on several factors, including:

- Number of DNs.

- Number of concurrent agents.
- Number of concurrent connections.
- Number of concurrent calls.
- Number of calls or messages per second.

Information about T-Server connection limits is provided in the [Genesys 7 Supported Operating Systems and Databases](#) document. Connection limits are determined by the platforms on which T-Servers run—T-Server itself does not set these limits.

The remaining factors are not limited by T-Servers, but could be limited by the switch and/or CTI interface. Unless specific exceptions are documented, T-Server can meet the performance capability of the switches it supports in each of these areas. The T-Server host environment and the network environment influences should also be taken into account.

Partitioned Application Link Configuration

MD110 prohibits multiple Application Links from monitoring the same devices concurrently. This prevents a “hot swap” between Application Links if a problem occurs. However, once an Application Link has been stopped, other Application Links can register for those devices.

It is also possible to partition the switch at the Application Link level, using the Application Link Security program provided by Ericsson, to lock the Application Links onto their own specific pool of devices available for registration.

See the switch documentation supplied by your vendor for more details.

Configuring Devices in Configuration Layer

This section describes how to configure devices in the Configuration Layer.

Procedure: Configuring Digital Devices

Summary

A digital handset will always have one ODN, but it could have no ADNs or many ADNs. Each ADN is a member of a queue, thus making an agent potentially available to many queues.

To tie all these devices together in the Configuration Layer, use the `Association` field for the ACD Positions (ADNs).

Start of procedure

1. For the following physical configuration:
 - ODN 5514
 - ADN 59500 member of Queue 5650
 - ADN 59700 member of Queue 5670in the Configuration Layer, define these objects as:
 - 5514—Extension
 - 59500, 59700—ACD Positions
 - 5650, 5670—ACD Queues
2. In the Association field for the ACD Positions, enter the Queue followed by the ODN (comma-separated) that this device belongs to:
 - ADN 59500 would have an association of 5650, 5514
 - ADN 59700 would have an association of 5670, 5514

End of procedure

VoIP Devices

VoIP devices must be configured as extension objects in the Genesys Configuration Layer and can be used only with emulated agents.

VoIP login/logout on the phoneset is not a CTI (Genesys) login—the Application Link does not issue login/logout events for VoIP.

VoIP login/logout corresponds only to an in-service/out-of-service event. In order to have agent functionality, VoIP devices must be in service, and then login must be performed using Genesys emulated agents.

Configuration option `acd-login` is not applicable for VoIP.

Routing Configuration

The current versions of the switch firmware and Application Link do not support software call routing as defined by the computer-supported telephony applications (CSTA) standard. The T-Server solution is to implement call routing by either:

- Emulating routing by using Hunt Groups.
- Using a CTI group as a single Routing Point.

Emulating Routing by Using Hunt Groups

Calls that arrive at a Hunt Group can never queue—they are immediately delivered to one of the Hunt Group members. If no members are available, the call is reported as busy. When you use Hunt Groups as Routing Points,

T-Server emulates routing events on the Hunt Group and hides all events for the Hunt Group members, even though the call is actually alerting on the members.

Procedure: **Using a Hunt Group as Genesys Routing Point/ external Routing Point**

Purpose: To enable T-Server to emulate routing events on the Hunt Group and hide all events for the Hunt Group members.

Start of procedure

1. Create Hunt Group members in the switch as device type `ITYPE = 0`. This device is a virtual ODN (that is, there is no physical device) and does not have associated ADNs.
2. Configure one of the following routing mechanisms for delivering calls from the Hunt Group members to the specified routing destination:
 - Redirect from the Hunt Group member.
 - Answer and transfer the call from the Hunt Group member.

The method that is used is determined by one of the following:

- Configuration option `ext-routing-min-digits` (see [page 245](#))
- Configuration option `supervised-route-timeout` (see [page 260](#))
- Extension `SUPERVISED_ROUTE` (see [page 181](#))

End of procedure

Notes: Supervised routing takes precedence where it is configured. Extension `GCTI_SUPERVISED_ROUTE` takes precedence over option `ext-routing-min-digits`.

Any CTI applications that try to register with T-Server for the Hunt Group members receive `EventError`.

Example—Configuring Hunt Groups as Routing Points

Table 10 shows an example configuration using Hunt Groups as Routing Points.

Table 10: Example Configuration

Switch Device Type/Value	Configuration Layer Object Type/Value	Switch-Specific Type	Association Field Value
Hunt Group/2500	Routing Point/2500	Not applicable	Not applicable
Hunt Group member/4000	Extension/4000	2	2500 (Routing Point)
Hunt Group member/4001	Extension/4001	2	2500 (Routing Point)

Using CTI Group as a Single Routing Point/ External Routing Point

While T-Server is not monitoring a CTI group, the switch is responsible for the distribution of calls arriving at this device. However, when T-Server starts to monitor such a device, it is Genesys' responsibility to distribute calls.

Note: A CTI group device cannot be used in conjunction with supervised routing and does not support route type reject.

Supervised Routing

When you use emulated routing, T-Server can determine whether calls are answered at the routing destination. If calls are not answered within the specified timeout, you can configure T-Server to recall the calls to the Routing Point and initiate rerouting. The supervision of the call travels with the call, so if a supervised call is routed to from one Routing Point to another, T-Server cancels the supervision on the first Routing Point. See configuration option supervised-route-timeout and the Extension attribute SUPERVISED_ROUTE for more details.

Note: When a call is routed using route type reject on an emulated Routing Point (using either supervised or nonsupervised routing) extension SUPERVISED_ROUTE needs to be sent with value 1 in order for T-Server to drop the call.

Device-Specific Behavior

This section describes device-specific behavior.

Initiating a Consult Call from an ODN or ADN

The consult call is always initiated from the ODN, regardless of the device type on which the primary call resides.

Completing Two-Step Conferences from ODNs or ADNs

When the primary call in a conference is established on the ADN, and the agent wishes to complete the conference (the consult call of conference is on the ODN), the agent must alternate the call before completing the conference.

When the primary call in a conference is established on the ODN, and the agent wishes to complete the conference, no actions are required before completing the conference.

The one exception to this rule is when the primary call is made by the ADN. In this situation, it is possible to complete the conference without alternating the call. However, the call that remains after conference completion is the consult call that was on the ODN—the primary call on the ADN is released.

One-Step Transfer from Specific Devices

In general, the following transfer mechanisms are supported by the switch:

- On ODNs the switch supports only `Initiate/Complete`, which corresponds to Genesys `TMuteTransfer`.
- On ADNs and analog devices, the switch supports only `Deflect`, which corresponds to Genesys `TSingleStepTransfer`.

Free Seating

MD110 supports agent functionality called Free Seating. The Free Seating feature allows an agent to log in to a device (analog, non-ACD digital DN). The existing directory number is placed out of service, and is replaced with the Free Seating directory number, which is now placed in service and logged in. When this (Free Seating) agent logs out, the Free Seating device is placed out of service and the original DN is placed back in service.

It is not possible to use a native CTI login request with Free Seating DNs.

Recommendations and Limitations

- Genesys recommends using digital DNs for Free Seating numbers. Genesys cannot guarantee correct single-step transfer behavior if analog devices are logged in to Free Seating numbers.

- Even if one device is logged in to a Free Seating number, it is still possible to log in to the same Free Seating number from a different device by entering the same security code. The Free Seating number is automatically logged out from the previous device and logged in to the most recent one.
- When a device is logged in to a Free Seating number, the agent is logged in to this number and the AgentID is the same as the Free Seating number.
- Forwarding:
 - You can set unconditional forwarding on an out-of-service Free Seating number via CTI. When the device is logged in to a Free Seating number, forwarding on an in-service Free Seating number remains in place.
 - You can set unconditional forwarding on an in-service Free Seating number. When the device is logged out from the Free Seating number, forwarding on the out-of-service Free Seating number remains in place.
 - You cannot set switch- or user-defined ForwardOnBusy or ForwardOnNoAnswer on Free Seating numbers.
 - You cannot set forwarding on a device that has been placed out-of-service by a Free Seating login.
 - If forwarding is set on an in-service device number (before Free Seating login), when a Free Seating number logs in to the device and it becomes out of service, forwarding remains in place.

Support for PBX Boss/Secretary Feature

Boss/Secretary mode on the MD110, involves the secretary having a preconfigured key allocated to alert the secretary when the boss receives a call.

The secretary hears ringing on the phone, but there is no CTI alerting event from the switch, so calls have to be answered manually.

After the call has been answered it can be manipulated via CTI as normal.

Support for Maintain Position in Queue

This feature ensures that a call's position in the queue is not lost while that call is placed on another device (IVR port) and then returned to the queue after treatment. This feature is applicable only for switch devices of type CTI Group that are configured as Routing Points in Genesys.

Configuration Options

The following configuration options are available:

enable-retain-in-queue

Default Value: false

Valid Values: false, true

Changes Take Effect: On arrival of a new call at the Routing Point

Specifies whether the Maintain Position in Queue feature is enabled (value = `true`) or disabled (value = `false`).

call-retain-in-queue

Default Value: `disabled`

Valid Values: `keep`, `divert`, `disabled`

Changes Take Effect: On arrival of a new call at the Routing Point

Values `keep` and `divert` instruct T-Server to apply the Maintain Position in Queue feature when the destination for routing is configured as a device with treatment (Voice Treatment Port) capabilities.

<code>keep</code>	T-Server follows PBX reporting and maintains the call queued on the Routing Point when the call is placed on a Treatment device.
<code>divert</code>	T-Server issue <code>EventDiverted</code> together with <code>EventRouteUsed</code> when the call is routed with the Maintain Position in Queue feature enabled.
<code>disabled</code>	The feature is switched off.

Note: For integration of the Maintain Position in Queue feature into existing URS/VTO environments, Genesys recommends configuring `call-retain-in-queue` with value `divert`.

The option can be configured in the main `TServer` section and applied for all nonemulated Routing Points controlled by T-Server. It can also be specified in the Routing Point `Annex` tab, in which case it overrides the setting for the global option. The value for the configuration option is also overridden by the `CallRetainInQueue` extension provided in `RequestRouteCall`.

queue-return-dly

Default Value: `1000`

Valid Values: `0-5000`

Changes Take Place: Immediately

Defines how long (in milliseconds) T-Server waits for a call to be returned to the queue where position in queue is being maintained. If the PBX does not report delivery within this interval (for example, because Call ID was changed), T-Server initiates call cleanup processing. See “retain-call-tout” on [page 262](#).

Returning a Call to CTI Ggroup Table 11 shows valid methods to return a call back to the CTI group after treatment.

Table 11: Methods of Returning a Call to CTI After Treatment

Option	Return Back (Established)	Drop Call	Return Back (Ringing)
keep	RequestReleaseCall Release on hardware level	RequestClearCall	Not Supported
divert	RequestSingleStepTransfer RequestMuteTransfer Release on hardware level	RequestReleaseCall RequestClearCall	RequestRedirectCall

Note: If a call is redirected or transferred from the treatment port to a device other than the originating CTI Group, then position in queue is released.



Chapter

7

Supported Functionality in T-Server for Ericsson MD110

This chapter describes the telephony functionality that T-Server for Ericsson MD110 supports and includes these sections:

- [Business-Call Handling, page 145](#)
- [Support for Emulated Agents, page 147](#)
- [Support for Emulated Predictive Dialing, page 155](#)
- [Support for No-Answer Supervision, page 157](#)
- [Smart OtherDN Handling, page 160](#)
- [Keep-Alive Feature, page 163](#)
- [VoIP Absence Code Handling, page 168](#)
- [T-Library Functionality, page 170](#)
- [Support for Agent Work Modes, page 180](#)
- [Use of the Extensions Attribute, page 180](#)
- [User Data Keys, page 186](#)
- [Error Messages, page 186](#)

Business-Call Handling

This section describes how T-Server handles different types of call

T-Server Call Classification

T-Server automatically assigns every call to one of three categories—*Business*, *Work-Related*, or *Private*. Based on this assignment, T-Server applies the appropriate business-call handling after the call is released.

Business Calls

T-Server automatically categorizes as a *business call* any call distributed to an agent either from a Queue or from a Routing Point. Use the following configuration options to define what additional calls to or from an agent are classified as business calls:

- `inbound-bsns-calls`
- `outbound-bsns-calls`
- `inherit-bsns-type`
- `internal-bsns-calls`
- `unknown-bsns-calls`

inbound-bsns-calls

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether all established inbound calls should be considered business calls.

outbound-bsns-calls

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether all established outbound calls should be considered business calls.

inherit-bsns-type

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Determines whether a consult call that is made from a business primary call should inherit the `business call` attribute.

internal-bsns-calls

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Determines whether T-Server considers internal calls made from or to any agent as business calls.

unknown-bsns-calls

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Determines whether T-Server considers calls of unknown call type made from or to any agent as business calls.

Work-Related Calls

T-Server categorizes as a *work-related* call any non-business call that an agent makes while in ACW. T-Server does not apply any automatic business-call handling after a work-related call.

Because emulated agents can make or receive a direct work-related call while in wrap-up time, T-Server pauses the emulated wrap-up timer for the duration of such a call.

If an agent receives a direct work-related call during legal-guard time, T-Server cancels the legal-guard timer and reapplies it at the end of the work-related call.

Private Calls

T-Server categorizes as a *private call* any call that does not fall into the business or work-related categories. T-Server does not apply any automatic business-call handling after a private call. If emulated agents receive a direct private call while in wrap-up or legal-guard time, the emulated wrap-up or legal-guard timer is not interrupted.

Support for Emulated Agents

T-Server provides a fully functional emulated-agent model that you can use either in addition to agent features available on the PBX or in place of them where they are not available on the PBX.

When this feature is used, T-Server emulates the following functionality:

- Login and logout
- Agent set ready
- Agent set not ready (using various work modes)
- Automatic after call work (ACW)
- After call work in idle
- Automatic legal-guard time to provide a minimum break between business related calls

Emulated Agent Login/Logout

You can configure T-Server to perform emulated login either always, never, or on a per-request basis. Use the following T-Server configuration options to configure emulated agent login:

- `emulate-login`
- `emulated-login-state`
- `agent-strict-id`

emulate-login

Default Value: `on-RP`

Valid Values: `true`, `false`, `on-RP`

Changes Take Effect: Immediately

Specifies whether T-Server performs emulated agent login when the login device is configured in the Configuration Layer as a device of type `extension`.

<code>true</code>	T-Server performs an emulated login.
<code>false</code>	T-Server passes a login request to the PBX.
<code>on-RP</code>	T-Server checks the Agent Group associated with the login request. If the Agent Group is a standard Routing Point the emulated login request succeeds. This value can only be set at the global level, and is available for backwards compatibility.

This value can be set in a number of places, and T-Server processes it in the order of precedence shown below, highest first. If the value is not present at the higher level, T-Server checks the next highest level, and so on.

1. In `RequestAgentLogin`, using attribute extension `EmulateLogin`.
2. In the Agent ID object on the Annex tab.
3. In the login device object on the Annex tab.
4. In the device representing an Agent Group object, on the Annex tab.
5. In the T-Server Application object, in the Tserver section.
6. Using an Agent Group corresponding to an object which is configured in the Configuration Layer as a device of type `Routing Point`.

emulated-login-state

Default Value: `not-ready`

Valid Values: `ready`, `not-ready`

Changes Take Effect: Immediately

When T-Server performs an emulated agent login and the client specifies an agent work mode other than `ManualIn` or `AutoIn`, T-Server uses this option to determine which event to distribute.

<code>not-ready</code>	T-Server distributes <code>EventAgentNotReady</code> after <code>EventAgentLogin</code> .
<code>ready</code>	T-Server distributes <code>EventAgentReady</code> after <code>EventAgentLogin</code> .

This option can be set in a number of places, and T-Server processes it in the order of precedence shown below, highest first. If the value is not present at the higher level, T-Server checks the next level, and so on.

1. In the Agent ID object on the Annex tab.
2. In the agent login device on the Annex tab.
3. In the login device representing an Agent Group during login, on the Annex tab.
4. In the T-Server Application object in the Tserver section.

agent-strict-id

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether, for emulated agents, T-Server allows any AgentID to be used during login (value `false`), or only those configured in Configuration Layer (value `true`).

Emulated Agent Ready/NotReady

Emulated agents can perform an emulated Ready or NotReady request regardless of whether they are on a call, subject to the rules governing work modes.

T-Server also reports any change in agent mode requested by the agent while remaining in a NotReady state (*self-transition*).

Note: Note that the *Genesys 7 Events and Models Reference Manual* and the *Voice Platform SDK 7.6 .NET (or Java) API Reference* define which agent state/agent mode transitions are permissible.

Emulated After-Call Work (ACW)

T-Server can apply emulated wrap-up (ACW) for agents after a business call is released, unless the agent is still involved in another business call (see “Business Calls” on [page 146](#)).

Timed and Untimed ACW

T-Server applies emulated ACW for an agent after any business call is released from an established state. T-Server automatically returns the agent to the Ready state at the end of a *timed* ACW period. The agent must return to the Ready state manually when the ACW period is *untimed*.

Events and Extensions

T-Server indicates the expected amount of ACW for an agent in `EventEstablished` using the extension `WrapUpTime`. It is not indicated in `EventRinging` because the value may change between call ringing and call

answer. Untimed ACW is indicated by the string value `untimed`, otherwise the value indicates the expected ACW period in seconds.

T-Server reports ACW using `EventAgentNotReady` with `workmode = 3` (`AgentAfterCallWork`) and indicates the amount of ACW it will apply using the extension `WrapUpTime`.

T-Server sends the `EventNotReady` (ACW) before the `EventReleased` at the end of the business call.

Emulated ACW Period

The amount of emulated ACW that T-Server applies (when required) after a business call is determined by the value in configuration option `wrap-up-time`.

Configuration option `untimed-wrap-up-value` determines which specific integer value of `wrap-up-time` indicates *untimed* ACW. To specify untimed ACW in request extensions or user data, you should use the string `untimed` instead. All positive integer values are treated as indicating timed ACW (in seconds). For backwards compatibility, the default value of `untimed-wrap-up-value` is `1000`.

Note: Changing the value of untimed ACW should be done with care, because it may affect the interpretation of all integer values of the option `wrap-up-time` in Configuration Manager. If lowered, it may change timed ACW to untimed, or disable ACW altogether. If raised it may change untimed or disabled ACW to timed ACW. The use of the new option (string) value `untimed` is encouraged where possible to minimize the impact of any future changes to the value of option `untimed-wrap-up-value`.

wrap-up-time

Default Value: `0`

Valid Value: Any positive integer, `untimed`

Changes Take Place: Immediately

Specifies the amount of wrap-up time (ACW) allocated to emulated agents at the end of a business call.

`0`

ACW is disabled

Exception: When set in the Annex tab of the Agent ID object, value `0` (zero) means T-Server will process from Step 4 in the processing order of precedence below.

Value greater than `0` but less than `untimed-wrap-up-value`

The number of seconds of timed ACW, after which T-Server returns the agent automatically to the Ready state.

Value equal to <code>untimed-wrap-up-value</code>	ACW is untimed and the agent must manually return to the Ready state.
Value greater than <code>untimed-wrap-up-value</code>	Disables ACW.
<code>untimed</code>	ACW is untimed and the agent must manually return to the Ready state.

This option can be set in a number of places, and T-Server processes it in the order of precedence shown below, highest first. If the value is not present at the higher level, T-Server checks the next level, and so on.

1. In `RequestAgentPendingACW`, in attribute extension `WrapUpTime` (applies to this agent only).
2. In `RequestACWInIdle`, in attribute extension `WrapUpTime` (applies to this agent only).
3. In the call, in user data `WrapUpTime` (limited to ISCC scenarios).
4. In a configuration object of type `ACD Queue` or `Routing Point`, on the Annex tab.
5. In `RequestAgentLogin`, in attribute extension `WrapUpTime` (applies to this agent only).
6. In the `Agent ID` object, in the Annex tab.
7. In the login device object, on the Annex tab.
8. Using an Agent Group corresponding to an object configured in the Configuration Layer as a device of type `ACD Queue`.
9. In the `T-Server Application` object.

untimed-wrap-up-value

Default Value: 1000

Valid Value: Any nonzero positive integer

Changes Take Effect: Immediately

Specifies the threshold at which the timing of ACW stops and manual intervention is required (*untimed ACW*).

wrap-up-threshold

Default Value: 0

Valid Values: Any positive integer

Changes Take Effect: Immediately

Specifies the minimum period (in seconds) that a business call must last before emulated ACW is applied at the end of the call.

Pending ACW

An agent can request emulated ACW, or override the period of (emulated) ACW to be applied to themselves, while on an established call. T-Server will apply the emulated ACW when the call is released. The agent sends `RequestAgentReady` with `workmode = 3` to request pending ACW while on an established call. The extension `WrapUpTime` indicates the amount of ACW that T-Server will apply, using the following parameters and rules:

- Extension missing—request is rejected
- Value = 0—ACW is disabled
- Value greater than 0—period of timed ACW in seconds
- Value = `untimed`—untimed ACW

If the request is successful, T-Server sends `EventAgentReady` with `workmode = 3` (ACW). T-Server will also indicate that the agent is in a pending ACW state by adding the extension `ReasonCode` with the new value `PendingACW`. It will also indicate the period of ACW to be applied using the `WrapUpTime` extension.

An agent may alter the period of pending ACW by sending a new `RequestAgentReady` with `workmode = 3`, using a different value for the `WrapUpTime` extension. If the request is successful, T-Server sends another `EventAgentReady` event, indicating the new value in the `WrapUpTime` extension.

Note: To enable this feature, the `WrapUpTime` extension must be enabled on the agent desktop.

ACW In Idle

An agent can activate wrap-up time on request when idle, by issuing a `RequestAgentNotReady` with `workmode = 3` (`AgentAfterCallWork`) to request emulated ACW while idle.

You can configure this feature in T-Server using the following options:

- `timed-acw-in-idle`
- `acw-in-idle-force-ready`

timed-acw-in-idle

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether T-Server applies the automatic wrap-up timer (using the wrap-up-time parameter) when an agent sends `RequestAgentNotReady`. With value `false`, T-Server does not automatically end manual wrap-up—the agent must return manually from ACW.

acw-in-idle-force-ready

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether, after timed automatic wrap-up (when you have set option `timed-acw-in-idle` to `true`), T-Server forces the agent to the Ready state. With value `false`, T-Server returns the agent to the state he or she was in prior to wrap-up.

Extending ACW

An agent can request an extension to the amount of emulated ACW for a call while in emulated ACW or in the legal-guard state.

The agent requests an extension to ACW by sending `RequestAgentNotReady` with `workmode = 3` (`AgentAfterCallWork`). T-Server determines the period of the extended ACW from the extension `WrapUpTime`, as follows:

- Value = 0—No change to ACW period, but T-Server reports how much ACW time remains.
- Value greater than 0—T-Server adds the given number of seconds to the timed ACW period. Untimed ACW remains unaffected.
- Value = `untimed`—T-Server applies untimed ACW.

T-Server sends `EventAgentNotReady` with `workmode = 3` (`AgentAfterCallWork`), reporting the newly extended amount of ACW using the extension `WrapUpTime`. If the agent was in the emulated legal-guard state, T-Server places the agent back into emulated ACW state.

The agent may extend the period of ACW as many times as desired. At the end of the extended timed ACW period, T-Server applies legal guard if any is configured. No legal guard is applied if the emulated ACW was untimed.

Calls While in Emulated ACW

T-Server's handling of an agent making or receiving a call while in emulated ACW is governed by the configuration option `backwds-compat-acw-behavior`.

backwds-compat-acw-behavior

Default Value: `true`

Valid Value: `true`, `false`

Changes Take Effect: Immediately

Specifies whether pre-7.5 behavior after-call work is enabled (value = `true`) or disabled (value = `false`), for backward compatibility.

With value `true`, if an agent receives or makes a business call while in emulated ACW, T-Server does the following:

1. Stops the ACW timer.

2. Forces the agent to the Ready state.
3. Restarts ACW (and the legal-guard timer) after the new business call is released.

If an agent makes or receives a work-related call while in ACW, T-Server does the following:

1. Suspends the ACW, but leaves the agent in the ACW state.
2. Resumes the ACW timer once the work-related call is released.

Note: A work-related call is one made by an agent while in ACW, or a consult call where the main call is either a business call or a work-related call.

After the ACW and any configured legal-guard time have been completed, the agent is forced to the Ready state.

If an agent makes or receives a private call during ACW, no action is taken and the ACW timer keeps running.

With value `false`, pre-7.5 behavior is used. In this case, T-Server forces the agent to the Ready state after the after-call work and legal-guard timer have been applied.

Emulated Legal-Guard Time

T-Server applies emulated legal-guard time for agents before they are about to be automatically set ready after any period of timed ACW or after the last business call is released where there is no ACW to be applied. It is a regulatory requirement in many countries to guarantee that agents have a break of a few seconds before the next call can arrive. No legal-guard time is applied if the ACW period was not timed or if the agent is not being placed into the Ready state.

T-Server reports legal guard using `EventAgentNotReady` with `workmode = 2` (`LegalGuard`). If an agent requests to be logged out during emulated legal-guard time, T-Server immediately logs the agent out.

If the agent requests to go to a `Not Ready` or `Ready` state during legal-guard time, T-Server terminates legal guard and transitions the agent to the requested state. If the agent requests to return to the ACW state, T-Server re-applies legal guard at the end of ACW, provided that the agent still requires it according to the above criteria.

The period of legal guard is determined by the following option:

legal-guard-time

Default Value: 0

Valid Value: Any integer from 0-30

Changes Take Effect: Immediately

Specifies a legal-guard time (in seconds) for emulated agents to postpone the transition to the Ready state after a business call. T-Server always considers a routed call a business call.

T-Server reports the legal-guard time using `EventAgentNotReady` with `workmode = 2 (LegalGuard)`.

If an agent requests to be logged out during emulated legal guard time, T-Server immediately logs the agent out.

If the agent requests to go to a NotReady or Ready state during legal-guard time, T-Server terminates legal-guard time and transitions the agent to the requested state. If the agent requests to return to the ACW state, then T-Server reapplies legal-guard time at the end of ACW provided that the agent still requires it according to the above criteria.

HA Synchronization

On startup and link re-establishment, the Hot Standby backup T-Server requests the primary T-Server to send details of all agents. The primary T-Server replies with all the information required for switchover, including all emulated and switch-based data.

From this point on, the primary T-Server also sends a similar synchronization message whenever an emulated agent's state changes.

This means that a higher level of synchronization between the two T-Servers is maintained at all times.

Support for Emulated Predictive Dialing

This feature enables Genesys Outbound Contact Server (OCS) to initiate calls without the use of the Call Progress Detection (CPD) Server and Dialogic hardware.

Note: This feature is not related to the predictive dialing algorithm OCS uses to determine when to make the next call. This feature only concerns the outbound-call mechanism. You cannot use Emulated Predictive Dialing with Dialogic hardware.

To enable the Predictive Dialing feature in T-Server, you must configure (in the Configuration Layer) a number of devices corresponding to the number of calls that can be made simultaneously. These devices are available as a pool for T-Server to use for predictive dialing. They are not associated with any specific dialing device (Queue or Routing Point). They are configured in the Genesys Configuration Layer with switch-specific type 4.

Because of a small discrepancy in the way the availability of dialing devices is calculated in T-Server and in OCS, Genesys recommends configuring extra

dialing devices. For example, if you plan to use five dialing devices in a campaign, configure six dialing devices in T-Server.

Limiting Distribution Time

Many countries forbid, by law, the queuing of more calls than there are available agents. The law in these countries states that such calls must be immediately dropped. T-Server does not handle this requirement for the duration of call distribution. The distribution mechanism must handle it.

If you use Universal Routing Server (URS) to distribute outbound calls to agents, set the `Timeout` option in the Strategy Target-Selection object to an appropriate value: for example, 1 second or 2 seconds.

Note: Your routing strategy is likely to fail if you set the value of `Timeout` to 0 (zero).

Once outbound calls have been successfully distributed to an agent, use the value of configuration option `prd-dist-call-ans-time` to limit the time that a call rings on an agent desktop without being answered.

If T-Server has no dialing devices available at the time of a `TMakePredictiveCall` request, it attempts to queue the request for the duration specified in option `max-pred-req-dly`. If a dialing device becomes available, T-Server makes the call. If not, T-Server rejects the request.

Call Progress Detection

T-Server's Emulated Predictive Dialing feature does not support call progress detection (CPD) to the same extent as Dialogic hardware. CPD is limited to normal switch signaling. In-band CPD is not supported. The following dialing results are supported:

- Answer
- No Answer
- Busy
- Dropped
- Wrong number (reported as `Sit Tone` by OC)
- Abandoned

Unsolicited Calls on Predictive Dialing Devices

An *unsolicited call* on a predictive dialing device is defined as:

- Any call delivered to a predictive dialing device.
- Any call originated without `TMakePredictiveCall`.

T-Server attempts to clear such unsolicited calls, in order to keep the predictive dialing device available. For delivered calls, T-Server answers and releases the call. For originated or established calls, T-Server releases the call.

Support for No-Answer Supervision

This section describes T-Server's No-Answer Supervision feature.

Agent No-Answer Supervision

This feature provides the following functionality:

- If an agent does not answer a call within a specified timeout, T-Server can divert the call to a sequence of overflow destinations. Alternatively, you can configure T-Server to return calls automatically to the last distribution device.
- If an agent fails to answer a call within a specified timeout, you can configure T-Server to either log out the agent or set the agent to `NotReady` to prevent further calls from arriving.

Configuration Options

T-Server provides three configuration options for defining the behavior of the Agent No-Answer Supervision feature:

- `agent-no-answer-timeout`
- `agent-no-answer-overflow`
- `agent-no-answer-action`

Extension No-Answer Supervision

The No-Answer Supervision feature includes devices of type `extension`. If a call is not answered on an extension within a specified timeout, T-Server can divert the call to a sequence of overflow destinations. Alternatively, you can configure T-Server to return calls automatically to the last distribution device.

Configuration Options

T-Server provides two configuration options for defining the behavior of No-Answer Supervision with devices of type `extension`:

- `extn-no-answer-timeout`
- `extn-no-answer-overflow`

Position No-Answer Supervision

The No-Answer Supervision feature includes devices of type `position`. If a call is not answered on a position within a specified timeout, T-Server can divert the call to a sequence of overflow destinations. Alternatively, you can configure T-Server to return calls automatically to the last distribution device.

Configuration Options

T-Server provides two configuration options for defining the behavior of No-Answer Supervision with devices of type `position`:

- `posn-no-answer-timeout`
- `posn-no-answer-overflow`

Configuration Options for Device-Specific Overrides

T-Server provides three configuration options with which you can configure device-specific overrides for individual devices. You set the values for these options on the Annex tab of the TServer section of the individual device in the Framework Configuration Layer. These are the options:

- `no-answer-timeout`
- `no-answer-overflow`
- `no-answer-action`

Extension Attributes for Overrides for Individual Calls

For all of the No-Answer Supervision options, you can specify the corresponding Extension attribute in `TRequestRouteCall`, to override the configured value for individual calls. This method allows the no-answer behavior to be determined in a routing strategy. These are the three extensions:

- `NO_ANSWER_TIMEOUT`
- `NO_ANSWER_OVERFLOW`
- `NO_ANSWER_ACTION`

Private Calls

You can also apply No-Answer Supervision to private calls, using configuration option `nas-private`:

nas-private

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Place: Immediately

Specifies whether No-Answer Supervision is enabled for private calls. When configured in the T-Server section, this value is the default value applied globally to all private calls.

Note: When set in the TServer section, this option defines the default value for all private calls. However, you can also set a value for this option on the Annex tab of DN of type extension or Agent ID in a section called TServer. When set there, this value overrides the default value for the specific DN.

Recall Scenarios

An additional configuration option allows you to configure No-Answer Supervision for recall scenarios.

recall-no-answer-timeout

Default Value: 15

Valid Values: Any integer from 0-600

Changes Take Place: Immediately

Defines the time that T-Server waits for a call to reappear on a device as a result of a recall (for example, a ringback waiting to be answered). When the timer expires, T-Server executes the actions defined by the relevant overflow option, as well as the action option for cases where an agent is logged in.

With value 0, there is no No-Answer Supervision for such calls.

This option can be defined either in the main Tserver section or in a section called Tserver on the Annex tab of any of the following configuration object types in Configuration Manager:

- Extension
- Position
- VT Port
- AgentID

Reporting

nas-indication

Default Value: none

Valid Values: none, ext, rsn

Changes Take Effect: Immediately

Specifies the reporting action in EventReleased when No-Answer Supervision overflows a call.

With value none, no reason or extension is provided in EventReleased.

With value ext, extension NO_ANSWER_TIMEOUT is supplied in EventReleased.

With value `rsn`, reason `NO_ANSWER_TIMEOUT` is supplied in `EventReleased`.

Smart OtherDN Handling

For T-Server clients that provide the Agent ID value as the `OtherDN` in requests to T-Server, T-Server can convert this `OtherDN` value using its knowledge of the association between the Agent ID and the DN to ensure the correct execution of the request by the switch. For switches expecting an Agent ID in the place of a DN for a particular operation, T-Server can convert the `OtherDN` value supplied by client to the Agent ID that the switch expects.

Configuration Options and Extension

The following configuration options and extension are provided to enable and disable this feature.

Configuration Options

convert-otherdn

Default Value: `+agentid +reserveddn +fwd`

Valid Values: `+/-agentid +/-reserveddn, +/-fwd`

Changes Take Effect: Immediately

Defines whether T-Server has to convert (if applicable) the value provided in request's `AttributeOtherDN`.

Value `+/-agentid` turns on/off either the conversion of the Agent ID value provided in the `OtherDN` attribute to the DN associated with this Agent, or the DN value to Agent ID value (where appropriate).

Value `+/-reserveddn` turns on/off the conversion of `OtherDN` for reserved DNs.

Value `+/-fwd` turns on/off conversion of `OtherDN` in request `TSetCallForward`.

dn-for-undesired-calls

Default Value: No default value

Valid Values: Any valid switch DN

Changes Take Effect: Immediately

Specifies the DN that T-Server uses as the request destination if the client provides a reserved DN in the request.

Note: You can set a value for this option on the `Annex` tab of appropriate DNs in a section called `TServer`. When set there, this value overrides the default value for the DN.

Extension

A new extension key `ConvertOtherDN` is also provided to enable this feature to be applied on a call-by-call basis.

Reporting of Other DN in Secret DN Scenario

When a caller has requested that CLID be withheld, T-Server will suppress the caller's DN appearing in the events for the other parties. The caller's number in the events for the destination DN will be represented by a configurable string that is displayed as `PRIVATE` by default. This ensures that applications do not receive confidential information, but does have the additional consequence of making the reporting model inconsistent when viewed globally by applications such as historical reporting. This in turn may also affect functions like billing.

The configuration option `clid-withheld-name` enables configuration of the string that replaces a withheld CLID value.

clid-withheld-name

Default Value: `PRIVATE`

Valid Values: Any string

Changes Take Effect: Immediately

Defines a name that replaces a withheld CLID. If no value is entered (empty string) the withheld CLID will be displayed.

Supported Requests

[Table 12](#) shows the requests that assume the use of the `OtherDN` value as a switch directory number, and can therefore support Smart OtherDN Handling.

Table 12: Requests That Support Smart OtherDN Handling

TRequest	Meaning of OtherDN Attribute	AgentID-to-DN Conversion	Reserved DN Conversion
TMakeCall	Call destination	Yes	Yes
TMakePredictiveCall ^a	Call destination	Yes	Yes
TRedirectCall	New destination for call	Yes	Yes
TInitiateTransfer	Call destination	Yes	Yes
TMuteTransfer	Call destination	Yes	Yes
TSingleStepTransfer	New destination for call	Yes	Yes
TInitiateConference	Call destination	Yes	Yes
TSingleStepConference	New destination for call	Yes	Yes

Table 12: Requests That Support Smart OtherDN Handling (Continued)

TRequest	Meaning of OtherDN Attribute	AgentID-to-DN Conversion	Reserved DN Conversion
TDeleteFromConference	Conference member to be deleted	Yes	Yes
TListenDisconnect	Request target	Yes	Yes
TListenReconnect	Request target	Yes	Yes
TCallSetForward ^b	Request target	Yes	Yes
TGetAccessNumber ^c	DN for which Access Number is requested	No	No
TSetCallAttributes ^c	Not specified	No	No
TReserveAgentAndGetAccessNumber ^c	DN for which Access Number is requested	No	No
TMonitorNextCall	Agent DN to be monitored	Yes	Not applicable
TCancelMonitoring	Agent DN that was monitored	Yes	Not applicable
TRouteCall ^d	New destination for call		
• RouteTypeUnknown		Yes	Yes
• RouteTypeDefault		Yes	Yes
• RouteTypeOverwriteDNIS		Yes	Yes
• RouteTypeAgentID		Yes ^e	Yes ^e

- a. TMakePredictiveCall assumes that the directory number should be outside the switch; however, this request could also support Smart OtherDN Handling.
- b. TCallSetForward has a separate flag in the configuration option for enabling conversion.
- c. T-Server cannot intercept these requests.
- d. Only the listed route types are applicable for OtherDN conversion.
- e. T-Server must perform Agent ID-to-DN conversion explicitly. The configuration option should be ignored.

Keep-Alive Feature

T-Server may not always receive timely notification when the CTI link stops functioning. In order for T-Server to detect link failure and initialize alarm and recovery procedures, T-Server usually needs to actively check the link's integrity. This is referred to as Keep-Alive or "KPL" functionality.

Keep-alive functionality involves sending a *KPL request* which elicits either a positive or negative response from the CTI link. The responses are counted in four cumulative counters. If the relevant counter reaches the maximum configured limit, T-Server either:

- Decrements the relevant warning/failure KPL tolerance counter
- Attempts to reconnect to the link
- Sends a warning message to Message Server

Three configuration options are available in the Link-Control section of T-Server:

- `kpl interval` sets the interval timer for KPL requests.
- `kpl-tolerance` sets the threshold at which T-Server either attempts to reconnect to the link or issues a warning message.
- `kpl-loss-rate` settings control how the four internal counters tracking both negative and positive KPL responses are incremented and decremented.

kpl-interval

Default Value: 10

Valid Value: Any integer from 0-600

Changes Take Effect: Immediately

Specifies a "keep-alive" interval (in seconds). To check network connectivity, T-Server issues a dummy CTI request at the interval specified when there is no other activity on the link. Value 0 (zero) disables this feature.

The value of this option may need to be increased to avoid false restarts if the switch is sometimes slow to respond, for example, during busy periods.

See also option `kpl-tolerance`.

kpl-tolerance

Default Value: 3

Valid Value: Any integer from 0-10

Changes Take Effect: Immediately

Specifies the threshold number of accumulated KPL request failures. When the threshold is reached T-Server may either consider the CTI link:

- To be lost—in which case T-Server tries to reconnect to it.
- To be unstable—in which case T-Server issues a warning message.

kpl-loss-rate

Default Value: 10, 100

Valid Values: Single integer or comma-separated pair of integers. The lower value in the pair is the failure value and the higher value is the warning rate.

Changes Take Effect: Immediately

Specifies how many KPL positive responses are needed to decrement either the failure or warning tolerance counter.

Value 0 (zero) disables this option.

Two comma-separated values means T-Server will calculate both the failure counter and the warning counter.

A single value means T-Server will calculate only the failure counter.

Note: This option has no effect if option `kpl-tolerance` has value 0. In that case, a single KPL failure will trigger a link restart.

Examples

Tables 13 through 18 in this section illustrate the cumulative effect of KPL responses on the tolerance and loss-rate counters, and what how T-Server reacts when thresholds are reached.

Tables 13 through 18 use the following configuration option values:

- `kpl-tolerance=3`
- `kpl-loss-rate=5, 15` where value 5 is the failure counter and value 15 is the warning counter

Table 13 shows how the KPL tolerance failure counter is decremented when the KPL loss-rate threshold is reached.

Table 13: Failure Counter—KPL Loss-Rate Threshold Reached

KPL Response (X/✓)	Failure Counter		Warning Counter	
	kpl-tolerance=3 X	kpl-loss-rate = 5, (15) ✓	kpl-tolerance=3 X	kpl-loss-rate = (5), 15 ✓
<i>Current values</i>	0	0	0	0
X	1	0	1	0
✓	1	1	1	1
2 x ✓	1	3	1	3
X	2	3	2	3

Table 13: Failure Counter—KPL Loss-Rate Threshold Reached (Continued)

	Failure Counter		Warning Counter	
2 x ✓	2	5	2	5
	kpl-loss-rate threshold (5) reached			
	Decrement KPL tolerance counter by 1	Reset KPL loss-rate counter to zero		
<i>New values</i>	1	0	2	5

Table 14 shows what happens when the KPL tolerance threshold is reached on the failure counter.

Table 14: Failure Counter—KPL Tolerance Threshold Reached

	Failure Counter		Warning Counter	
KPL Response Result (X/✓)	kpl-tolerance=3 x	kpl-loss-rate = 5, (15) ✓	kpl-tolerance=3 x	kpl-loss-rate = (5), 15 ✓
<i>Current values</i>	1	1	0	10
x	2	1	1	10
x	3	1	2	10
	kpl-tolerance threshold reached.			
	T-Server initiates reconnection to CTI link and all counters are reset to 0.			
<i>New values</i>	0	0	0	0

Table 15 shows what how the KPL tolerance warning counter is decremented on when the KPL loss-rate threshold is reached.

Table 15: Warning Counter—KPL Loss-RateThreshold Reached—Tolerance Counter Decremented

	Failure Counter		Warning Counter	
KPL Response Result (X/✓)	kpl-tolerance=3 X	kpl-loss-rate = 5, (15) ✓	kpl-tolerance=3 X	kpl-loss-rate = (5), 15 ✓
<i>Current values</i>	0	3	1	13
X	1	3	2	13
✓	1	4	2	14
✓	1	5	2	15
	KPL loss-rate threshold reached.KPL tolerance counter decremented by 1 and loss-rate counter reset.		KPL loss-rate threshold reached.KPL tolerance counter decremented by 1 and loss-rate counter reset.	
<i>New values</i>	0	0	1	0

Table 16 shows what happens when the KPL tolerance threshold is reached on the warning counter.

Table 16: Warning Counter—KPL Tolerance Threshold Reached—T-Server Sends Warning Message to Message Server

	Failure Counter		Warning Counter	
KPL Response Result (X/✓)	kpl-tolerance=3 X	kpl-loss-rate = 5, (15) ✓	kpl-tolerance=3 X	kpl-loss-rate = (5), 15 ✓
<i>Current values</i>	0	0	2	10
X	1	0	3	10
			kpl-tolerance threshold (3) reached	
			Reset kpl-tolerance counter to 0	

Table 16: Warning Counter—KPL Tolerance Threshold Reached—T-Server Sends Warning Message to Message Server

	Failure Counter		Warning Counter	
			T-Server sends warning message to Message Server	
<i>New values</i>	1	0	0	See table 17 and 18.

Tables 17 and 18 show how the warning counters will behave depending on whether the next KPL response is positive or negative.

Table 17: Warning Counters- KPL Loss-Rate Value After Positive KPL Response

	Failure Counter		Warning Counter	
KPL Response Result (X/✓)	kpl-tolerance=3 X	kpl-loss-rate = 5, (15) ✓	kpl-tolerance=3 X	kpl-loss-rate = (5), 15 ✓
<i>Current values</i>	1	0	0	10
✓	1	1	0	0
				Note: The positive KPL response, after the warning message was sent, results in the counter being reset to 0, instead of incrementing to 11.

Table 18: Warning Counters—KPL Loss-Rate Value After Negative KPL Response

KPL Response Result (X/✓)	Failure Counter		Warning Counter	
	kpl- tolerance=3 X	kpl-loss-rate = 5, (15) ✓	kpl- tolerance=3 X	kpl-loss-rate = (5), 15 ✓
<i>Current values</i>	1	0	0	10
X	2	0	1	11
				Note: The negative KPL response, after the warning message was sent, results in the counter incrementing to 11.

VoIP Absence Code Handling

An emulated agent logged in on a VoIP phone can be made unavailable to new calls by using switch Absence codes.

When Emulated Agent Synchronization is enabled, T-Server will accept ReasonCode and ReasonCodeEnd extensions in TAgentNotReady, and use the provided values to communicate with the PBX to set VoIP phone Absence codes. The same extensions in EventAgentNotReady are used to report Absence codes set on the VoIP phone.

Enabling VoIP Agent Synchronization Feature

To enable VoIP agent synchronization, you must set the value of the T-Server synchronization feature for processing by T-Server in the following order of precedence, highest first:

1. In extension SyncEmuAgent, in request TAgentLogin on a VoIP DN.
2. On the Annex tab of an Agent ID object that will be used on a VoIP DN.
3. On the Annex tab of a Extension object corresponding to a VoIP DN.

Note: This feature is for VoIP DNs only—do not attempt to configure this for any other device type.

When this feature is enabled, Absence codes from the physical DN are mapped to `ready/notready` on an emulated agent. When no emulated agent is logged in, the codes are mapped to `DND`.

When this feature is disabled in the T-Server configuration, T-Server continues to map Absence codes to `DND` functions.

Availability This feature is available on BC 12 and BC 13.

Configuring Absence Codes

The MD110 switch uses 10 absence codes (for VoIP DNs) numbered 0–9. By default, on the switch, the absence codes 0–3 express a 24-hour clock time (the time of return from absence), and codes 4–9 express a calendar date (the date of return from absence).

To report on absence codes in T-Server, the switch codes are mapped to the T-Server configuration option `reason-code`.

To report on absence periods in T-Server, the value(s) of `reason-code` are mapped to the T-Server configuration options `absence-0` to `absence-9`.

Configuration Options

reason-code

Default Value: 2

Valid Values: Any integer from 0–9

Changes Take Effect: Immediately

Equates to the reason why the agent is in a `notready` state. This value is passed to the switch in `TAgentLogin (ManualIn)` or `TAgentSetNotReady`.

The option can be set in the following places, in order of precedence, from the highest to the lowest:

1. In extension `ReasonCode` in requests `TAgentLogin (ManualIn)` and `TAgentSetNotReady` on a VoIP DN.
2. In the `TServer` section on the Annex tab of an `Agent ID` object that will be used on a VoIP DN.
3. In the `TServer` section on the Annex tab of an `Extension` object corresponding to a VoIP DN.
4. In the `TServer` section of the application.

absence-0 ... absence-9

Default Value: None

Valid Value: Any integer

Changes Take Effect: Immediately

Each absence-*n* option enables users to define a default number of minutes or days for the corresponding reason code.

absence-0 ...	Absence period expressed in minutes
absence-3	
absence-4 ...	Absence period expressed in days
absence-9	

These options are set in in the reason-code-end section of T-Server.

**Overriding Default
Absence Period**

Extension ReasonCodeEnd is used when a value that is different from those specified in the absence-*n* options is required. Extension ReasonCodeEnd can be used in requests TAgentLogin (ManualIn) and TAgentSetNotReady on a VoIP DN. The following values can be used:

0-3	Defines expiration in minutes, or defines a specific time (in hh:mm format).
4-9	Defines expiration in days, or defines a specific date (in MM/DD format)

T-Library Functionality

Table 19 presents the T-Library functionality supported in the T-Server for Ericsson MD110. The table entries use these notations:

N—Not supported

Y—Supported

I—Supported, but reserved for Genesys Engineering

E—Event only supported

In Table 19, 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 7 Events and Models Reference Manual*.

Table 19 reflects only that switch functionality used by Genesys software and might not include the complete set of events offered by the switch.

Certain requests in Table 19 are reserved for internal use and are listed here merely for completeness of information.

Notes describing specific functionalities appear at the end of the table.

Table 19: Supported 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		EventRegistered	Y
TUnregisterAddress		EventUnregistered	Y
TRegisterAll			I
TUnregisterAll			I
TRegisterClient			I
Call-Handling Requests			
TMakeCall ^a	Regular	EventDialing	Y
	DirectAgent		N
	SupervisorAssist		N
	Priority		N
	DirectPriority		N
TAnswerCall			Y
TReleaseCall		EventReleased	Y
TClearCall		EventReleased	Y ^b
THoldCall		EventHeld	Y

Table 19: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TRetrieveCall		EventRetrieved	Y
TRedirectCall		EventReleased	Y
TMakePredictiveCall		EventDialing*, EventQueued	Y
Transfer/Conference Requests			
TInitiateTransfer		EventHeld, EventDialing*	Y
TCompleteTransfer		EventReleased*, EventPartyChanged	Y
TInitiateConference		EventHeld, EventDialing*	Y
TCompleteConference		EventReleased*, EventRetrieved, EventPartyChanged, EventPartyAdded	Y
TDeleteFromConference		EventPartyDeleted*, EventReleased	Y
TReconnectCall		EventReleased, EventRetrieved*	Y
TAlternateCall		EventHeld*, EventRetrieved	Y
TMergeCalls	ForTransfer	EventReleased*, EventPartyChanged	N
	ForConference	EventReleased*, EventRetrieved, EventPartyChanged, EventPartyAdded	N
TMuteTransfer		EventHeld, EventDialing*, EventReleased, EventPartyChanged	Y
TSingleStepTransfer		EventReleased*, EventPartyChanged	Y
TSingleStepConference		EventRinging*, EventEstablished	N

Table 19: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
Call-Routing Requests			
TRouteCall ^a	Unknown ^c	EventRouteUsed	Y
	Default		Y
	Label		N
	OverwriteDNIS		Y
	DDD		N
	IDDD		N
	Direct		N
	Reject ^{d e}		Y
	Announcement		N
	PostFeature		N
	DirectAgent		N
	Priority		N
	DirectPriority		N
	AgentID		Y
	CallDisconnect		Y

Table 19: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
Call-Treatment Requests			
TApplyTreatment	Unknown	(EventTreatmentApplied + EventTreatmentEnd)/EventTreatmentNotApplied	N
	IVR		N
	Music		N
	RingBack		N
	Silence		N
	Busy		N
	CollectDigits		N
	PlayAnnouncement		N
	PlayAnnouncementAnd-Digits		N
	VerifyDigits		N
	RecordUserAnnouncement		N
	DeleteUserAnnouncement		N
	CancelCall		N
	PlayApplication		N
	SetDefaultRoute		N
	TextToSpeech		N
	TextToSpeechAndDigits		N
	FastBusy		N
	RAN		N
TGiveMusicTreatment		EventTreatmentApplied	N
TGiveRingBackTreatment		EventTreatmentApplied	N
TGiveSilenceTreatment		EventTreatmentApplied	N

Table 19: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
DTMF (Dual-Tone Multifrequency) Requests			
TCollectDigits		EventDigitsCollected	N
TSendDTMF		EventDTMFSent	Y
Voice-Mail Requests			
TOpenVoiceFile		EventVoiceFileOpened	N
TCloseVoiceFile		EventVoiceFileClosed	N
TLoginMailBox		EventMailBoxLogin	N
TLogoutMailBox		EventMailBoxLogout	N
TPlayVoice		EventVoiceFileEndPlay	N
Agent & DN Feature Requests			
TAgentLogin ^f	WorkModeUnknown	EventAgentLogin	Y
	ManualIn		Y
	AutoIn/Legal Guard		Y
	AfterCallWork		N
	AuxWork		N
	Walk Away		Y
	Return Back		Y
	NoCallDisconnect		N
TAgentLogout		EventAgentLogout	Y
TAgentSetIdleReason		EventAgentIdleReasonSet	N
TAgentSetReady		EventAgentReady	Y
TAgentSetNotReady ^d	WorkModeUnknown	EventAgentNotReady	Y
	ManualIn		N
	AutoIn/Legal Guard		Y
	AfterCallWork		Y

Table 19: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
	AuxWork		N
	Walk Away		Y
	Return Back		Y
	NoCallDisconnect		N
TMonitorNextCall	OneCall	EventMonitoringNextCall	N
	AllCalls		N
TCancelMonitoring		EventMonitoringCanceled	N
TStartCallMonitoring			I
TStopCallMonitoring			I
TCallSetForward	None	EventForwardSet	Y
	Unconditional		Y
	OnBusy		Y ^g
	OnNoAnswer		Y ^e
	OnBusyAndNoAnswer		N
	SendAllCalls		N
TCallCancelForward	None	EventForwardCancel	Y
	Unconditional		Y
	OnBusy		Y
	OnNoAnswer		Y
	OnBusyAndNoAnswer		N
	SendAllCalls		N
TSetMuteOff		EventMuteOff	N
TSetMuteOn		EventMuteOn	N
TListenDisconnect		EventListenDisconnected	N
TListenReconnect		EventListenReconnected	N

Table 19: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TSetDNDOOn		EventDNDOOn	Y
TSetDNDOOff		EventDNDOOff	Y
TSetMessageWaitingOn		EventMessageWaitingOn	N
TSetMessageWaitingOff		EventMessageWaitingOff	N
TTransactionMonitoring		EventTransactionStatus	Y
Query Requests			
TQuerySwitch	DateTime	EventSwitchInfo	N
	ClassifierStat		N
TQueryCall	PartiesQuery	EventPartyInfo	N
	StatusQuery		Y
TQueryAddress	AddressStatus	EventAddressInfo	Y
	MessageWaitingStatus		N
	AssociationStatus		N
	CallForwardingStatus		Y
	AgentStatus		Y
	NumberOfAgentsInQueue		Y
	NumberOfAvailableAgentsInQueue		Y
	NumberOfCallsInQueue		Y
	AddressType		Y
	CallsQuery		Y
	SendAllCallsStatus		Y
	QueueLoginAudit		Y
	NumberOfIdleTrunks		N
	NumberOfTrunksInUse		N
	DatabaseValue		N

Table 19: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
	DNStatus		Y
	QueueStatus		Y
TQueryLocation	AllLocations	EventLocationInfo	I
	LocationData		I
	MonitorLocation		I
	CancelMonitorLocation		I
	MonitorAllLocations		I
	CancelMonitorAllLocations		I
	LocationMonitorCanceled		I
	AllLocationsMonitor Canceled		I
TQueryServer		EventServerInfo	Y
User-Data Requests			
TAttachUserData		EventAttachedDataChanged	Y
TUpdateUserData		EventAttachedDataChanged	Y
TDeleteUserData TDeletePair		EventAttachedDataChanged	Y
TDeleteAllUserData		EventAttachedDataChanged	Y
ISCC (Inter-Server Call Control) Requests			
TGetAccessNumber		EventAnswerAccessNumber	Y
TCancelReqGetAccess- Number		EventReqGetAccessNumber Canceled	Y
Special Requests			
TReserveAgent		EventAgentReserved	I
TReserveAgentAndGet- AccessNumber			I

Table 19: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TSendEvent TDistributeEvent		EventACK	I
TSendEventEx		EventACK	I
TSetCallAttributes TSetCallInfo		EventCallInfoChanged	Y
TSendUserEvent TDistributeUserEvent		EventACK	Y
TPrivateService		EventPrivateInfo	Y
TLostBackupConnection			N
Network 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

- If unsupported subtypes are used, no error is returned.
- Supported on when Maintain Position in Queue feature is enabled.
- Supports any type of Routing Point.
- Subtype Reject is supported only for routing from Hunting Groups.
- From 7.6, subtype Reject no longer requires the SupervisedRoute = 1 extension to clear the call, because T-Server intelligently clears the call in the best manner for the switch.
- If unsupported subtypes are used, T-Server by default uses WorkModeUnknown.

- g. The switch does not support setting of a user-defined forwarding destination. A destination preconfigured in the switch can be set via CTI. The `otherDN` value should be empty when request `ForwardOnNoAnswer/Busy`.
- h. All T-Servers support NAT/C requests with `AttributeHomeLocation`, provided that this attribute identifies a network location that is capable of processing such requests.

Support for Agent Work Modes

[Table 20](#) indicates the types of agent work modes that T-Server for Ericsson MD110 supports.

Table 20: Supported Agent Work Modes

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

- a. Valid only if extension `WrapUpTime` is provided in the request. See “Pending ACW” on [page 152](#).

Use of the Extensions Attribute

The T-Server for the Ericsson MD110 switch supports the use of the `Extensions` attribute as documented in the *Genesys 7 Events and Models Reference Manual*.

T-Server also supports the Extensions described in [Table 21](#).

Table 21: Use of the Extensions Attribute

Extension		Used In	Description
Key	Type		
GCTI_SUPERVISED_ROUTE	integer string	TRouteCall	Specifies the use of <code>Diver</code> t (value of 0) or <code>Transfer</code> (nonzero value) services to route a call.
GCTI_KEY_NO	integer string	TPrivateService	Uses values from 0–255. Invokes the MD110 <code>Press Programmable Key</code> escape service. Users are responsible for determining which services are invoked by the programmable keys.
dnd	integer	EventAddressInfo	Provides the status of the DND state on the device.
fwd	string	EventRegistered EventAddressInfo	Provides the status of the Forwarding state on the device. Valid values are: <ul style="list-style-type: none"> • unk • off • on • fwd dst dn • selective
NO_ANSWER_TIMEOUT	string	TRouteCall	If set, the value of this extension overrides any value set in any of the following configuration options for the current call: <ul style="list-style-type: none"> • no-answer-timeout • agent-no-answer-timeout • extn-no-answer-timeout • posn-no-answer-timeout

Table 21: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
NO_ANSWER_ACTION	string	TRouteCall	<p>If set, the value of this extension overrides any value set in any of the following configuration options for the current call:</p> <ul style="list-style-type: none"> no-answer-action agent-no-answer-action
NO_ANSWER_OVERFLOW	comma-separated list	TRouteCall	<p>If set, the value of this extension overrides any value set in any of the following configuration options for the current call:</p> <ul style="list-style-type: none"> no-answer-overflow agent-no-answer-overflow extn-no-answer-overflow posn-no-answer-overflow
SUPERVISED_ROUTE	string	TRouteCall	Overrides the value of configuration option supervised-route-timeout for individual calls.
ConvertOtherDN	integer or string	See “Smart OtherDN Handling” on page 160 .	<p>If set, the value of this extension overrides the value set in configuration option convert-otherdn.</p> <p>Value 0 disables all conversions for the call.</p> <p>Value 1 forces the relevant conversion for the call.</p>

Table 21: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
CallRetainInQueue	string	TRouteCall	If set, the value of this extension overrides the value set in configuration option <code>call-retain-in-queue</code> .
	string	Call-related events	Appears in event reporting when Maintain Position in Queue is applied. Values <code>keep</code> and <code>divert</code> indicate successful feature execution and inform which reporting model is in use.
ReasonCode	integer	EventRegistered EventAddressInfo EventAgentNotReady TAgentNotReady TAgentLogin	In <code>EventRegistered</code> and <code>EventAddressInfo</code> (DN Status), provides the Absence code when an emulated agent is logged in and emulated agent synchronization with the PBX is enabled. In <code>EventAgentNotReady</code> , provides the Absence code name as requested by client (when emulated agent synchronization with the PBX is enabled). In <code>TAgentLogin</code> (Workmode = Manual In) and <code>TAgentNotReady</code> , the client may supply the desired Absence code and T-Server will apply it on the VoIP phone (when emulated agent synchronization with the PBX is enabled).

Table 21: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
ReasonCodeEnd	string	EventRegistered, EventAddressInfo EventAgentNotReady TAgentNotReady	<p>In EventRegistered and EventAddressInfo (DN Status), provides the Absence code expiration time.</p> <p>In EventAgentNotReady, provides the Absence code expiration time.</p> <p>In TAgentNotReady, the client may supply the desired Absence code expiration time and T-Server will apply it on the VoIP phone</p> <p>Note: Extensions are provided in events only when emulated agent synchronization with the PBX is enabled.</p> <p>Note: Neither T-Server nor the PBX validates the value.</p>
BusinessCall	integer	All call-related events	<p>0—Private call</p> <p>1—Business call</p> <p>2—Work-related call</p>
SwitchSpecificType	string or integer	TRegisterAddress	<p>Specifies the switch-specific type T-Server uses when the created DN is not specified in the Configuration Layer.</p> <p>T-Server verifies the combination device type/switch specific type combination in the same manner as it does for a DN configured in the Configuration Layer.</p> <p>Note: Values in this extension key that are unknown to T-Server are processed as 0 (zero).</p>

Table 21: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
EmulateLogin	string	TAgentLogin	With value yes, T-Server performs an emulated login. With value no, T-Server attempts a real login.
EmulateLogin	string	EventAgentLogin EventAddressInfo EventRegistered	Value yes indicates that the T-Server has performed an emulated login.
WrapUpTime	integer	TAgentLogin	Specifies the amount of emulated wrap-up time (in seconds) allocated to this agent at the end of a business call. This value is effective for the duration of this agent's login session. It can be overridden by the value in the WrapUpTime extension in TAgentNotReady.
WrapUpTime	integer	TAgentNotReady	Specifies the amount of emulated wrap-up time (in seconds) allocated to this agent at the end of a business call. This value is effective only for the lifespan of this request.

User Data Keys

The T-Server for the Ericsson MD110 switch supports the user data key described in [Table 22](#).

Table 22: User Data Keys

Extension		Used In	Description
Key	Type		
GCTI_CSTA_CORRELATOR	string	AssociateData Call-related events	Maximum of 20 alphanumeric characters (0-9, A-Z) When the switch provides a Private Data field in call-related events, a data key GCTI_CSTA_CORRELATOR is automatically added to a call's user data. The meaning of the key is "switch-provided user data," for example, CLID or Customer ID. If there are existing keys in user data with the same name, data is rewritten accordingly with the latest value provided in the CSTA event from the switch.
WrapUpTime	integer	Call-related requests	Specifies the amount of emulated wrap-up time allocated to all agents at the end of the business call.

Error Messages

[Table 23](#) presents the complete set of error messages T-Server distributes in `EventError`.

Table 23: Error Messages

Code	Description
T-Server-Defined Errors	
40	No additional licenses

Table 23: Error Messages (Continued)

Code	Description
41	Client has not registered for DN
42	Resource is already seized
43	Object is already in requested state
50	Unknown error
51	Unsupported operation
52	Internal error
53	Invalid attribute
54	Switch not connected
55	Incorrect protocol version
56	Invalid connection ID
57	Timeout expired
58	Out of service
59	DN not configured in Configuration Manager
71	Invalid Called DN
88	Origination DN not specified
96	Cannot complete conference
97	Cannot initiate transfer
98	Cannot complete transfer
99	Cannot retrieve original signal
100	Unknown cause
105	Information element missing
109	Link down or bad link specified
111	Too many outstanding requests
118	Requested service unavailable
119	Invalid password

Table 23: Error Messages (Continued)

Code	Description
123	DN for association does not exist
128	Invalid DN type for DN registration
132	Invalid link ID
133	Link already established
147	No link responding
148	Facility already enabled
149	Facility already disabled
164	Invalid system command
166	Resource unavailable
168	Invalid origination address
169	Invalid destination request
171	Switch cannot retrieve call
172	Switch cannot complete transfer
173	Switch cannot complete conference
174	Cannot complete answer call
175	Switch cannot release call
177	Target DN invalid
179	Feature could not be invoked
185	Set is in invalid state for invocation
186	Set is in target state
191	Agent ID IE is missing or invalid
192	Agent ID is invalid
202	Another application has acquired the resource
220	No internal resource available
221	Service not available on device

Table 23: Error Messages (Continued)

Code	Description
223	Invalid parameter passed to function
231	DN is busy
236	Timeout performing operation
256	API restricted from monitor
259	Invalid password
263	Must be logged on to use this command
302	Invalid DTMF string
323	No answer at DN
380	Interdigit timeout occurred
402	Invalid route address
452	No trunk for outbound calls
477	Invalid Call ID
496	Invalid call state
503	Network failed to deliver outbound call
504	Network rejected outbound call
506	Invalid teleset state
527	Agent ID already in use
627	Unknown information element detected
700	Invalid login request
701	Invalid logout request
704	Invalid make call request
705	Route request is invalid
706	Invalid mute transfer request
708	Invalid initiate transfer request
710	Invalid complete transfer request

Table 23: Error Messages (Continued)

Code	Description
711	Invalid retrieve request
712	Cannot find route point in call
714	Invalid Call_ID
717	Agent not logged in
742	Invalid DN
749	Agent already logged in
750	Extension in use
804	Invalid Call_ID
ISCC (Inter Server Call Control) Errors	
1000	Invalid or missing server location name
1001	Remote server disconnected
1002	Remote server has not processed request
1004	Remote link disconnected
1005	External routing feature not initiated
1006	No free CDNs
1007	No access number
1008	TCS feature is not initiated
1009	Invalid route type
1010	Invalid request
1011	No primary server was found on location
1012	Location is invalid or missing
1013	Timeout performing requested transaction
1014	No configured access resources are found
1015	No registered access resources are found
1016	Client is not authorized

Table 23: Error Messages (Continued)

Code	Description
1017	Invalid transaction type
1018	Invalid or missing transaction data
1019	Invalid location query request
1020	Invalid origin location
Operational Errors	
1110	Duplicate invocation (packet missed)
1111	Unrecognized operation (packet transmission error)
1112	Mistyped argument (packet transmission error)
1113	Resource limitation
1114	Initiator releasing
1115	Unrecognized link ID
1116	Unexpected linked response
1117	Unexpected child operation
1120	Unrecognized invocation
1121	Result response unexpected
1122	Mistyped result
1130	Unrecognized invocation
1131	Unexpected error response
1132	Unrecognized error
1133	Unexpected error
1134	Mistyped parameter
1140	Generic
1141	Request incompatible with object
1142	Value is out of range
1143	Object not known

Table 23: Error Messages (Continued)

Code	Description
1144	Invalid calling device
1145	Invalid called device
1146	Invalid forwarding destination
1147	Request caused privilege violation on device
1148	Request caused privilege violation on called device
1149	Request caused privilege violation on calling device
1150	Invalid call identifier
1151	Invalid device identifier
1152	Invalid CSTA connection identifier
1153	Invalid call destination
1154	Invalid feature requested
1155	Invalid allocation state
1156	Invalid cross-reference identifier
1157	Invalid object type provided in the request
1158	Security violation
State-Incompatibility Errors	
1160	Generic
1161	Invalid object state
1162	Invalid connection ID
1163	No active call
1164	No held call
1165	No call to clear
1166	No connection to clear
1167	No call to answer
1168	No call to complete

Table 23: Error Messages (Continued)

Code	Description
System Resource–Availability Errors	
1170	Generic
1171	Service is busy
1172	Resource is busy
1173	Resource is out of service
1174	Network busy
1175	Network out of service
1176	Overall monitor limit exceeded
1177	Conference member limit exceeded
Subscribed Resource–Availability Errors	
1180	Generic
1181	Object monitor limit exceeded
1182	Trunk limit exceeded
1183	Outstanding request limit exceeded
Performance-Management Errors	
1185	Generic
1186	Performance limit exceeded
Security Errors	
1190	Unspecified
1191	Sequence number violated
1192	Timestamp violated
1193	PAC violated
1194	Seal violated
1700	The agent is already reserved by another server

Table 23: Error Messages (Continued)

Code	Description
Network Attended Transfer/Conference Errors	
1901	Unexpected request TNetworkConsult.
1902	Unexpected request TNetworkAlternate.
1903	Unexpected request TNetworkReconnect.
1904	Unexpected request TNetworkTransfer.
1905	Unexpected request TNetworkMerge.
1906	Unexpected request TNetworkSingleStepTransfer.
1907	Unexpected request TNetworkPrivateService.
1908	Unexpected message.



Chapter

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 195](#)
- [Mandatory Options, page 196](#)
- [Log Section, page 196](#)
- [Log-Extended Section, page 210](#)
- [Log-Filter Section, page 212](#)
- [Log-Filter-Data Section, page 213](#)
- [Common Section, page 213](#)
- [Changes from 7.5 to 7.6, page 214](#)

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 it is otherwise specified in this document or in the documentation for your application, you set common configuration options in Configuration Manager in the corresponding sections on the `Options` tab of the `Application` object.

Warning! Configuration section names, configuration option names, and predefined option values are case-sensitive. Type them in the Configuration Manager interface 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 202](#).

Note: For definitions of the Standard, Interaction, Trace, and Debug log levels, refer to the *Framework 7.6 Deployment Guide* or to *Framework 7.6 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 202](#)). 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>1–100</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-file

Default 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_convertDefault Value: `Local`

Valid Values:

- | | |
|--------------------|--|
| <code>local</code> | 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. |
| <code>utc</code> | 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_formatDefault Value: `time`

Valid Values:

<code>time</code>	The time string is formatted according to the HH:MM:SS.sss (hours, minutes, seconds, and milliseconds) format.
<code>locale</code>	The time string is formatted according to the system's locale.
<code>ISO8601</code>	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-attributesDefault Value: `false`

Valid Values:

<code>true</code>	Attaches extended attributes, if any exist, to a log event sent to log output.
<code>false</code>	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 7.6 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-pointDefault 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 202](#)). 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 202](#).

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 the `compatible-output-priority` option unless you have specific reasons to use the 6.x log output logic—that is, to mimic the output priority as implemented in releases 6.x. Setting this option to `true` affects log consistency.

Log Output Options

To configure log outputs, set log level options (`all`, `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 206](#).

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

Warnings!

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

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 <code>network</code> output enables an application to send log events of the <code>Standard</code> , <code>Interaction</code> , and <code>Trace</code> levels to Message Server. <code>Debug</code> -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.

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 (stdout).
<code>stderr</code>	Log events are sent to the Standard error output (stderr).
<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 (stdout).
<code>stderr</code>	Log events are sent to the Standard error output (stderr).
<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):

stdout	Log events are sent to the Standard output (stdout).
stderr	Log events are sent to the Standard error output (stderr).
memory	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
[filename]	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the Debug level and higher (that is, log events of the Standard, Interaction, Trace, and Debug levels). The log output types must be separated by a comma when more than one output is configured—for example:

```
debug = stderr, /usr/local/genesys/logfile
```

Note: Debug-level log events are never sent to Message Server or stored in the Log Database.

Log File Extensions

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

- *.log—Assigned to log files when you configure output to a log file. For example, if you set `standard = confservlog` for Configuration Server, it prints log messages into a text file called `confservlog.<time_stamp>.log`.
- *.qsp—Assigned to temporary (spool) files when you configure output to the network but the network is temporarily unavailable. For example, if you set `standard = network` for Configuration Server, it prints log messages into a file called `confserv.<time_stamp>.qsp` during the time the network is not available.
- *.snapshot.log—Assigned to files that contain the output snapshot when you configure output to a log file. The file contains the last log messages that an application generates before it terminates abnormally. For example,

if you set `standard = confservlog` for Configuration Server, it prints the last log message into a file called `confserv.<time_stamp>.snapshot.log` in case of failure.

Note: Provide `*.snapshot.log` files to Genesys 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. Use this configuration when trying to reproduce an application's failure. The memory log file will contain a snapshot of the application's log at the moment of failure; this should help you and Genesys Technical Support identify the reason for the 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 following options 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:

<code>alarm</code>	The log level of log event <code><eventID></code> is set to <code>Alarm</code> .
<code>standard</code>	The log level of log event <code><eventID></code> is set to <code>Standard</code> .
<code>interaction</code>	The log level of log event <code><eventID></code> is set to <code>Interaction</code> .
<code>trace</code>	The log level of log event <code><eventID></code> is set to <code>Trace</code> .
<code>debug</code>	The log level of log event <code><eventID></code> is set to <code>Debug</code> .
<code>none</code>	Log event <code><eventID></code> 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` (see [page 212](#)).

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 *URS 7.6 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-disableDefault 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

This section must be called `log-filter`.

default-filter-typeDefault Value: `copy`

Valid Values:

<code>copy</code>	The keys and values of the KVList pairs are copied to the log.
<code>hide</code>	The keys of the KVList pairs are copied to the log; the values are replaced with strings of asterisks.
<code>skip</code>	The KVList pairs are not copied to the log.

Changes Take Effect: Immediately

Specifies the default way of presenting KVList information (including `UserData`, `Extensions`, and `Reasons`) in the log. The selected option will be applied to the attributes of all KVList pairs except the ones that are explicitly defined in the `log-filter-data` section.

Example

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

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

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

Log-Filter-Data Section

This section must be called `log-filter-data`.

<key name>

Default Value: `copy`

Valid Values:

<code>copy</code>	The key and value of the given KVList pair are copied to the log.
<code>hide</code>	The key of the given KVList pair is copied to the log; the value is replaced with a string of asterisks.
<code>skip</code>	The KVList pair is not copied to the log.

Changes Take Effect: Immediately

Specifies the way of presenting the KVList pair defined by the key name in the log. Specification of this option supersedes the default way of KVList presentation as defined in the `log-filter` section for the given KVList pair.

Note: If the T-Server common configuration option `log-trace-flag` is set to `-udata`, it will disable writing of user data to the log regardless of settings of any options in the `log-filter-data` section.

Example

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

Here is an example of the log with option `PASSWORD` set to `hide`:

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

Common Section

This section must be called `common`.

enable-async-dnsDefault 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-delayDefault 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 7.5 to 7.6

Table 24 provides all the changes to common configuration options between release 7.5 and the latest 7.6 release.

Table 24: Common Log Option Changes from 7.5 to 7.6

Option Name	Option Values	Type of Change	Details
Log Section			
Use the following options only when requested by Genesys Technical Support.			
<code>x-conn-debug-open</code>	<code>0, 1</code>	New	See the description on page 207 .
<code>x-conn-debug-select</code>	<code>0, 1</code>	New	See the description on page 208 .

Table 24: Common Log Option Changes from 7.5 to 7.6 (Continued)

Option Name	Option Values	Type of Change	Details
x-conn-debug-timers	0, 1	New	See the description on page 208 .
x-conn-debug-write	0, 1	New	See the description on page 208 .
x-conn-debug-security	0, 1	New	See the description on page 208 .
x-conn-debug-api	0, 1	New	See the description on page 209 .
x-conn-debug-dns	0, 1	New	See the description on page 209 .
x-conn-debug-all	0, 1	New	See the description on page 209 .
Extended Log Section (New Section)			
level-reassign-<eventID>	alarm, standard, interaction, trace, debug, none	New	See the description on page 210 .
level-reassign-disable	true, false	New	See the description on page 212 .
Common Section (New Section)			
Use the following options only when requested by Genesys Technical Support.			
enable-async-dns	off, on	New	Use only with T-Servers. See the description on page 214 .
rebind-delay	10–600	New	See the description on page 214 .



Chapter

9

T-Server Common Configuration Options

This chapter describes the configuration options that are common to all T-Server types. It contains the following sections:

- [Setting Configuration Options, page 217](#)
- [Mandatory Options, page 218](#)
- [T-Server Section, page 218](#)
- [License Section, page 223](#)
- [Agent-Reservation Section, page 225](#)
- [Multi-Site Support Section, page 226](#)
- [Translation Rules Section, page 235](#)
- [Backup-Synchronization Section, page 236](#)
- [Call-Cleanup Section, page 237](#)
- [Security Section, page 239](#)
- [Timeout Value Format, page 239](#)
- [Option Changes from Release 7.5 to 7.6, page 240](#)

T-Server also supports common log options described in Chapter 8, “Common Configuration Options,” on [page 195](#).

Setting Configuration Options

Unless it is specified otherwise, you set configuration options in Configuration Manager in the corresponding sections on the `Options` tab for the T-Server Application object.

Mandatory Options

Except as noted for certain environments, the configuration of common options is not required for basic T-Server operation.

T-Server Section

The T-Server section contains the configuration options that are used to support the core features common to all T-Servers.

TServer 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: `false`

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.

Note: Use of this option can negatively impact T-Server processing speed.

background-timeout

Default Value: 60 msec

Valid Values: See “Timeout Value Format” on [page 239](#).

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 checks whether a client provides the correct name and password of a tenant. If it does, 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.

Note: To make T-Server compatible with 3.x and 5.x clients, set the `check-tenant-profile` option to `false`.

compatibility-port

Default Value: 0

Valid Values: 0 or any valid TCP/IP port

Changes Take Effect: After T-Server has reconnected to the link

Specifies the TCP/IP port that 3.x clients use to establish connections with T-Server. Connections to this port are accepted only if T-Server has a connection with the switch. If set to 0 (zero), this port is not used.

Note: Starting with release 7.5, 3.x clients are no longer supported. You can use this option for backward compatibility with the previous T-Server releases.

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.

log-trace-flags

Default Value: +iscc, +cfg\$dn, -cfgserv, +passwd, +udata, -devlink, -sw, -req, -callops, -conn, -client

Valid Values (in any combination):

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

Changes Take Effect: Immediately

Specifies—using a space-, comma- or semicolon-separated list—the types of information that are written to the log files.

management-port

Default Value: 0

Valid Values: 0 or any valid TCP/IP port

Changes Take Effect: After T-Server is restarted

Specifies the TCP/IP port that management agents use to communicate with T-Server. If set to 0 (zero), this port is not used.

merged-user-data

Default Value: main-only

Valid Values:

main-only	T-Server attaches user data from the remaining call only.
merged-only	T-Server attaches user data from the merging call.
merged-over-main	T-Server attaches user data from the remaining and the merging call. In the event of equal keys, T-Server uses data from the merging call.
main-over-merged	T-Server attaches data from the remaining and the merging call. In the event of equal keys, T-Server uses data from the remaining call.

Changes Take Effect: Immediately

Specifies the data that is attached to the resulting call after a call transfer, conference, or merge completion.

Note: The option setting does not affect the resulting data for merging calls if the `consult-user-data` option is set to `joint`. (See “consult-user-data” on [page 220](#).)

server-id

Default Value: An integer equal to the `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 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 224](#).

license This section must be called `license`.

Notes: T-Server also supports the `license-file` option described in the *Genesys 7 Licensing Guide*.

The `License` section is not applicable to Network T-Server for DTAG.

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: `0` or string `max`

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 2, “DNs and Agent Logins,” [page 45](#).

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 225](#).

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
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-licences = max	tserver_sdn = 500	500 seat-related DNs
num-sdn-licences = max		

Example 2

If...		Then...
Options Settings	License File Settings	License Checkout
num-of-licences = 1000	tserver_sdn = 500	500 seat-related DNs
num-sdn-licences = max		

Example 3

If...		Then...
Options Settings	License File Settings	License Checkout
num-of-licences = 1000	tserver_sdn = 600	400 seat-related DNs
num-sdn-licences = 400		

Example 4

If...		Then...
Options Settings	License File Settings	License Checkout
num-of-licences = max	tserver_sdn = 5000	1000 seat-related DNs
num-sdn-licences = 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 32](#) section for details on this feature.

agent-reservation 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.

reject-subsequent-request

Default Value: `true`

Valid Values:

- `true` T-Server rejects subsequent requests.
- `false` A subsequent request prolongs the current reservation made by the same client application for the same agent.

Changes Take Effect: Immediately

Specifies whether T-Server rejects subsequent requests from the same client application, for an agent reservation for the same Agent object that is currently reserved.

Note: Genesys does not recommend setting this option to `false` in a multi-site environment in which remote locations use the Agent-Reservation feature.

request-collection-time

Default Value: `100 msec`

Valid Values: See “Timeout Value Format” on [page 239](#).

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 239](#).

Changes Take Effect: Immediately

Specifies the default interval that an AgentDN is reserved to receive a routed call from a remote T-Server. During this interval, the agent cannot be reserved again.

Multi-Site Support Section

The Multi-Site Support 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 are grouped with related options that support the same functionality (such as those for Transfer Connect Service or the ISCC/Call Overflow feature).

extrouter This 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 239](#).

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-changesDefault Value: `false`

Valid Values:

- | | |
|--------------------|-------------------------------------|
| <code>true</code> | EventPartyChanged is generated. |
| <code>false</code> | EventPartyChanged is not generated. |

Changes Take Effect: Immediately

Specifies whether the destination T-Server generates `EventPartyChanged` for the incoming call when the resulting `ConnID` attribute is different from the `ConnID` attribute of an instance of the same call at the origination location.

use-data-fromDefault Value: `active`

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-voi`, `reroute`, `direct-callid`, `direct-voi`, `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 79](#) 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-uu, direct-callid, direct-network-callid, direct-uu, 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 239](#).

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 239](#).

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 239](#).

Changes Take Effect: For the next request for remote service

Specifies the time interval that a T-Server at the origination location waits for a notification of routing service availability from the destination location.

Counting starts when the T-Server sends a request for remote service to the destination site.

resource-allocation-mode

Default Value: `circular`

Valid Values:

- `home` T-Server takes an alphabetized (or numerically sequential) list of configured DNs and reserves the first available DN from the top of the list for each new request. For example, if the first DN is not available, the second DN is allocated for a new request. If the first DN is freed by the time the next request comes, the first DN is allocated for this next request.
- `circular` T-Server takes the same list of configured DNs, but reserves a subsequent DN for each subsequent request. For example, when the first request comes, T-Server allocates the first DN; when the second request comes, T-Server allocates the second DN; and so on. T-Server does not reuse the first DN until reaching the end of the DN list.

Changes Take Effect: Immediately

Specifies the manner in which T-Server allocates resources (that is, DNs of the `External Routing Point` type and Access Resources with `Resource Type dn is`) for multi-site transaction requests.

resource-load-maximum

Default 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 239](#).

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.

<code>always</code>	The TCS feature is used for every call.
<code>app-defined</code>	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 <code>TC-type</code> key and a nonempty string value to the <code>UserData</code> 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 239](#).

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 239](#).

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 239](#).

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-allDefault 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-featureDefault 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-toutDefault Value: `10 sec`Valid Values: See “Timeout Value Format” on [page 239](#).

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-idDefault 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 (formerly DMS-100).

Event Propagation Option

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.

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 88](#). See “Configuring Number Translation” on [page 94](#) 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
```

Backup-Synchronization Section

The Backup-Synchronization section contains the configuration options that are used to support a high-availability (hot standby redundancy type) configuration.

backup-sync 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-traceDefault Value: `off`

Valid Values:

<code>off, false, no</code>	No trace (default).
<code>local, on, true, yes</code>	Trace on this T-Server side only.
<code>remote</code>	Trace on the redundant T-Server side only.
<code>full, both</code>	Full trace (on both sides).

Changes Take Effect: Immediately

Specifies whether the option is active, and to what level the trace is performed. This option applies only if the `protocol` option is set to `addp`.

protocolDefault Value: `default`

Valid Values:

<code>default</code>	The feature is not active.
<code>addp</code>	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-toutDefault Value: `20 sec`Valid Values: See “Timeout Value Format” on [page 239](#).

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 7.6 Management Layer User’s Guide*.

call-cleanup This section must be called `call-cleanup`.

cleanup-idle-toutDefault Value: `0`Valid Values: See “Timeout Value Format” on [page 239](#).

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.

notify-idle-tout

Default Value: 0

Valid Values: See “Timeout Value Format” on [page 239](#).

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 239](#).

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 checking.

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.

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 7.6 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
```

Option Changes from Release 7.5 to 7.6

[Table 26](#) lists the configuration options that:

- Are new or changed in the 7.6 release of T-Server
- Have been added or changed since the most recent 7.5 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 7.5 to 7.6

Option Name	Option Values	Type of Change	Details
TServer Section			
ani-distribution	inbound-calls-only, all-calls, suppressed	New	See the option description on page 218 .
use-data-from	active, original, current, active-data-original-call	New value	New option value, active-data-original-call. See the option description on page 228 .
compatibility-port	0 or any valid TCP/IP port	Obsolete	See the option description on page 219 .
backup-sync Section			
network-provided-address	true, false	Obsolete	



Chapter

10

Configuration Options in T-Server for Ericsson MD110

This chapter describes the configuration options unique to the T-Server for Ericsson MD110 and includes these sections:

- [Mandatory Options, page 241](#)
- [T-Server Section, page 242](#)
- [Reason-code-end Section, page 266](#)
- [Switch-Specific Type Section, page 266](#)
- [Annex Tab Options, page 266](#)
- [Application-Link Section, page 269](#)
- [Link-Control Section, page 270](#)
- [Changes from 7.5 to 7.6, page 274](#)

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
Application-Link Section		
app-link-version	3.0.200.	Specifies the version of the Application Link. See description on page 270 .
Link Control Section		
hostname	No default value	Specifies the host of the link according to the switch configuration. See description on page 270 .
port	No default value	Specifies the TCP/IP port of the link according to the switch configuration. See description on page 270 .

T-Server Section

This section must be called TServer.

connid-update

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

With value `true`, T-Server distributes `EventPartyChanged` in response to the silent connection ID update performed by ISCC for scenarios where ISCC cannot recognize “topology loop.”

default-destination

Default Value: `default`

Valid Value: A string of destination DNs separated by the pipe (|) symbol

Changes Take Effect: Immediately

Specifies the default destination to which calls are routed in exception situations.

queue-cleanupDefault Value: `false`Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether T-Server deletes unstable calls (in queued or ringing state) after the time set in the `call-expire-time` option expires.

switch-agent-idDefault Value: `false`Valid Values: `true`, `false`

`false` When logging in an agent via CTI, the `AgentID` can be any value and is mandatory. The password field is an optional value. However, the `AgentID` value specified is only used by T-Server internally for reporting to its clients. T-Server passes the value of the ODN (the `DIR` value in the manual unlock code below) to the Application Link.

If a password has been specified in the CTI login, T-Server passes it to the Application Link. If no password is specified, T-Server by default sends the value of the ODN to the Application Link (Auth No. in the manual unlock code in the Note below).

`true` With value `true`, T-Server, when logging in an agent via CTI, sends to the Application Link the `AgentID` specified in the CTI login (that is, not the ODN value). Password behavior is the same as above.

Changes Take Effect: Immediately

Specifies whether T-Server passes the `AgentID` specified in the client request to the switch. This option is only useful for nonstandard switch configuration, and `AgentID` must be a decimal value. Do not change the default value unless only the `AgentID` configured on the switch must be used.

To manually unlock a phone, use the code `#FC*DIR*Auth No#`, where `FC` is the country-specific unlock code.

Note: At T-Server start/restart, T-Server queries the switch for device status. If the device status is reported by the switch as logged in, T-Server will by default use the ODN value for that device as the agent login ID attribute, which T-Server will then distribute to its clients.

Upon receiving a `RequestAgentLogin` from a client where the device is logged out, T-Server passes the ODN value to the switch in the login request, but communicates with the client using the agent login ID value from the client's `RequestAgentLogin`.

Hence any T-Server start/restart will compromise the monitoring and reporting of agent status, unless agent login IDs in the Genesys environment are set up as identical to the ODN value.

agentid-as-password

Default Value: `false`

Valid Values: `true`, `false`

`true` With value `true`, T-Server gets the password value from the CSTA logged-on event and represents it as attribute `AgentID` for any events it distributes to clients registered for this device.

`false` With value `false` (or with an empty password field) T-Server uses the ODN number to report `AgentID` to its clients.

Changes Take Effect: Immediately

Used when a T-Server receives from the link a CSTA logged-on event that is unrelated to any `RequestAgentLogin` made by that T-Server. This CSTA logged-on may be in response to a request made from another CTI application on the same Application Link (for example, another T-Server).

Note: This option is only relevant for agent login events received by T-Server from the Application Link, that have not been in response to a CTI request made by that T-Server. This login event may, however, be in response to a request made from another CTI application on the same Application Link (for example, another T-Server)

adn-login

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Prior to release 6.5.301.00, with value `false`, T-Server did not permit `RequestAgentLogin` via the ADN. From release 6.5.301.00 onward, T-Server permits `RequestAgentLogin` from any associated devices (ODNs and ADNs). T-Server distributes an `EventAgentLogin` for all the associated devices.

With value `true`, T-Server distributes `EventAgentLogin` only for the device from which the `RequestAgentLogin` was initiated.

Regardless of this option setting, on `RequestAgentLogout`, T-Server distributes `EventAgentLogout` for all associated devices.

acd-login

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies which authorization code T-Server passes to the Application Link after a login request. This code varies according to the presence or absence of an ACD on the switch.

Note: When set in the TServer section, this option defines the default value for all ODN-only extensions. However, you can also set a value for this option on the Annex tab of ODN-only extensions, in a section called TServer. When set there, this value overrides the default value for the specific extension. Configuring this value on a per-extension basis in this way replaces the pre-7.1 functionality provided by using switch-specific type 9.

password-delimiter

Default Value: %

Valid Value: Any character not a number

Changes Take Effect: Immediately

Specifies a character used to separate the password parameter when it is transmitted as part of either AgentID or Queue.

ext-routing-min-digits

Default Value: 25

Valid Value: Any integer from 0-25

Changes Take Effect: Immediately

Specifies the number of digits in the destination number according to which T-Server decides how to route the call. If the number of digits in the destination number is less than or equal to the value specified, T-Server routes the call internally (using `Diver tCall`). If the number of digits in the destination number is greater than the value specified, T-Server routes the call externally (using `TransferCall`).

Note: This option is not applicable for routing based on CTI groups (see “Using CTI Group as a Single Routing Point/ External Routing Point” on [page 140](#)).

ext-xfer-min-digits

Default Value: 25

Valid Value: Any integer from 0-25

Changes Take Place: Immediately

Specifies how T-Server performs transfers from CAS ports. If the destination number length is greater than, or equal to, the value defined in this option, T-Server uses two-step transfer from the CAS port instead of single-step transfer using `Def lect`.

Note: Supervised routing, when configured, takes precedence.

ext-redir-min-digits

Default Value: 0

Valid Value: Any integer from 0-25

Changes Take Place: Immediately

Specifies the number of digits in the destination number, according to which T-Server decides how to perform `RequestRedirectCall` to an unmonitored (unknown) destination. If the number of digits in the destination number is less than the value specified, T-Server will try to execute the `Deflect` service towards the PBX. If the number of digits in the destination number is equal to or greater than the value specified, T-Server will try to perform `Emulated Divert` (answering the call and using two-step transfer to the requested destination).

Default value is 0 for compatibility with future service extensions.

With the default value set, T-Server always performs an emulated redirect.

Note: This option is only applicable to devices from which you can answer transfer on the same line. Therefore it does not apply to ADNs, analog or CAS devices.

Warning! Should the redirect fail for any reason, the whole call will be cleared.

route-consult-call

Default Value: true

Valid Values: true, false

Changes Take Effect: Immediately

With value true, T-Server sends an `EventRouteRequest` to Interaction Router prior to transfer completion. This option is used in transfer-to-Routing-Point scenarios.

application-link

Default Value: No default value

Valid Value: Any valid name

Changes Take Effect: Immediately

Specifies the section name containing the Application Link control options for that link. You must specify a value for this option if the default name of the section is changed.

link-control

Default Value: link-control

Valid Value: Any valid name

Changes Take Effect: Immediately

Specifies the section name containing the link control options. You must specify a value for this option if the default name of the section is changed.

switch-domain-check

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether T-Server monitors the switch domain to determine the source of calls from unmonitored devices. With value `true`, T-Server monitors the switch domain and determines whether a call from an unmonitored device is an inbound call or an internal call. If DPNSS trunks are used, Genesys recommends setting this value to `false`.

analog-transfer

Default Value: 1

Valid Values: 1, 2

Changes Take Effect: Immediately

Defines the type of native transfer method T-Server uses for analog devices. Value 1 indicates that T-Server uses `TSingleStepTransfer` (switch service `Deflect`). Value 2 indicates that T-Server uses `TMuteTransfer` (switch service `Initiate/Complete`).

Note: Consult your switch vendor before setting this value.

force-snapshot-interval

Default Value: 1000

Valid Value: Any integer from 100–5000

Defines a timeout (in milliseconds) after which T-Server queries the switch for the status of devices after completion of transfers in which the primary call was on the ADN and the consult call was on the ODN.

login-code

Default Value: 73

Valid Value: Any valid Ericsson market code

Changes Take Effect: Immediately

This value must be set to the same value as the ACD Login/Logout code in the market-dependent feature of the switch configuration. If not, login does not operate. T-Server performs no validation on this code.

use-makecall-login

Default Value: `false`

Valid Values: true, false

Changes Take Place: Immediately

true T-Server uses MakeCall services to log agents in and out.

false T-Server uses CSTA services to log agents in and out.

Determines the method T-Server uses to log agents in and out.

block-failed-dn-tout

Default Value: 5000

Valid Values: Any integer from 0-10000

Defines the timeout (in milliseconds) after which T-Server blocks any requests for devices where error TERR_CSTA_UNSPECIFIED is received. Value 0 indicates no timeout, and therefore no blocking.

force-pbx-ag-state

Default Value: true

Valid Values: true, false

Changes Take Place: Immediately

true T-Server in some scenarios sends SetFeature requests unnecessarily, just to ensure that device is really in the requested state.

false T-Server relies only on information stored in T-Server (based on query reporting or real-time reporting).

Determines whether T-Server uses available workarounds for ODN synchronization problems and makes decisions to send SetFeature requests to the ODN, based on the T-Server's internal agent-state tracking.

redistr-dly

Default Value: 500

Valid Value: Any integer from 0-2000

Changes Take Place: Immediately

Specifies the delay (in milliseconds) applied by T-Server in order to ascertain the reason why an alerting call on an ADN was cleared (remote release or ACD Queue redistribution). With value 0 (zero), EventAbandoned is distributed for all scenarios. With a value greater than 0, T-Server distributes EventReleased if ACD Queue redistribution is detected during this delay.

Note: Configure this option with care as call volume can affect the time Application Link takes to distribute call information.

min-route-dly

Default Value: 250

Valid Value: Any integer from 0-5000

Changes Take Place: Immediately

Specifies the delay (in milliseconds) that T-Server applies to the processing of `RequestRouteCall`. T-Server applies this delay only for scenarios where the time difference between `EventRouteRequest` and `RequestRouteCall` is less than defined in this option. Genesys recommends values in the range 500-1000.

vto-onhook-delay

Default Value: 1000

Valid Value: Any integer from 0-10000

Changes Take Effect: Immediately

Specifies the delay (in milliseconds) that T-Server applies before sending `EventOnHook` after transferring a call. This option is valid for all devices configured in the Configuration Layer with switch-specific type 8. This delay allows the VTO enough time to receive the `EventReleased` before becoming available. Value 0 disables this function.

Note: When analog devices are used in conjunction with Voice Treatment Option (VTO) or Outbound Contact (OC), there is a delay between the CTI and the physical `Released` state, which means that after a CTI `OnHook` event, the physical port is still in use for a brief period. The Application Link generates `Service Initiated` and `Connection Cleared` with a new `CallID`, and this prompts T-Server to create and release a new call.

unknown-xfer-merge-udata

Default Value: false

Valid Values: true, false

Changes Take Place: Immediately

With value true, T-Server copies the user data from the current monitored call to the call transferred from an unmonitored destination. Because the primary call was hitherto unknown, normal user data inheritance mechanisms cannot be used.

inbound-bsns-calls

Default Value: false

Valid Values: true, false

Changes Take Effect: Immediately

Specifies whether T-Server should consider all established inbound calls on an emulated agent as business calls.

outbound-bsns-calls

Default Value: false

Valid Values: true, false

Changes Take Effect: Immediately

Specifies whether T-Server should consider all established outbound calls on a emulated agent as business calls after being established.

inherit-bsns-type

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Determines whether a consult call that is made from a business primary call should inherit the `business call` attribute.

internal-bsns-calls

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Determines whether T-Server considers internal calls made from or to any agent as business calls.

unknown-bsns-calls

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Determines whether T-Server considers calls of unknown call type made from or to any agent as business calls.

legal-guard-time

Default Value: `0`

Valid Value: Any integer from `0-10`

Changes Take Effect: Immediately

Specifies a legal-guard time (in seconds) for emulated agents to postpone the transition to the `Ready` state after a business call. T-Server always considers a routed call a business call.

timed-acw-in-idle

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether T-Server applies the automatic wrap-up timer (using the `wrap-up-time` parameter) when an agent sends `RequestAgentNotReady`. With value `false`, T-Server does not automatically end manual wrap-up—the agent must return manually from ACW.

acw-in-idle-force-ready

Default Value: true

Valid Values: true, false

Changes Take Effect: Immediately

Specifies whether, after timed manual wrap-up (when you have set option `timed-cwk-in-idle` to true), T-Server forces the agent to the Ready state. With value false, T-Server returns the agent to the state he or she was in prior to wrap-up.

emulate-login

Default Value: on-RP

Valid Values: true, false, on-RP

Changes Take Effect: Immediately

Specifies whether T-Server performs emulated agent login when the login device is configured in the Configuration Layer as a device of type `extension`.

true	T-Server performs an emulated login.
false	T-Server passes a login request to the PBX.
on-RP	T-Server checks the Agent Group associated with the login request. If the Agent Group is a standard Routing Point the emulated login request succeeds. This value can be set only at the global level, and is available for backwards compatibility.

This option can be set in a number of places, and T-Server processes it in the following order of precedence highest first. If the value is not present at the higher level, T-Server checks the next highest level, and so on.

1. In `RequestAgentLogin`, using attribute extension `EmulateLogin`.
2. In the Agent ID object, on the Annex tab.
3. In the login device object, on the Annex tab.
4. In the device representing an Agent Group object, on the Annex tab.
5. In the T-Server Application object, in the Tserver section.
6. In an Agent Group corresponding to an object which is configured in the Configuration Layer as a device of type `Routing Point`.

emulated-login-state

Default Value: ready

Valid Values: ready, not-ready

Changes Take Effect: Immediately

When T-Server performs an emulated agent login and the client specifies an agent work mode other than `ManualIn` or `AutoIn`, T-Server uses this option to determine which event to distribute.

not-ready	T-Server distributes <code>EventAgentNotReady</code> after <code>EventAgentLogin</code> .
-----------	---

ready T-Server distributes `EventAgentReady` after `EventAgentLogin`.

This option can be set in a number of places, and T-Server processes it in the following order of precedence, highest first. If the value is not present at the higher level, T-Server checks the next level, and so on.

1. In the Agent ID object, on the Annex tab.
2. In the agent login device, on the Annex tab.
3. In the login device representing an Agent Group during login, on the Annex tab.
4. In the T-Server Application object, in the Tserver section.

agent-strict-id

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether, for emulated agents, T-Server allows any AgentID to be used during login (value `false`), or only those configured in the Configuration Layer (value `true`).

sync-emu-agent

Default Value: `off`

Valid Values: `on`, `off`

Changes Take Effect: Immediately

Reserved for Genesys Engineering.

See “VoIP Absence Code Handling” on [page 168](#).

wrap-up-time

Default Value: 0

Valid Value: Any positive integer, untimed

Changes Take Effect: Immediately

Specifies the amount of wrap-up time (ACW) allocated to emulated agents at the end of a business call.

0	ACW is disabled Exception: When set on the Annex tab of the Agent ID object, value 0 (zero) means T-Server will process from Step 4 in the processing order of precedence below.
Value greater than 0 but less than untimed-wrap-up-value	The number of seconds of timed ACW, after which T-Server returns the agent to the Ready state.
Value equal to untimed-wrap-up-value	ACW is untimed and the agent must manually return to the Ready state.
Value greater than untimed-wrap-up-value	ACW is disabled.
untimed	ACW is untimed and the agent must manually return to the Ready state.

This option can be set in a number of places, and T-Server processes it in the following order of precedence, highest first. If the value is not present at the higher level, T-Server checks the next level, and so on.

1. In RequestAgentPendingACW, in attribute extension WrapUpTime (applies to this agent only).
2. In RequestACWInIdle, in attribute extension WrapUpTime (applies to this agent only).
3. In the call, in user data WrapUpTime (limited to ISCC scenarios).
4. In a configuration object of type ACD Queue or Routing Point, on the Annex tab.
5. In RequestAgentLogin, in attribute extension WrapUpTime (applies to this agent only).
6. In the Agent ID object, on the Annex tab.
7. In the login device object, on the Annex tab.
8. In the Agent Group corresponding to an object configured in the Configuration Layer as a device of type ACD Queue.
9. In the T-Server Application object.

untimed-wrap-up-value

Default Value: 1000

Valid Value: Any nonzero positive integer

Changes Take Effect: Immediately

Specifies the threshold at which the timing of ACW stops and manual intervention is required (*untimed ACW*).

wrap-up-threshold

Default Value: 0

Valid Values: Any positive integer

Changes Take Effect: Immediately

Specifies the minimum period (in seconds) that a business call must last before emulated ACW is applied at the end of the call.

backwds-compat-acw-behavior

Default Value: true

Valid Value: true, false

Changes Take Effect: Immediately

Specifies whether pre-7.5 behavior after-call work is enabled (value = true) or disabled (value = false), for backward compatibility.

With value true, if an agent receives or makes a business call while in emulated ACW, T-Server does the following:

1. Stops the ACW timer.
2. Forces the agent to the Ready state.
3. Restarts ACW (and the legal-guard timer) after the new business call is released.

If an agent makes or receives a work-related call while in ACW, T-Server does the following:

1. Suspends the ACW, but leaves the agent in the ACW state.
2. Resumes the ACW timer once the work-related call is released.

Note: A *work-related* call is a call made by an agent while in ACW, or a consult call where the main call is either a business call or a work-related call.

After the ACW and any configured legal-guard time have been completed, the agent is forced to the Ready state.

If an agent makes or receives a private call during ACW, no action is taken and the ACW timer keeps running.

With value false, pre-7.5 behavior is used. In this case, T-Server forces the agent to the Ready state after the after-call work and legal-guard timer have been applied.

override-switch-acw

Default Value: `false`

Valid Value: `true`, `false`

Changes Take Effect: Immediately

Reserved for Genesys Engineering.

agent-group

Default Value: `None`

Valid Value: Any agent group value

Changes Take Effect: At the next agent login session

Specifies a value for a virtual group that will be used for T-Server reporting.

T-Server obtains the value for this option in the following order of precedence:

1. In the `TServer` section of the `Annex` tab of the `AgentID` object.
2. In the `TServer` section of the `Annex` tab of the `DN` object.
3. In the main `TServer` section.

agent-no-answer-timeout

Default Value: `15`

Valid Value: Any integer from `0-600`

Changes Take Effect: Immediately

Defines the default time (in seconds) that T-Server allows for a logged-in agent (real or emulated) to answer a call before executing the actions defined in options `agent-no-answer-overflow` and `agent-no-answer-action`. Value `0` (zero) disables the Agent No-Answer Supervision feature. See also extension `NO_ANSWER_TIMEOUT`.

Notes: When you set a value for this option on the `Annex` tab of an `Agent ID` object in the Configuration Layer, that value overrides, for that agent, the value of this option set in the `TServer` section.

Because this T-Server supports supervised routing, the value defined for option `supervised-route-timeout` overrides the value defined for `agent-no-answer-timeout` for supervised routed calls. If a call is delivered to a device using supervised routing, and the routing timeout expires, T-Server does not apply the specified no-answer overflow. If the call is routed to an agent, T-Server does apply the specified no-answer action after the supervised-routing timeout expires.

agent-no-answer-overflow

Default Value: No default value

Valid Values: `none`, `recall`, `release`, any valid overflow destination, in a comma-separated list

<code>none</code>	T-Server does not attempt to overflow a call on an agent desktop when <code>agent-no-answer-timeout</code> expires. T-Server treats this value as the end of a list. Subsequent values are not executed.
<code>recall</code>	T-Server returns the call to the last distribution device (the device reported in the <code>ThisQueue</code> attribute of the call) when <code>agent-no-answer-timeout</code> expires.
<code>release</code>	T-Server releases the call.
Any valid overflow destination	T-Server returns the call to the specified destination when <code>agent-no-answer-timeout</code> expires.

Changes Take Effect: Immediately

Specifies a sequence of overflow destinations that T-Server attempts to overflow to when the time specified in option `agent-no-answer-timeout` expires. T-Server attempts to overflow in the order specified in the list. If all overflow attempts fail, T-Server abandons overflow. See also extension `NO_ANSWER_OVERFLOW`. If the list of overflow destinations contains the value `recall` and the call was not distributed, T-Server skips to the next destination in the list.

Note: When you set a value for option `no-answer-overflow` on the Annex tab of an Agent ID object in the Configuration Layer, that value overrides, for that agent, the value of `agent-no-answer-overflow` in the TServer section.

agent-no-answer-action

Default Value: `none`

Valid Values: `none`, `notready`, `logout`

<code>none</code>	T-Server takes no action on agents when calls are not answered.
<code>notready</code>	T-Server sets agents to <code>NotReady</code> when calls are not answered.
<code>logout</code>	T-Server automatically logs out agents when calls are not answered.

Changes Take Effect: Immediately

Defines T-Server's default action if a logged-in agent (real or emulated) fails to answer a call within the time defined in `agent-no-answer-timeout`. See also extension `NO_ANSWER_ACTION`.

Note: When you set a value for option `no-answer-action` on the Annex tab of an Agent ID object in the Configuration Layer, that value overrides, for that agent, the value of `agent-no-answer-action` in the TServer section.

extn-no-answer-timeout

Default Value: 15

Valid Value: Any integer from 0-600

Changes Take Effect: Immediately

Defines the default no-answer timeout (in seconds) that T-Server applies to any device of type `extension`. When the timeout ends, T-Server executes the actions defined in option `extn-no-answer-overflow`.

Value 0 (zero) deactivates no-answer supervision for devices of type `extension`. See also extension `NO_ANSWER_TIMEOUT`.

Note: When you set a value for option `no-answer-timeout` on the Annex tab of an `Extension` object in the Configuration Layer, that value overrides, for that extension, the value of `extn-no-answer-timeout` in the TServer section.

posn-no-answer-timeout

Default Value: 15

Valid Value: Any integer from 0-600

Changes Take Effect: Immediately

Defines the default no-answer timeout (in seconds) that T-Server applies to any device of type `position`. When the timeout ends, T-Server executes the actions defined in option `posn-no-answer-overflow`.

Value 0 (zero) deactivates no-answer supervision for devices of type `position`. See also extension `NO_ANSWER_TIMEOUT`.

Note: When you set a value for option `no-answer-timeout` on the Annex tab of a `Position` object in the Configuration Layer, that value overrides, for that position, the value of `posn-no-answer-timeout` in the TServer section.

extn-no-answer-overflow

Default Value: No default value

Valid Values: `none`, `recall`, `release`, any valid overflow destination, in a comma-separated list

<code>none</code>	T-Server does not attempt to overflow a call on an extension when <code>extn-no-answer-timeout</code> expires. T-Server treats this value as the end of a list. Subsequent values are not executed.
<code>recall</code>	T-Server returns the call to the last distribution device (the device reported in the <code>ThisQueue</code> attribute of the call) when <code>extn-no-answer-timeout</code> expires.

<code>release</code>	T-Server releases the call.
Any valid overflow destination	T-Server returns the call to the specified destination when <code>extn-no-answer-timeout</code> expires.

Changes Take Effect: Immediately

Specifies a sequence of overflow destinations that T-Server attempts to overflow to when the time specified in option `extn-no-answer-timeout` has expired.

T-Server attempts to overflow in the order specified in the list. If all overflow attempts fail, T-Server abandons overflow. See also extension `NO_ANSWER_OVERFLOW`.

If the list of overflow destinations contains the value `recall`, and the call was not distributed, T-Server skips to the next destination in the list.

Note: If you set a value for option `no-answer-overflow` on the Annex tab of any `Extension` object in the Configuration Layer, that value overrides, for that extension, the value of `extn-no-answer-overflow` in the `TServer` section.

posn-no-answer-overflow

Default Value: No default value.

Valid Values: `none`, `recall`, `release`, any valid overflow destination, in a comma-separated list

<code>none</code>	T-Server does not attempt to overflow a call on a position when <code>posn-no-answer-timeout</code> expires. T-Server treats this value as the end of a list. Subsequent values are not executed.
<code>recall</code>	T-Server returns the call to the last distribution device (the device reported in the <code>ThisQueue</code> attribute of the call) when <code>posn-no-answer-timeout</code> expires.
<code>release</code>	T-Server releases the call..
Any valid overflow destination	T-Server returns the call to the specified destination when <code>posn-no-answer-timeout</code> expires.

Changes Take Effect: Immediately

Specifies a sequence of overflow destinations that T-Server attempts to overflow to when the time specified in option `posn-no-answer-timeout` expires. T-Server attempts to overflow in the order specified in the list. If all overflow attempts fail, T-Server abandons overflow. See also extension `NO_ANSWER_OVERFLOW`.

If the list of overflow destinations contains the value `recall` and the call was not distributed, T-Server skips to the next destination in the list.

Note: If you set a value for option `no-answer-overflow` on the Annex tab of any Position object in Configuration Layer, that value overrides, for that position, the value of `posn-no-answer-overflow` in the TServer section.

nas-private

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Place: Immediately

Specifies whether No-Answer Supervision is enabled for private calls. When configured in the TServer section, this value is the default value applied globally to all private calls.

Note: When set in the TServer section, this option defines the default value for all private calls. However, you can also set a value for this option on the Annex tab of DN's of type `extension` or `Agent ID`, in a section called TServer. When set there, this value overrides the default value for the specific DN.

recall-no-answer-timeout

Default Value: 15

Valid Value: Any integer from 0-600

Changes Take Place: Immediately

Defines the time that T-Server waits for a call to reappear on a device as a result of a recall (for example, a ringback waiting to be answered). When the timer expires, T-Server executes the actions defined by the relevant overflow option, as well as the action option for cases where an agent is logged in.

With value 0, there is no No-Answer Supervision for such calls.

This option can be defined either in the main Tserver section or in a section called Tserver on the Annex tab of any of the following configuration object types in Configuration Manager:

- Extension
- Position
- VT Port
- AgentID

nas-indication

Default Value: `none`

Valid Values: `none`, `ext`, `rsn`

Changes Take Effect: Immediately

Specifies the reporting action in EventReleased when No-Answer Supervision overflows a call.

With value `none`, no reason or extension is provided in `EventReleased`.

With value `ext`, extension `NO_ANSWER_TIMEOUT` is supplied in `EventReleased`.

With value `rsn`, reason `NO_ANSWER_TIMEOUT` is supplied in `EventReleased`.

prd-dist-call-ans-time

Default Value: 0

Valid Value: Any integer from 0-10

Changes Take Effect: Immediately

Specifies the interval (in seconds) during which an agent can answer a predictive call before T-Server abandons it. With value 0 (zero), T-Server does not automatically abandon the call, which then rings on the agent desktop until it is answered.

When an emulated predictive dial is made from an emulated Routing Point, and options `prd-dist-call-ans-time` and `supervised-route-timeout` are set, the value in `prd-dist-call-ans-time` takes precedence. For predictive dialing to work, you must set values greater than 0 (zero) for both options.

Note: When set in the TServer section, this option defines the default value to be applied for all agents. However, this option can also be set on the Annex tab of any queue or Routing Point that is to be used as the origination device for a predictive call, in a section called TServer. When set there, this value overrides the value of the global T-Server option for all calls that are originated from that queue or Routing Point.

max-pred-req-delay

Default Value: 3

Valid Value: Any integer from 0-10

Changes Take Effect: Immediately

Defines the maximum time (in seconds) that T-Server waits for a free dialing resource to become available before rejecting a `TMakePredictiveCall` request.

supervised-route-timeout

Default Value: 5

Valid Value: Any integer from 0-600

Changes Take Effect: Immediately

Specifies the interval (in seconds) that T-Server waits for a call routed from an emulated Routing Point using supervised routing to be answered. If the call is not answered within the period specified, T-Server recalls the call to the Routing Point and initiates rerouting. Value 0 (zero) deactivates this feature. See also `agent-no-answer-timeout`. For predictive dialing to work, you must set values greater than 0 (zero) for both this option and `prd-dist-call-ans-time`.

This timeout should be set to a value higher than the system latency.

Notes: You can use extension `SUPERVISED_ROUTE` to override the value of this configuration option on a call-by-call basis.

When set in the `TServer` section, this option defines the default value for all Routing Points. However, you can also set a value for this option on the `Annex` tab of DNs of type `Routing Point` in a section called `TServer`. When set there, this value overrides the default value for the specific Routing Point. You can also use `Extension` attribute `SUPERVISED_ROUTE` to override the value of this configuration option on a call-by-call basis.

consult-supervised-rt

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether T-Server allows supervised routing of consultation calls. With value `false`, T-Server forces nonsupervised routing for consultation calls, regardless of configuration option or call-by-call settings.

Note: When set in the `TServer` section, this option defines the default value for all Routing Points. However, you can also set a value for this option on the `Annex` tab of DNs of type `Routing Point` in a section called `TServer`. When set there, this value overrides the default value for the specific Routing Point.

route-request-attempt

Default Value: `0`

Valid Value: Any integer from `0-10`

Changes Take Effect: Immediately.

Specifies the number of times T-Server attempts to send route requests to Interaction Router if the first attempt is unsuccessful. For calls waiting to be routed, T-Server keeps the previous value. Please refer to the Genesys Universal Routing Server documentation for further details.

correct-connid

Default Value: `true`

Valid Value: `true`, `false`

Changes Take Effect: Immediately

With value `true`, T-Server corrects wrong connection IDs provided by the application in CTI requests. Value `false` disables this feature.

correct-rqid

Default Value: `true`

Valid Value: `true`, `false`

Changes Take Effect: Immediately

With value `true`, T-Server corrects the ID of CTI requests provided by the application. Value `false` disables this feature.

callback-dn

Default Value: `CallbackDN`

Valid Value: Any string that does not correspond to an existing internal device

Changes Take Effect: Immediately

Defines the value of the third-party DN used in reporting the switch CallBack scenario as an emulated single-step transfer.

Note: Genesys recommends that Callback DNs should not be used in dialing plans, because call flow and T-Server behavior become unpredictable if they are.

retain-call-tout

Default Value: 1

Valid Value: Any integer from 0-3600

Changes Take Effect: Immediately

Specifies the interval (in seconds) that T-Server waits before deleting information about calls that are completed, but for which it has received no notification from the switch.

expire-call-tout (removed)

Default Value: 60

Valid Value: Any integer from 0-1440

Changes Take Effect: Immediately

Specifies the interval (in minutes) that T-Server waits before considering calls *expired* (that is, for which it has received no event from the switch) and deleting them.

convert-otherdn

Default Value: `+agentid +reserveddn +fwd`

Valid Values: `+/-agentid +/-reserveddn, +/-fwd`

Changes Take Effect: Immediately

Defines whether T-Server has to convert (if applicable) the value provided in request's `AttributeOtherDN`.

Value `+/-agentid` turns on/off either the conversion of the Agent ID value provided in the `OtherDN` attribute to the DN associated with this Agent, or the DN value to Agent ID value (where appropriate).

Value `+/-reserveddn` turns on/off the conversion of `OtherDN` for reserved DNs.

Value `+/-fwd` turns on/off conversion of `OtherDN` in request `TSetCallForward`.

dn-for-undesired-calls

Default Value: No default value

Valid Values: Any valid switch DN

Changes Take Effect: Immediately

Specifies the DN that T-Server uses as the request destination if the client provides a reserved DN in the request.

Note: You can set a value for this option on the Annex tab of appropriate DN's in a section called `tServer`. When set there, this value overrides the default value for the DN.

accept-dn-type

Default Value: `+extension +position +acdqueue +routedn`

Valid Values: `+/-extension +/-position +/-acdqueue +/-routedn`

Changes Take Effect: Immediately

Defines the supported set of device types that are not configured in Configuration Layer but that T-Server can register.

`+/-extension` Accepts or rejects registration on DN of type `extension` (`AddressTypeDN`)

`+/-position` Accepts or rejects registration on DN of type `position` (`AddressTypePosition`)

`+/-acdqueue` Accepts or rejects registration on DN of type `ACD Queue` (`AddressTypeQueue`)

`+/-routedn` Accepts or rejects registration on DN of type `Routing Point` (`AddressTypeRouteDN`)

default-dn-type

Default Value: `none`

Valid Values: `none, extension, position, acdqueue, routedn`

Changes Take Effect: Immediately

Defines the value that T-Server applies for `AttributeAddressType` when the client does not provide that attribute or provides value `AddressTypeUnknown`.

`none` T-Server assigns DN type using PBX-provided information

`extension` T-Server uses value `AddressTypeDN`

`position` T-Server uses value `AddressTypePosition`

`acdqueue` T-Server uses value `AddressTypeQueue`

`routedn` T-Server uses value `AddressTypeRouteDN`

dn-del-mode

Default Value: `idle`

Valid Values: `never, idle, force`

Changes Take Effect: Immediately

Defines how T-Server handles device and device-related information when the DN is not configured in the Configuration Layer and there are no clients registered on that DN.

<code>never</code>	T-Server does not unregister the DN with the PBX and device related information is never deleted from T-Server memory.
<code>idle</code>	T-Server unregisters the DN with the PBX and device-related information is deleted from T-Server memory as soon as there are no more calls on this device.
<code>force</code>	T-Server unregisters DN with the PBX and device-related information is deleted from T-Server memory regardless of the calls that existed on that DN.

Note: Timeout Value Format—T-Server applies a defined delay before unregistering the DN after the last call has left that DN. Value `idle` is equivalent to setting the Timeout Value to 0 (zero).

clid-withheld-name

Default Value: `PRIVATE`

Valid Values: Any string

Changes Take Effect: Immediately

Defines a name that replaces a withheld CLID. If no value is entered (empty string) the withheld CLID will be displayed.

login-pincode

Default Value: `1111`

Valid Values: `global`, `DN`, `Agent ID`

Changes Take Effect: Immediately

Specifies the value that is passed to the PBX during ACD login in the `<PINCODE>` variable. This feature allows:

- All parameters passed to T-Server to be ignored
- The `<AUTHCODE>` part of the login to default to the DN of the place the agent is logging into
- The `<PINCODE>` part of the login string to default to a value defined in this option.

<code>global</code>	T-Server passes an arbitrary string (for example, <code>1111</code> , or <code>1234</code>) in the <code><PINCODE></code> part of the login.
<code>DN</code>	T-Server passes the DN number in the <code><PINCODE></code> part of the login.
<code>Agent ID</code>	T-Server passes the Agent ID in the <code><PINCODE></code> part of the login.

enable-retain-in-queue

Default Value: `false`

Valid Values: `false`, `true`

Changes Take Effect: On arrival of a new call at the Routing Point

Specifies whether the Maintain Position in Queue feature is enabled (value = `true`) or disabled (value = `false`).

call-retain-in-queue

Default Value: `disabled`

Valid Values: `disabled`, `keep`, `divert`

Changes Take Effect: On arrival of a new call at the Routing Point

Values `keep` and `divert` instruct T-Server to apply the Maintain Position in Queue feature when the destination for routing is configured as a device with treatment (Voice Treatment Port) capabilities.

With value `keep`, T-Server follows PBX reporting and maintains the call queued on the Routing Point when the call is placed on a Treatment device.

With value `divert`, T-Server issues `EventDiverted` and `EventRouteUsed` when the call is routed with the Maintain Position in Queue feature enabled.

With value `disabled`, the feature is switched off.

queue-return-dly

Default Value: `1000`

Valid Values: `0-5000`

Changes Take Place: Immediately

Defines how long (in milliseconds) T-Server waits for a call to be returned to the queue where position in queue is being maintained. If the PBX does not report delivery within this interval (for example, because Call ID was changed), T-Server initiates call cleanup processing. See “retain-call-tout” on [page 262](#).

This option is provided to enable a workaround to scenarios in which VTO ports can become locked.

reason-code

Default Value: `2`

Valid Values: Any integer from `0-9`

Changes Take Effect: Immediately

Equates to the reason why the agent is in a `notready` state. This value is passed to the switch in `TAgentLogin (ManualIn)` or `TAgentSetNotReady`.

The option can be set in the following places in order of precedence, from the highest to the lowest:

1. In extension `ReasonCode`, in requests `TAgentLogin (ManualIn)` and `TAgentSetNotReady` on a VoIP DN.

2. In the TServer section on the Annex tab of an Agent ID object that will be used on a VoIP DN.
3. In the TServer section on the Annex tab of an Extension object corresponding to a VoIP DN.
4. In the TServer section of the application.

Reason-code-end Section

This section must be called `reason-code-end`.

absence-0 ... absence-9

Default Value: None

Valid Value: Any integer

Changes Take Effect: Immediately

Each `absence-n` option enables users to define a default number of minutes or days for the corresponding reason code.

<code>absence-0 ...</code>	Absence period expressed in minutes
<code>absence-3</code>	
<code>absence-4 ...</code>	Absence period expressed in days
<code>absence-9</code>	

Switch-Specific Type Section

This section must be called `SwitchSpecificType`.

extension

Default Value: 0

Valid Value: Switch-specific types for DN of type `Extension` supported by T-Server

Changes Take Effect: Immediately

Defines the switch-specific type that T-Server uses for registration of DN's of type `extension` (`AddressTypeDN`) that are not configured in Configuration Layer.

Annex Tab Options

You can set the following configuration options in the TServer section of the Annex tab of the relevant object.

no-answer-timeout

Default Value: Same as value in corresponding global option

Valid Value: Any integer from 0-600

Changes Take Effect: Immediately

Defines the time (in seconds) that T-Server waits for a call that is ringing on the device in question to be answered. When the timer expires, T-Server applies the appropriate overflow, and, in the case of agents, the appropriate logout or NotReady action.

Value 0 (zero) deactivates no-answer supervision for this device.

When set, this option overrides any of the following global T-Server configuration options for the object where it has been set (depending on type of configuration object):

- agent-no-answer-timeout if defined for an Agent ID object
- extn-no-answer-timeout if defined for an Extension object
- posn-no-answer-timeout if defined for a Position object

Note: You can only set this configuration option in the TServer section of the Annex tab of the relevant configuration object in the Configuration Layer. You cannot define it in the main TServer configuration section.

no-answer-overflow

Default Value: No default value

Valid Values: none, recall, release, default, any valid overflow destination, in a comma-separated list

none	T-Server does not attempt to overflow a call on an agent desktop when agent-no-answer-timeout expires.
recall	T-Server returns the call to the last distribution device (the device reported in the ThisQueue attribute of the call) when agent-no-answer-timeout expires.
release	T-Server releases the call.
default	T-Server stops execution of the current overflow sequence and continues with the T-Server default overflow sequence defined by the relevant overflow option in the main TServer section.
Any valid overflow destination	T-Server returns the call to the specified destination when agent-no-answer-timeout expires.

Changes Take Effect: Immediately

Overrides any of the following global T-Server configuration options for the object where it has been set (depending on the type of configuration object):

- agent-no-answer-overflow if defined for an Agent ID object
- extn-no-answer-overflow if defined for an Extension object
- posn-no-answer-overflow if defined for a Position object

T-Server attempts to apply the overflow in the order that is listed. If the first overflow destination fails, then T-Server attempts the next one in the list. If all overflow destinations in the list fail, then T-Server abandons overflow. If the list of overflow destinations contains the value `recall`, and the call was not distributed, T-Server skips to the next destination in the list.

Note: You can set this configuration option only in the `TServer` section of the Annex tab of the relevant configuration object in the Configuration Layer. You cannot define it in the main `TServer` configuration section.

no-answer-action

Default Value: `none`

Valid Values: `none`, `notready`, `logout`

`none` T-Server takes no action on agents when business calls are not answered.

`notready` T-Server sets agents to `NotReady` when business calls are not answered.

`logout` T-Server automatically logs out agents when business calls are not answered.

Changes Take Effect: Immediately

Overrides the global T-Server configuration option `agent-no-answer-action` for that agent.

This option is defined in a section called `TServer` on the Annex tab of any Agent ID object in the Configuration Layer. If an emulated or real PABX agent receives a T-Server business call and the agent fails to answer the call within the time defined in option `agent-no-answer-timeout`, the `no-answer-action` option determines the action T-Server performs on this agent.

If a call is abandoned before either `agent-no-answer-timeout` or `no-answer-timeout` or `supervised-route-timeout` expires (depending on which timer is applicable), T-Server performs no action on this agent.

Note: You can set this configuration option only in the `TServer` section of the Annex tab of the relevant configuration object in the Configuration Layer. You cannot define it in the main `TServer` configuration section.

acd-login

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies which authorization code T-Server passes to the Application Link after a login request.

When you set this option in the Annex tab of the appropriate extension, the extension behaves the same way as for switch-specific type 9 in pre-7.1 releases of T-Server.

nas-private

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Place: Immediately

Specifies whether No-Answer Supervision is enabled for private calls. When configured in the T-Server section, this value is the default value applied globally to all private calls.

Note: When set in the TServer section, this option defines the default value for all private calls. However, you can also set a value for this option on the Annex tab of DN's of type extension or Agent ID, in a section called TServer. When set there, this value overrides the default value for the specific DN.

Application-Link Section

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

max-outstanding-login-rq

Default Value: 4

Valid Value: Any integer from 1-1000

Determines the maximum number of login requests that T-Server processes simultaneously for devices configured with switch-specific type 10 (for Smart Monitoring). Once the maximum number of outstanding login requests has been reached, T-Server rejects any subsequent login attempt. T-Server verifies this option setting and sets the value to no more than half of the overall outstanding requests per host (defined by configuration option `max-outstanding-rq`).

Note: This feature is not HA-compatible. Any queued login requests are lost when T-Server roles are changed.

max-outstanding-dev-rq

Default Value: 1

Valid Value: 1

Changes Take Effect: Immediately

Specifies the maximum number of requests that can be sent to the switch on behalf of a phone set before receiving any confirmations or rejections. Any additional requests are queued.

Note: The Ericsson MD110 switch supports only one request per device.

app-link-version

Default Value: 3.0.200

Valid Values: See description.

Changes Take Effect: Immediately

Specifies the version of the Ericsson Application Link currently in use. It should be written in the format:

- 3.0.XYY for BC9
- 4.0.X[X]YY for BC10/11/12

where X[X] is the Service Pack and YY is the Hot Fix level. For example:

- AL4 SP8 HF4 = 4.0.804
- AL4 SP10 HF3 = 4.0.1003

Note: This option is used to enable T-Server to work with the different fixes and features that have been updated through the years with the Application Link. Genesys cannot guarantee T-Server will operate correctly if this option is not set correctly.

Link-Control Section

The section name must be `link-control`.

hostname

Default Value: Mandatory field. No default value.

Valid Value: 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.

port

Default Value: Mandatory field. No default value.

Valid Value: 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.

call-rq-gap

Default Value: 250

Valid Value: Any integer from 0-1000

Changes Take Place: Immediately

Specifies (in milliseconds) the length of delay applied to a request issued against a busy call (a call that has another request working on it already). This prevents race conditions on the different call legs.

Set the value of this option to a time longer than the usual response time for a request from the switch.

rq-gap

Default Value: 0

Valid Value: Any integer from 0-1000

Changes Take Effect: Immediately

Specifies the minimum interval (in milliseconds) between succeeding CTI requests sent over the link. You can adjust the value to meet CTI-link load and performance requirements.

rq-expire-tout

Default Value: 10000

Valid Value: Any integer from 0-30000

Changes Take Effect: Immediately

Specifies the interval (in milliseconds) that T-Server waits before deleting pending requests (requests for which it has received no notification from the switch) from clients.

This timeout should be set to a value higher than the system latency.

reg-interval

Default Value: 60

Valid Values: Any integer from 0-600

Changes Take Effect: Immediately

Specifies the time interval (in seconds) for the Start Monitor request to be re-sent to the switch if the initial request fails. Value 0 (zero) switches this feature off.

kpl-interval

Default Value: 10

Valid Value: Any integer from 0-600

Changes Take Effect: Immediately

Specifies a “keep-alive” interval (in seconds). To check network connectivity, T-Server issues a dummy CTI request at the interval specified when there is no other activity on the link. Value 0 (zero) disables this feature.

The value of this option may need to be increased to avoid false restarts if the switch is sometimes slow to respond, for example, during busy periods.

See also option `kpl-tolerance`.

kpl-tolerance

Default Value: 3

Valid Value: Any integer from 0-10

Changes Take Effect: Immediately

Specifies the threshold number of accumulated KPL request failures. When the threshold is reached T-Server may either consider the CTI link:

- To be lost—in which case T-Server tries to reconnect to it.
- To be unstable—in which case T-Server issues a warning message.

kpl-loss-rate

Default Value: 10, 100

Valid Values: Single integer or comma-separated pair of integers. The first value in the pair is the failure value.

Changes Take Effect: Immediately

Specifies how many KPL positive responses are needed to decrement either the failure or warning tolerance counter.

Value 0 (zero) disables this option.

Two comma-separated values means T-Server will calculate both the failure counter and the warning counter.

A single value means T-Server will calculate only the failure counter.

Note: This option has no effect if option `kpl-tolerance` has value 0. In that case, a single KPL failure will trigger a link restart.

See “Keep-Alive Feature” on [page 163](#) for more details.

max-outstanding

Default Value: 16

Valid Value: Any integer from 1-100

Changes Take Effect: Immediately

Specifies the maximum number of outstanding sent requests awaiting the response from the link.

call-max-outstanding

Default Value: 1

Valid Value: Any integer

Changes Take Effect: Immediately

Specifies the maximum number of requests that can be outstanding against a call at any time.

Note: Genesys does not recommend changing this option as switch/link behavior cannot be predicted.

ha-sync-dly-lnk-conn

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: At T-Server start/restart

Determines whether the backup T-Server delays sending of `EventLinkConnected` until it has been notified that T-Server synchronization has completed. With value `true`, the backup T-Server sends `EventLinkConnected` once it has completed switch synchronization (that is, after all calls are cleared in the primary T-Server). With value `false`, there is no delay in sending `EventLinkConnected`, and synchronization takes place as for pre-7.1 T-Servers.

restart-period

Default Value: `20`

Valid Values: `0-600`

Changes Take Effect: Immediately

Specifies the interval (in seconds) that T-Server waits between attempts to reconnect to the switch when the link fails. Value `0` (zero) means T-Server does not try to reconnect unless the link configuration is changed.

restart-cleanup-limit

Default Value: `0`

Valid Values: Any integer

Changes Take Effect: Immediately

Defines the maximum number of reconnect attempts for calls (and possibly agent logins) in T-Server during link outage. Value `0` zero means all the calls are deleted immediately after the link failure. See also option `restart-period`.

quiet-cleanup

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Disables the events that T-Server would otherwise send to clients during clean-up to notify them about the deleted calls. With value `true`, the T-Server clients are supposed to drop all the calls upon `EventLinkDisconnected` without waiting for T-Server notification. See also the option `restart-cleanup-limit`.

quiet-startup

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Disables the events that T-Server would otherwise send to clients during link start-up to notify clients about the changes that occurred during the link outage. With value `true`, clients should query the T-Server after the `EventLinkConnected`.

restart-cleanup-dly

Default Value: `0`

Valid Values: Any integer

Changes Take Effect: Immediately

Specifies the delay (in seconds) for T-Server to keep “unreliable” calls after link startup. This delay allows T-Server to salvage calls that existed before the link failure (for which any events were received) if T-Server was unable to verify their existence using snapshot. Value `0` (zero) means any nonverified calls are cleared up immediately after completion of link startup.

reg-delay

Default Value: `1000`

Valid Values: `0–5000`

Changes Take Effect: Immediately

Defines the time (in milliseconds) that T-Server waits for the `DN Created` notification from Configuration Server before it starts processing the registration request from the client as a request for a DN not configured in the Configuration Layer.

reg-silent

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

With value `true`, T-Server reports `EventRegistered` for “on-demand” registration with the PBX when the procedure is completed.

With value `false`, T-Server reports `EventRegistered` as early as possible during the PBX registration procedure.

Note: Reserved for Genesys Engineering.

Changes from 7.5 to 7.6

Table 28 lists the configuration options that:

- Are new or changed in the 7.6 release of T-Server.
- Have been added or changed since the most recent 7.5 release of this document.

If a configuration option has been replaced with another that enables the same functionality, the new option name and location in this chapter are noted..

Table 28: Changes from 7.5 to 7.6

Option Name	Details
T-Server Section	
expire-call-tout	Removed in 7.6. See description on page 262 .
clid-withheld-name	Introduced in 7.6. See description on page 264 .
queue-return-dly	Introduced in 7.6. See description on page 265 .
wrap-up-threshold	Introduced in 7.6. See description on page 254 .
nas-indication	Introduced in 7.6. See description on page 259 .
Link-Control Section	
kpl-loss-rate	Introduced in 7.6. See description on page 272 .



Chapter

11

High Availability (HA)

This chapter presents switch-specific High Availability (HA) information for the T-Server for Ericsson MD110. This chapter contains the following section:

- [Genesys HA with Single Application Link, page 277](#)

Genesys HA with Single Application Link

The MD110 switch only allows devices to be monitored by CTI applications across one Application Link at a time. This means that you can only configure a Genesys HA environment for a single Application Link.

Hot and warm standby are both supported.

Hot-Standby Mode

Figure 12 shows the architecture of Hot Standby HA mode.

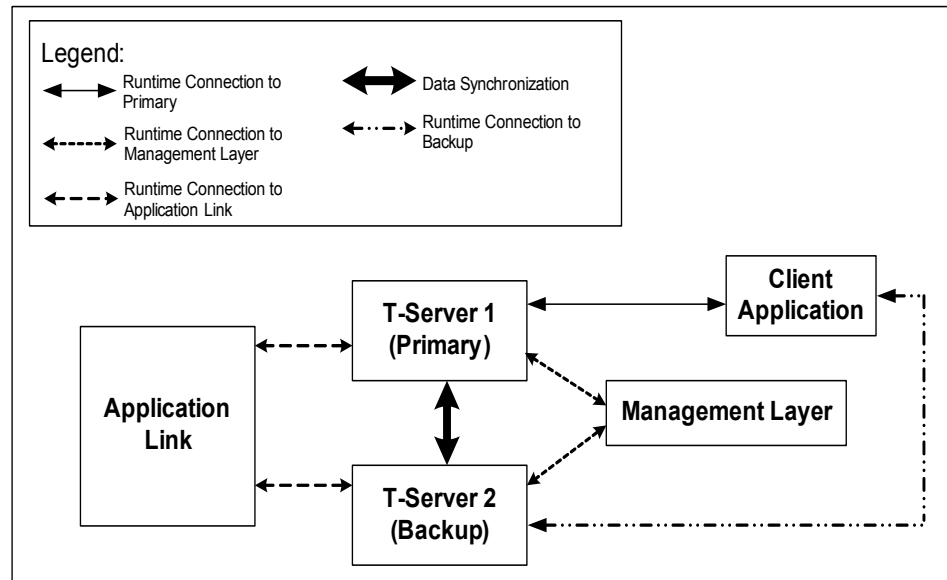


Figure 12: MD110 Hot-Standby Mode Architecture

Warm-Standby Mode

Figure 13 shows the architecture of Warm Standby HA mode.

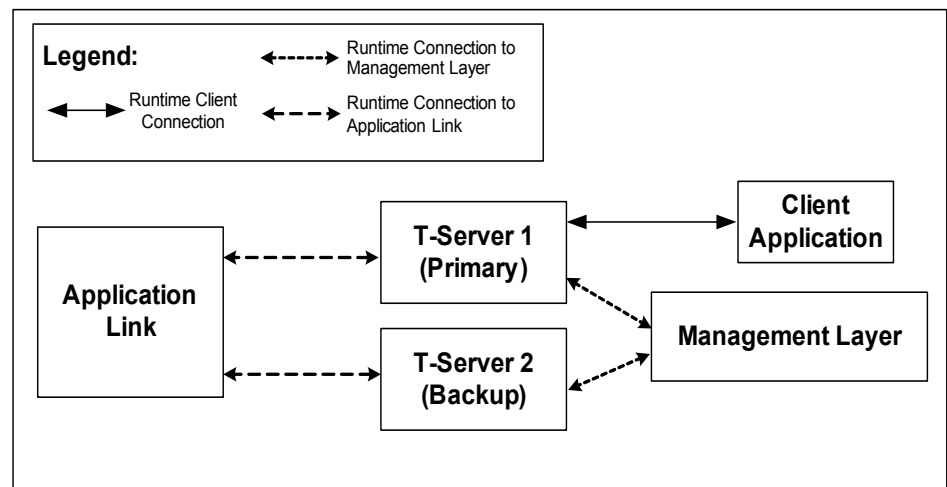


Figure 13: MD110 Warm-Standby Mode Architecture



Chapter

12 Ericsson MD110 Call Flows

This chapter presents switch-specific call-model information for Ericsson MD110 scenarios for which a greater level of detail may be required than is supplied in the *Genesys 7 Events and Models Reference Manual* and *Voice Platform SDK 7.6 .NET (or Java) API Reference*. It contains the following section:

- [ADN-ODN Conference Call Flow, page 279](#)

ADN-ODN Conference Call Flow

This section describes T-Server reporting for conference completion on the Ericsson MD110.

Description

Conference completion is invoked either via the Conference CSTA service or manually on the phone. For all the following scenarios, conference completion is performed on DN2-1 and DN2-2, which represent different lines on the same Ericsson phone (ADN-ODN or ADN-ADN).

The Ericsson switch may apply restrictions to the completion of the service. However, those restrictions are not in the scope of this document. In any case, restrictions do not affecting reporting.

All scenarios are equally applicable for service completion achieved either manually or via CTI.

Service Completion After Alternating the Call

Figure 14 on [page 280](#) shows service completion after alternating a call.

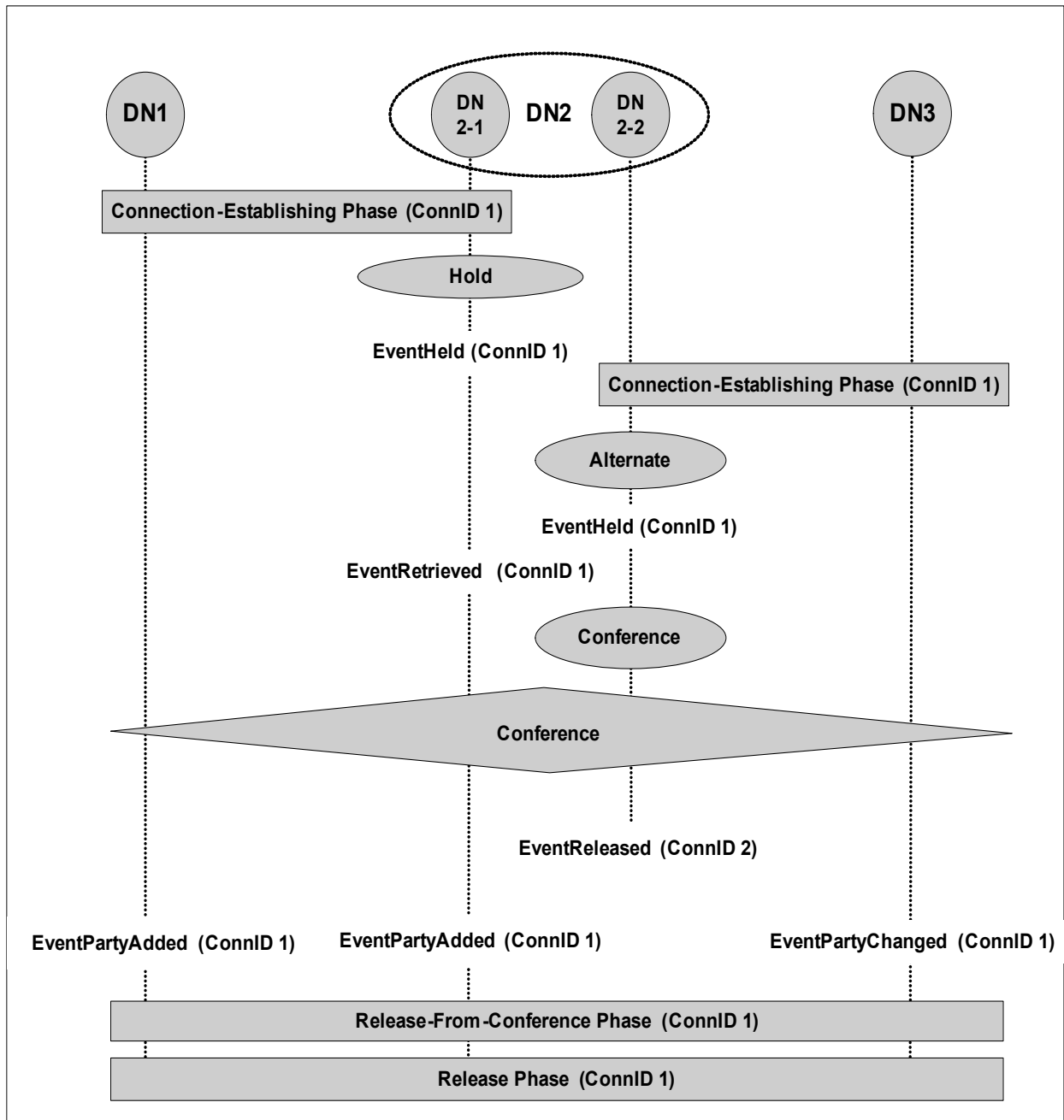


Figure 14: Service Completion After Alternating the Call

Table 29: Call Flow Before Conference Completion - With Alternate

DN1	DN2-1	DN2-2	DN3
Scenarios Before Conference Completion			
		Conference	
		EventReleased	
		ConnId Conn2	
		ThisDN DN2-2	
		OtherDN DN3	
		CallState Conferenced	
EventPartyAdded	EventPartyAdded		EventPartyChanged
ThisDN DN1	ThisDN DN2-1		ThisDN DN3
OtherDN DN3	OtherDN DN3		PreviousConnId Conn2
OtherDNRole NewParty	OtherDNRole NewParty		ThirdPartyDN DN2-1
ThirdPartyDN DN2-1	ThirdPartyDN DN2-1		ThirdPartyRole ConferencedBy
ThirdPartyRole AddedBy	ThirdPartyRole AddedBy		CallState Conferenced
CallState Conferenced	CallState Conferenced		
Release from Conference Phase			

Service Completion Without Alternating the Call

Figure 15 shows service completion without alternating the call.

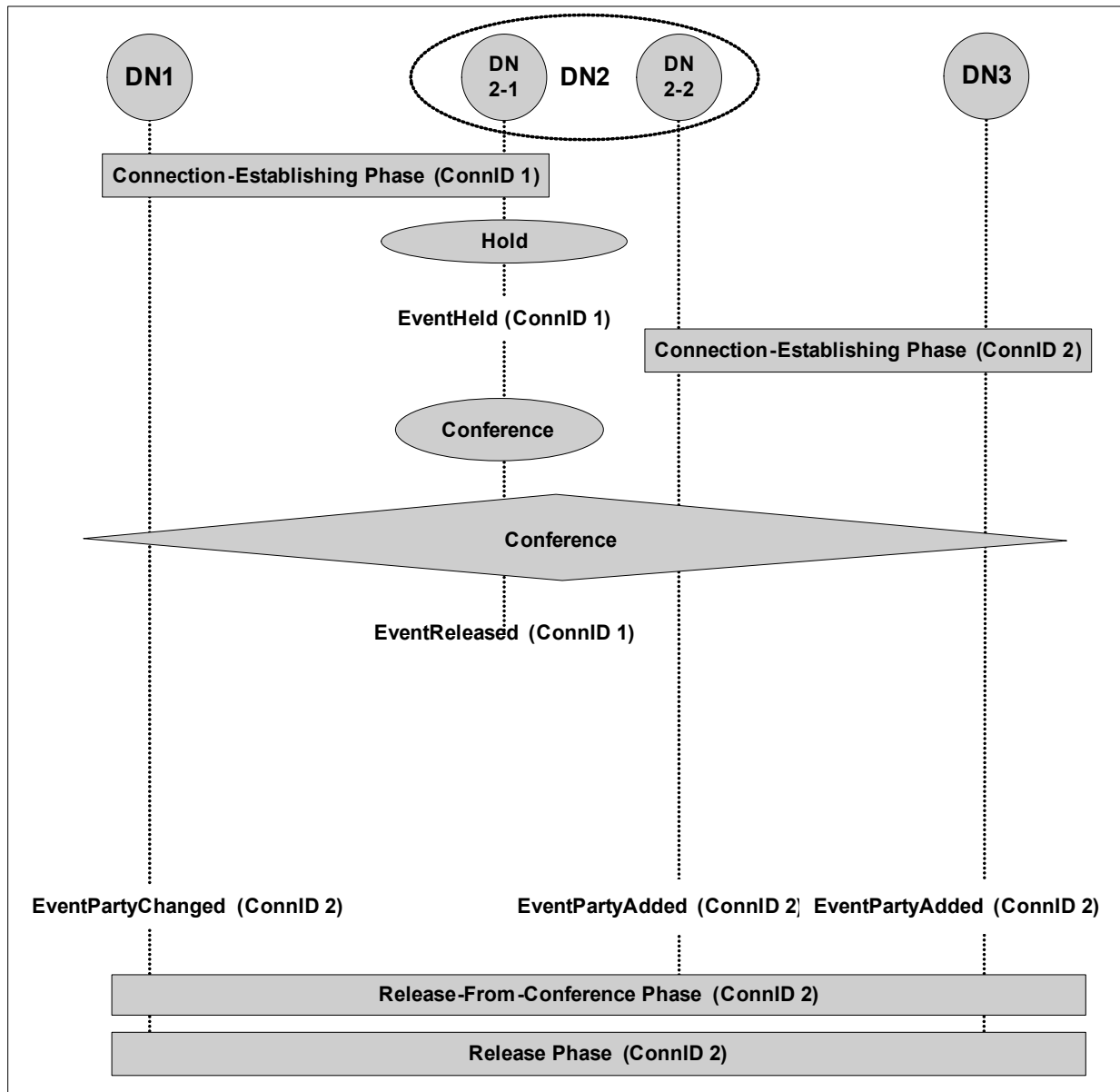


Figure 15: Service Completion Without Alternating the Call

Table 30: Call Flow Before Conference Completion - Without Alternate

DN1	DN21	DN22	DN3
Scenarios Before Conference Completion			
	Conference		
	EventReleased		
	ConnId Conn2		
	ThisDN DN21		
	OtherDN DN1		
	CallState Conferenced		
EventPartyChanged		EventPartyAdded	EventPartyAdded
ThisDN DN1		ThisDN DN22	ThisDN DN3
PreviousConnId conn1		OtherDN DN1	OtherDN DN1
ThirdPartyDN DN22		OtherDNRole NewParty	OtherDNRole NewParty
ThirdPartyRole ConferencedBy		ThirdPartyDN DN22	ThirdPartyDN DN22
CallState Conferenced		ThirdPartyRole AddedBy	ThirdPartyRole AddedBy
EventPartyChanged		CallState Conferenced	CallState Conferenced
Release from Conference Phase			



Index

A

- absence codes 170, 266
- absence-n
 - configuration options 170, 266
- accept-dn-type
 - configuration options 263
- Access Code
 - configuration. 106
 - defined. 44, 105
- acd-login
 - configuration options 244, 268
- acw-in-idle-force-ready
 - configuration options 153
 - emulated agents. 153
- ADDP. 60
- addp-remote-timeout
 - common configuration option 236
- addp-timeout
 - common configuration option 236
- addp-trace
 - common configuration option 237
- ADN-login
 - configuration options 268
- adn-login
 - configuration options 244, 268
- Advanced Disconnect Detection Protocol 28
- agent answer supervision 181, 255
- Agent Login objects 45
- agent no-answer supervision. 157
- agent reservation
 - defined. 32
- agent-group
 - configuration options 255
 - emulated agent options 255
- agentid-as-password
 - configuration options 244
- agent-no-answer-action
 - configuration options 256, 257
- agent-no-answer-overflow
 - configuration options 255
- agent-no-answer-timeout
 - configuration options 255
- Agent-Reservation section
 - common configuration options 225–226
- agent-strict-id
 - configuration options 149, 252
- all
 - common log option 203
- analog-transfer
 - configuration options 247
- ANI. 72
- ani-distribution
 - common configuration option 218
- annex tab
 - configuration options 158
- annex tab options
 - no-answer-action 268
 - no-answer-overflow 267
 - no-answer-timeout 266
- app
 - command line parameter 117
- Application objects
 - multi-site operation 103
- application-link
 - configuration options 246
- Application-Link Section
 - configuration options 269
- app-link-version
 - configuration options 270
- audience
 - defining 12

B

- background-processing
 - common configuration option 218
- background-timeout
 - common configuration option 219
- backup servers. 51
- backup-sync
 - configuration section 60

Backup-Synchronization section	
common configuration options	236–237
block-failed-dn-tout	
configuration options	248
buffering	
common log option	196

C

callback-dn	
configuration options	262
Call-Cleanup section	
common configuration options	237–239
call-max-outstanding	
configuration options	272
call-retain-in-queue	
configuration options	143, 265
call-rq-gap	
configuration options	271
cast-type	
common configuration option	71, 228
CDN	77
changes from 7.5 to 7.6	
common configuration options	214
configuration options	240, 274
chapter summaries	
defining	12
check-point	
common log option	200
check-tenant-profile	
common configuration option	219
cleanup-idle-tout	
common configuration option	237
clid-withheld-name	
configuration options	161, 264
Code property	106, 107
cof-ci-defer-create	
common configuration option	233
cof-ci-defer-delete	
common configuration option	233
cof-ci-req-tout	
common configuration option	86, 233
cof-ci-wait-all	
common configuration option	234
cof-feature	
common configuration option	234
cof-rci-tout	
common configuration option	234
command line parameters	117
app	117
host	117
I	118
lmspath	118
nco X/Y	118
port	117
V	118
commenting on this document	17
common configuration options	196–215
addp-remote-timeout	236
addp-timeout	236
addp-trace	237
Agent-Reservation section	225–226
ani-distribution	218
background-processing	218
background-timeout	219
Backup-Synchronization section	236–237
Call-Cleanup section	237–239
cast-type	228
changes from 7.5 to 7.6	214
check-tenant-profile	219
cleanup-idle-tout	237
cof-ci-defer-create	233
cof-ci-defer-delete	233
cof-ci-req-tout	233
cof-ci-wait-all	234
cof-feature	234
cof-rci-tout	234
common section	213–214
compatibility-port	219
consult-user-data	220
customer-id	220
default-dn	229
direct-digits-key	229
dn-for-unexpected-calls	230
enable-async-dns	214
event-propagation	235
inbound-translator-<n>	235
License section	223–225
local-node-id	234
log section	196–209
log-extended section	210–212
log-filter section	212
log-filter-data section	213
log-trace-flags	221
management-port	221
mandatory	196
match-call-once	227
merged-user-data	221
Multi-Site Support section	226–235
network-request-timeout	230
notify-idle-tout	238
num-of-licenses	223
num-sdn-licenses	223
periodic-check-tout	238
protocol	237
rebind-delay	214
reconnect-tout	227
register-attempts	230
register-tout	230
reject-subsequent-request	226
report-connid-changes	228
request-collection-time	226

request-tout	230
reservation-time	226
resource-allocation-mode	231
resource-load-maximum	231
route-dn	231
rule-<n>	235
Security section	239
server-id	222
setting	195, 217
sync-reconnect-tout	237
tcs-queue	232
tcs-use	232
timeout	232
timeout value format	239–240
Translation Rules section	235
T-Server section	218–222
use-data-from	228
use-implicit-access-numbers	232
user-data-limit	222
common log options	196–213
all	203
buffering	196
check-point	200
compatible-output-priority	201
debug	205
default-filter-type	212
expire	197
interaction	204
keep-startup-file	198
<key name>	213
level-reassign-<eventID>	210
level-reassign-disable	212
log section	196–209
log-extended section	210–212
log-filter section	212
log-filter-data section	213
mandatory options	196
memory	201
memory-storage-size	201
message_format	199
messagefile	198
print-attributes	200
segment	197
setting	195
spool	201
standard	203
time_convert	199
time_format	200
trace	204
verbose	196
x-conn-debug-all	209
x-conn-debug-api	209
x-conn-debug-dns	209
x-conn-debug-open	207
x-conn-debug-security	208
x-conn-debug-select	208
x-conn-debug-timers	208
x-conn-debug-write	208
common options	
common log options	196–213
common section	213–214
mandatory options	196
common section	
common options	213–214
compatibility-port	
common configuration option	219
compatible-output-priority	
common log option	201
Configuration Manager	
configuring T-Server	46
multiple ports	47
configuration options	
absence-n	170, 266
accept-dn-type	263
acd-login	244, 268
acw-in-idle-force-ready	153
ADN-login	268
adn-login	244, 268
agent answer supervision	255
agent-group	255
agentid-as-password	244
agent-no-answer-action	256
agent-no-answer-overflow	255
agent-no-answer-timeout	255
agent-strict-id	149, 252
analog-transfer	247
application-link	246
app-link-version	270
block-failed-dn-tout	248
callback-dn	262
call-max-outstanding	272
call-retain-in-queue	143, 265
call-rq-gap	271
changes from 7.5 to 7.6	240, 274
clid-withheld-name	161, 264
common log options	196–213
common options	196–215
connid-update	242
consult-supervised-rt	261
convert-otherdn	160, 262
correct-connid	261
correct-rqid	261
default-destination	242
default-dn-type	263
dn-del-mode	263
dn-for-undesired-calls	160, 263
emulated-login-state	148, 251
emulate-login	148, 251
enable-retain-in-queue	142, 265
expire-call-tout	262
expire-call-tout (removed)	262
extension	266

- extn-no-answer-action 257
- extn-no-answer-overflow 257
- ext-redir-min-digits 246
- ext-routing-min-digits 245
- ext-xfer-min-digits 245
- force-pbx-ag-state 248
- force-snapshot-interval 247
- ha-sync-dly-lnk-conn 273
- hostname 270
- inbound-bsns-calls 146, 249
- inherit-bsns-type 146, 250
- internal-bsns-calls 146, 250
- kpl-interval 163, 271
- kpl-loss-rate 164, 272
- kpl-tolerance 163, 272
- legal-guard-time 154, 250
- link-control 246
- link-max-outstanding-dev-rq 269
- login-code 247
- login-pincode 264
- mandatory
 - common 196
- max-outstanding 272
- max-outstanding-device-rq 269
- max-outstanding-login-rq 269
- max-pred-req-delay 260
- min-route-dly 248
- nas-indication 159, 259
- nas-private 158, 259, 269
- no-answer supervision 256, 257, 258, 266
- old-call-in-acw-behavior 153, 254
- outbound-bsns-calls 146, 249
- override-switch-acw 255
- password-delimiter 245
- port 270
- posn-no-answer-overflow 258
- posn-no-answer-timeout 257, 266
- prd-dist-call-ans-time 260
- queue-cleanup 243
- queue-return-dly 143, 265
- quiet-cleanup 273
- quiet-startup 273
- reason-code 169, 265
- recall-no-answer-timeout 159, 259
- redistr-dly 248
- reg-delay 274
- reg-interval 271
- reg-silent 274
- restart-cleanup-dly 274
- restart-cleanup-limit 273
- restart-period 273
- retain-call-tout 262
- route-consult-call 246
- route-request-attempt 261
- rq-expire-tout 271
- rq-gap 271

- setting
 - common 195
- supervised-route-timeout 260
- switch-agent-id 243
- switch-domain-check 247
- sync-emu-agent 252
- timed-acw-in-idle 152, 250
- T-Server for Ericsson MD110 241
- unknown-bsns-calls 146, 250
- unknown-xfer-merge-udata 249
- untimed-wrap-up-value 151, 253
- use-makecall-login 247
- vto-onhook-delay 249
- wrap-up-threshold 151, 254
- wrap-up-time 150, 253
- configuring
 - high availability
 - T-Server 59–61
 - multi-site operation 103–116
 - steps 103
 - T-Server 46
 - multiple ports 47
- configuring devices in Configuration Layer 137
- configuring digital devices 137
- connid-update
 - configuration options 242
- consult-supervised-rt
 - configuration options 261
- consult-user-data
 - common configuration option 220
- convert-otherdn
 - configuration options 160, 262
- correct-connid
 - configuration options 261
- correct-rqid
 - configuration options 261
- CTI-Link Section
 - configuration options 270
- customer-id
 - common configuration option 220

D

- data keys 186
- debug
 - common log option 205
- Default Access Code
 - configuration 105
 - defined 105
- default-destination
 - configuration options 242
- default-dn
 - common configuration option 229
- default-dn-type
 - configuration options 263

- default-filter-type
 - common log option 212
- destination location 65
- destination T-Server 70
- device-specific behavior 141
- direct-ani
 - ISCC transaction type 72, 79
- direct-callid
 - ISCC transaction type 72, 79
- direct-digits
 - transaction type 79
- direct-digits-key
 - common configuration option 229
- direct-network-callid
 - ISCC transaction type 73, 79
- direct-notoken
 - ISCC transaction type 74, 79
- direct-uui
 - ISCC transaction type 73, 79
- DN objects 45
- dn-del-mode
 - configuration options 263
- dn-for-undesired-calls
 - configuration options 160, 263
- dn-for-unexpected-calls
 - common configuration option 230
- dnis-pool
 - in load-balancing mode 75
 - ISCC transaction type 68, 74, 79
- DNs
 - configuring for multi-sites 110
- document
 - conventions 14
 - errors, commenting on 17
 - version number 14

E

- emulated agent options
 - agent-group 255
 - old-call-in-acw-behavior 153, 254
 - sync-emu-agent 252
 - untimed-wrap-up-value 151, 253
- emulated agents 147–155
 - acw-in-idle-force-ready 153
 - agent-strict-id 149, 252
 - inbound-bsns-calls 146, 249
 - legal-guard-time 154, 250
 - outbound-bsns-calls 146, 249
 - timed-acw-in-idle 152, 250
 - wrap-up-time 150, 253
- emulated predictive dialing 155
- emulated supervised routing
 - consult-supervised-rt 261
 - supervised-route-timeout 260

- emulated-login-state
 - configuration options 148, 251
- emulate-login
 - configuration options 148, 251
- enable-async-dns
 - common configuration option 214
- enable-retain-in-queue
 - configuration options 142, 265
- Error Messages
 - T-Server for Ericsson MD110 186
- Event Propagation
 - defined 97
- EventAttachedDataChanged 97
- event-propagation
 - common configuration option 235
- expire
 - common log option 197
- expire-call-tout
 - configuration options 262
- expire-call-tout (removed)
 - configuration options 262
- extension
 - configuration options 266
- extension no-answer supervision 157
- extensions 158
- extensions attribute 180
- extn-no-answer-action 257
- extn-no-answer-overflow
 - configuration options 257
- ext-redir-min-digits
 - configuration options 246
- extrouter
 - configuration section 95, 100, 104
- ext-routing-min-digits
 - configuration options 245
- ext-xfer-min-digits
 - configuration options 245

F

- figures
 - hot standby redundancy 54
 - Multiple-to-Point mode 78
 - Point-to-Point mode 77
 - steps in ISCC/Call Overflow 85
- force-pbx-ag-state
 - configuration options 248
- force-snapshot-interval
 - configuration options 247
- Free Seating 141

H**HA**

See also high availability

See hot standby

HA configuration 51–61

HA Proxy

starting. 124, 125

ha-sync-dly-lnk-conn

configuration options 273

high-availability configuration 51–61

host

command line parameter 117

hostname

configuration options 270

hot standby. 29, 51

defined. 29

figure 54

T-Server configuration. 58

I**inbound-bsns-calls**

configuration options 146, 249

emulated agents. 146, 249

inbound-translator-<n>

common configuration option 235

inherit-bsns-type

configuration options 146, 250

Inter Server Call Control 65–83

Inter Server Call Control/Call Overflow. 83–87

interaction

common log option 204

internal-bsns-calls

configuration options 146, 250

ISCC

destination T-Server. 70

origination T-Server. 70

ISCC transaction types. 67, 70

direct-ani. 72, 79

direct-callid 72, 79

direct-digits 79

direct-network-callid 73, 79

direct-notoken 74, 79

direct-uui. 73, 79

dnis-pool. 74, 79

in load-balancing mode 75

pullback 76, 79

reroute. 76, 79

route. 77, 79

route-uui. 78

supported 79

ISCC/COF

supported 84

iscc-xaction-type 67

K**keep-startup-file**

common log option 198

<key name>

common log option 213

keys

. 186

Known Limitations 131, 277, 279

kpl-interval

configuration options 163, 271

kpl-loss-rate

configuration options 164, 272

kpl-tolerance

configuration options 163, 272

L**I**

command line parameter 118

legal-guard-time

configuration options 154, 250

emulated agents 154, 250

level-reassign-<eventID>

common log option 210

level-reassign-disable

common log option 212

License section

common configuration options 223–225

link-control

configuration options 246

Link-Control Section 270

link-max-outstanding-dev-rq

configuration options 269

lmspath

command line parameter 118

local-node-id

common configuration option 234

location parameter 66

log configuration options. 196–213

log section

common log options 196–209

log-extended section

common log options 210–212

log-filter section

common log options 212

log-filter-data section

common log options 213

login-code

configuration options 247

login-pincode

configuration options 264

log-trace-flags

common configuration option 221

M

Management Layer	40
management-port	
common configuration option	221
Mandatory Options	
configuration options	241
mandatory options	
common configuration options	218
match-call-once	
common configuration option	227
max-outstanding	
configuration options	272
max-outstanding-device-rq	
configuration options	269
max-outstanding-login-rq	
configuration options	269
max-pred-req-delay	
configuration options	260
Media Layer	40
memory	
common log option	201
memory-storage-size	
common log option	201
merged-user-data	
common configuration option	221
message_format	
common log option	199
messagefile	
common log option	198
min-route-dly	
configuration options	248
Multiple-to-One mode	78
Multiple-to-Point mode	78
Multi-Site Support section	
common configuration options	226–235

N

nas-indication	
configuration options	159, 259
nas-private	
configuration options	158, 259, 269
NAT/C feature	95
nco X/Y	
command line parameter	118
network attended transfer/conference	95
network objects	40
network-request-timeout	
common configuration option	230
no-answer supervision	157, 256, 257, 258, 266
agents	157
device-specific overrides	158
extensions	157
overrides for individual calls	158
positions	158

no-answer-action	158
annex tab options	268
no-answer-overflow	158
annex tab options	267
no-answer-timeout	158
annex tab options	266
notify-idle-tout	
common configuration option	238
Number Translation feature	87–95
number translation rules	88
num-of-licenses	
common configuration option	223
num-sdn-licenses	
common configuration option	223

O

objects	
Agent Logins	45
DNs	45
network	40
Switches	44
Switching Offices	44
telephony	40
old-call-in-acw-behavior	
configuration options	153, 254
emulated agent options	153, 254
One-to-One mode	77
origination location	65
origination T-Server	70
outbound-bsns-calls	
configuration options	146, 249
emulated agents	146, 249
override-switch-acw	
configuration options	255

P

partitioned switch configuration	137
password-delimiter	
configuration options	245
periodic-check-tout	
common configuration option	238
Point-to-Point mode	77
port	
command line parameter	117
configuration options	270
position no-answer supervision	158
posn-no-answer-overflow	
configuration options	258
posn-no-answer-timeout	
configuration options	257, 266
prd-dist-call-ans-time	
configuration options	260
predictive dialing	155

primary servers 51
 print-attributes
 common log option 200
 protocol
 common configuration option 237
 pullback
 ISCC transaction type 76, 79

Q

queue-cleanup
 configuration options 243
 queue-return-dly
 configuration options 143, 265
 quiet-cleanup
 configuration options 273
 quiet-startup
 configuration options 273

R

reason codes 170, 266
 reason-code
 configuration options 169, 265
 rebind-delay
 common configuration option 214
 recall-no-answer-timeout
 configuration options 159, 259
 reconnect-tout
 common configuration option 227
 redistrib-dly
 configuration options 248
 redundancy
 hot standby 29, 51
 warm standby 29, 51
 redundancy types 55, 56, 58
 hot standby 29
 reg-delay
 configuration options 274
 reg-interval
 configuration options 271
 register-attempts
 common configuration option 230
 register-tout
 common configuration option 230
 reg-silent
 configuration options 274
 reject-subsequent-request
 common configuration option 226
 report-connid-changes
 common configuration option 228
 request-collection-time
 common configuration option 226
 request-tout
 common configuration option 230

ISCC configuration option 67
 reroute
 ISCC transaction type 76, 79
 reservation-time
 common configuration option 226
 resource-allocation-mode
 common configuration option 231
 resource-load-maximum
 common configuration option 231
 restart-cleanup-dly
 configuration options 274
 restart-cleanup-limit
 configuration options 273
 restart-period
 configuration options 273
 retain-call-tout
 configuration options 262
 route
 ISCC transaction type 68, 77, 79, 110
 route-consult-call
 configuration options 246
 route-dn
 common configuration option 231
 route-request-attempt
 configuration options 261
 route-uuui
 ISCC transaction type 78
 routing
 Inter Server Call Control 70–83
 rq-expire-tout
 configuration options 271
 rq-gap
 configuration options 271
 rule-<n>
 common configuration option 235
 run.bat 121
 run.sh 120

S

Security section
 common configuration options 239
 segment
 common log option 197
 server-id
 common configuration option 222
 setting
 common configuration options 217
 setting configuration options
 common 195
 Setting the DN Properties
 T-Server for Ericsson MD110 134
 spool
 common log option 201
 standard
 common log option 203

- starting
 - HA Proxy 124
 - T-Server 125
- supervised-route-timeout
 - configuration options 260
- supported agent work modes
 - supported functionality 180
- Supported Functionality
 - T-Server for Ericsson MD110 145
- supported functionality
 - supported agent work modes 180
- Switch objects 44
 - multi-site operation 103
- switch/CTI environments 136
- switch-agent-id
 - configuration options 243
- switch-domain-check
 - configuration options 247
- Switching Office objects 44
 - multi-site operation 104, 105, 106, 110
- Switch-Specific Configuration
 - T-Server for Ericsson MD110 131, 279
- sync-emu-agent
 - configuration options 252
 - emulated agent options 252
- sync-reconnect-tout
 - common configuration option 237

T

- Table 180
- Target ISCC
 - Access Code configuration 107
 - Default Access Code configuration 106
- tcs-queue
 - common configuration option 232
- tcs-use
 - common configuration option 232
- telephony objects 40
- time_convert
 - common log option 199
- time_format
 - common log option 200
- timed-acw-in-idle
 - configuration options 152, 250
 - emulated agents 152, 250
- timeout
 - common configuration option 68, 232
 - ISCC configuration option 68
- timeout value format
 - common configuration options 239–240
- TInitiateConference 66
- TInitiateTransfer 66
- TMakeCall 66
- TMuteTransfer 66

- trace
 - common log option 204
- transaction types (ISCC) 67, 70
 - supported 79
- transfer connect service 82
- Translation Rules section
 - common configuration options 235
- TRouteCall 66
- trunk lines 77, 78
- T-Server
 - configuring Application objects 46
 - for multi-sites 103
 - configuring redundancy 56
 - HA 58
 - high availability 58
 - hot standby 58
 - multi-site operation 103–116
 - redundancy 55, 56, 58
 - starting 125, 126
 - using Configuration Manager 46
 - multiple ports 47
 - warm standby 56
- T-Server for Ericsson MD110
 - Application-Link Section
 - configuration options 269
 - changes from 7.5 to 7.6
 - configuration options 274
 - configuration options 241
 - CTI-Link Section
 - configuration options 270
 - Error Messages 186
 - Mandatory Options
 - configuration options 241
 - Setting the DN Properties 134
 - Supported Functionality 145
 - Switch-Specific Configuration 131, 279
 - T-Server Section
 - configuration options 242
- T-Server Section
 - configuration options 242
- T-Server section
 - common configuration options 218–222
- TSingleStepTransfer 66
- TXRouteType 67
- typographical styles 14

U

- UNIX
 - installing T-Server 42, 48
 - starting applications 121
 - starting HA Proxy 124
 - starting T-Server 126
 - starting with run.sh 120

unknown-bsns-calls
 configuration options 146, 250
 unknown-xfer-merge-udata
 configuration options 249
 untimed-wrap-up-value
 configuration options 151, 253
 emulated agent options 151, 253
 use-data-from
 common configuration option 228
 use-implicit-access-numbers
 common configuration option 232
 use-makecall-login
 configuration options 247
 user data keys 186
 user data propagation 97
 user-data-limit
 common configuration option 222

V

V
 command line parameters 118
 VDN 77
 verbose
 common log option 196
 version numbering
 document 14
 vto-onhook-delay
 configuration options 249

W

warm standby 29, 51
 figure 52
 T-Server configuration 56
 Windows
 installing T-Server 43, 49
 starting applications 121
 starting HA Proxy 125
 starting T-Server 126
 starting with run.bat 121
 wrap-up-threshold
 configuration options 151, 254
 wrap-up-time
 configuration options 150, 253
 emulated agents 150, 253

X

x-conn-debug-all
 common log option 209
 x-conn-debug-api
 common log option 209

x-conn-debug-dns
 common log option 209
 x-conn-debug-open
 common log option 207
 x-conn-debug-security
 common log option 208
 x-conn-debug-select
 common log option 208
 x-conn-debug-timers
 common log option 208
 x-conn-debug-write
 common log option 208