



Framework 8.1

T-Server for Alcatel A4400/OXE

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 © 1997–2012 Genesys Telecommunications Laboratories, Inc. All rights reserved.

About Genesys

Genesys is the world's leading provider of customer service and contact software - with more than 4,000 customers in 80 countries. Drawing on its more than 20 years of customer service innovation and experience, Genesys is uniquely positioned to help companies bring their people, insights and customer channels together to effectively drive today's customer conversation. Genesys software directs more than 100 million interactions every day, maximizing the value of customer engagement and differentiating the experience by driving personalization and multi-channel customer service - and extending customer service across the enterprise to optimize processes and the performance of customer-facing employees. Go to www.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 company names and logos may be trademarks or registered trademarks of their respective holders. © 2012 Genesys Telecommunications Laboratories, Inc. All rights reserved.

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 regional numbers provided on [page 17](#). For complete contact information and procedures, refer to the [Genesys Technical Support Guide](#).

Ordering and Licensing Information

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

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 81fr_dep-ts_a4400_03-2012_v8.1.001.02



Table of Contents

List of Procedures	11
Preface	15
	About T-Server for Alcatel A4400/OXE	15
	Making Comments on This Document	16
	Contacting Genesys Technical Support.....	17
	Document Change History	17
Part 1	T-Server Deployment	19
	New for All T-Servers in 8.1	19
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	27
	Redundant T-Servers	28
	Multi-Site Support	32
	Agent Reservation	32
	Client Connections	33
	Next Steps	33
Chapter 2	T-Server General Deployment.....	35
	Prerequisites.....	35
	Software Requirements	35
	Hardware and Network Environment Requirements	37
	Licensing Requirements	37
	About Configuration Options.....	39
	Deployment Sequence	40
	Deployment of T-Server.....	40
	Configuration of Telephony Objects	40

	Configuration of T-Server.....	43
	Installation of T-Server.....	44
	Next Steps	47
Chapter 3	High-Availability Deployment.....	49
	Warm Standby Redundancy Type	50
	Hot Standby Redundancy Type	51
	Prerequisites.....	53
	Requirements.....	53
	Synchronization Between Redundant T-Servers	53
	Warm Standby Deployment.....	54
	General Order of Deployment.....	54
	Modification of T-Servers for Warm Standby	55
	Warm Standby Installation of Redundant T-Servers	56
	Hot Standby Deployment.....	56
	General Order of Deployment.....	56
	Modification of T-Servers for Hot Standby	57
	Hot Standby Installation of Redundant T-Servers	60
	Next Steps	60
Chapter 4	Multi-Site Support.....	61
	Multi-Site Fundamentals	62
	ISCC Call Data Transfer Service	63
	ISCC Call Flows.....	64
	ISCC Transaction Types	70
	T-Server Transaction Type Support.....	78
	Transfer Connect Service Feature.....	82
	ISCC/Call Overflow Feature	83
	Number Translation Feature.....	87
	Number Translation Rules	88
	Network Attended Transfer/Conference Feature.....	95
	Event Propagation Feature.....	97
	User Data Propagation	98
	Party Events Propagation	99
	Switch Partitioning	100
	Event Propagation Configuration	101
	ISCC Transaction Monitoring Feature	104
	Configuring Multi-Site Support.....	104
	Applications	105
	Switches and Access Codes	106
	DNs.....	112
	Configuration Examples.....	117
	Next Steps	118

Chapter 5	Starting and Stopping T-Server Components	119
	Command-Line Parameters	119
	Starting and Stopping with the Management Layer	121
	Starting with Startup Files	122
	Starting Manually	123
	HA Proxy	126
	T-Server	127
	Verifying Successful Startup	129
	Stopping Manually	129
	Starting and Stopping with Windows Services Manager	130
	Next Steps	130
 Part 2	 T-Server Configuration	 131
	New in T-Server for Alcatel A4400/OXE	132
 Chapter 6	 Alcatel A4400 Switch-Specific Configuration	 135
	Known Limitations	135
	Changes in Reporting Behavior from 7.x to 8.1	137
	Switch Terminology	139
	Support of Switch/CTI Environments	141
	Setting DN Properties	141
	Routing Without Using Switch-Routing Services	144
	Configuring Switch Timers	145
	Configuring Extensions in the PBX	146
	Configuring CCD Agents	147
	Definitions	147
	Agent Login	147
	Preassigned and Supervisor Agents	148
	CCD Agent/Supervisor–Related T-Server Features	152
	Advanced Agent Features	158
	Agent Substitution	159
	Configuring CCD Objects	161
	Network Call ID Matching	164
	Routing Using Emulated Routing Points	164
	Configuring Hunt Groups/Virtual Devices (PBX)	165
	Configuring Hunt Groups/Virtual Devices in the Configuration Layer	166
	Configuring External Routing Point Virtual Z Members in Alcatel A4400	166
	Configuring the External Routing Point and Virtual Z Devices in the Configuration Layer	167
	Integrating Routing Points in the CCD	169

Chapter 7	Supported T-Server Features	171
	Support for Business-Call Handling.....	172
	T-Server Call Classification.....	172
	Feature Configuration	174
	Support for Emulated Agents	175
	Emulated Agent Login/Logout	175
	Emulated Agent Ready/NotReady.....	176
	Emulated After-Call Work (ACW).....	176
	Feature Configuration	179
	HA Synchronization	180
	Emulated Wrap-Up Time for CCD Agents	180
	Support for Predictive Dialing	182
	Voice Activity Detection	183
	Configuring Predictive Dialing in the Configuration Layer	186
	Feature Configuration	186
	Limiting Call-Distribution Time	187
	Configuring OCS to Use Predictive Dialing	188
	CCO Agent Reservation Feature	188
	Set Reservation	188
	Reset Reservation	189
	CCO Call Tag	189
	Make CCO Call.....	189
	Support for Emulated Predictive Dialing.....	189
	Limiting Distribution Time.....	190
	Call Progress Detection	190
	Unsolicited Calls on Predictive Dialing Devices.....	191
	Feature Configuration	191
	Support for No-Answer Supervision	191
	Agent No-Answer Supervision	191
	Extension No-Answer Supervision	192
	Position No-Answer Supervision	192
	Feature Configuration	192
	Support for Smart OtherDN Handling.....	194
	Feature Configuration	194
	Supported Requests	194
	Support for the Keep-Alive Feature	196
	Support for Call Type Prediction.....	196
	Feature Configuration	196
	Support for Call Release Tracking.....	197
	DN-Based Reporting.....	197
	Call-Based Reporting.....	197
	Feature Configuration	198
	Support for Failed Route Notification.....	198

HA Considerations	198
Feature Configuration	198
Support for Link Bandwidth Monitoring	199
High and Low Watermarks	199
HA Considerations	200
Feature Configuration	200
Request Handling Enhancements	200
Feature Configuration	200
Hot-Standby HA Synchronization	201
Feature Configuration	202
Support for Boss/Secretary Functionality	202
Support for A4400/OXE Spatial Redundancy	202
Support for Outbound Caller ID	203
Support for Inter Server Call Control	204
ISCC Routing Strategies	204
ISCC/Call Overflow	206
Feature Configuration	207
Configuring the GCID for ISCC Call Tracking	207
Support for Account Codes	209
Indexing by Key Name	209
Entering Account Codes Manually	210
Entering Account Codes by CTI Request	210
Setting Keys in Call-Related Requests	210
Feature Configuration	211
Walk-Away Codes	211
Account Code as a Private Event/Service	212
T-Library Functionality	212
CTI-Supported Functionality for SIP Extensions	223
Support for Agent Work Modes	226
Private Services and Events	227
Use of the Extensions Attribute	231
Extension Filtering	251
User Data Keys	251
Reasons Keys	253
T-Server Error Messages	253
 Chapter 8	
Supported High Availability (HA) Configurations	263
Genesys HA Configuration	263
Hot Standby Mode	263
Warm Standby Mode	264
HA and PBX Licensing Issues	265
Configuration	265

Chapter 9	Using Alcatel A4400 Routing Services Interface.....	267
	RSI Description.....	267
	RSI and CSTA Interfaces.....	268
	Configuring RSI in the Alcatel A4400	268
	RSI Agent Functionality	270
	Call Routing.....	270
	Treatment Support	270
	Treatments.....	274
	Collect Digits Treatment.....	274
	Play Announcement Treatment	276
	Play Announcement and Collect Digits Treatment	281
	Music Treatment	281
	Busy Treatment.....	282
	Ringback Treatment.....	282
	Silence Treatment.....	282
	Cancel Call Treatment	282
	IVR Treatment.....	283
	RSI Reroute Behavior.....	284
	Feature Configuration	284
	RSI Reroute Authorization	284
	Voice Guides in the Alcatel A4400 PBX	285
	Checking Voice Guides.....	286
	Private Data in Route Requests on RSI	287
	Feature Configuration	287
Chapter 10	Alcatel A4400 Call Flows	289
	Call Queued with Automatic Camp-On.....	289
	Call Queued with Automatic Camp-On.....	289
	Call Parking/Unparking	293
	Description.....	293
Chapter 11	Connecting GVP:EE 6.5.5 to Alcatel A4400/OXE.....	297
	PBX Configuration	297
	Device Configuration	298
	Dialogic Configuration for DMV Cards.....	299
	The FCD File	299
	The CDP File	300
	The PDK.CFG File.....	300
	Dialogic Configuration Manager	301
	Troubleshooting	301
	Dialogic Configuration for JCT-E1 Cards.....	305

	Upgrade to Global Call Protocols Package 4.1	305
	Install PTR 27176	305
	Dialogic Configuration Manager Settings	305
	Adapt the Protocol File to Work with A4400	307
	Defining the Protocol File To Be Used In GVP	308
Chapter 12	Troubleshooting	309
	Troubleshooting Procedures.....	309
	CSTA Licenses	309
	CPU Usage	311
	CSTA Protocol Errors.....	312
	Network Issues	312
	Timer Issues	313
Chapter 13	Common Configuration Options	315
	Setting Configuration Options.....	315
	Mandatory Options	316
	log Section.....	316
	Log Output Options.....	322
	Examples	326
	Debug Log Options	327
	log-extended Section	330
	log-filter Section	332
	log-filter-data Section.....	332
	security Section	333
	sml Section	333
	common Section.....	335
	Changes from 8.0 to 8.1	335
Chapter 14	T-Server Common Configuration Options	337
	Setting Configuration Options.....	337
	Mandatory Options	338
	TServer Section	338
	license Section	343
	agent-reservation Section.....	346
	extrouter Section	347
	ISCC Transaction Options	349
	Transfer Connect Service Options	353
	ISCC/COF Options	354
	Event Propagation Options	356
	Number Translation Option	357

	GVP Integration Option.....	358
	backup-sync Section	358
	call-cleanup Section	360
	Translation Rules Section.....	361
	security Section	362
	Timeout Value Format	362
	Changes from Release 8.0 to 8.1	363
Chapter 15	Configuration Options in T-Server for Alcatel A4400/OXE.....	365
	Application-Level Options.....	365
	Mandatory Options	366
	TServer Section	366
	SwitchSpecificType Section	400
	Link-tcp Section	401
	Link-Control Section	401
	Lang-Map Section.....	406
	Ext-Filter Section	407
	DN-Level Options	408
	TServer Section	408
	Changes from 8.0 to 8.1	411
Supplements	Related Documentation Resources	413
	Document Conventions	415
Index	419



List of Procedures

Configuring T-Server	43
Configuring multiple ports	44
Installing T-Server on UNIX	45
Installing T-Server on Windows	46
Verifying the installation of T-Server	47
Modifying the primary T-Server configuration for warm standby	55
Modifying the backup T-Server configuration for warm standby	56
Modifying the primary T-Server configuration for hot standby	57
Modifying the backup T-Server configuration for hot standby	59
Activating Transfer Connect Service	83
Configuring Number Translation	95
Activating Event Propagation: basic configuration	102
Modifying Event Propagation: advanced configuration	102
Configuring T-Server Applications	105
Configuring Default Access Codes	107
Configuring Access Codes	108
Configuring access resources for the route transaction type	112
Configuring access resources for the dnis-pool transaction type	114
Configuring access resources for direct-* transaction types	114
Configuring access resources for ISCC/COF	115
Configuring access resources for non-unique ANI	115
Modifying DNs for isolated switch partitioning	116
Configuring T-Server to start with the Management Layer	121
Starting T-Server on UNIX with a startup file	122
Starting T-Server on Windows with a startup file	123
Starting HA Proxy on UNIX manually	127
Starting HA Proxy on Windows manually	127
Starting T-Server on UNIX manually	128
Starting T-Server on Windows manually	128

Stopping T-Server on UNIX manually	129
Stopping T-Server on Windows manually	129
Setting A4400 timers to support link functions	145
Configuring an extension in the PBX for agent login	146
Configuring CCD agent devices in the PBX	151
Configuring multiline devices in the PBX	151
Configuring Headset mode for agent handsets	152
Overriding Headset mode in extensions.	153
Configuring supervisor call behavior in T-Server	153
Overriding supervisor call settings in extensions	154
Configuring/canceling a supervisor help request	154
Configuring supervisor listening/step-in	155
Overriding supervisor listening/step-in settings in extensions	156
Configuring unavailable/withdrawal types in processing groups	157
Overriding unavailable/withdrawal types using extensions	158
Configuring smart monitoring	158
Activating agent substitution for use with Genesys routing for PBX releases higher than R5.0-d2.314.7	160
Configuring CCD Pilots in Configuration Layer	161
Configuring CCD Pilots in the PBX.	162
Configuring CCD Queues in Configuration Layer.	162
Configuring CCD/RSI processing groups in Configuration Layer.	163
Configuring CCD processing groups in the PBX	163
Configuring a Hunt Group in Alcatel A4400	165
Configuring Hunt Group virtual Z members in Alcatel A4400.	165
Configure the Hunt Group and virtual Z devices in the Configuration Layer	166
Configuring External Routing Point virtual Z members in Alcatel A4400	166
Configuring the External Routing Point and virtual Z devices in the Configuration Layer	167
Configuring a Routing Point as part of PBX CCD.	170
Configuring manual wrap-up time for CCD agents in the PBX.	181
Configuring wrap-up time for CCD agents in the Configuration Layer	181
Activating VAD in the Alcatel A4400.	183
Checking GPA DSP status.	184
Configuring virtual Z dialing devices in the PBX.	185

Configuring virtual Z dialing devices in the PBX without VAD	185
Configuring VAD-Related Timers in the Alcatel A4400	186
Configuring devices for predictive dialing in the Configuration Layer. .	186
Configuring the CCD to limit call distribution time	187
Configuring URS to limit call distribution time.	187
Configuring OCS to use predictive dialing	188
Providing ISCC Call Tracking.	207
Creating a new RSI in the PBX	268
Configuring RSI in the Configuration Layer	269
Configuring a GPA Board in the Alcatel A4400 PBX	285
Configuring a VG listening prefix	286
Checking GPA Voice Guide status.	286
Checking the Number of CSTA Licenses Available	309
Checking the licensing feature descriptions	310
Checking the Number of CSTA Licenses Used	310
Analyzing 100 Percent CPU Usage	311
Monitoring CPU Load	311
Identifying CSTA protocol errors	312
Identifying whether CTI link is isolated	313



Preface

Welcome to the *Framework 8.1 T-Server for Alcatel A4400/OXE 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 Alcatel A4400/OXE. The reference information includes, but is not limited to, configuration options, limitations, and switch-specific functionality. You must configure the configuration objects and options described in this document in the Framework Configuration Layer.

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

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

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

This preface contains the following sections:

- [About T-Server for Alcatel A4400/OXE, page 15](#)
- [Making Comments on This Document, page 16](#)
- [Contacting Genesys Technical Support, page 17](#)
- [Document Change History, page 17](#)

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on [page 413](#).

About T-Server for Alcatel A4400/OXE

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.

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

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

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

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

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

Making Comments on This Document

If you especially like or dislike anything about this document, feel free to e-mail your comments to Techpubs.webadmin@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 scope of this document only and to the way in which the information is presented. Contact your Genesys Account Representative or 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.

Contacting Genesys Technical Support

If you have purchased support directly from Genesys, contact Genesys Technical Support at the regional numbers below.

Note: The following contact information was correct at time of publication. For the most up-to-date contact information, see the [Contact Information](#) on the Tech Support website. Before contacting technical support, refer to the *Genesys Technical Support Guide* for complete contact information and procedures.

Genesys Technical Support Contact Information

Region	Telephone	E-Mail
North America and Latin America	+888-369-5555 (toll-free) +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-1276-45-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp
India	000-800-100-7136 (toll-free) +61-7-3368-6868	support@genesyslab.com.au
Malaysia	1-800-814-472 (toll-free) +61-7-3368-6868	support@genesyslab.com.au

Document Change History

This version of the *Framework 8.1 T-Server for Alcatel A4400/OXE Deployment Guide* has been updated with the following:

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



Part

1

T-Server Deployment

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

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

- Chapter 1, “T-Server Fundamentals,” on [page 21](#), describes T-Server, its place in the Framework 8 architecture, T-Server redundancy, and multi-site issues. It stops short of providing configuration and installation information.
- Chapter 2, “T-Server General Deployment,” on [page 35](#), presents configuration and installation procedures for all T-Servers.
- Chapter 3, “High-Availability Deployment,” on [page 49](#), addresses high availability (HA).
- Chapter 4, “Multi-Site Support,” on [page 61](#), details the variations available for T-Server implementations across geographical locations.
- Chapter 5, “Starting and Stopping T-Server Components,” on [page 119](#), describes how, and in what order, to start up T-Server among other Framework components. It also provides possible stopping commands.

New for All T-Servers in 8.1

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

- T-Server no longer connects to applications that have disabled status in the configuration environment.
- The default value of the background-processing configuration option has been changed to true. See “background-processing” on [page 338](#) for details.

- T-Server now supports the Unresponsive Process Detection feature. The following configuration options enable this feature:
 - “heartbeat-period” on [page 333](#)
 - “hangup-restart” on [page 334](#)

For more information, refer to the *Framework 8.0 Management Layer User’s Guide*.

- T-Server now supports IPv6. For more information, refer to the *Framework 8.1 Deployment Guide*.
- T-Server now supports vSphere 4 Hypervisor.
- T-Server now supports Acrezzo FLEXNet Publisher v11.9 license manager.

Notes: • Configuration option changes common to all T-Servers are described in “Changes from Release 8.0 to 8.1” on [page 363](#).

- For information about the new features that are available in your T-Server in the initial 8.1 release, see Part Two of this document.



Chapter

1

T-Server Fundamentals

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

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

- If you are an experienced user of T-Server, start with “New for All T-Servers in 8.1” on [page 19](#), 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 27](#)
- [Redundant T-Servers, page 28](#)
- [Multi-Site Support, page 32](#)
- [Agent Reservation, page 32](#)
- [Client Connections, page 33](#)
- [Next Steps, page 33](#)

Learning About T-Server

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

Framework and Media Layer Architecture

[Figure 1](#) illustrates the position Framework holds in a Genesys solution.

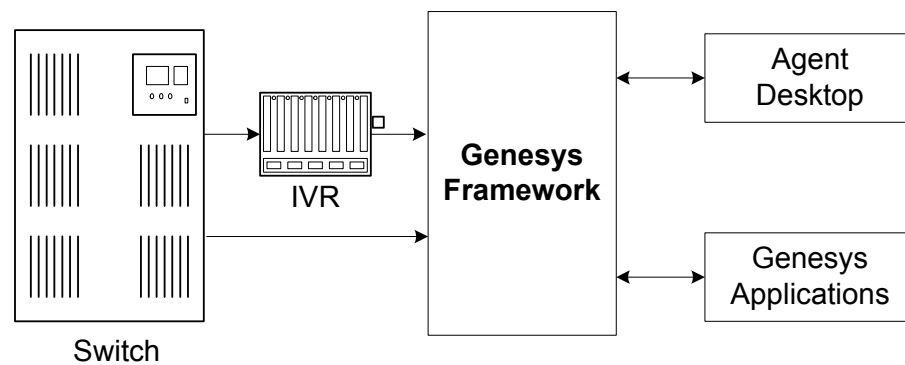


Figure 1: Framework in a Genesys Solution

Moving a bit deeper, [Figure 2](#) presents the various layers of the Framework architecture.

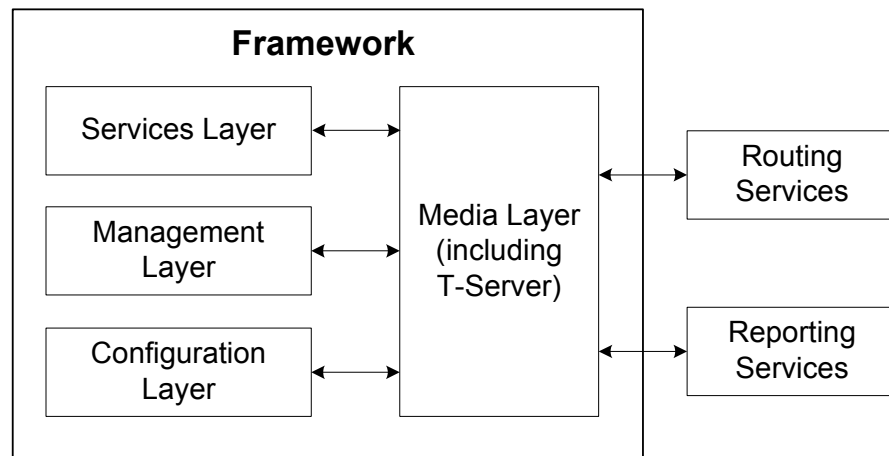


Figure 2: The Media Layer in the Framework Architecture

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

[Figure 3](#) presents the generalized architecture of the Media Layer.

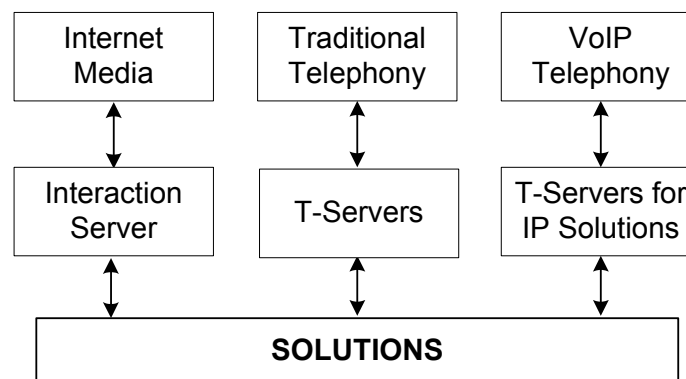


Figure 3: Media Layer Architecture

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

One or more components in practically every solution are T-Server clients. Solutions comprise a number of different Genesys software packages, from collections of components for various types of routing to those that allow for

outbound dialing to still others. Framework in general, and T-Server in particular, enable these solutions to function in your enterprise.

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

T-Server Requests and Events

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

Details of T-Server Functionality

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

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

Bridging

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

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

Refer to the *Genesys Events and Models Reference Manual* for complete information on all T-Server events and call models and to the

TServer.Requests portion of the *Voice Platform SDK 8.x .NET (or Java) API Reference* for technical details of T-Library functions.

Messaging

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

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

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

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

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

Interaction Tracking

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

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

Difference and Likeness Across T-Servers

Although Figure 3 on [page 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 your T-Server will not work with another switch.) Thus, all T-Servers play a common role in the architecture, but their specific features differ from implementation to implementation, based on the media device in use.

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

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

T-Server Functional Steps During a Sample Call

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

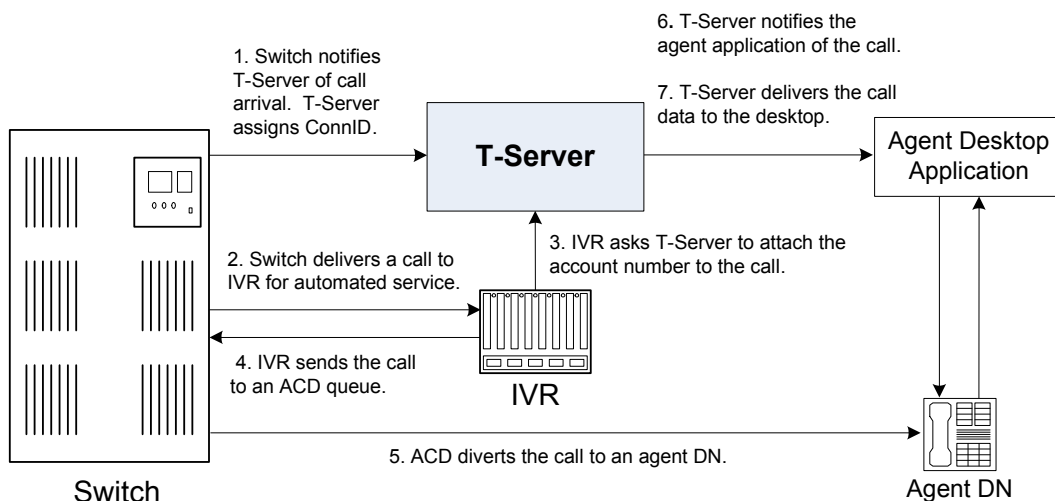


Figure 4: Functional T-Server Steps

Step 1

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

Step 2

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

Step 3

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

Step 4

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

Step 5

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

Step 6

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

Step 7

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

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

Advanced Disconnect Detection Protocol

Since the 6.0 release of T-Server, the Advanced Disconnect Detection Protocol (ADDP) has replaced the Keep-Alive Protocol (KPL) as the method to detect

failures for certain T-Server connections, including connections between two T-Servers and between a T-Server and its clients.

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

ADDP applies only to connections between Genesys software components.

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

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

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

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

Redundant T-Servers

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

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

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

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

Support for Hot Standby Redundancy in Various T-Servers

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

[Table 1](#) only summarizes hot standby redundancy support in various T-Servers. For detailed, up-to-date information on the subject, see the *Genesys Supported Media Interfaces Reference Manual* 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
Aastra MXONE CSTA I	Yes	No	—
Alcatel A4200/OXO	Yes	No	—
Alcatel A4400/OXE	Yes	No	—
Aspect ACD	Yes	No	—
Avaya Communication Manager	Yes	No ^a	—
Avaya INDeX	Yes	No	—
Avaya TSAPI	Yes	No	—
Cisco UCCE	Yes	No	—
Cisco Unified Communications Manager	Yes	No	—
DataVoice Dharma	Yes	No	—
Digitro AXS/20	Yes	No	—

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

T-Server Type	Hot Standby Supported	HA Proxy Required	Number of HA Proxy Components
EADS Intecom M6880	Yes	No	—
EADS Telecom M6500	Yes	No	—
eOn eQueue	Yes	No	—
Fujitsu F9600	Yes	No	—
Huawei C&C08	Yes	No	—
Huawei NGN	Yes	No	—
Mitel MiTAI	Yes	No	—
NEC NEAX/APEX	Yes	No	—
Nortel Communication Server 2000/2100	Yes	Yes ^b , No ^c	1 per link
Nortel Communication Server 1000 with SCCS/MLS	Yes	No	—
Philips Sopho iS3000	Yes	No ^d	1
Radvision iContact	No	—	—
Samsung IP-PCX IAP	Yes	No	—
Siemens Hicom 300/HiPath 4000 CSTA I	Yes	No	—
Siemens HiPath 3000	Yes	No	—
Siemens HiPath 4000 CSTA III	Yes	No	—
Siemens HiPath DX	Yes	No	—
SIP Server	Yes	No	—
Spectrum	Yes	No	—
Tadiran Coral	Yes	No	—
Teltronics 20-20	Yes	Yes	1
Tenovis Integral 33/55	Yes	No	—
Network T-Servers^e			
AT&T	No	—	—

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
Concert	No	—	—
CRSP	No	—	—
DTAG	No	—	—
GenSpec	No	—	—
ISCP	No	—	—
IVR Server, using network configuration	Yes	—	—
KPN	No	—	—
MCI	No	—	—
NGSN	No	—	—
Network SIP Server	No	—	—
Sprint	No	—	—
SR3511	No	—	—
Stentor	No	—	—

- a. With release 7.1, T-Server for Avaya Communication Manager no longer uses HA Proxy for its support of hot standby. Earlier releases of this T-Server require two HA Proxies to support hot standby.
- b. For T-Server for Nortel Communication Server 2000/2100 in high-availability (hot standby) configuration, Genesys recommends that you use link version 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 61](#).

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 63](#)), 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 8.x .NET (or Java) API Reference* for more details on this function from the client’s point of view.

In addition to invoking the `TReserveAgent` function, you can customize the Agent Reservation feature by configuring options in the `T-Server Application` object. See “agent-reservation Section” on [page 346](#) in the “T-Server Common Configuration Options” chapter in Part Two for more details.

Starting with version 8.1, T-Server supports Agent Reservation failure optimization, to ensure that only agent reservation requests of the highest priority are collected. T-Server responds immediately with the `EventError` message to existing or new reservation requests of a lower priority while collecting the agent reservation requests of the highest priority only. This functionality is controlled with the `collect-lower-priority-requests` configuration option (see [page 346](#)).

Client Connections

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

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

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

Next Steps

Now that you have gained a general understanding of the roles and features available with T-Servers, you are ready to learn how T-Servers are installed and configured. That information is presented in the next few chapters of this *Deployment Guide*. So unless you are already familiar with T-Server deployment and operation procedures, continue with Chapter 2, “T-Server General Deployment,” on [page 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](#)
- [Deployment of T-Server, page 40](#)
- [Next Steps, page 47](#)

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

Prerequisites

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

Software Requirements

Framework Components

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

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

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

Media Layer and LCA

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

Supported Platforms

Refer to the *Genesys Supported Operating Environment Reference Manual* for the list of operating systems and database systems supported in Genesys releases 6.x, 7.x, and 8.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 8.x Security Deployment Guide*.

Hardware and Network Environment Requirements

Hosting

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

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

Installation Privileges

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

Server Locations

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

Supported Platforms

Refer to the *Genesys Supported Media Interfaces Reference Manual* for the list of supported switch and PBX versions. You can find this document on the Genesys 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 Licensing Guide* available on the Genesys Documentation Library DVD.

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

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

Licensing Basic Implementations

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

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

Licensing HA Implementations

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

Licensing Multi-Site Implementations

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

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

Configuring License Files

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

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

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

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

About Configuration Options

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

Configuration options common to all T-Servers, independent of switch type, are described in Chapter 14, “T-Server Common Configuration Options,” on [page 337](#). *T-Server-specific* configuration options are described in a separate chapter. T-Server also supports unified Genesys log options, as described in the “Common Configuration Options” chapter.

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

Deployment Sequence

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

Task Summary: T-Server Deployment Sequence

Objective	Related Procedures and Actions
1. Deploy Configuration Layer objects and ensure Configuration Manager is running.	See the <i>Framework 8.1 Deployment Guide</i> for details.
2. Deploy Network objects (such as Host objects).	See the <i>Framework 8.1 Deployment Guide</i> for details.
3. Deploy the Management Layer.	See the <i>Framework 8.1 Deployment Guide</i> for details.
4. Test your configuration and installation.	See Chapter 5, “Starting and Stopping T-Server Components,” on page 119 .

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

Deployment of T-Server

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

Configuration of Telephony Objects

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

Recommendations

Genesys recommends registering (configuring) only those entities you plan to use in the current configuration. The more data there is in the Configuration

Database, the longer it takes for the CTI setup to start, and the longer it will take to process configuration data. Remember that adding configuration objects to the Genesys Configuration Database does not cause any interruption in contact center operation.

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

Switching Offices

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

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

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

Switches

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

Two types of access codes exist in a Genesys configuration:

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

See Chapter 4, “Multi-Site Support,” on [page 61](#), for step-by-step instructions.

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

DNs and Agent Logins

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

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

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

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

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

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

Multi-Site Operations

See the section, “Configuring Multi-Site Support” on [page 104](#), for information on setting up DNs for multi-site operations.

Configuration of T-Server

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

Recommendations

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

Procedure: Configuring T-Server

Start of procedure

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

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

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

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

5. In a multi-site environment, you must complete additional T-Server configuration steps to support multi-site operations; see Chapter 4, “Multi-Site Support,” on [page 61](#).

End of procedure

Next Steps

- See “Installation of T-Server” on [page 44](#).

Procedure: Configuring multiple ports

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

Start of procedure

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

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

- e. Click OK.
5. Click OK to save the new configuration.

End of procedure

Installation of T-Server

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

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

- `media_layer\<switch>\windows` for Windows installations, where `<switch>` is your switch name.

Procedure: Installing T-Server on UNIX

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

Start of procedure

1. In the directory to which the T-Server installation package was copied, locate a shell script called `install.sh`.
2. Run this script from the command prompt by typing `sh` and the file name. For example: `sh install.sh`.
3. When prompted, confirm the host name of the computer on which T-Server is to be installed.
4. When prompted, specify the host and port of Configuration Server.
5. When prompted, enter the user name and password to access Configuration Server.
6. When prompted, select the T-Server application you configured in “Configuring T-Server” on [page 43](#) from the list of applications.
7. Specify the destination directory into which T-Server is to be installed, with the full path to it.
8. If the target installation directory has files in it, do one of the following:
 - Type 1 to back up all the files in the directory (recommended).
 - Type 2 to overwrite only the files in this installation package. Use this option only if the installation being upgraded operates properly.
 - Type 3 to erase all files in this directory before continuing with the installation.

The list of file names will appear on the screen as the files are copied to the destination directory.
9. If asked which version of the product to install, the 32-bit or the 64-bit, choose the one appropriate to your environment.
10. If asked about the license information that T-Server is to use: specify either the full path to, and the name of, the license file, or the license server parameters.

11. As soon as the installation process is finished, a message appears announcing that installation was successful. The process places T-Server in the directory with the name specified during the installation.

End of procedure

Next Steps

- To verify manual installation, go to “Verifying the installation of T-Server” on [page 47](#).
- To test your configuration and installation, go to Chapter 5, “Starting and Stopping T-Server Components,” on [page 119](#), and try it out.
- To configure and install redundant T-Servers, see Chapter 3, “High-Availability Deployment,” on [page 49](#).
- To install T-Servers for a multi-site environment, proceed to Chapter 4, “Multi-Site Support,” on [page 61](#).

Procedure: Installing T-Server on Windows

Start of procedure

1. In the directory to which the T-Server installation package was copied, locate and double-click `Setup.exe` to start the installation.
2. When prompted, specify the connection parameters to the Configuration Server associated with this T-Server.
3. When prompted, select the T-Server Application you configured in “Configuring T-Server” on [page 43](#) from the list of applications.
4. Specify the license information that T-Server is to use: either the full path to, and the name of, the license file, or the license server parameters.
5. Specify the destination directory into which T-Server is to be installed.
6. Click `Install` to begin the installation.
7. Click `Finish` to complete the installation.

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

End of procedure

Next Steps

- To verify manual installation, go to “Verifying the installation of T-Server” on [page 47](#).

- To test your configuration and installation, go to Chapter 5, “Starting and Stopping T-Server Components,” on [page 119](#), and try it out.
- To configure and install redundant T-Servers, see Chapter 3, “High-Availability Deployment,” on [page 49](#).
- To install T-Servers for a multi-site environment, proceed to Chapter 4, “Multi-Site Support,” on [page 61](#).

Procedure:

Verifying the installation of T-Server

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

Prerequisites

- [Procedure: Installing T-Server on UNIX, on page 45](#)
- [Procedure: Installing T-Server on Windows, on page 46](#)

Start of procedure

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

End of procedure

Next Steps

At this point, you have configured and installed T-Server using Configuration Manager. If you want to test your configuration and installation, go to Chapter 5, “Starting and Stopping T-Server Components,” on [page 119](#), and try it out. Otherwise, if you want to configure and install redundant T-Servers, see Chapter 3, “High-Availability Deployment,” on [page 49](#). If you want to install T-Servers for a multi-site environment, proceed to Chapter 4, “Multi-Site Support,” on [page 61](#).

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 29](#) 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 50](#)
- [Hot Standby Redundancy Type, page 51](#)
- [Prerequisites, page 53](#)
- [Warm Standby Deployment, page 54](#)
- [Hot Standby Deployment, page 56](#)
- [Next Steps, page 60](#)

Warm Standby Redundancy Type

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

Warm Standby Redundancy Architecture

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

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

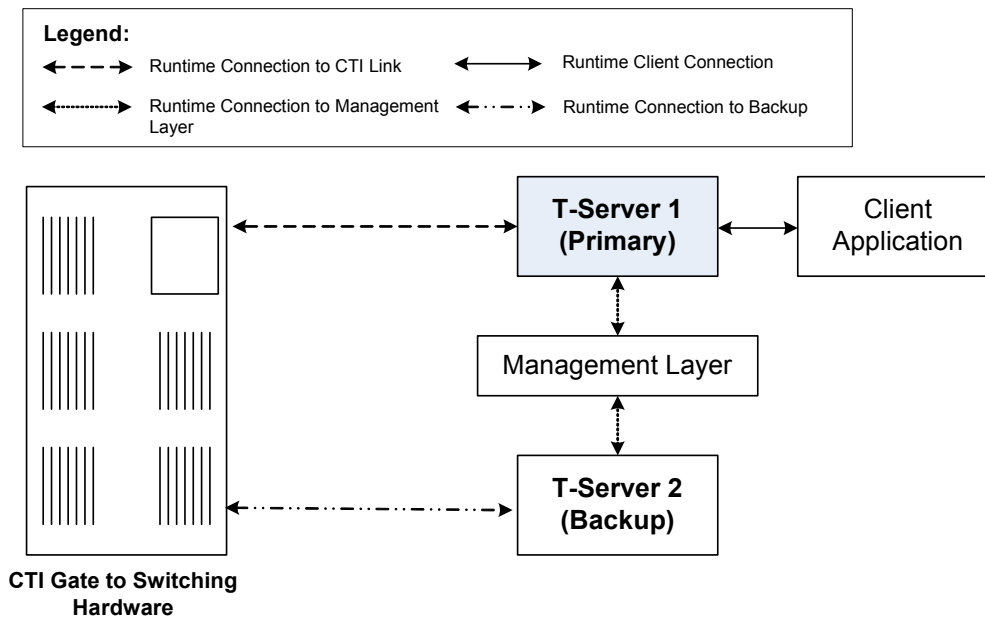


Figure 5: Warm Standby Redundancy Architecture

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

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

Hot Standby Redundancy Type

Genesys uses the expression *hot standby* to describe the redundancy type in which a backup server application remains initialized, clients connect to both the primary and backup servers at startup, and the backup server data is synchronized from the primary server. Data synchronization and existing client connections to the backup guarantee higher availability of a component. (See Figure 6 on [page 52](#).)

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

Hot Standby Redundancy Architecture

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

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

The details of hot standby redundancy implementation between T-Servers and their switches vary depending on switch support for multiple CTI links. If your T-Server supports hot standby (see Table 1 on [page 29](#)), refer to Part Two for detailed information on the available hot standby schema.

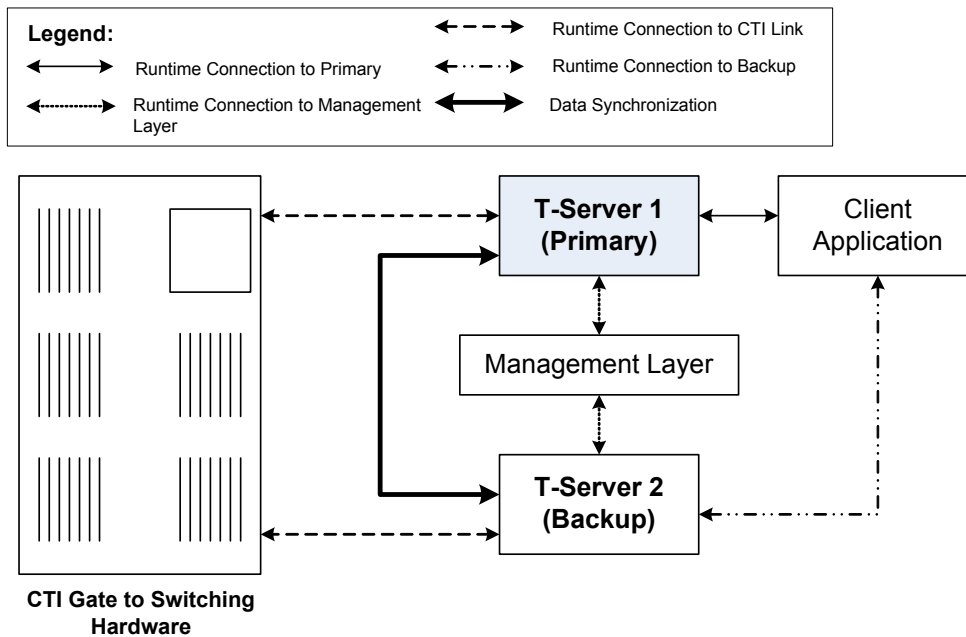


Figure 6: Hot Standby Redundancy Architecture

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 63](#) for ISCC feature descriptions.

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

However, keep the following hot standby limitations in mind:

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

Prerequisites

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

Requirements

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

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

Synchronization Between Redundant T-Servers

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

Configuration Warnings

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

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

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

Warm Standby Deployment

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

General Order of Deployment

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

1. Configure two T-Server `Application` objects as described in “Configuration of T-Server” on [page 43](#).
2. Make sure the `Switch` object is configured for the switch these T-Servers should serve, as described in “Configuration of T-Server” on [page 43](#).
3. Modify the configuration of the primary and backup T-Servers as instructed in the following sections.

After completing the configuration steps, ensure that both T-Servers are installed (see [page 56](#)).

Modification of T-Servers for Warm Standby

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

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

Procedure:

Modifying the primary T-Server configuration for warm standby

Start of procedure

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

End of procedure

Next Steps

- [Procedure: Modifying the backup T-Server configuration for warm standby](#), on page 56

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

Start of procedure

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

End of procedure

Warm Standby Installation of Redundant T-Servers

The installation of a redundant T-Server is the same as that for the stand-alone T-Server. If you have not installed the primary and backup T-Servers yet, follow the instructions in “Installation of T-Server” on [page 44](#) for both installations.

Hot Standby Deployment

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

General Order of Deployment

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

1. Configure two T-Server Applications objects as described in “Configuring T-Server” on [page 43](#).

2. Make sure the `Switch` object is configured for the switch these T-Servers should serve, as described in “Configuration of Telephony Objects” on [page 40](#).
3. Modify the configuration of the primary and backup T-Servers as instructed in the following sections.

After completing the configuration steps, ensure that both T-Servers are installed (see [page 60](#)).

Table 1 on [page 29](#) summarizes hot standby redundancy support in various T-Servers. For detailed, up-to-date information on the subject, see the *Genesys Supported Media Interfaces Reference Manual* located on the Technical Support website at

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

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 44](#).

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

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

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

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

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

End of procedure

Next Steps

- [Procedure: Modifying the backup T-Server configuration for hot standby, on page 59](#)

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 44](#).

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

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

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

End of procedure

Hot Standby Installation of Redundant T-Servers

The installation of a redundant T-Server is the same as that for the stand-alone T-Server. If you have not installed the primary and backup T-Servers yet, follow instructions in “Installation of T-Server” on [page 44](#) for both installations.

Next Steps

At this point, you have learned how to configure and install redundant T-Servers. Go to Chapter 5, “Starting and Stopping T-Server Components,” on [page 119](#), to test your configuration and installation, or continue with Chapter 4, “Multi-Site Support,” on [page 61](#), for more possibilities.

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 62](#)
- [ISCC Call Data Transfer Service, page 63](#)
- [ISCC/Call Overflow 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 104](#)
- [Configuring Multi-Site Support, page 104](#)
- [Next Steps, page 118](#)

Note: Each switch/T-Server combination offers different multi-site options. For details describing your specific switch/T-Server environment, refer to Chapter 14, “T-Server Common Configuration Options,” on [page 337](#).

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, CallType, and CallHistory). The following T-Server features support this capability:
 - ISCC Call Data Transfer Service (active external routing)—when requested by a T-Server client by specifying the desired destination in the location parameter, and also with various ISCC strategies performed by direct dial or by using the Transfer Connect Service. See “ISCC Transaction Types” on [page 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 98](#))
 - 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 call topology loops and multiple reappearances of the same call instance. This approach ensures that all T-Servers involved with the call report the same ConnID, and also optimizes telephony trunk allocation by preventing trunk tromboning.

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

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

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

ISCC Call Data Transfer Service

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

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

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

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

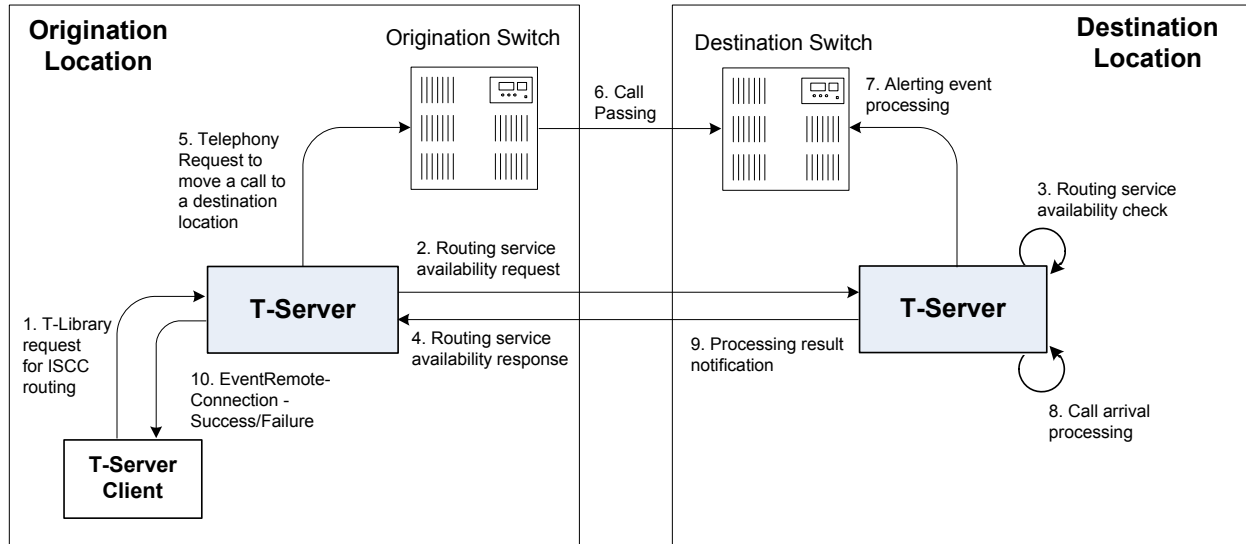


Figure 7: Steps in the ISCC Process

ISCC Call Flows

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

Step 1

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

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

Step 2

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

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

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

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

Note: For more information on Access Codes and Default Access Code, see “Switches and Access Codes” on [page 106](#).

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

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

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

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

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

Step 3

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

Note: The `resource-allocation-mode` and `resource-load-maximum` configuration options determine how resources are allocated. For option descriptions, refer to Chapter 14, “T-Server Common Configuration Options,” on [page 337](#) for option descriptions.

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

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

Step 4

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

Step 5

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

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

Step 6

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

Step 7

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

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

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

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

Step 8

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

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

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

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

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

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

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

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

Step 9

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

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

Step 10

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

Client-Controlled ISCC Call Flow

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

Step 1

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

Step 2

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

Step 3

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

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

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

Step 4

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

Step 5

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

Step 6

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

Step 7

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

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

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

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

ISCC Transaction Types

As switches of different types provide calls with different sets of information parameters, a single mechanism for passing call data between the switches is not feasible in some cases. Therefore, the ISCC feature supports a number of mechanisms for passing call data along with calls between locations. This section describes ISCC transaction type principles, identifies which transaction types are supported for each T-Server, and defines each transaction type (beginning with “direct-ani” on [page 71](#)).

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 71](#). 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 14, “T-Server Common Configuration Options,” on [page 337](#) 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 71](#)
- `direct-notoken`, [page 73](#)
- `dnis-pool`, [page 74](#)
- `pullback`, [page 75](#)
- `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 72](#)
- `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 call attribute is taken as the parameter for call matching. Properly configured switches and trunks can keep the ANI attribute when a call is transferred over the network. T-Server can use this network feature for call matching.

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

Typically, the ANI attribute represents the original call identifier (customer phone number), which guarantees that the attribute remains unique. However, you can use the `non-unique-ani` resource type to block ISCC from matching calls based on an ANI that is known to be non-unique. (See “Configuring access resources for non-unique ANI” on [page 115](#) for details.)

direct-callid

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

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

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

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

direct-network-callid

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

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

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

direct-uui

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

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

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

direct-notoken

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

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

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

dnis-pool

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

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

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

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

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

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

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

In Load-Balancing Mode

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

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

pullback

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

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

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

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

reroute

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

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

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

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

route

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

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

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

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

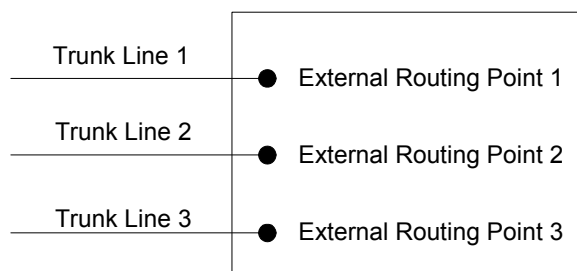


Figure 8: Point-to-Point Trunk Configuration

Note: Dedicated DNs of the External Routing Point type must be configured in a switch. See “Configuring Multi-Site Support” on [page 104](#).

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](#).

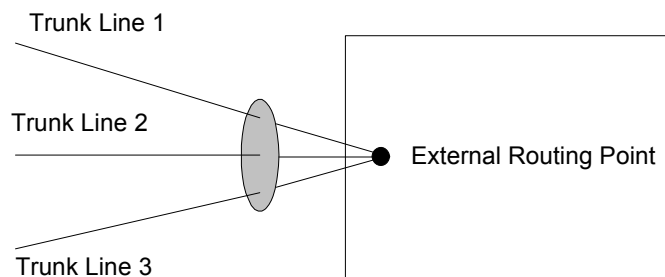


Figure 9: Multiple-to-Point Trunk Configuration

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. You must also ensure that the trunks involved do not drop this data.

T-Server Transaction Type Support

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

Table 3: T-Server Support of Transaction Types

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

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

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

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

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

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

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

- Not supported in the case of function `TRouteCall` on a Virtual Routing Point: a Routing Point can be simulated using a hunt group with calls being deflected or transferred from the hunt-group member when routing. When a two-step (typically mute) transfer is used on such a hunt-group member, `CallID` and `ANI` usually change; thus, the `direct-callid` and `direct-ani` types do not work.
- 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 `UUUI` 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.
- Supported only for `TRouteCall` requests made from a Native Routing Point.
- Not supported if a `TMakeCall` request is made.
- Not supported if a `TInitiateTransfer` or `TInitiateConference` request is made from an outgoing call on a device.
- SIP Server supports the `direct-uuui` type.

Transfer Connect Service Feature

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

Procedure: Activating Transfer Connect Service

Start of procedure

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

ISCC uses this DN as an intermediate step when sending calls to the remote location. The DN that is configured as `tcs-queue` receives attached data indicating the Feature Access Code (FAC) needed to reach the remote site. After a call is directed to the DN with data, a monitoring application takes the data and generates the required DTMF (dual-tone multifrequency) tones to redirect the call through the network to the remote location.

5. When you are finished, click Apply.
6. Click OK to save your changes and exit the Properties dialog box.

End of procedure

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

ISCC/Call Overflow Feature

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Note: When the ISCC/COF feature is in use, the Number Translation feature becomes active. For more information on feature configuration, see “Number Translation Feature” on [page 87](#).

ISCC/COF Call Flow

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

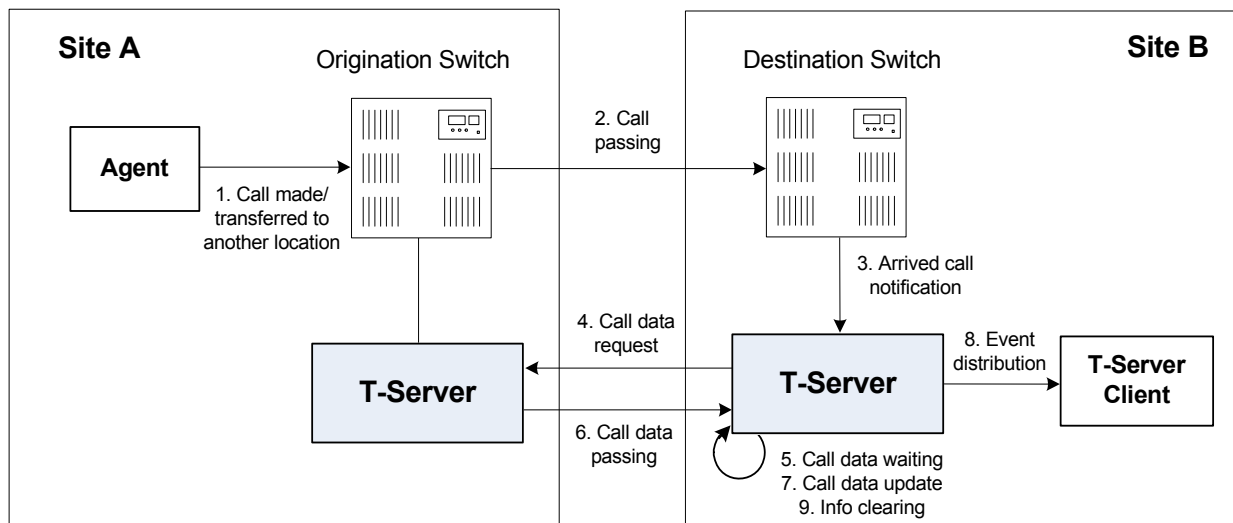


Figure 10: Steps in the ISCC/COF Process

Step 1

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

Step 2

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

Step 3

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

Step 4

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

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

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

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

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

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

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

Step 5

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

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

Step 6

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

Step 7

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

Step 8

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

Step 9

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

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

Number Translation Feature

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

The translation process involves two algorithms, one for rule selection and the other for the actual translation. Through the first algorithm, T-Server selects a rule that will be used for number translation. Through the second algorithm, T-Server translates the number according to the selected rule definition. See “Number Translation Rules” on [page 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 95](#).

Number Translation Rules

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

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

Using ABNF for Rules

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

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

Common Syntax Notations

Syntax notations common to many of these rules include:

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

Component Notations

Component notations include:

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

where:

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

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

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

where:

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

where:

- `ALPHA` indicates that letters can be used in the name for the rule option.
- `DIGIT` indicates that numbers can be used in the name for the rule option.
- `"-"` indicates that a dash (-) can also be used in the option name, for example, `rule-01`.
- `in-pattern = 1*(digit-part / abstract-group)`

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 = 3222332.

T-Server formats the output string as 3222332.

Example 4 T-Server receives input number 19253227676.

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

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

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

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

T-Server formats the output string as 919253227676.

Example 5 T-Server receives input number 4089137676.

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

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

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

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

T-Server formats the output string as 804079137676.

Example 6 T-Server receives input number 011441112223333.

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

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

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

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

T-Server formats the output string as 9011441112223333.

Procedure: Configuring Number Translation

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

Overview

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

Start of procedure

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

For the option description and its valid values, see Chapter 14, “T-Server Common Configuration Options,” on [page 337](#).

6. When you are finished, click Apply.
7. Click OK to save your changes and exit the Properties dialog box.

End of procedure

Network Attended Transfer/Conference Feature

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

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

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

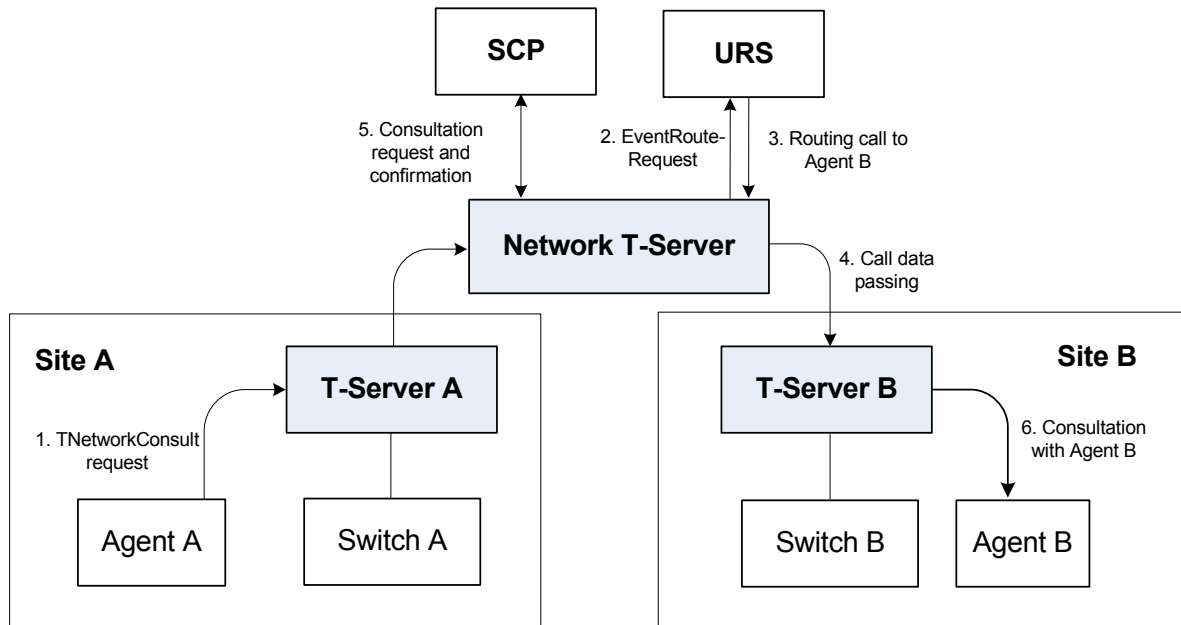


Figure 11: Steps in the NAT/C Process in URS-Controlled Mode

Step 1

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

Step 2

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

Step 3

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

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

Step 4

The Network T-Server proceeds to obtain the access number from T-Server B, and passes the call data to T-Server B. (See “ISCC Call Data Transfer Service” on [page 63](#) for details.)

Step 5

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

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

Step 6

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

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

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

Event Propagation Feature

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

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

User Data Propagation

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

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

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

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

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

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

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

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

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

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

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

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

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

Party Events Propagation

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

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

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

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

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

Switch Partitioning

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

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

Site A

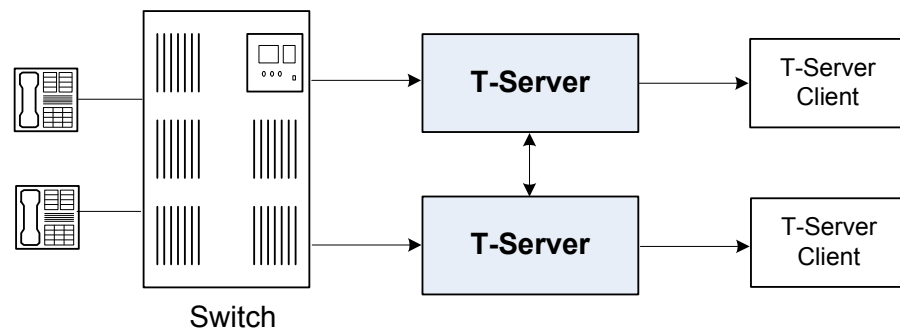


Figure 12: Switch Partitioning Architecture

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

In addition, T-Server supports `LocalCallType` and `PropagatedCallType` attributes, which depend on the [propagated-call-type](#) configuration option setting for reporting. See the option description on [page 342](#).

To control race conditions that may occur in the switch-partitioned environment, use the `epp-tout` configuration option (see [page 357](#)).

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

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

[Table 5](#) shows the T-Server types that support switch partitioning.

Table 5: T-Server Support for Switch Partitioning

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

Event Propagation Configuration

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

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

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

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

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

Procedure:**Activating Event Propagation: basic configuration**

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

Start of procedure

1. Open the T-Server Application's Properties dialog box.
2. Click the Options tab.
3. Open the extrouter section.
4. Set the [event-propagation](#) option to the list value.
This setting enables User Data propagation. If you need to enable Party Events propagation, perform Step 5.
5. Set the [use-data-from](#) option to the current value.
This setting enables Party Events propagation.
For the option description and its valid values, see Chapter 14, “T-Server Common Configuration Options,” on [page 337](#).
6. When you are finished, click Apply.
7. Click OK to save your changes and exit the Properties dialog box.

End of procedure**Next Steps**

- For advanced feature configuration, do the following procedure:
[Procedure: Modifying Event Propagation: advanced configuration](#), on [page 102](#)

Procedure:**Modifying Event Propagation: advanced configuration**

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

Prerequisites

- [Procedure: Activating Event Propagation: basic configuration](#), on [page 102](#)

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 107](#) for instructions. If no Access Codes are configured, see [page 108](#) for instructions.

4. In the Switch Access Code Properties dialog box that opens, specify a value for the ISCC Protocol Parameters field as follows:
 - To enable distribution of both user data associated with the call and call-party-associated events¹, type:
propagate=yes
which is the default value.
 - To enable distribution of user data associated with the call and disable distribution of call-party-associated events, type:
propagate=udata
 - To disable distribution of user data associated with the call and enable distribution of call-party-associated events, type:

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

- propagate=party
 - To disable distribution of both user data associated with the call and call-party-associated events, type:
propagate=no
- 5. Click OK to save configuration updates and close the Switch Access Code Properties dialog box.
- 6. Click Apply and OK to save configuration updates and close the Switch Properties dialog box.

End of procedure

ISCC Transaction Monitoring Feature

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

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

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

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

Configuring Multi-Site Support

Prior to configuring T-Server to support multi-site operation, you must read the "Licensing Requirements" on [page 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 name of each T-Server application, port assignments, and switch names), and use Configuration Manager to create and partially configure each T-Server object. Review multi-site option values in the “extrouter Section” on [page 347](#) 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 112](#) for details.

Applications

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

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

Procedure: Configuring T-Server Applications

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

Start of procedure

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

- Connection Protocol
 - Local Timeout
 - Remote Timeout
 - Trace Mode
4. Click the Options tab. Create a new section called extrouter or open an existing section with this name.

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

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

Note: For a list of options and valid values, see “extrouter Section” on [page 347](#), in the “T-Server Common Configuration Options” chapter in Part Two of this document.

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

End of procedure

Next Steps

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

Switches and Access Codes

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

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

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

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

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

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

Note: When migrating from previous releases of T-Servers to 8.1, or when using T-Servers of different releases (including 8.1) in the same environment, see “Compatibility Notes” on [page 111](#).

Procedure: Configuring Default Access Codes

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

Prerequisites

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

Start of procedure

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

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

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

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

End of procedure

Next Steps

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

Procedure: Configuring Access Codes

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

Prerequisites

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

Start of procedure

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

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

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

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

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

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

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

Table 6: Target Type: ISCC Protocol Parameters

ISCC Protocol Parameters	Description
dnis-tail=<number-of-digits>	Where number-of-digits is the number of significant DNIS digits (last digits) used for call matching. 0 (zero) matches all digits.
propagate=<yes, udata, party, no>	Default is yes. For more information, see “Modifying Event Propagation: advanced configuration” on page 102 .
direct-network-callid=<>	For configuration information, see Part Two of this document. (Use Table 4 on page 84 to determine if your T-Server supports the direct-network-callid transaction type.)

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

Table 7: Target Type: ISCC Call Overflow Parameters

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

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

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

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

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

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

8. When you are finished, click Apply.

End of procedure

Next Steps

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

Compatibility Notes

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

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

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

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

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

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

These values are fully compatible with the transaction types supported in T-Server release 5.1.

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

DNs

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

Procedure: Configuring access resources for the route transaction type

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

Prerequisites

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

Start of procedure

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

In determining an access number for the Routing Point, T-Server composes it of the values of the following properties (in the order listed):

- a. Access number (if specified).
- b. Switch access code from the switch of the origination party to the switch to which the Routing Point belongs, concatenated with its **Association** (if the **Association** value is specified).
- c. Switch access code from the switch of the origination party to the switch to which the Routing Point belongs, concatenated with the number for the DN.
- d. Default access code of the switch to which the Routing Point belongs, concatenated with its **Association** (if the **Association** value is specified).
- e. Default access code of the switch to which the Routing Point belongs, concatenated with the number for the DN.

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

6. When you are finished, click **Apply**.

End of procedure

Procedure:

Configuring access resources for the dnis-pool transaction type

Purpose: To configure dedicated DN's required for the dnis-pool transaction type.

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's **Properties** dialog box, specify the number of the configured DN as the value of the **Number** field. This value must be a dialable number on the switch.
3. Select **Access Resource** as the **Type** field and type **dnis** as the value of the **Resource Type** field on the **Advanced** tab.
4. Click the **Access Numbers** tab. Click **Add** and specify these **Access Number** parameters:
 - Origination switch.
 - Access number that must be dialed to reach this DN from the origination switch.

An access number for the access resource is determined in the same manner as for the route access resource.

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

End of procedure

Procedure:

Configuring access resources for direct-* transaction types

Start of procedure

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

You can select the **Use Override** check box on the **Advanced** tab to indicate whether the override value should be used instead of the number value to dial to the DN. You must specify this value if the DN has a different DN name and dialable number. In fact, this value is required for T-Servers for some switch

types—such as Aspect ACD, Nortel Communication Server 2000/2100, and Spectrum.

End of procedure

Procedure: Configuring access resources for ISCC/COF

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

Start of procedure

Note: Use Table 4 on [page 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.
2. On the **General** tab of the **DN Properties** dialog box, enter the name of the configured DN in the **Number** field.

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

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

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

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

End of procedure

Procedure: Configuring access resources for non-unique ANI

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

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

Start of procedure

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

End of procedure

Procedure:**Modifying DNs for isolated switch partitioning**

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

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

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

Start of procedure

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

7. When you are finished, click Apply.

End of procedure

Configuration Examples

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

Multiple Transaction Types

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

In this example, you configure an origination and a destination switch for as described in “Switches and Access Codes” on [page 106](#).

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 112](#).
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 108](#).

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 108](#).

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

Next Steps

Continue with Chapter 5, “Starting and Stopping T-Server Components,” on [page 119](#) to test your configuration and installation.

5

Starting and Stopping T-Server Components

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

- [Command-Line Parameters, page 119](#)
- [Starting and Stopping with the Management Layer, page 121](#)
- [Starting with Startup Files, page 122](#)
- [Starting Manually, page 123](#)
- [Verifying Successful Startup, page 129](#)
- [Stopping Manually, page 129](#)
- [Starting and Stopping with Windows Services Manager, page 130](#)
- [Next Steps, page 130](#)

Command-Line Parameters

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

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

Common command-line parameters are as follows:

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

-l	<p>The license address. Use for the server applications that check out technical licenses. Can be either of the following:</p> <ul style="list-style-type: none"> • The full path to, and the exact name of, the license file used by an application. For example, -l /opt/mlink/license/license.dat. • The host name and port of the license server, as specified in the SERVER line of the license file, in the port@host format. For example, -l 7260@ctiserver. <p>Note: Specifying the License Manager's host and port parameter eliminates the need to store a copy of a license file on all computers running licensed applications.</p>
-V	<p>The version of a Framework component. Note that specifying this parameter does not start an application, but returns its version number instead. You can use either uppercase or lowercase.</p>
-nco X/Y	<p>The Nonstop Operation feature is activated; X exceptions occurring within Y seconds do not cause an application to exit. If the specified number of exceptions is exceeded within the specified number of seconds, the application exits or, if so configured, the Management Layer restarts the application. If the -nco parameter is not specified, the default value of 6 exceptions handled in 10 seconds applies. To disable the Nonstop Operation feature, specify -nco 0 when starting the application.</p>
-lmspath	<p>The full path to log messages files (the common file named common.lms and the application-specific file with the extension *.lms) that an application uses to generate log events. This parameter is used when the common and application-specific log message files are located in a directory other than the application's working directory, such as when the application's working directory differs from the directory to which the application is originally installed.</p> <p>Note that if the full path to the executable file is specified in the startup command-line (for instance, c:\gcti\multiserver.exe), the path specified for the executable file is used for locating the *.lms files, and the value of the lmspath parameter is ignored.</p>
- transport-port <port number>	<p><port number> is the port number that a client will use for its TCP/IP connection to Configuration Server. See the Client-Side Port Definition section in the <i>Genesys 8.x Security Deployment Guide</i> for more information.</p>
- transport-address <IP address>	<p><IP address> is the IP address that a client will use for its TCP/IP connection to Configuration Server. See the Client-Side Port Definition section in the <i>Genesys 8.x Security Deployment Guide</i> for more information.</p>

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

Starting and Stopping with the Management Layer

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

Start of procedure

1. Open the T-Server Application's Properties dialog box.
2. Click the Start Info tab.
3. Specify the directory where the application is installed and/or is to run as the Working Directory.
4. Specify the name of the executable file as the command-line.
5. Specify command-line parameters as the Command-Line Arguments.
The command-line parameters common to Framework server components are described on [page 119](#).
6. When you are finished, click Apply.
7. Click OK to save your changes and exit the Properties dialog box.

End of procedure

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

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

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

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

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

Starting with Startup Files

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

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

When using a startup file, verify that the startup parameters the installation script inserted in the startup file are correct. Use the following instructions for UNIX and Windows to start those application for which startup files are created. See the appropriate sections in “Starting Manually” on [page 123](#) 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 119](#).

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

`-app "T-Server Nortel"`

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

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

Table 9: T-Server and HA Proxy Executable Names

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

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

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

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

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

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

HA Proxy

If you do not use HA Proxy in your Genesys implementation, proceed to “T-Server” on [page 127](#).

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 119](#).

Procedure: Starting HA Proxy on UNIX manually

Start of procedure

1. Go to the directory where HA Proxy is installed and type the following command-line:
`ha_proxy_<switch> -host <Configuration Server host>
 -port <Configuration Server port> -app <HA Proxy Application>`
2. Replace `ha_proxy_<switch>` with the correct HA Proxy executable name, which depends on the type of the switch used.
 Table 9 on [page 124](#) lists HA Proxy executable names for supported switches.

End of procedure

Procedure: Starting HA Proxy on Windows manually

Start of procedure

1. Start HA Proxy from either the Start menu or the MS-DOS window. If using the MS-DOS window, go to the directory where HA Proxy is installed and type the following command-line:
`ha_proxy_<switch>.exe -host <Configuration Server host> -port
 <Configuration Server port> -app <HA Proxy Application>`
2. Replace `ha_proxy_<switch>.exe` with the correct HA Proxy executable name, which depends on the type of the switch used.
 Table 9 on [page 124](#) 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 119](#).

Procedure: Starting T-Server on UNIX manually

Start of procedure

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

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

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

Table 9 on [page 124](#) lists T-Server executable names for supported switches.

End of procedure

Procedure: Starting T-Server on Windows manually

Start of procedure

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

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

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

Table 9 on [page 124](#) lists T-Server executable names for supported switches.

End of procedure

Verifying Successful Startup

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

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

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

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

Stopping Manually

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

Procedure: Stopping T-Server on UNIX manually

Start of procedure

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

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

End of procedure

Procedure: Stopping T-Server on Windows manually

Start of procedure

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

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

End of procedure

Starting and Stopping with Windows Services Manager

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

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

where the command-line parameters common to Framework server components are described on [page 119](#) and

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

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

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

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

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

Next Steps

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



Part

2

T-Server Configuration

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

- Chapter 6, “Alcatel A4400 Switch-Specific Configuration,” on [page 135](#), 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 T-Server Features,” on [page 171](#), describes which features this T-Server supports, including T-Library functionality, use of the Extensions attribute, and error messages.
- Chapter 8, “Supported High Availability (HA) Configurations,” on [page 263](#), provides switch-specific information about high availability.
- Chapter 9, “Using Alcatel A4400 Routing Services Interface,” on [page 267](#), describes how to use the A4400’s Routing Services Interface.
- Chapter 10, “Alcatel A4400 Call Flows,” on [page 289](#) gives additional detail about T-Server-specific calls flows where they are not covered in Genesys call model documentation.
- Chapter 11, “Connecting GVP:EE 6.5.5 to Alcatel A4400/OXE,” on [page 297](#) describes how to configure GVP for the A4400/OXE.
- Chapter 12, “Troubleshooting,” on [page 309](#), describes how to resolve some of the problems you may experience with this T-Server.
- Chapter 13, “Common Configuration Options,” on [page 315](#), describes the log configuration options that are common to all Genesys server applications.
- Chapter 14, “T-Server Common Configuration Options,” on [page 337](#), describes the configuration options that are common to all T-Server types, including options for multi-site configuration.

- Chapter 15, “Configuration Options in T-Server for Alcatel A4400/OXE,” on [page 365](#), describes configuration options specific to this T-Server, including the link-related options—those that address the interface between T-Server and the switch.

New in T-Server for Alcatel A4400/OXE

The following new features are available in the initial 8.1 release or have been added in the most recent 8.0 release of T Server for Alcatel A4400/OXE:

- **Support to add GCTI_PARTY_NAME and GCTI_OTHER_DEVICE_NAME information to calls.** Starting with version 8.0.002.02, GCTI_PARTY_NAME and GCTI_OTHER_DEVICE_NAME information is now added to calls delivered to a device when it is provided by the PBX.
- **Support for Calling Party Name.** Starting with version 8.0.002.02, the private event Calling Party Name is supported when reported by the switch.
- **Support to redefine the call state that T-Server reports in EventReleased in a TMakePredictiveCall.** Starting with version 8.0.002.02, users can now redefine the call state that T-Server reports in EventReleased in a TMakePredictiveCall scenario when the call fails or is dropped.
- **Support to set Voice Activity Detection state.** Starting with version 8.0.002.04, T-Server now allows a Voice Activity Detection state to be set for a predictive call (when TMakePredictiveCall is requested).
- **Support for enhanced ISCC/COF flexible call matching.** Starting with version 8.0.100.07, T-Server now supports enhanced ISCC/COF flexible call matching.
- **Support to report the PBX UTC time.** Starting with version 8.0.100.07, T-Server now reports a new key PBXTimeStamp in AttributeExtensions that corresponds to the PBX UTC time in seconds, if such information is provided by the PBX.
- **Support for Calling Line Name information reporting.** Starting with version 8.0.100.07, T-Server now reports information regarding Calling Line Name in respective events in a key CallerID of AttributeExtensions if such information is provided by the PBX.
- **Support for the suppression of the reporting for call routing events.** Starting with version 8.0.100.07, T-Server now suppresses the reporting of call routing events in the scenario, where the OXE VAD feature is used for outbound dialing and Outbound Contact Server (OCS) is configured to request T-Server to automatically overflow calls detected as Fax and/or Answering Machine to another destination.

- **Support for DNs of type Route Queue.** Starting with version 8.0.100.07, T-Server now supports configuring DNs of type Route Queue in the Configuration Layer.
- **Support for LMS event.** Starting with version 8.0.100.07, T-Server now supports the following new LMS event that is generated if T-Server does not enable routing capabilities on a routing device.

```
36104|STANDARD|GCTI_RQM_004|Device [device] route register failed,  
error code [error code]([error code string])
```
- **Support to provide the pickup device number if provided by the PBX.** Starting with version 8.0.100.07, T-Server now reports a CallPickup key in AttributeExtensions with the value of the pickup device number, if the value is provided by the PBX; otherwise it puts the value unknown in the EventReleased message if the message is generated as a result of call pickup.
- **Support for account codes.** Starting with version 8.0.100.07, T-Server now supports the `acw-retain-call` and `acw-retain-lock` configuration options to enable the account codes functionality. See “Support for Account Codes” on [page 209](#).
- **Support for ISCC call tracking.** Starting with version 8.0.100.07, T-Server can now be configured to use the Global CallID (GCID) for ISCC call tracking. See “Support for Inter Server Call Control” on [page 204](#).
- **Support for operating systems.** T-Server is now supported on the following platforms:
 - AIX 7.1 64-bit
 - HP-UX Itanium (version 11i v3)
 - Red Hat Enterprise Linux 5 64-bit

Note: Configuration option changes that apply to your T-Server are described in “Changes from 8.0 to 8.1” on [page 411](#).

6

Alcatel A4400 Switch-Specific Configuration

This chapter presents switch-specific reference information for configuring T-Server for the Alcatel A4400 switch. It contains these sections:

- [Known Limitations, page 135](#)
- [Changes in Reporting Behavior from 7.x to 8.1, page 137](#)
- [Switch Terminology, page 139](#)
- [Support of Switch/CTI Environments, page 141](#)
- [Setting DN Properties, page 141](#)
- [Configuring Switch Timers, page 145](#)
- [Configuring Extensions in the PBX, page 146](#)
- [Configuring CCD Agents, page 147](#)
- [Advanced Agent Features, page 158](#)
- [Configuring CCD Objects, page 161](#)
- [Network Call ID Matching, page 164](#)
- [Routing Using Emulated Routing Points, page 164](#)

Known Limitations

1. The PBX does not support Do Not Disturb (DND) functionality through CTI, so setting DND can only be done from the phoneset. The PBX also provides no CSTA reporting to T-Server when DND is set manually on a telset.
2. T-Server does not support the use of parallel Hunt Groups.

3. When reporting is required for Hunt Group members, the Hunt Group itself must not be configured in Configuration Manager.
4. Partitioned-switch configuration is supported by T-Server with the restriction that RSI DNs must be assigned to a single T-Server—they cannot be shared between two T-Servers except in an HA configuration.
5. When prefix is used in `TMakeCall` to activate a switch feature, this prefix must begin with a `*` or `#`. This means the prefix must be configured accordingly on the switch.
6. T-Server does not issue `EventOnHook/EventOffHook` for devices configured as PCM ports.
7. The PBX does not send any events to T-Server when forwarding is set manually on any device.
8. Forwarding status cannot be preserved between `Login/Logout` states when agent substitution is set to `true`.
9. If a DN has `CallForward` set before login, and then login occurs, forwarding is cancelled.
10. Predictive dialing in HA configurations only works with releases 7.0.1x of T-Server and higher. It is not supported in prior releases.
11. T-Server does not support CCD Routing Pilots (CCD Pilots with the routing flag enabled).
12. Call scenarios and call tracking are not accurate unless all devices taking part in call distribution are monitored.
13. If the progress of `SingleStepTransfer` is interrupted by any request, T-Server may not report the scenario correctly.
14. When the switch partitioning feature is used, the following prerequisites are needed for ISCC bounced calls:
One physical switch configured in two partitions in Configuration Manager, and the value of the `extrouter` configuration option, `match-call-once`, should be set to `false`.
In the following scenario, for `EventRinging` on DN3, the `ConnID` is different from the `ConnID` used in the rest of the call. This is because of a limitation in ISCC with emulated single-step transfer. However, call control is maintained and any attached user data will be maintained.
 - DN1 on partition 1 calls DN2 on partition 2.
 - DN2 single-step transfers to DN3 back on partition 1.
15. Starting with release 8.0, when a device manually releases an outbound call (a call established with an external party), the `Extensions` attribute `ReleasingParty` will be `2 Remote` if the reporting of the releasing party is enabled in T-Server (when the value of the configuration option, `releasing-party-report` is set to `true`).

This is because the switch sends exactly the same `ConnectionCleared` event regardless of whether the external party releases the call or whether local party releases manually.

16. With Outbound Contact Server (OCS) release 7.6 or lower, OCS looks for `EventAbandoned` with a call state of `dropped` in order to correctly update a call if it is cleared as a result of the T-Server configuration option, `prd-dist-call-ans-time`.
Starting with T-Server release 8.0, T-Server reports `EventReleased` with call state `dropped` in this scenario.
When using OCS version 7.6 or lower with T-Server version 8.0, the configuration option, `prd-dist-call-ans-time`, must be set to 0 (zero) in order ensure correct call results on OCS.
17. When a Direct ANI strategy is used (when the value of the `extrouter` configuration option `cast-type` is set to `direct-ani`) and a call is initiated, or a consultation call is initiated, from an agent in substitution mode, T-Server is not able to match calls. The reason for this is that the call is made from a substituted device (ProACD number), and delivered with the ANI of the agent device, which do not match.
18. Only VPN overflow is supported between the T-Server nodes when targeting the CCD Pilot or Routing Services Interface (RSI) Routing Points. Private to public overflow is not supported.
19. Network optimization is not supported with CCD or RSI.
20. When using several A4400 PBXs connected through an ABC-F network, the date and time must be synchronized on all PBXs in order to ensure the proper PBX CSTA reporting to T-Server.
21. Routing from a RSI through DPNSS or DASS trunks is not supported.
22. T-Server 8.x supports only PBX versions 7.x and higher.

Changes in Reporting Behavior from 7.x to 8.1

1. The high-availability (HA) synchronization of emulated agents is not supported if the primary T-Server is version 8.0 and the backup T-Server is version 7.6, or earlier. Both primary and backup T-Servers must be version 8.0, or higher, to support this feature.
2. Starting in release 8.0, T-Server reports the `CallStateOk` attribute in `EventRinging` at the forwarding destination in the following scenario.
 - Immediate forwarding is set on a device.
 - A call from a RSI is routed to the device with forwarding set.
 - The call is delivered to the forwarding destination.

In version 7.6, or earlier, T-Server reported the `CallState` attribute in `EventRinging` as `Forwarded` on the forwarding destination in this same scenario.

3. Starting in release 8.0, when the `OnNoAnswer` subtype of a `TCallSetForward` request is invoked by the switch, T-Server reports the `CallState` attribute for `EventRinging` on the forwarding destination as `Redirected`. Previously in this scenario (for release 7.6 and earlier), T-Server reported the `CallState` attribute as `Forwarded`.
4. T-Server is not able to reliably generate an `EventError` in the following scenario:
 - A supervised route is made from an emulated Routing Point.
 - The supervised route is not answered.
 - A switchover is performed.
 - T-Server recalls the call to the Routing Point when the value set for the `supervised-route-timeout` configuration option expires.

This only occurs with a switchover when T-Server is running in high-availability (HA) with an 8.0 primary T-Server, and a 7.x backup T-Server.

5. In a scenario where a call is redirected to an internal device that has forwarding to another internal device, the `CallState` attribute in `EventRinging` as `Redirected` when the call arrives at the forwarding destination. This change applies to version 7.6 and later. Previously in this scenario, T-Server reported the `CallState` attribute as `Forwarded`.
6. When running a passive 7.6 T-Server, and performing a supervised emulated Routing Point, reporting for the passive T-Server is exactly the same as the main (control) T-Server.
Starting with release 8.0, the reporting is different in the passive T-Server since the transfer off the Hunt Group member is reported by the switch. For this reason, T-Server reports `EventDialing`, `EventEstablished`, and `EventReleased` events on the emulated Routing Point and `EventPartyChanged` on the routing destination.
7. When a call is made to an ACD Queue with several agents logged in, and the first agent does not answer, the switch overflows the call to another logged in agent. Prior to release 8.0, T-Server reported `EventReleased` on the agent that did not answer the call with a `CallState` attribute of `Forwarded`, and `EventRinging` for the new agent with a `CallState` attribute of `Forwarded`. Starting with release 8.0, T-Server reports `EventReleased` with a `CallState` attribute of `NoAnswer` and `EventRinging` with a `CallState` attribute of `Redirected`.
8. When using the switch hold call functionality with the setting `Do Not Ringback`, reporting is different when a held call is retrieved. Previously in release 7.6, or earlier, T-Server reported `EventPrivateInfo`

and EventRinging with a CallState attribute of 0k.

In release 8.0, or higher, T-Server reports EventPrivateInfo and EventRinging with CallState attribute of Redirected.

9. T-Server is not able to report predictive dialing situations correctly in the following scenario:
 - The primary T-Server is version 8.0.
 - The backup T-Server is version 7.6, or earlier.
 - The backup T-Server then becomes the primary T-Server.
10. The PBX provides limitations in reporting for the following scenario:
 - An internal call is made to an unmonitored device;
 - which then transfers to a RSI;
 - and completes the transfer while the call is on the RSI.

For this reason, with release 8.0 of T-Server, the origination DN receives two EventPartyChanged events. The first contains no value for the OtherDN attribute. The other contains the Routing Services Interface (RSI).

Release 7.6, and earlier, of T-Server reported only one EventPartyChanged on the origination DN with the OtherDN attribute reported as the trunk (prefixed with 'T').

Switch Terminology

Table 10 compares relevant A4400 switch terminology with Genesys terminology.

Table 10: Switch Terminology Comparison

Genesys Term	Alcatel A4400 Term
ACD	CCD
ACD Position	CCD Business Agent
ACD Queue	Super-queue (not a switch device) CCD Queue CCD Guide CCD Pilot CCD Processing Group RSI Processing Group
Agent ID used in CTI login request	Agent ID Signature

Table 10: Switch Terminology Comparison (Continued)

Genesys Term	Alcatel A4400 Term
Extension	ProACD device (including multiline device) Predictive dialing device PCM port Analog port Virtual device for routing Voice Activity Detection (VAD)
Position	Supervisor
Voice Treatment Port	Analog port PCM port
Trunk (unmonitored)	Unmonitored trunk
Trunk (monitored)	Not applicable
Routing Point	RSI Hunting Group Virtual device for routing
Group DN	Not applicable
Predictive dialing device	PCM port VAD port or device
Emulated Routing Point	Hunting Group
Emulated Routing Point member	Virtual Hunting Group member
Logon	Logon
Logoff	Logoff
Ready	Withdrawal (toggle)
NotReady	Withdrawal (toggle)
AfterCallWork	Wrap-up
ReasonCode	Unavailable/Withdrawal type

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 Supported Operating Environment Reference Manual](#). 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.

Setting DN Properties

[Table 11](#) indicates how to set the Genesys DN properties for the Alcatel A4400 PBX.

Table 11: Setting the Genesys DN Properties

Switch Device Type	Genesys DN Type	Switch-Specific Type	Association	Register	Comments
Station (including multiline)	Extension	Not applicable	Not applicable	See description	To avoid monitoring both stations and business agents on the switch, configure type extension in Local mode. Login service can still be used to activate an agent on this extension.
CCD Business Agent	ACD Position	Not applicable	Not applicable	True	

Table 11: Setting the Genesys DN Properties (Continued)

Switch Device Type	Genesys DN Type	Switch-Specific Type	Association	Register	Comments
CCD Pilot	ACD Queue	1	Not applicable	True	PBX does not provide reporting for Statistic pilots.
CCD Queue/CCD Guides RSI Processing Group	ACD Queue	1	Not applicable	False	<p>You can configure T-Server to provide extended reporting for the path of a call through the CCD, including the CCD Queue and CCD Guides to which the call is currently connected. To receive guides reporting, define the following devices in the Configuration Layer:</p> <ul style="list-style-type: none"> • Waiting Guides; 0WG1-0WG6 • IVR-in-Q Guides; 0IVR1-0IVR6 • Interactive Automated Attendant; 0IAA1-0IAA6 • After Guide 6; 0NOMORE • Presentation Guide; 0PRG • Blocking Guide; 0CLO
RSI ^a	Routing Point	2	Not applicable	True	Only Alcatel A4400 release 4.2, and higher, supports this type of routing.
Virtual Z	External Routing Point	1	Empty	True	A single virtual device used as an External Routing Point where only a single call is allowed on the Routing Point at any time.

Table 11: Setting the Genesys DN Properties (Continued)

Switch Device Type	Genesys DN Type	Switch-Specific Type	Association	Register	Comments
Predictive Dialing Device	Extension	4	Not applicable	True	Virtual device for emulation of TMakePredictiveCall. From switch release R4.1 onwards, you can associate this device type with the PBX VAD feature, to provide on-board call progress detection for TMakePredictiveCall. Previous switch releases provide only a limited set of call results. For more information, see “Support for Predictive Dialing” on page 182 .
PCM Port	Extension VTO Port	7	Not applicable	True	Used for virtual devices associated with time slots on a PCM (CAS) interface. Note: The Auto-Originate feature is not available for this type of devices from release 7.0.2 onwards.
Analog Port	Extension VTO Port	8	Not applicable	True	Provides special support for caller hang-up scenarios for analog IVR devices and analog dialing devices used by CPDServer.
Supervisor	Position	5	Not applicable	True	Provided for backward compatibility only. Configure type supervisor as a normal agent.

- a. Support for routing using pilots with the Routing Enabled feature is discontinued. All references to routing with pilots imply routing with Routing Services Interface (RSI).

Routing Without Using Switch-Routing Services

To perform routing independent of switch-routing services, use an emulated Routing Point implemented in T-Server for Alcatel A4400. The telephone extensions that belong to the Routing Point correspond with fictive devices (virtual devices) on the switch. T-Server reports all events for these fictive devices as if they were happening to the Routing Point.

When a call arrives on a fictive device on the switch, T-Server queries a route to use from an application registered for the Routing Point. In response, the client sends a `TRouteCall`. T-Server behavior is determined by the setting of options `supervised-route` and `supervised-route-timeout`. See [page 395](#).

[Table 12](#) details the configuration for an emulated Routing Point. For more information, see [pages 164](#).

Table 12: Routing Without Using Switch-Routing Services

Switch Device Type	DN Type	Switch-Specific Type	Association	Register	Comments
Virtual Device for Routing	Extension	2	Directory number of the Routing Point used for this extension.		Extension used to provide routing services on behalf of a Routing Point.
		6	Directory number of the Routing Point used for this extension.		As for type 2, except that <code>EventRouteRequest</code> is generated only after the device is put into the <code>Established</code> state.
	Routing Point	Not applicable	Not applicable		Implemented in T-Server.
Hunting Group	Routing Point	Not applicable	Not applicable		The Hunting Group with virtual device members is used to emulate routing services.

Configuring Switch Timers

Procedure:

Setting A4400 timers to support link functions

Purpose: To set switch timers to support link functions.

Start of procedure

1. In the PBX configuration, navigate to mgr/System/Go down hierarchy/Timers.
2. Use the recommended value for each timer as described in [Table 13](#).

Table 13: Timer Configuration on Alcatel A4400

Timer Configuration on the Alcatel A4400		
Timer Number	Recommended Value	Description
7		This timer controls how long the PBX waits before reconnecting a business consultation call that has not been answered. The default value is 30 seconds.
10	6	This timer defines the waiting time before connecting a call to an agent who is in headset mode.
12		This timer controls how long after a call (made to a wrong number and released by the PSTN) that the PBX generates a connection cleared event with a CallCancelled cause.
42	2	This timer controls the interval after a call is established, after which a consultation call can be made. If transfers or conferences are failing, a possible cause could be that this timer is set for too long a period. If not set to 2, a consultation call on behalf of a set that first did an external call may not work correctly.
95	2	This timer displays ISDN total costs. Set to 2 to display the ISDN total costs. If not set to 2, the monitoring of external incoming calls may not work correctly.

Table 13: Timer Configuration on Alcatel A4400 (Continued)

Timer Configuration on the Alcatel A4400		
137	0	This timer is used for charging information. If you are not using this feature, ensure that the timer is set to 0. Otherwise, after every inbound ISDN call, the PBX will send an additional originated/connection cleared event after it has reported the usual connection cleared events.
194	Any value from 1–99	Timer to avoid rebounds. If not set within the range specified, incorrect monitoring may occur.
238	2	Timer to play back the guide (CCD). If not set to 2, there is a delay when an agent requests NotReady (Withdrawal) using the key on the phone.
239	2	Timer to play back the busy tone (CCD). If not set to 2, there is a delay before the PBX actually puts a phone on-hook when a call is released while the phone is in Hands-Free mode.
282		This timer controls the maximum length of an announcement. The default value is 60 seconds.
313	10	This timer enables answer machine detection. A value of 0 disables this timer.
314	10	This timer controls the waiting duration for starting voice detection.
315	8	This timer controls the maximum duration of silence after voice detection.

End of procedure

Configuring Extensions in the PBX

An extension on the Alcatel A4400 is a nonagent telephony device, which includes both analog and digital ACD-authorized phone sets.

Procedure:

Configuring an extension in the PBX for agent login

Purpose: To configure an extension for agent login.

Start of procedure

1. In the PBX configuration, navigate to mgr/Users/Create.
2. Set the value of ACD Station to ACD-authorized phone set. This enables agents to log in.
An ACD-authorized phone set is an ACD-enabled extension.

End of procedure

Configuring CCD Agents

This section describes how to configure CCD agents.

Definitions

[Table 14](#) gives some definitions.

Table 14: CCD Agents—Definitions

Term	Definition
Agent in the CCD and RSI	You can attach an agent to several processing groups in the CCD, but an agent is only assigned to one processing group at any time. To be part of the system, an agent must log in on an ACD-authorized phone set. During this operation, the agent selects, or is automatically assigned to, a processing group (authentication by password may be requested).
Self-assigning agent	Self-assigning agents can select their own processing group from those to which they are attached.
Preassigned agent	When a supervisor controls an agent's entry into one of the processing groups to which the agent is attached, the agent is known as preassigned.
Processing group for the CCD and RSI	The processing group is a group device on the PBX on which agents log in and become available for the CCD. Although you can assign agents to several processing groups, they can only be logged in to one at any time.

Agent Login

When an agent logs in on a device, the extension (ACD-authorized device) is taken out of service, and the agent (Position) is taken into service on the same physical handset. See also “Agent Substitution” on [page 159](#).

Example

Agent 3000 wants to log in to processing group 3101 from extension 1000; agent 3000 has password 1234. The desktop application must be registered for both 1000 and 3000. (Processing group 3101 is not necessarily configured in Configuration Manager.)

The following login request must be issued from extension 1000:

```
RequestAgentLogin
AttrThisQueue '3101' (Processing Group)
AttrThisDN '1000' (Login Extension)
AttrAgentID '3000' (AgentID)
AttrPassword '1234'
```

Extension 1000 is taken out of service, and 3000 is taken into service on the same handset.

Note: The identity of the pilot that can distribute calls to this agent depends on the ACD configuration in the PBX

Preassigned and Supervisor Agents

The Alcatel A4400 PBX has a feature called Preassigned Agents. An agent who is preassigned can log in to the CCD, but cannot log in to a processing group (PG). Only a supervisor using the Call Center Supervisor (CCS) application can assign such an agent to a group. When an agent is logged in, but not assigned to a processing group, he or she is in the Preassigned state.

Note: Agents configured as supervisors in the PBX can enter a processing group from the Preassigned state and can move between groups, both manually and using CTI requests.

Table 15 defines the login/logout behavior of preassigned and supervisor agents when the preassign-agent-compat option is set to true.

Table 15: Behavior of Preassigned and Supervisor Agents (preassign-agent-compat = true)

Supervisor/ Preassigned Agent State	Request	Event	State on PBX After Request
Logged out	TAgentLogin ThisQueue = ""	EventAgentLogin ThisQueue not provided GCTI_PREASSIGNED_ AGENT=1	Logged in, preassigned (not entered in PG)
	Supervisor only TAgentLogin ThisQueue = PG	EventAgentLogin ThisQueue = PG	Logged in, entered in PG
Logged in, preassigned (not entered in PG)	Supervisor only TAgentLogin ThisQueue = PG	EventAgentLogin ThisQueue = PG	Logged in, entered in PG
	TAgentLogout ThisQueue = ""	EventAgentLogout	Logged out
Logged in and entered in PG	<i>Supervisor only</i> TAgentLogin ThisQueue = PG2	EventAgentLogout ThisQueue = PG EventAgentLogin ThisQueue = PG2	Logged in, entered in PG2.
	TAgentLogout ThisQueue = ""	EventAgentLogout ThisQueue = PG	Logged out
	TAgentLogout ThisQueue = PG	EventAgentLogout ThisQueue = PG EventAgentLogin	Logged in, preassigned (not entered in PG)

Table 16 defines the login/logout behavior of preassigned and supervisor agents when the `preassign-agent-compat` option is set to `false`.

Table 16: Behavior of Preassigned and Supervisor Agents (`preassign-agent-compat = false`)

Supervisor/ Preassigned Agent State	Request	Event	State on PBX After Request
Logged out	TAgentLogin ThisQueue = ""	EventAgentLogin ThisQueue not provided GCTI_PREASSIGNED_ AGENT=1	Logged in, preassigned (not entered in PG)
	Supervisor only TAgentLogin ThisQueue = PG	EventAgentLogin ThisQueue = PG	Logged in, entered in PG
Logged in, preassigned (not entered in PG)	Supervisor only TAgentLogin ThisQueue = PG	EventQueueLogout This Queue = "" EventAgentLogin ThisQueue = PG	Logged in, entered in PG
	TAgentLogout ThisQueue = ""	EventAgentLogout	Logged out
Logged in and entered in PG	<i>Supervisor only</i> TAgentLogin ThisQueue = PG2	EventAgentLogin ThisQueue = "" EventQueueLogout ThisQueue = PG1 EventAgentLogin ThisQueue = PG2 EventQueueLogout ThisQueue = ""	Logged in, entered in PG2.
	TAgentLogout ThisQueue = ""	EventAgentLogout ThisQueue = PG	Logged out
	TAgentLogout ThisQueue = PG	EventAgentLogin ThisQueue = "" EventQueueLogout ThisQueue = PG	Logged in, preassigned (not entered in PG)

Procedure: Configuring CCD agent devices in the PBX

Summary

You can assign an agent device to one or several processing groups on the PBX. The processing group is a device on the switch where agents log in and become available for the ACD.

Start of procedure

1. In the PBX configuration, navigate to mgr/Users/Create.
2. Assign the agent device to one or more processing groups.
3. Configure the agent device as shown in [Table 17](#). Although you can assign the agent device to several processing groups, the agent can be logged in to only one at any time.

Table 17: Agent Device Configuration

Entries	Value	Description
Directory Number	Agent Number	The Agent Number (ID) in the PBX.
Shelf/board/equipment address	255/255/255	An agent is not assigned to any hardware unless he or she logs in.
ACD Station	Agent/Supervisor	Defines whether this is a normal agent or supervisor.

End of procedure

Procedure: Configuring multiline devices in the PBX

Purpose: Configuring any proACD device as a multiline device.

Start of procedure

1. In the PBX configuration, navigate to mgr/Users/Progr. Keys.
2. Set the appropriate keys on the proACD device to assigned as multiline
3. Select Function = Multi-Line.

4. Enter the number of the proACD device for the Directory Number.

Note: This same number needs to be entered for every multiline key being configured.

5. Navigate to mgr/Applications/CCD/CCD RSI system parameters/, and then navigate to System Option - Attendant enquiry call on 2 lines.
6. Set the option System Option - Attendant enquiry call on 2 lines to True.
7. Navigate to mgr/Specific Telephone Services then to the option Enquir.call mod.on multilines.
 - A value of True means that a consultation call is initiated from the same multiline key being occupied by the primary call.
 - A value of False means that a consultation call is initiated from another idle multiline key

End of procedure

CCD Agent/Supervisor–Related T-Server Features

Procedure: Configuring Headset mode for agent handsets

Purpose: To enable agents to activate the Headset mode on the handset at login.

Start of procedure

1. In the PBX configuration, navigate to mgr/Users/Go down hierarchy/Progr. Keys
2. For every agent who is going to use this feature, configure a Headset button in the PBX by setting the value one of the programmable function keys headset.
3. Set the default Headset mode for this agent by setting the value of the `headset-mode` configuration option.
 If the value is set to true, agents are automatically logged in with the Headset mode (where a Headset button is configured).
 If the value is set to false, agents are logged in with the Headset mode switched off.

End of procedure

Procedure: Overriding Headset mode in extensions

Start of procedure

To override the default behavior of Headset mode defined by the `headset-mode` option, configure the `Extensions` attribute `GCTI_HEADSET_MODE` in any `TAgentLogin` request to have:

- A nonzero value (integer), to activate the Headset mode.
- A zero value, to deactivate the Headset mode.

End of procedure

Supervisor Help Services

Procedure: Configuring supervisor call behavior in T-Server

Purpose: To enable agents to request a connection to an available supervisor.

Summary

When this request is made, the PBX ignores the call destination in the request and selects an available supervisor for the call instead. If there is no supervisor agent logged in or available, the PBX rejects the request.

Start of procedure

1. In the PBX configuration, navigate to `mgr/Users/Go down hierarchy/?????`
2. To enable agents to request a connection to an available supervisor, set the value of the `supervisor-call-enable` configuration option to `true`.
3. To force all agent calls to be sent to an available supervisor, regardless if the number dialed, set the value of the `supervisor-call` configuration option to `true`.

End of procedure

Procedure: Overriding supervisor call settings in extensions

Start of procedure

To override the default settings defined in the [supervisor-call](#) configuration option, configure the `Extensions` attribute `GCTI_SUPERVISOR_CALL` to have:

- A nonzero (integer) value to send the call to a supervisor.
- A zero value to send the call to the defined destination.

End of procedure

Note: The `Extensions` attribute `GCTI_SUPERVISOR_CALL` can be used in the following requests:

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

Procedure: Configuring/canceling a supervisor help request

Purpose: To enable a CCD agent to request assistance from an available CCD supervisor.

Start of procedure

1. The agent requests assistance from an available CCD supervisor using a `TRequestPrivateService` request with Service Number 6 (see [“Request Supervisor Help”](#) in Table 24 on [page 227](#)).
2. When an agent requests supervisor assistance, the CCD finds an available supervisor. If successful, both the requesting agent and the selected supervisor are notified using an `EventPrivateInfo` event with Service Number 6 (see [“Supervisor Assistance Request Event”](#) in Table 24 on [page 227](#)). The `otherDN` attribute of the event contains the selected supervisor or the requesting agent, respectively.
3. The supervisor can initiate assistance by using either `“”` ([Configuring supervisor listening/step-in, page 155](#)) or [“Permanent Listening”](#) ([page 157](#)).

4. The agent can cancel a request for assistance by using a `TRequestPrivateService` request with Service Number 7 (see “[Cancel Requested Supervisor Help](#)” in Table 24 on [page 227](#)). This request must contain the `Extensions` attribute `GCTI_OTHER_DN` set to the number of the supervisor. Both agent and supervisor are notified by an `EventPrivateInfo` event with Service Number 7.

Note: The supervisor can use the same request with Service Number 7 to reject the service request from the agent. In this case, `GCTI_OTHER_DN` must be set to the number of the requesting agent.

End of procedure

Procedure: Configuring supervisor listening/step-in

Purpose: To enable the supervisor listening and step-in functionality.

Summary

T-Server provides agent-monitoring functionality—the Supervisor Step-In feature—for supervisor agents in the PBX. This feature allows supervisors to participate in an agent business call in three different ways:

- Listen—Supervisor can hear what is being said but cannot participate in the conversation.
- Restricted intrusion—Supervisor can hear what is being said and can talk to the agent without the customer being able to hear what is said.
- Full intrusion—Supervisor is actively participating in a conference with the agent and the customer.

Start of procedure

1. Enable the PBX `SingleStepConference` function by programming the ACD Listening key for every supervisor agent. Navigate to `mgr/Users/Go` down `hierarchy/Progr. Keys`.
2. For every supervisor agent, set the value of `function` to `acd listening`.
3. In the `Tserver` section of T-Server, set the value of the [supervisor-step-in](#) configuration option to `true`. With this value, all single-step conference calls are made in Supervisor Step-In mode.

4. Defines the type of supervisor participation by setting the value of configuration option `participation-type` to either `active` or `silent`. This defines how supervisors participate in conference calls created by the `TSingleStepConference` service.

End of procedure

Procedure: Overriding supervisor listening/step-in settings in extensions

Purpose: To override the default settings for supervisor listening/step-in for individual calls by using extensions in `TSingleStepConference`.

Start of procedure

1. To override the default settings defined in option `supervisor-step-in` for an individual call, configure the `Extensions` attribute key `GCTI_SUPERVISOR_STEP_IN` to have either of the following values:
 - 1—step-in mode only
 - 0 (zero)—listening mode only
2. To override the default settings defined in option `participation-type` for an individual call, configure the `Extensions` attribute key `GCTI_PARTICIPATION_TYPE` to have either of the following values:
 - 1—Full intrusion. The supervisor fully participates in the call.
 - 0 (zero)—Restricted intrusion. The supervisor hears everything and can talk to the agent without the customer hearing.

End of procedure

Monitoring Mode

When a supervisor activates the Supervisor Step-In feature, T-Server notifies clients of the Monitoring mode, using the `Extensions` attribute `GCTI_ACTIVE_MONITORING`. See [Table 18](#). This `Extensions` attribute is reported in all events until the supervisor is no longer involved in the call. The supervisor

can change the Monitoring mode manually on the handset. T-Server reports this change with `EventCallInfoChanged`.

Table 18: Monitoring Mode Extension GCTI_ACTIVE_MONITORING

Extension		Used In	Description
Key	Type		
GCTI_ACTIVE_MONITORING	Integer	Call-related events	Defines the Monitoring mode. Valid values: 0—Silent monitoring (listen). 1—Active monitoring (intrude/restricted).

Permanent Listening

The Permanent Listening feature allows a CCD supervisor to show the status of an agent on the display of the phone set (no voice path is connected). Permanent listening is activated by issuing `TRequestPrivateService` with Service Number 15 (see “[Activate Permanent Listening](#)” in Table 24 on [page 227](#)).

Procedure: Configuring unavailable/withdrawal types in processing groups

Purpose: To set up codes for agents to specify reasons for withdrawing from processing groups.

Summary

You can define up to nine withdrawal types for each processing group (PG) on the PBX. If an agent withdraws from the PG, he or she is asked for this withdrawal type. Use the withdrawal types to provide a more accurate reason for withdrawing from the PG.

To use this feature, the requested withdrawal type must be configured in the PG where the requesting agent is logged in.

Start of procedure

1. Navigate to `mgr/Applications/CCD/ProcessingGroup/Consult.modify`.
2. In the `unavailable/withdrawal type` field, specify the number (1–9) of withdrawal types you want to display on the agent phonesets.

3. For each withdrawal type you specify, in the `Display unavailable/withdrawal type x` field, add the text to display on the phoneset when this type is invoked by the agent.

End of procedure

Procedure: Overriding unavailable/withdrawal types using extensions

Start of procedure

You can override the default setting for unavailable/withdrawal types for an individual call by using the `Extensions` attribute `GCTI_NOT_READY_ACTIVATION` in `TAgentSetNotReady`. T-Server reports the withdrawal type using the same `Extensions` attribute in `EventAgentNotReady`.

End of procedure

Advanced Agent Features

Procedure: Configuring smart monitoring

Purpose: To reduce the number of PBX CSTA licenses required by enabling T-Server to stop monitoring on devices that are out of service.

Summary

Normally, T-Server must be registered for both the extension (ACD-authorized device) and the position (Agent Device) to be able to receive events correctly from the PBX. This, however, uses two CSTA licenses in the PBX, even though one of the devices will always be out of service, depending on the login state of the agent.

Smart monitoring enables T-Server to stop monitoring on the device that is out of service. This means that when the agent logs in, T-Server starts monitoring the agent device and stops monitoring the ACD-authorized device, and vice versa. As the out-of-service device used for the agent or agent device is not monitored, only one CSTA license is necessary for every physical phone set.

Start of procedure

1. To reduce the number of CSTA licenses required, in the Tserver section of T-Server, set the value of the `agent-smart-monitor` configuration option to `true`.
2. To monitor only logged-in agents that are configured as a `Position` object in Configuration layer, set the value of `agent-smart-monitor` to `strict`. This only has effect if the value of the `agent-substitute` configuration option is set to `true`.
See “agent-smart-monitor” on [page 373](#).
3. When this feature is enabled, configure the PGs where the agents log in to withdraw the agent at logon. Otherwise, when calls are waiting in the queue, the agent may not get events for the first call because he or she receives it before monitoring can be started.

End of procedure

Restrictions

As there may be a short period of time when the agent logs in where two licenses are in use, you should have at least one extra license available in the PBX (that is, a minimum of number of DNs + 1).

Agent Substitution

Note: In release 7.1 of T-Server, Agent Substitution is the default mode of the switch. For backward compatibility, you can deactivate it by setting the value for the `agent-substitute` configuration option to `false`.

When Agent Substitution is activated (normal operation), T-Server sends all call-related events on the ACD-authorized device when an agent is logged in. This means that T-Server clients only need to register for the ACD-authorized device and not the agent device.

Although T-Server is reporting all events for the agent on the ACD-authorized device extension, calls must still be made to the agent and not to the ACD-authorized device. See “agent-substitute” on [page 374](#).

Note: T-Server rejects all client registration requests for devices configured as ACD Positions in Configuration Layer.

When Agent Substitution is deactivated, an agent is logged in by sending a login request on a ACD-authorized device, and then the agent device is taken into service. No call-related events are received on the ACD-authorized device. This behavior is retained for backward compatibility.

Procedure:

Activating agent substitution for use with Genesys routing for PBX releases higher than R5.0-d2.314.7

Start of procedure

1. Navigate to mgr/Applications/CCD/CCD_RSI system parameters/Review_Modify/ProACD forwarded to agent at logon.
2. Activate the CCD option, ProACD forwarded to agent at logon. When this option is activated, you do not need to use the URS option, `use_agent_id`.

End of procedure

Note: If the Smart OtherDN Handling feature is used (see “convert-otherdn” on [page 377](#)), you do not need to use the URS option `use_agent_id`.

Extensions

When Agent Substitution is activated, T-Server also attempts to perform the necessary attribute substitutions in call-related events.

T-Server uses the following three additional attributes for call-related events to provide the presubstitution value of the corresponding event attribute:

- `GCTI_SUB_THIS_DN`—Corresponds to `AttributeThisDN`.
- `GCTI_SUB_OTHER_DN`—Corresponds to `AttributeOtherDN`.
- `GCTI_SUB_THIRD_DN`—Corresponds to `AttributeThirdPartyDN`.

T-Server provides an `Extensions` attribute if the value has been substituted by T-Server. For example—if `AttributeOtherDN` has been substituted in an event, T-Server provides the `Extensions` attribute `GCTI_SUB_OTHER_DN` in that event.

Configuring CCD Objects

The Alcatel ACD is made up of a three-tier system, allowing a flexible call-distribution system to be created. The call flow is as follows:

1. Calls enter the ACD on the pilot device.
2. The pilot can be configured to distribute calls to any combination of queues.
3. ACD Queues, in turn, distribute to any combination of processing groups where agents are logged in.

At least one agent must be logged in to an associated processing group; otherwise, all calls to the pilot are rejected. Alternatively, you can configure a dissuasion group to be always open for calls.

Note: To get consistent reporting, you only have to configure the pilot in Configuration Layer. T-Server can provide information about queues and processing groups, if required. [Configuring CCD Pilots in Configuration Layer, page 161](#).

Procedure:

Configuring CCD Pilots in Configuration Layer

Purpose: To configure the CCD Pilot to provide information about how the call was distributed.

Start of procedure

In the Configuration Layer, complete the following fields for the CCD pilot device:

- Type—ACD Queue
- Number—<pilot number>. The number of the pilot as configured on the PBX
- Register—true. T-Server should register for this device on the PBX.

End of procedure

Procedure: Configuring CCD Pilots in the PBX

Start of procedure

1. Navigate to mgr/Applications/CCD/Pilot.
2. Set Directory Number to <pilot number>.
3. Configure Pilot Supervised Transfer according to whether consult calls will (true) or will not (false) be distributed before transfer is completed.
4. Configure ABC Local Call Allowed according to whether direct calls through private ABC networks from other Alcatel A4400s are allowed or not allowed to this pilot.
5. Configure AutoWrapUp Timer. The PBX can apply this after-call work period after a business call. The unit is 100 msec
6. Configure Time Between Two Calls. The PBX can apply a mandatory break after a business call. If AutoWrapUp Timer has been configured, it will be applied before this pause. The unit is 100 msec.
7. Set Pilot Routing Flag to false. Value true will not work with T-Server.

End of procedure

Procedure: Configuring CCD Queues in Configuration Layer

Summary

To get information about how the call was distributed, you can configure the presentation guide, queues, blocking guide, and all the waiting guides in Configuration Layer. If any of these is configured, T-Server will send Queued/Diverted events if a call is queued on this device.

Start of procedure

In the Configuration Layer, configure the following fields for the CCD queue object.

- Type—ACD Queue
- Number—Queue/Guide number. The following are valid numbers:
 - <Any queue>
 - ØPRG = presentation guide
 - ØCLO = blocking guide
 - ØWG1-6 = waiting guides
 - ØIVR1-6 = IVR guides for IVR-in-Queue

- 0IAA1-6 = interactive/automatic attendant
- Register—No (not checked)

End of procedure**Configuring CCD Queues in the PBX**

No special considerations need to be taken into account when configuring queues on the Alcatel A4400.

Procedure:**Configuring CCD/RSI processing groups in Configuration Layer**

Purpose: The CCD processing group (PG) is the device in the PBX where CCD agents are logged in. By configuring the PG in Configuration Layer, you can query the number of agents logged in.

You do not have to configure PGs in Configuration Layer for call reporting to be consistent.

Start of procedure

In the Configuration Layer, configure the following fields for the CCD/RSI Processing Groups object.

- Type—ACD Queue
- Number—<PG number>. The number of the PG as configured on the PBX.
- Register—false. T-Server should not register for this device on the PBX.

End of procedure

Procedure:**Configuring CCD processing groups in the PBX****Start of procedure**

1. Navigate to mgr/Applications/CCD/Pilot.
2. Configure Directory Number with the <PG number>.
3. Configure Withdrawal After Logon to either true or false. This determines the state of the agent after login.

4. Configure `Pilot Direct Call` to have the `<Pilot Directory Number>`. This determines if an incoming external call that arrives directly at an agent who is logged in to this PG is considered a business call. The pause and wrap-up timers defined in the configuration for this pilot are applied to the agent after such a call is released.
5. Configure `Outgoing ACD Calls`. This defines if a direct outgoing external call made by an agent who is logged in to this PG is considered a business call (`true`). This means that the pause and wrap-up timers configured in the pilot specified in the `Pilot Direct Call` option are applied when the call is released.
6. Configure the `Withdrawal Type` with a value from 0 to 9. If 1-9, the agent is asked for a withdrawal type when withdrawing from the PG. The withdrawal type can be selected by using `TAgentNotReady` with `Extensions` attribute `GCTI_NOT_READY_ACTIVATION`. This option defines how many withdrawal types are presented to the agent.
7. Configure `Display Withdrawal Type <n>` with the text that will be shown in the agent phoneset display if the agent withdraws manually (not by CTI).

End of procedure

Network Call ID Matching

In order to fully support the Call Overflow feature, T-Server supports the Network Call ID Matching feature. To activate this feature, the value of the `default-network-call-id-matching` option must be set to `ts-gcid` as indicated on [page 356](#).

Routing Using Emulated Routing Points

T-Server can use Hunt Group (HG) devices on the PBX to provide routing functionality. The Hunt Groups have virtual device (virtual Z device) members. The number of virtual devices assigned to such a Hunt Group must be greater than the maximum number of calls expected to be queuing on the Routing Point at the same time.

Note: For PBX version 4.2 or later, T-Server supports routing using RSI Routing Points. See [page 267](#).

Configuring Hunt Groups/Virtual Devices (PBX)

The following procedures describe the configurations required in the PBX to create a Hunt Group with virtual Z device members that can be used for Genesys routing.

Procedure: Configuring a Hunt Group in Alcatel A4400

Start of procedure

1. Navigate to mgr/Groups/Hunting Group/Create.
2. Configure the Directory Number to have the Hunt Group number, in the format <HG number>.
3. Set the value of the configuration option, Authorized Camp-On Calls %..
A value of 0 (zero) stops calls from queueing on the Hunt Group device if no member devices are available.
If you set a value greater than 0 (zero), the PBX allows calls to be queued if all Hunt Group members are busy. However, the PBX does not notify T-Server about such calls until they are finally delivered to a Hunt Group member. This means that there can be calls in the system that T-Server does not know about.

End of procedure

Procedure: Configuring Hunt Group virtual Z members in Alcatel A4400

Start of procedure

1. Navigate to mgr/Users/Create.
2. Set the following values:
 - Directory Number—<VZ Number>
 - Shelf/Board/Equipment Address—255/255/255
A virtual device is not given a physical address in the PBX.
 - Set Type—ANALOG
Specifies virtual devices of type analog.
 - Hunting Group Dir no—<HG Number>
Defines the Hunt Group with which this virtual device is to be associated.
 - Ghost Z—True Specifies that this is a virtual device.

End of procedure

Configuring Hunt Groups/Virtual Devices in the Configuration Layer

Procedure:

Configure the Hunt Group and virtual Z devices in the Configuration Layer

Start of procedure

1. Configure the PBX Hunt Group device as a Genesys Routing Point with the following parameters:
 - Number—<HG number>
 - Association—Not applicable
 - Switch-specific type—1 (default value)
2. Configure the PBX virtual Z device as a Genesys Extension with the following parameters:
 - Number—<VZ number>
 - Association—<HG number>
 - Switch-specific type—2
 - Register—checked

End of procedure

Configuring External Routing Point Virtual Z Members in Alcatel A4400

Procedure:

Configuring External Routing Point virtual Z members in Alcatel A4400

Start of procedure

1. Navigate to mgr/Users/Create.
2. Set the following values:
 - Directory Number—<VZ Number>
 - Shelf/Board/Equipment Address—255/255/255
A virtual device is not given a physical address in the PBX.
 - Set Type—ANALOG
Specifies virtual devices of type analog.

- Ghost Z—True Specifies that this is a virtual device.

End of procedure

Configuring the External Routing Point and Virtual Z Devices in the Configuration Layer

Procedure: Configuring the External Routing Point and virtual Z devices in the Configuration Layer

Start of procedure

Configure the PBX virtual Z device as a Genesys External Routing Point with the following parameters:

- Number—<VZ number>
- Association—Not applicable
- Switch-specific type—1 (default value)
- Register—checked

End of procedure

Routing Failure Scenarios

In some cases the routing of a call may fail. To enable you to handle such situations, T-Server provides the `route-request-attempts` configuration option.

Routing to External Destinations

A customer could be routed to an external or unmonitored destination where the call is not answered, forcing the customer to hang up and call again.

As T-Server emulates routing to external or unmonitored destinations by using the Transfer services on the PBX, T-Server can also monitor whether the call is answered, before completing the routing. This ensures that a customer is not routed to a destination where there will be no answer. [Figure 13](#) illustrates this situation.

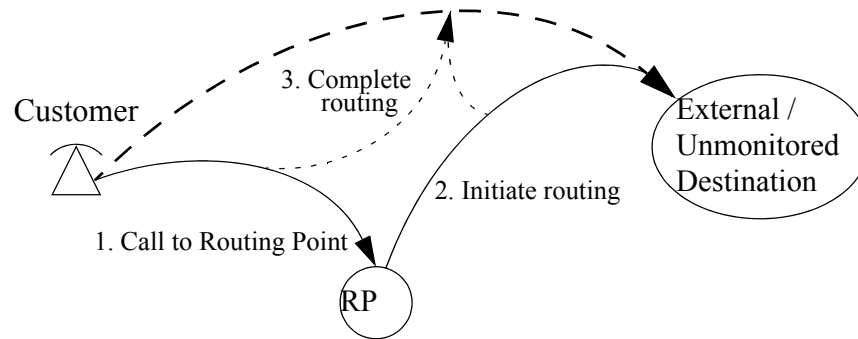


Figure 13: Supervised Routing to External Destinations

Use the following two configuration options to define how T-Server should behave in this situation:

- `imm-trf-route-external`
- `supervised-route-timeout`

Routing Consultation Calls

T-Server emulates routing to external or unmonitored destinations by using the Transfer services on the PBX. This means that you cannot route a consultation call in this case. To do so, T-Server would have to put the initial consultation call on hold and initiate a second chained consultation call. The PBX does not support this. Therefore, T-Server cannot route such a call before transfer is completed to the Routing Point. Figure 14 illustrates this situation.

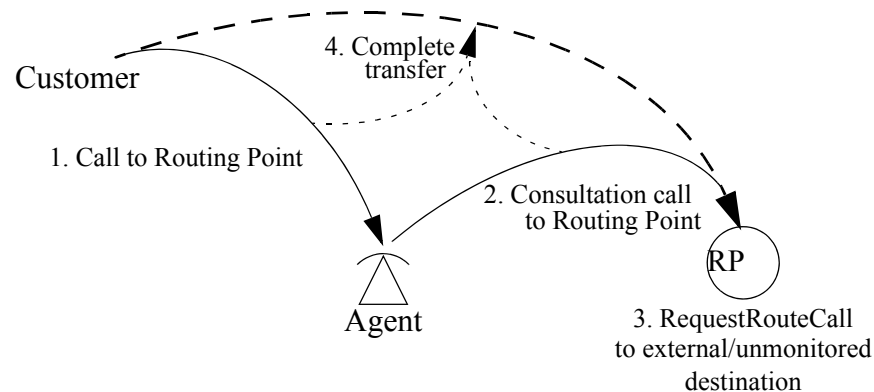


Figure 14: Routing Consultation Calls

T-Server provides two configuration options to handle what should happen after Step 3 in Figure 14.

- `auto-transfer-to-route`
- `route-handover-timeout`

Supervised Routing to CCD Pilots

You can define the way calls are routed from a Routing Point to a pilot by using the `Extensions` attribute `GCTI_SUPERVISED_ROUTE` in a `TRouteCall` request. When T-Server uses the supervised route method, T-Server uses the Transfer service to route the call to Pilots. In this case, T-Server checks whether the destination Pilot is available. If not, T-Server reissues a `EventRouteRequest` event.

By setting the `Extensions` attribute `GCTI_SUPERVISED_ROUTE` to `0` in `TRouteCall`, T-Server is forced to use the Divert service instead of the Transfer service to route the call.

Note: Use the `Extensions` attribute key `GCTI_SUPERVISED_ROUTE` when routing to Pilot, unknown, or unmonitored destinations.

Integrating Routing Points in the CCD

As described in the previous section, T-Server uses Hunt Groups with virtual member devices on the PBX to provide routing functionality. This section describes how you can make a Routing Point part of the Alcatel A4400 CCD, which offers some advantages:

- A presentation guide can be played before calls enter routing.
- If a call is not routed within 30 seconds, it is pulled back to the CCD and redistributed to another CCD queue.

Figure 15 shows how such a CCD configuration looks.

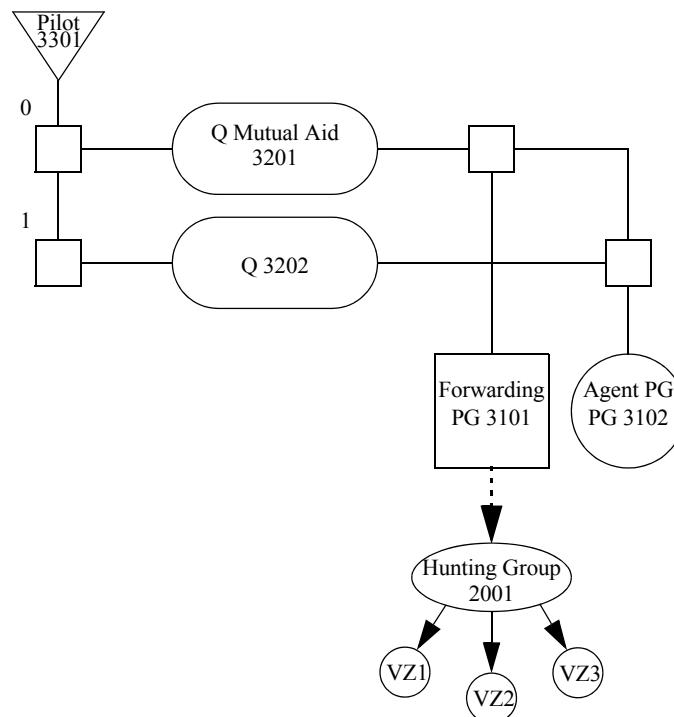


Figure 15: CCD Configuration

Call Process

If a call enters the CCD on Pilot 3301, it first goes to the presentation guide, and then is distributed to Hunting Group 2001 through queue 3201. If the call is not routed within 30 seconds, the call is pulled back to the CCD, and can then be redistributed through ACD Queue 3202 to an agent in processing group 3102.

Procedure: Configuring a Routing Point as part of PBX CCD

Start of procedure

1. Configure the Hunt Group in the Alcatel A4400 as described in “Configuring a Hunt Group in Alcatel A4400” on [page 165](#).
2. Configure a forwarding processing group. Set the following values:
 - Directory Number—<PG number>
(3101)
 - Type—Forwarding
This is a forwarding processing group.
 - Forwarding Directory Number—<HG number>
(2001)
3. Configure the Mutual Aid queue. Set the following values:
 - Directory Number—<queue number>
(3201)
 - Type—Mutual Aid
Specifies a mutual aid queue.
 - Distribution direction 0—<PG number>
(3101)
4. Configure the pilot to distribute calls, giving higher priority to the Mutual Aid Queue than to the normal queue.
5. Configure all devices as you normally would in the Configuration Layer. The minimum configuration consists of the pilot, the Routing Point, and the agents.

Note: If you omit the pilot from the Configuration Layer, T-Server cannot maintain the connection ID when the call is redistributed.

End of procedure

7

Supported T-Server Features

This chapter describes the telephony functionality supported by the T-Server for Alcatel A4400/OXE. It includes the following sections:

- [Support for Business-Call Handling, page 172](#)
- [Support for Emulated Agents, page 175](#)
- [Support for Predictive Dialing, page 182](#)
- [Support for Emulated Predictive Dialing, page 189](#)
- [Support for No-Answer Supervision, page 191](#)
- [Support for Smart OtherDN Handling, page 194](#)
- [Support for the Keep-Alive Feature, page 196](#)
- [Support for Call Type Prediction, page 196](#)
- [Support for Call Release Tracking, page 197](#)
- [Support for Failed Route Notification, page 198](#)
- [Support for Link Bandwidth Monitoring, page 199](#)
- [Request Handling Enhancements, page 200](#)
- [Hot-Standby HA Synchronization, page 201](#)
- [Support for Boss/Secretary Functionality, page 202](#)
- [Support for A4400/OXE Spatial Redundancy, page 202](#)
- [Support for Outbound Caller ID, page 203](#)
- [Support for Inter Server Call Control, page 204](#)
- [Support for Account Codes, page 209](#)
- [T-Library Functionality, page 212](#)
- [Support for Agent Work Modes, page 226](#)
- [Private Services and Events, page 227](#)
- [Use of the Extensions Attribute, page 231](#)
- [Extension Filtering, page 251](#)
- [User Data Keys, page 251](#)

- [Reasons Keys, page 253](#)
- [T-Server Error Messages, page 253](#)

Support for Business-Call Handling

This section describes how T-Server handles different types of calls:

T-Server Call Classification

T-Server automatically assigns every call to one of three categories:

- business
- work-related
- private

Based on this assignment, T-Server applies the appropriate business-call handling after the call is released.

Business Call Type Configuration

T-Server uses the following criteria to determine the business call type of a call. The criteria are given in order of precedence, highest to lowest:

- T-Server supports a request `Extensions` attribute to define the business call type upon call initiation or answer.
- T-Server uses the originator agent state to determine if the call is work related.
- The DN-level configuration option, `bsns-call-type`, specifies the business call type for calls that pass through or arrive at a distribution device. If the call passes through a distribution device and there is no DN-level option present, the call will be classified as a business call as long as this is enabled by the option `bsns-call-dev-types`. Automatic classification of calls as business calls on distribution DNs can be disabled by the application option `bsns-call-dev-types`. If the corresponding flag of this option is disabled, calls passing distribution DNs of that type will not change their respective business classification. The request extension and DN-level option will still be effected. Distribution devices include the following device types:
 - Routing Point
 - ACD Queue
 - Routing Queue
 - External Routing Point
- T-Server supports options that are configured on the `Application` level to define whether specific call types (inbound, outbound, internal, or unknown) are to be classified as business calls.

To specify the business call type for the new call, the `Extensions` attribute `BusinessCallType` can be attached to the following requests:

- `TMakeCall`
- `TInitiateTransfer`
- `TMuteTransfer`
- `TInitiateConference`
- `TMakePredictiveCall`
- `TAnswerCall`
- `TRouteCall`

This `Extensions` attribute takes precedence over all other business call type configuration options. The `Extensions` attribute can also be attached to `TAnswerCall` to specify the business call type for the answering party and call.

If the call passes through more than one distribution device, then the usual rules for assigning a business call type are followed. Once set, the business call type cannot be overridden unless it is changed to be a business call.

When the configuration options `inbound-bsns-calls`, `internal-bsns-calls`, and `outbound-bsns-calls` are set at the `Application` level, they control whether the call type of the associated calls are to be classified as business calls.

T-Server will not classify the business type of the call using these options until the destination is known. Also these options are not be used to set the originating party's business type as business until after the `Dialing` event has been reported. (This is to ensure that Genesys reporting is consistent regardless of the switch reported order of events).

The private request, `TSetBusinessCall` allows T-Server clients to set the business call type of an existing call to be business. T-Server will respond to a successful request by distributing the private event `EventBusinessCallSet`.

Call Transformation

Business type assignment at any stage does not affect parties already present in the call, except for outbound calls that are configured to be treated as business calls. In this case, the dialing party will become a business party as soon as T-Server determines that the call is an outbound call. An outbound work-related calls remains work-related regardless of the option setting.

Calls that are already classified as private or work-related can only be promoted to the business type. Changing between private and work-related is not possible, regardless of device settings and request extensions.

Call transfers or conference calls result in the business classification propagation from the transferring or conferencing main party to all be participants of the secondary call.

Business Calls

T-Server automatically categorizes as a *business call* any call distributed to an agent either from a ACD Queue or from a Routing Point.

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.

Feature Configuration

The following configuration options define what additional calls to or from an agent are classified as business calls:

- `agent-only-private-calls`
- `bsns-call-dev-types`
- `bsns-call-type`
- `inbound-bsns-calls`
- `inherit-bsns-type`
- `internal-bsns-calls`
- `outbound-bsns-calls`
- `unknown-bsns-calls`

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.

Agent Logout on Client Unregistering from DN

In some scenarios (such as a power failure/disconnection or when a desktop stops responding), agents may still receive calls but be unable to handle them. To prevent this problem, T-Server can be configured to automatically logout the agent in such circumstances.

When a client desktop or application disconnects from the T-Server while an agent is still logged in, the T-Server receives a notification that the application is unregistering from the agent's DN. Also, the T-Server is able to uniquely identify the client application which sends a T-Library request, including TAgentLogin and TRegisterAddress.

The T-Server can associate the client application (the one that sends the initial TAgentLogin request) with the agent and automatically log that agent out when the client application unregisters the agent DN while the agent is still logged in. The initial TAgentLogin request is the one which first logs the agent in.

HA Considerations

If the T-Server is running in HA mode, a client connecting to one T-Server will be connected to both with the same session ID. Therefore the client's session ID must be used as part of the association data to ensure consistency across the primary and backup T-Servers. The primary T-Server will send a HA synchronization message to the backup when there is a change in client associations.

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: The *Genesys Events and Models Reference Manual* and the *Voice Platform SDK 8.x .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 174](#)).

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 `Extensions` attribute `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 `Extensions` attribute `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 the [wrap-up-time](#) configuration option.

The [untimed-wrap-up-value](#) configuration option determines which specific integer value of wrap-up-time indicates *untimed* ACW. To specify untimed ACW in request `Extensions` or `UserData`, 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 can affect the interpretation of all integer values of the `wrap-up-time` configuration option. If the value is lowered, it may change timed ACW to untimed, or disable ACW altogether. If the value is raised, it may change untimed, or disabled ACW to timed ACW. The use of the option (string) value `untimed` is encouraged where possible to minimize the impact of any future changes to the value of the configuration option, `untimed-wrap-up-value`.

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 applies the emulated ACW when the call is released. The agent sends `TAgentReady` with `workmode = 3` to request pending ACW while on an established call. The `Extensions` attribute, `WrapUpTime`, indicates the amount of ACW that T-Server applies, 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 also indicates that the agent is in a pending ACW state by adding the `Extensions` attribute, `ReasonCode`, with the new value `PendingACW`. It also indicates the period of ACW to be applied using the `Extensions` attribute, `WrapUpTime`.

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

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

ACW In Idle

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

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 `TAgentNotReady` with `workmode = 3` (`AgentAfterCallWork`). T-Server determines the period of the extended ACW from the `Extensions` attribute, `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 `Extensions` attribute, `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 `backwds-compat-acw-behavior` configuration option.

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.

Feature Configuration

Enabling or Disabling the Emulated Agent Login

The following configuration options enable or disable the emulated agent login/logout functionality:

- `agent-emu-login-on-call`
- `agent-strict-id`
- `emulate-login`
- `emulated-login-state`

Enabling or Disabling the Automatic Agent Logout

The following configuration options enable or disable the automatic logout of the agent:

- `agent-logout-on-unreg`
- `agent-logout-reassoc`

Enabling or Disabling the Emulated ACW Period

The following configuration options enable or disable the emulated ACW period functionality:

- `untimed-wrap-up-value`
- `wrap-up-threshold`
- `wrap-up-time`

Enabling or Disabling Pending ACW Functionality

The following configuration options enable or disable the pending ACW functionality:

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

Enabling or Disabling the Emulated ACW Functionality

The `backwds-compat-acw-behavior` configuration option enables or disables the; emulated ACW functionality.

Determining the Period of Legal Guard Time

The following configuration options determine the period of legal guard time:

- `legal-guard-time`
- `real-agent-pause-time`

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.

Emulated Wrap-Up Time for CCD Agents

In many cases, the CCD does not apply business-call rules to a call that was routed directly to an agent because it does not recognize the call as a business call. In this case the CCD applies no wrap-up or legal guard time after such a call is released.

This feature allows T-Server to apply the emulated wrap-up and legal guard time for agents after such a call is released.

After a T-Server business call (not a CCD business call) is released on a CCD agent, T-Server immediately applies CCD wrap-up on this agent to stop new calls arriving to this agent.

Note: If a CCD agent releases a T-Server business call while the CCD is waiting to distribute a call to this agent, T-Server cannot apply this feature.

If an agent requests withdrawal manually, there is a two- to three-second delay before the PBX sends the CSTA event. If pause/guard time expires during this delay, T-Server sets the agent back to the Ready state.

Procedure: Configuring manual wrap-up time for CCD agents in the PBX

Summary

For this feature to work, T-Server must apply a wrap-up timer on the PBX when a T-Server business call is released.

Start of procedure

1. Navigate to Mgr/Applications/CCD/Processing Group.
2. Configure Wrap-Up Idle Timer to equal the combined duration (in seconds) of T-Server configuration options, `real-agent-pause-time` and `legal-guard-time`, plus a small amount extra. For example—if `real-agent-pause-time` = 30 seconds and `legal-guard-time` is set to 5 seconds, configure this option to 38 seconds.

End of procedure

Next Steps

Procedure: Configuring wrap-up time for CCD agents in the Configuration Layer

Start of procedure

Set the value of the `real-agent-pause-time` option.

If the values set range from 0-999, T-Server sends `EventAgentNotReady` with `workmode` = 3 (`AgentAfterCallWork`) when a routed call is released. At the end of the wrap-up time, T-Server applies legal-guard time if it is configured, otherwise T-Server sends `EventAgentReady`.

If the value is 1000, an “eternal” wrap-up is defined. T-Server sends `EventAgentNotReady` with `workmode` = 3 (`AgentAfterCallWork`) when a routed call is released. T-Server does not automatically return the agent to the Ready state.

This timer is only applied if a routed call was established on the agent

End of procedure

Interrupting CCD Agent Wrap-Up Time

Emulated wrap-up time on CCD agents can be interrupted in several ways. The effect of such interruptions is described below.

Direct Call from the Agent

When an agent makes a direct calls during the emulated wrap-up time, he or she is set directly to the Ready state before the call is initiated.

Agent Logs Out

T-Server immediately logs out the agent.

Agent Requests Ready

T-Server immediately interrupts the emulated wrap-up time and sends `EventAgentReady`. If legal-guard time is set, T-Server delays `EventAgentReady` for that duration.

Agent Requests Not Ready (Withdraw)

The agent is withdrawn.

Support for Predictive Dialing

The Alcatel A4400 T-Server Predictive Dialing feature allows the Outbound Contact Server (OCS) to initiate calls without the use of Call Progress Detection (CPD) Server and Dialogic cards.

You can use the PBX feature Voice Activity Detection (VAD), *instead of Dialogic cards*, together with T-Server Predictive Dialing to provide full CPD capabilities. See “Voice Activity Detection” on [page 183](#).

This chapter describes how to enable T-Server Predictive Dialing and how to configure OCS to use this feature to make outbound calls.

Note: This feature is *not* related to the predictive dialing algorithm that OCS uses to determine when to make the next call. The T-Server Predictive Dialing feature only involves the way in which the outbound calls are initiated.

To enable T-Server Predictive Dialing, you must configure a number of virtual devices (virtual Zs) in the PBX and in Configuration Layer. The number of virtual Zs corresponds to the number of calls that can be initiated simultaneously. These virtual Zs are available as a pool for T-Server to use for predictive dialing. They are not associated with any specific origination DN (Pilot or Routing Point) or outbound campaign.

Voice Activity Detection

Alcatel A4400 R4.1 introduced a PBX feature called Voice Activity Detection (VAD) for use with the T-Server Predictive Dialing feature. This provides native on-board CPD, which means that you can use the Genesys OCS without Dialogic hardware.

Note: VAD requires licenses and hardware for the PBX. Please consult your Alcatel representative for details.

Although you can use the T-Server Predictive Dialing function with earlier versions of the PBX, CPD is not available and only the following dialing results are reported:

- Answer
- No Answer
- Busy
- Dropped
- Wrong number (dialing result reported as Sit Tone)
- Abandoned

With VAD, the following dialing results are also available:

- Fax detected
- Answering machine detected

Procedure: Activating VAD in the Alcatel A4400

To activate VAD on the PBX, you must install a GPA board. This provides the DSP functionality for performing the call-result analysis. The following steps describe how to install and configure the GPA board for use with VAD, observing the safety guidelines described in your PBX documentation.

Note: Only a qualified Alcatel engineer should make changes to PBX configuration.

Start of procedure

1. Install the RAM card in the GPA board.
2. Insert the GPA board in your PBX, observing the safety guidelines described in your PBX documentation.
3. Configure the GPA board in the system:
 - a. Locate mgr/Shelf/Go Down Hierarchy/Create.

- b. Select board type GPA.
 4. Configure the GPA board DSP function to be used with VGs:
 - a. Locate mgr/Shelf/Go Down Hierarchy/Go Down Hierarchy/Gpa Dsp program/Create.
 - b. Select a DSP function that includes the required number of VAD functions.

End of procedure

A GPA board can handle a maximum of 5 DSP functions with 9 VAD devices per DSP. Therefore, there is a maximum of 45 VAD devices available on a GPA board, so up to 45 simultaneous outbound calls can be made.

A GPA2 board has 2 DSP functions. 1 DSP function is dedicated for use by Voice Guides (VG). The remaining DPS consists of 4 virtual DSPs, with 15 VAD devices per virtual DSP. Therefore, there is a maximum of 60 VAD devices available on a GPA2 board, so up to 60 simultaneous outbound calls can be made.

Note: VAD is not supported over ABC-F trunks. As a result, outbound calls cannot be made with an ARS prefix.

Procedure: Checking GPA DSP status

Purpose: To check the voice guide status on each GPA board in the system.

Start of procedure

At the command prompt, enter:

```
dspstat <crystal (0-19)> <cpl (0-28)>
```

where crystal and cpl are the name and the location, respectively, of the GPA board in question.

End of procedure

The output from this command provides detailed information about the DPSs configured on the specified GPA board.

Note: Because of a minor difference in the way the availability of dialing devices is calculated in T-Server and in OCS, you should configure extra dialing devices. For example, if five dialing devices are used in a campaign, then configure six dialing devices in T-Server/PBX.

Procedure: Configuring virtual Z dialing devices in the PBX

Purpose: To create dialing devices in the PBX for use with the Predictive Dialing feature. You can create six such dialing devices for every VAD function that you created in the GPA DSP Program.

Start of procedure

1. Navigate to mgr/Users/Create.
2. Set the following values for the device:
 - Directory Number—<VZ Number>
 - Shelf/Board/Equipment Address—255/255/255
A virtual device is not given a physical address in the PBX
 - Set Type—ANALOG
Specifies virtual devices of type analog.
 - Ghost Z—True
Specifies that this is a virtual device.
 - Ghost Z Feature—CC0
Specifies this device as a CCO device.
 - VAD use for Ghost Z—True
Activates the VAD function for this device.

End of procedure

Procedure: Configuring virtual Z dialing devices in the PBX without VAD

Purpose: To create dialing devices in the PBX for use with the Predictive Dialing feature where there is no VAD on the PBX.

Start of procedure

1. Navigate to mgr/Users/Create.
2. Set the following values for the device:
 - Directory Number—<VZ Number>
 - Shelf/Board/Equipment Address—255/255/255
A virtual device is not given a physical address in the PBX
 - Set Type—ANALOG
Specifies virtual devices of type analog.

- Ghost Z—True
Specifies that this is a virtual device.
- Ghost Z Feature—None
- VAD use for Ghost Z—False

End of procedure

Procedure: Configuring VAD-Related Timers in the Alcatel A4400

Start of procedure

1. Navigate to mgr/System/Go Down Hierarchy/Timers.
2. Ensure that the values of timers 267 and 268 are set to the PBX default values.
Timer 267 determines the duration that VAD should wait for a tone after a call has been answered.
Timer 268 determines how long an outbound call from a VAD dialing device rings before VAD classifies the call as not answered.

End of procedure

Configuring Predictive Dialing in the Configuration Layer

Procedure: Configuring devices for predictive dialing in the Configuration Layer

Start of procedure

Set the following values for the device:

- Type—Extension
- Number—<Virtual Device Number>
- Switch-specific type—4 (on Advanced tab of DN Properties window)

End of procedure

Feature Configuration

You can also use the following options to adjust the behavior of the Predictive Dialing feature in T-Server:

- `predictive-delay-time`

- `prd-dist-call-ans-time`
- `max-pred-req-delay`

Limiting Call-Distribution Time

If Outbound Contact is making more calls than there are available agents, many countries legally forbid the queuing of these calls. The law in these countries requires that such calls be dropped immediately.

T-Server does not handle this requirement; the distribution mechanism must be configured to handle it.

Procedure: Configuring the CCD to limit call distribution time

Start of procedure

1. Navigate to `mgr/Application/CCD/Queue`.
2. Set the option `Max Waiting Time` to 0 (zero) in the Queue configuration. The CCD can overflow the call to a dissuasion group if no agent is available.

End of procedure

Procedure: Configuring URS to limit call distribution time

Start of procedure

In the routing strategy, set the option `Timeout` in the Target Selection object to an appropriate value, for example, 1 or 2 seconds.

End of procedure

Note: The routing strategy is likely to fail if the timeout is set to 0 (zero)

Configuring OCS to Use Predictive Dialing

Procedure:

Configuring OCS to use predictive dialing

Purpose: To configure OCS to use T-Server predictive dialing instead of CPD Server to originate outbound calls. Normally, an outbound campaign is associated with a specific CPD Server in Configuration Layer.

Start of procedure:

1. Configure OCS to have no CPD server by ensuring that the Campaign Group in Configuration Manager does not have any associated CPD Server in the **Connections** tab of the **Properties** dialog.
2. Ensure that the value of the **Number of CPD Ports** parameter on the **Advanced** tab of the Campaign Group corresponds with the number of dialing devices dedicated to this Campaign Group.

In addition, the OCS configuration `channel_num` option can be configured to limit the number of dialing devices used for the given OCS or switch.

End of procedure

CCO Agent Reservation Feature

The Alcatel A4400 provides functionality that allows agents to be reserved for CCO calls. This functionality is used with the CCO feature, which can be provided for the Alcatel A4400. Agent Reservation uses the Genesys Outbound Contact with predictive dialing and, like VAD, is available from Alcatel A4400 R4.1.

To reserve an agent for a CCO call on the Alcatel A4400, you must set the outgoing `ACD call` option to `true` in the processing group where the agent is logged in.

Set Reservation

To reserve an agent for a CCO call by CTI, send this request:

`TAgentNotReady (AfterCallWork)`

- `Extensions` attribute key `GCTI_SET_RESERVATION` = 1
- `Extensions` attribute key `GCTI_CC_TREATMENT_TYPE` = 1

Reset Reservation

To cancel agent reservation by CTI, send this request:

TAgentNotReady (AfterCallWork)

- Extensions attribute key GCTI_RESET_RESERVATION = 1
- Extensions attribute key GCTI_CC_TREATMENT_TYPE = 1

CCO Call Tag

To identify a call as a CCO call, set the following extensions:

- Extensions attribute key GCTI_ACR_STATUS_1 = 1
- Extensions attribute key GCTI_EXPERT_EVALUATION_LEVEL_1 = 1 (for example)

To identify a call as a CCO call, set the following UserData:

- Extensions attribute key GCTI_SKILL_NUMBER_1 = 80

Make CCO Call

A CCO call can be made in several ways:

- A reserved agent can make a direct outbound call with the CCO call extensions.
- Make an outbound call from a virtual Z that has Ghost Z Feature = CCO, and transfer the call to a pilot configured with the Advanced Call Routing (ACR) feature in the PBX. For this call to be distributed, the CCD agents must also have the ACR skill Audio Outbound (80).
- Make an outbound call from any device such as a Dialogic port, and transfer the call directly to a Statistic Pilot, which has the CCO call profile and forwards the call to an ACR pilot.

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 (ACD 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 the 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 the configuration option, `max-pred-req-delay`. 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.

Feature Configuration

Configuring the Distribution Time

The following configuration options control the distribution time:

- `max-pred-req-delay`
- `prd-dist-call-ans-time`

Controlling Unsolicited Call on Predictive Dialing Devices

The `dn-for-undesired-calls` configuration option controls the unsolicited calls on predictive dialing devices functionality.

Support for No-Answer Supervision

This section describes the 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 send calls back automatically to the last distribution device.
- If an agent fails to answer a call that is ringing within a specified timeout, you can configure T-Server to either log out the agent or set the agent `NotReady`, to prevent further calls from arriving.

Extension No-Answer Supervision

The No-Answer Supervision feature includes devices of type `Extension` and provides the following functionality:

- 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 send calls back automatically to the last distribution device.

Position No-Answer Supervision

The No-Answer Supervision feature includes devices of type `ACD Position` and provides the following functionality:

- If a call is not answered on a `ACD Position` within a specified timeout, T-Server can divert the call to a sequence of overflow destinations. Alternatively, you can configure T-Server to send calls back automatically to the last distribution device.

Feature Configuration

Enabling or Disabling the Agent No-Answer Supervision Feature

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

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

Enabling or Disabling the Agent No-Answer Supervision Feature for Devices of type Extension

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

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

Enabling or Disabling the Agent No-Answer Supervision Feature for Devices of type ACD Positions

T-Server provides the following two configuration options for defining the behavior of No-Answer Supervision with devices of type ACD Position:

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

Enabling or Disabling the Agent No-Answer Supervision Feature for Device-Specific Overrides

T-Server provides the following 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 Configuration Layer. These are the options:

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

Extension Attributes for Overrides for Individual Calls

For all of the No-Answer Supervision options, you can specify the corresponding Extensions 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. The following are the three Extensions attributes:

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

Note: It is not necessary to specify all Extensions attributes. T-Server uses the configured values for any Extensions attribute not specified.

Enabling or Disabling the Agent No-Answer Supervision Feature for Private Calls

The configuration `nas-private` option applies No-Answer Supervision to private calls.

Enabling or Disabling the Agent No-Answer Supervision Feature for Reporting

The `nas-indication` configuration option enables or disables No-Answer Supervision for reporting.

Support for 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.

Feature Configuration

The following configuration option and `Extensions` attribute key support the Smart OtherDN Handling feature:

- `convert-otherdn` (configuration option)
- `ConvertOtherDN` (`Extensions` attribute key)

Supported Requests

Table 19 shows the requests that assume the use of the `otherDN` value as a switch directory number known to T-Server, and that can therefore support Smart OtherDN Handling.

Table 19: 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

Table 19: Requests That Support Smart OtherDN Handling (Continued)

TRequest	Meaning of OtherDN Attribute	AgentID-to-DN Conversion	Reserved DN Conversion
TSingleStepTransfer	New destination for call	Yes	Yes
TInitiateConference	Call destination	Yes	Yes
TSingleStepConference	New destination for call	Yes	Yes
TDeleteFromConference	Conference member to be deleted	Yes	Yes
TListenDisconnect	Request target	No	No
TListenReconnect	Request target	No	No
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 • RouteTypeUnknown • RouteTypeDefault • RouteTypeOverwriteDNIS • RouteTypeAgentID	New destination for call		
		Yes	Yes
		Yes	Yes
		Yes	Yes
		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.

Support for the 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 a counter, which is decreased on positive response and increased on a negative or missing response. If the counter reaches the maximum configured limit, T-Server attempts to reconnect to the link.

Support for Call Type Prediction

T-Servers use CTI-provided information to assign a call type to a call. On occasions when the CTI information is either insufficient or arrives too late for T-Server to assign a definite call type, T-Server uses a call type prediction procedure to assign a call type on a "best possible guess" basis.

[Table 20](#) shows how T-Server assigns call types in different scenarios.

Table 20: Call Type Prediction

Call Direction/ OtherDN	External	Internal	Unknown
Incoming	CallTypeInbound	CallTypeInternal	CallTypeUnknown
Outgoing	CallTypeOutbound	CallTypeInternal	CallTypeUnknown

Feature Configuration

The `call-type-by-dn` configuration option enables or disables the Call Type Prediction feature.

Support for Call Release Tracking

T-Server provides information about which party initiated the release of a call. This can be valuable for different applications to provide historical and real-time call reporting.

The following T-Library SDK call models can be reported in this way:

- Normal call release
- Abnormal call release
- Call release from a conference
- Rejection of an alerting call
- Release for a failed or blocked call to a busy destination

DN-Based Reporting

In DN-based reporting, information about the call release initiator will be reported in the `AttributeExtension` using extension key `ReleasingParty` in `EventReleased` and `EventAbandoned` events, when those events are distributed.

One of the following values will be reported in the `ReleasingParty` key:

- 1 Local—The call is released because the `ThisDN` value in the `EventReleased` was requesting the release.
- 2 Remote—The call is released because the other party (which is remote to `ThisDN`) in the `EventReleased` or `EventAbandoned` events was requesting release operation.
- 3 Unknown—The call is released, but T-Server cannot determine the release initiator.

Call-Based Reporting

Independently of DN-based reporting, T-server provides the call release initiator in `AttributeCtrlParty` for `EventCallPartyDeleted` and `EventCallDeleted` events. For scenarios where T-Server cannot provide the release initiator, `AttributeCtrlParty` will not appear in event reporting.

T-Server will provide `AttributeCtrlParty` reporting (for the party that initiated the call release) either:

- When the call is released using a GCTI request and T-Server is aware of the result of the requested operation, or;
- The PBX CTI protocol provides reliable information about the identity of party that released.

Feature Configuration

The `releasing-party-report` configuration option enables or disables the Call Release Tracking feature.

Support for Failed Route Notification

T-Server supports a variety of alarm messages for unsuccessful routing scenarios.

When this feature is enabled, a failed route timer is set using the interval defined in the `route-failure-alarm-period` configuration option. Each routing failure reported during this period is added to a counter. If this counter exceeds a “high water mark” threshold value defined by the `route-failure-alarm-high-wm` configuration option, T-Server sets a route failure alarm condition, and resets the counter.

The alarm condition is cleared when fewer route failures than configured in the `route-failure-alarm-low-wm` configuration option are recorded and there is also no more than the number of route failures configured in `route-failure-alarm-high-wm` in one complete period (configured in `route-failure-alarm-period`).

Setting the value of the `route-failure-alarm-period` configuration option to 0 (zero) disables the feature.

HA Considerations

Only the primary T-Server maintains the failed routing counter. The backup T-server will not run the `route-failure-alarm-period` timer, and so keeps the routing failure alarm in the canceled state.

On switchover from primary role to backup role, T-Server stops the `route-failure-alarm-period` timer and clears any alarm internally, without sending any LMS message.

On switchover from backup role to primary role, T-Server starts the `route-failure-alarm-period` timer and starts counting route requests and routing failures.

Feature Configuration

The following configuration options control the Failed Route Notification feature:

- `route-failure-alarm-high-wm`
- `route-failure-alarm-low-wm`
- `route-failure-alarm-period`

Support for Link Bandwidth Monitoring

T-Server provides bandwidth monitoring on a CTI link and can notify the Genesys Management Layer when Configuration Layer limits are exceeded.

When configured high or low thresholds are reached, T-Server sends alarm messages LINK_ALARM_HIGH LMS or LINK_ALARM_LOW LMS, as appropriate.

High and Low Watermarks

The configuration option, `link-alarm-high`, specified as a percentage of the `use-link-bandwidth` value, defines an upper threshold bandwidth value, which when breached, raises a LINK_ALARM_HIGH LMS message.

The configuration option `link-alarm-low`, specified as a percentage of the `use-link-bandwidth` value, defines a lower threshold bandwidth value, which when breached, raises a LINK_ALARM_LOW LMS message.

LMS Messages

High alarm

STANDARD Link bandwidth: %d1 requests per second exceeds alarm threshold %d2 requests per second on CTI link ID %d3

Attributes:

%d1 represents the measured requests sent on the link

%d2 represents the current link-alarm-high option setting

%d3 represents the CTI Link ID

Low alarm

STANDARD Link bandwidth: %d1 requests per second dropped below alarm threshold %d2 requests per second on CTI link ID %d3

Attributes:

%d1 represents the measured requests sent on the link

%d2 represents the current link-alarm-low option setting

%d3 represents the CTI Link ID

Note: The text description for this message is slightly misleading because the LMS message is created if the set request rate is reached. This is so that users can use 0 (zero) as a value and therefore a low watermark LMS message will still be created. This is different from high watermark handling, where the value must be exceeded to create the LMS message.

If both high and low alarm values are set to 0 (zero), generation of alarms is disabled.

HA Considerations

If the primary T-Server is at the high watermark prior to a switchover, its state is not transferred to the backup T-Server.

Feature Configuration

The following configuration options control the Link Bandwidth Monitoring feature:

- `link-alarm-high`
- `link-alarm-low`
- `use-link-bandwidth`

Request Handling Enhancements

T-Server supports two major enhancements to queue handling: request conflict resolution and a new device queue.

Requests submitted by different clients are treated no differently to requests submitted by the same client. For this reason, having multiple clients controlling the same device can result in unexpected behavior.

Note: While this configuration is supported, it should be recognized that there is no special handling for multiple clients.

Feature Configuration

The following configuration options control the request handling enhancements:

- `call-rq-gap`
- `device-rq-gap`
- `max-outstanding`
- `rq-conflict-check`
- `rq-gap`

Hot-Standby HA Synchronization

This section describes how T-Server supports hot-standby HA synchronization.

Figure 16 shows the process of successful detection of T-Server synchronization. The primary T-Server is assumed to have successfully completed switch synchronization.

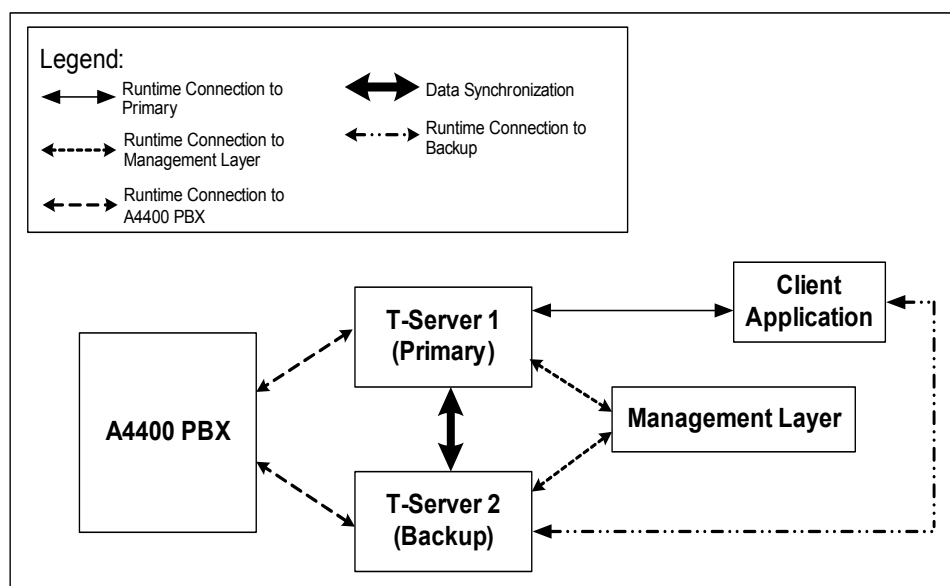


Figure 16: Successful Hot-Standby HA T-Server Synchronization

Primary T-Server Still in Start-up Phase

If the primary T-Server is still in the process of switch synchronization when it receives a `Backup Ready` message from the backup T-Server, the primary T-Server sends the `Full Sync Done` message immediately. This allows the backup T-Server to send `EventLinkConnected` and become available. The Management Layer then sets the backup T-Server as the new primary, and vice versa. Once the old primary T-Server finishes switch synchronization, it then initiates T-Server synchronization with the new primary T-Server as shown in Figure 16.

Primary T-Server's Link with the Switch is Down

If the primary T-Server has lost communication with the switch when it receives a `Backup Ready` message from the other T-Server, then it sends the `Full Sync Done` message immediately. It can be assumed to have lost synchronization with the switch itself and there is no guarantee that it will recover communication with the link, which the backup T-Server currently has.

Backup T-Server Fails During Synchronization

If the backup T-Server fails while waiting for synchronization, then the primary T-Server stops the synchronization process.

Primary T-Server Fails During Synchronization

If the primary T-Server fails while waiting for synchronization, then the backup T-Server sends `EventLinkConnected` immediately.

Call Synchronization Between T-Servers

An integral part of T-Server synchronization is the synchronization of the connection IDs of the calls between the T-Servers. It is the connection IDs of calls created by the backup T-Server during the switch synchronization phase that differ from those in the primary T-Server—those created afterwards are synchronized by the normal HA mechanism. When the primary T-Server receives the `Backup Ready` message from the backup T-Server, it tags all current calls. Once all tagged calls have been released, the primary T-Server can be certain that the connection IDs for all current calls have been synchronized with the backup T-Server because they were created after the backup T-Server completed its startup phase. If no further T-Server synchronization is required, the primary T-Server sends the `Full Sync Done` message to the backup T-Server.

Feature Configuration

The `ha-sync-dly-lnk-conn` configuration option controls of the Hot- Standby HA Synchronization feature.

Support for Boss/Secretary Functionality

Boss/Secretary functionality on the PBX cannot be supported by CTI, because no events are generated. Please refer to your switch documentation for details of this feature.

Support for A4400/OXE Spatial Redundancy

To enable Spatial Redundancy two link sections (`Link-n-name`) must be configured in the T-Server options, and their corresponding sections created in the `Options` tab. One link points to the “role address” used by one Appliance

Server/Media Gateway; the other link points to the “role address” used by the second Appliance Server/Media Gateway.

At link startup, T-Server polls both links until it receives a response from one. It then closes the other link. If the active link goes down, T-Server then reopens both links and keeps polling until it receives a response.

If a PBX switchover occurs, T-Server momentarily loses connection to all links. After the link has been re-established the PBX provides no information for any calls that were ongoing before the switchover, and any calls received during the switchover. Thus, after the switchover is completed, T-Server is unable to report the correct device states or call states for any devices involved in such calls. All calls that arrive after the switchover are handled correctly.

It is possible in some instances for the T-Server and PBX to lose agent state synchronization immediately after a switchover. If this occurs, T-Server attempts to recover, but a manual agent-state change on the phoneset may still be required.

If the communication between the primary and backup Appliance Server/Media Gateway is lost, then one of them enters the Pseudo-Main mode. In this mode the PBX allows T-Server to connect and start up, but no requests are accepted and no events are sent to T-Server.

To provide extra resilience in the event of primary T-Server failure, you can configure the primary and the backup T-servers in different subnets, each T-Server polling both Appliance Servers.

Please see the relevant switch documentation for full details of this feature.

Support for Outbound Caller ID

T-Server supports Outbound Caller ID. Several new keys in the Extensions attribute are introduced to be used in `TMakePredictiveCall`, `TMakeCall` and `TRouteCall` (RSI only):

<code>CPNDigits</code>	Holds the digits according to numbering/dialing plan format that will be passed as a call line identity (CLI).
<code>CPNPlan</code>	Holds the number type (integer or string). Valid values are 0–7.
<code>CPNType</code>	Holds the numbering plan (integer or string). Valid values are 0–15.
<code>CPNPresentation</code>	Holds the presentation indicator (integer or string). Valid values are 0–3.
<code>CPNScreening</code>	Holds screening indicator (integer or string). Valid values are 0–3.

T-Server will check Extension key `CPNDigits` and add any private data therein in the corresponding request to the PBX. The remaining extensions are optional and if values are absent or incorrect T-Server uses default value 0.

Value representations should correspond to those defined in the PBX configuration.

Note: The use of this functionality may be subject to statutory constraints as determined by the telecommunications regulatory bodies in different jurisdictions. Therefore, use of this functionality must be agreed in advance by the relevant regulatory body for your jurisdiction.

Support for Inter Server Call Control

This section does *not* provide a complete guide to configuring Inter Server Call Control (ISCC) functionality, formerly known as External Routing. It only describes that functionality as implemented in T-Server for Alcatel A4400.

ISCC allows T-Servers on different sites to interchange `Connection ID` and `UserData` information when a call is passed from one PBX to another.

ISCC Routing Strategies

The T-Server for Alcatel A4400 supports several types of ISCC routing strategy. The specific strategy determines what method is used to identify the correct call at the remote site to ensure that the `Connection ID` and `UserData` are attached to the call.

You must use the `cast-type` configuration option—located in the `extrouter` section of the T-Server Application object in the Configuration Layer—to define the method ISCC uses to identify the correct call at the remote site.

The following list describes the `cast-type` values you can use with the T-Server for Alcatel A4400.

`cast-type = route-notoken (default)`

This strategy requires that you configure one or more External Routing Points in the local DNs. ISCC at the origination site will substitute the External Routing Point for the actual destination in the `MakeCall/Transfer/Conference/Route` request. When the call arrives at the External Routing Point at the remote site, ISCC will attach the correct connection ID and user data to the call and route the call to the final destination.

`cast-type = direct-uui`

This strategy uses correlator data to identify the call when it arrives at the remote site. Do not use this strategy if you are using correlator data for other

purposes in your environment. Although T-Server attempts to reattach the original correlator data at the remote site, success is not guaranteed.

Note: This strategy only works in cases where correlator data can be transported to the remote site and where the call scenario is such that the PBX allows correlator data to be attached to the call. Please check the restrictions for correlator data for the relevant PBX version.

cast-type = direct-ani

This strategy uses the ANI field of the call to identify it at the remote site.

Note:

- This strategy does not work for `RequestRouteCall` when you are using Emulated Routing Points.
- When a Direct ANI strategy is used (extrouter option `cast-type` is set to `direct-ani`) and a call is initiated, or a consultation call is initiated, from an agent in substitution mode, T-Server will not be able to match calls. The reason for this is that the call is made from a substituted device (ProACD number), and delivered with the ANI of the agent device, which do not match.

cast-type = direct-notoken

This strategy assigns the connection ID and user data to the next call that arrives at the destination.

Note: If another call arrives at the destination between the time that the call data is transferred to the remote site and the ISCC call actually arrives at the destination, then the call data will be assigned to the wrong call.

cast-type = route-uui

This strategy 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.

cast-type = direct-callid

With this strategy, 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.

Note: Cast type `direct-callid` is only supported for partitioned configurations.

cast-type = 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 information 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 the `Target Type` and `ISCC Protocol Parameters` fields of the corresponding `Switch Access Code` in the Configuration Layer.

To configure this:

1. Set `cast-type` to `direct-network-callid`.
2. Configure ABC trunks between nodes.
3. In the `Access Code` tab of the switch, set `direct-network-callid=ts-gcid` in the `ISCC Protocol Parameters` box.

Note: This `cast-type` is only supported with `TRouteCall`.

ISCC/Call Overflow

This section does *not* provide a complete guide to ISCC/Call Overflow (formerly known as External Call Handling) functionality. It only describes that functionality as implemented in T-Server for Alcatel A4400/OXE.

The ISCC/Call Overflow feature in T-Server handles calls delivered between sites by means other than ISCC. The Alcatel A4400 supports this feature as of PBX R4.1.1, which introduces the Global CallID feature for calls made in an ABC-F network. This feature permits the unique identification of calls between sites in the network.

T-Server uses the Global CallID feature with external call handling to attach the correct `ConnectionID` and `UserData` at the remote side.

When T-Server first encounters a new `Global CallID` (a call arrives from outside), the ISCC/Call Overflow feature seizes the T-Server events and broadcasts the `Global CallID` to all connected Alcatel A4400 T-Servers. If one of the remote T-Servers replies, it means that the particular `Global CallID` is known at that site and the remote T-Server sends the `ConnectionID` and `UserData`. The events that were seized are updated with this information and sent to the T-Server clients.

Feature Configuration

Use the following configuration options, located in the `extrouter` section of the T-Server Application object in the Configuration Layer, to configure ISCC/Call Overflow.

- `cof-feature`
- `cof-ci-req-tout`
- `cof-rci-tout`
- `cof-ci-wait-all`
- `cof-ci-defer-delete`
- `cof-ci-defer-create`

Configuring the GCID for ISCC Call Tracking

When Alcatel A4400/OXE switches are networked using the ABC-F trunk protocol, the Global CallID (GCID) reported in CSTA events to T-Server is maintained by Alcatel A4400/OXE even when calls pass between nodes in the network. In such deployments, it is recommended to configure T-Server to use the GCID for ISCC call tracking.

T-Server provides two mechanisms for ISCC call tracking. In order to ensure that all calls made between Alcatel A4400/OXE nodes are correctly tracked by T-Server, it is necessary to enable and configure both of these mechanisms in T-Server:

- **ISCC/extrouter:** The Inter Server Call Control (ISCC) functionality mechanism is used for calls that are routed between nodes using Genesys routing.
- **ISCC/cof:** The Call Overflow mechanism is used for all calls that pass between nodes by other means. This provides, for example, a correct multi-site call tracking for calls that are transferred manually by agents.

Procedure: Providing ISCC Call Tracking

Purpose: To set the two mechanisms needed for ISCC call tracking.

Start of procedure

1. Ensure Alcatel A4400/OXE switches are networked using ABC-F trunk protocol.

Notes:

- Alcatel A4400/OXE provides several overflow mechanisms in case of network congestion between nodes. When using Genesys T-Server only the only overflow mechanism that is supported is VPN Overflow.
- Network optimization is not supported with CCD or RSI.
- When using several Alcatel A4400/OXE PBXs connected through an ABC-F network the date and time must be synchronized on all PBXs in order to ensure proper PBX CSTA reporting to T-Server.

2. Configure ISCC/extrouter. See Chapter 9 on [page 104](#) for details of how to configure T-Server connections and access codes to enable ISCC.

As ABC-F networks use closed numbering where each number is unique in the network, no trunk access codes are required to call between nodes. Therefore, the Switch Access Code objects in Configuration Manager must be set up with a blank access code values (see the Code field in [Figure 17](#)).

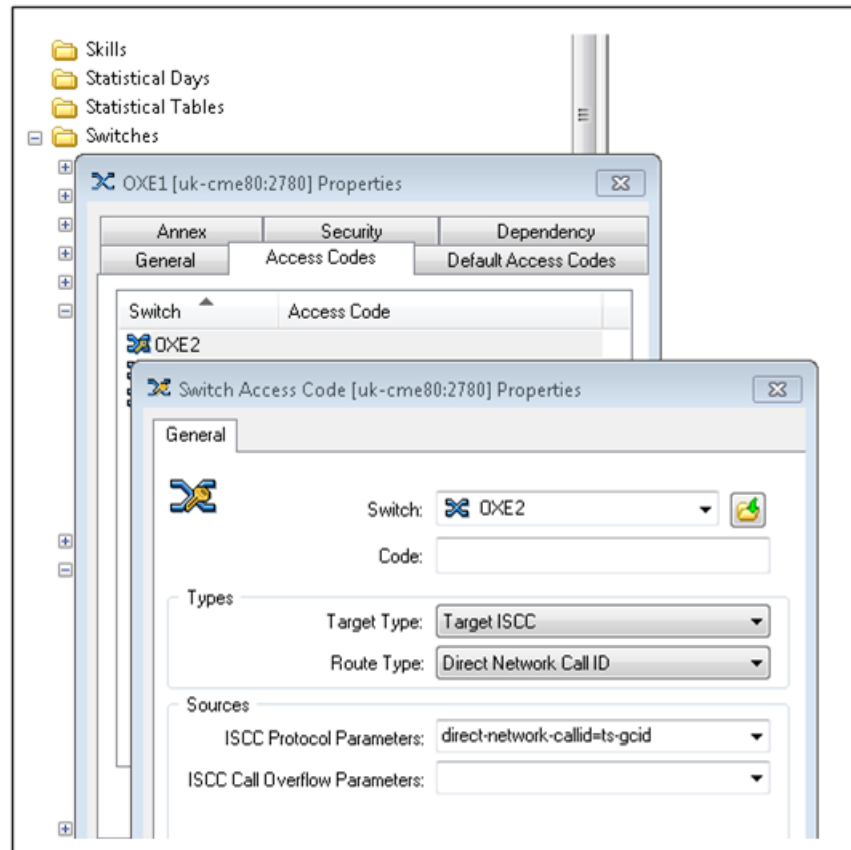


Figure 17: Setting GCID for Tracking Calls

3. Configure ISCC/cof:

Call Overflow must be enabled on all T-Servers in the network, and has to be enabled as follows in both the primary and the backup T-Servers. In the `extrouter` section of the T-Server application, set the following configuration options to the values shown below:

- Set the value for `default-network-call-id-matching` to `ts-gcid`.
- Set the value for `cof-feature` to `true`.

End of procedure

Support for Account Codes

T-Server distinguishes between two situations for the Account Code feature:

- Account codes that are entered during a call or during ACW when the released call is left are treated as account codes.
- Account codes that are entered while the agent is idle are interpreted as walk-away codes.

Any account codes entered during a call are treated by T-Server as call account codes. T-Server reports such account codes using attached user-data with configurable keys. Optionally, T-Server can report an account code as an `Extensions` attribute key, instead of `UserData`, to minimize interference with the other components, such as ISCC and other user applications.

While Interaction Concentrator can pick up event `Extensions` the same as the `UserData`, Stat Server can only work with the `UserData`.

Indexing by Key Name

In scenarios where multiple account codes are required, it is possible to turn on key name indexing. When indexing is enabled, T-Server should attach each subsequent account code and increment the index part of the key.

The index is an incremental integer attached to the configured key name after an underscore, starting with 1 (for example—`AccountCode_1`, `AccountCode_2`, and so on). T-Server only attaches unique codes that are not yet attached. T-Server keeps non-indexed key (for example—`AccountCode`) updated with the last received value irrespective of whether indexing is enabled or not.

In case multiple calls exist on the device, T-Server attaches the code to the last active call. For this reason, T-Server has to keep a historically ordered stack of active calls on the device so that last active call can be easily and reliably identified.

The data key name is set by the configuration option, `accode-name`. If the configured name is different from `AccountCode`, the name for multiple account codes is also changed and represented as "`<value of "accode-name">_<N>`".

Entering Account Codes Manually

If account codes are entered manually on the phoneset (or by some means other than a CTI request from T-Server), T-Server verbosely attaches the user-data key under the configured name to the call. This user-data appears in all subsequent call related events on all devices that are involved in the call. If Extensions are chosen as the account code holder, these are reported with the next call-related event.

Note: If configuration option, `accode-data`, is set to any value other than `udata`, then T-Server does not attach account codes to the user data.

Entering Account Codes by CTI Request

Explicit attaching of account codes by CTI request (Attach/Update UserData) is supported. You are responsible for setting the correct key as configured.

When T-Server receives such a CTI request, it sends this account code to the PBX for the requested DN after the requested call function is complete, and only for the requested DN. In this case, T-Server immediately replies with an `EventAttachedDataChanged` as event user-data is handled by the T-Server common part, and this event is sent automatically. T-Server suppresses reporting of any associated acknowledgment or error from the PBX.

Setting Keys in Call-Related Requests

T-Server allows these keys to be set in any call-related request that supports `UserData` (for example—`TMakeCall` with `UserData`), or in the `Extensions` attribute of any call-related request. If such a request is received, T-Server reports the account code as normal and additionally, it sends the account code to the PBX as soon as the call request to the PBX is successfully completed. T-Server suppresses reporting of any associated acknowledgment or error from the PBX. T-Server does not proceed with the account code update, if the original call-related request fails.

In scenarios using `TReleaseCall`, `TRedirectCall`, `TRouteCall`, `TSingleStepTransfer`, and `TCompleteTransfer`, T-Server attaches the account code first, and then performs a call release or transfer. This is because there is no chance to report, or maybe even attach, the account code after the call is released or transferred. When using `TMuteTransfer`, T-Server attaches the account code after the `EventDialing` event, if the dial-out was successful.

Notes:

- It is not possible to set a call associated account code for a call until the call has been initiated.
- It is not possible to set a call associated account code for a finished call after the call has been released.

T-Server sends an `EventUserEvent` message immediately upon receipt of an account code from the PBX. It is possible to set an account code by CTI during this phase as described above using a `TUpdateUserData` request or a `TPrivateService` request.

Feature Configuration

The following configuration options and `Extensions` attribute key enable T-Server to keep connection IDs alive after a call is released, if there is automatic ACW. This allows agents to enter call-related account codes during the ACW phase:

- | | |
|------------------------------|--|
| Configuration Options | <ul style="list-style-type: none"> • <code>accode-data</code> • <code>accode-name</code> • <code>accode-privateservice</code> • <code>acw-retain-call</code> • <code>acw-retain-lock</code> |
|------------------------------|--|

Extensions Key The default `Extensions` attribute key is `AccountCode`.

Walk-Away Codes

Any account code entered while an agent is idle and not-ready is treated by T-Server as a walk-away code. T-Server reports these walk-away codes using an `EventAgentNotReady` event with the predefined `Extensions` attribute key `ReasonCode` in the `AttributeExtensions` field.

Examples of Different Walk-Away Scenarios

1. Idle and Ready agent:

If the account code is entered manually on the phoneset while the agent is idle and in the Ready state, T-Server either ignores this event from the PBX or reports it using a `TPrivateService` request.
2. Idle and Not-Ready agent:
 - If the account code is entered manually on the phoneset while an agent is in the Not-Ready state, T-Server resends an `EventAgentNotReady` event with the `Extensions` attribute key `ReasonCode`.
 - If an agent who is ready sends a `TAgentNotReady` with the `Extensions` attribute key `ReasonCode` in the `AttributeExtension` field, T-Server immediately sets the agent to the NotReady state and sets the account code on the PBX. T-Server acknowledges the request by sending an `EventAgentNotReady` event with the corresponding `AttributeExtension`.

- If an idle and not-ready agent sends a `TAgentNotReady` request with the `Extensions` attribute key `ReasonCode`, T-Server only sets the account code on the PBX and reports the `EventAgentNotReady` event with the corresponding `AttributeReason`. T-Server does not change the actual agent state on the PBX in this case.

Account Code as a Private Event/Service

You can set and report account code using a `TPrivateService` request and a `EventPrivateInfo` event. The `Extensions` attribute should contain the account code in question, reported or requested. An account code can only be set through the CTI on an established call.

Currently, T-Server uses the same ID (51971) for the request and event. The `Extensions` attribute name is defined by the configuration option, `acccode-name`.

T-Library Functionality

[Table 21](#) presents T-Library functionality supported by T-Server for Alcatel A4400/OXE switch. The table entries use these notations:

- **N**—Not supported
- **Y**—Supported
- **I**—Supported, but reserved for Genesys Engineering
- **E**—Event only supported

In [Table 21](#), when a set of events is sent in response to a single request, the events are listed in an arbitrary order. An asterisk (*) indicates the event that contains the same Reference ID as the request. For more information, refer to the *Genesys Events and Models Reference Manual*.

[Table 21](#) reflects only that switch functionality that Genesys software uses and might not include the complete set of events that the switch offers.

Certain requests listed in [Table 21](#) are reserved for Genesys Engineering and are listed here merely for completeness of information.

Detailed notes for specific functionalities may appear at the end of a table.

Table 21: Supported T-Library Functionality

Feature Request/Subtype	Supported	Corresponding Events/Comments
General Requests		
TOpenServer	I	EventServerConnected
TOpenServerEx	I	EventServerConnected

Table 21: Supported T-Library Functionality (Continued)

Feature Request/Subtype	Supported	Corresponding Events/Comments
TCloseServer	I	EventServerDisconnected
TSetInputMask	I	EventACK Internal service. CSTA Set Monitor Filter service is not used.
TDispatch	I	Not applicable
TScanServer	I	Not applicable
TScanServerEx	I	Not applicable
Registration Requests		
TRegisterAddress	Y	EventRegistered. Monitor begins at startup, even if no clients are connected. Restrictions apply.
TUnregisterAddress	Y	EventUnregistered. Monitors are stopped only when a device is disabled, deleted, or reconfigured into Local mode.
Call-Handling Requests		
TMakeCall	Y	<p>EventDialing</p> <p>An analog extension must have Hands-Free mode turned on. Note the following possibilities:</p> <ul style="list-style-type: none"> CTI_INHIBIT_PROGRESS_TONE, GCTI_AUTO_ORIGINATE, GCTI_SUPERVISOR_CALL, and GCTI_REQUESTING_DEVICE keys can be used in request. ACR List can also be defined. An account code defined by option accode-name could be used. CPNxxx keys could be used. <p>Call unparking is supported from release 7.0.204.</p> <p>Note: When using a prefix in TMakeCall in order to activate a switch feature this prefix must begin with a * or #. This means the prefix must be configured accordingly on the switch.</p>
TMakeCall	N	
• Regular		
• DirectAgent	N	
• SupervisorAssist	N	

Table 21: Supported T-Library Functionality (Continued)

Feature Request/Subtype	Supported	Corresponding Events/Comments
• Priority	N	
• DirectPriority	N	
TAnswerCall	Y	Analog extension must have Hands-Free mode turned on.
TReleaseCall	Y	EventReleased Target must be in Dialing or Established state. Extensions attribute key GCTI_RECONNECT_GUIDE_LEVEL can be used in this request.
TClearCall	N	
THoldCall	Y	EventHeld Only for T0/T2 calls
TRetrieveCall	Y	EventRetrieved
TRedirectCall	Y	EventReleased
TMakePredictiveCall	Y	EventDialing*, EventQueued Emulation of TMakePredictiveCall is supported (you must define Predictive Dialing devices in the Configuration Layer to use this service). The GCTI_CSTA_CORRELATOR and ACR List keys can be used in this request. CPNxxx extensions can also be used.
Transfer/Conference Requests		
TInitiateTransfer	Y	EventHeld, EventDialing* GCTI_INHIBIT_PROGRESS_TONE, GCTI_INHIBIT_HOLD_TONE, GCTI_PRIORITARY_TRANSFER, GCTI_SUPERVISOR_CALL, and GCTI_SUPERVISED_TRANSFER Extensions attribute keys can be used in the request. ACR List can also be defined.
TCompleteTransfer	Y	EventReleased*, EventPartyChanged
TInitiateConference ^a	Y	EventHeld, EventDialing* GCTI_INHIBIT_PROGRESS_TONE, GCTI_INHIBIT_HOLD_TONE, GCTI_PRIORITARY_TRANSFER, GCTI_SUPERVISOR_CALL, and GCTI_SUPERVISED_TRANSFER keys can be used in the request. ACR List can also be defined.

Table 21: Supported T-Library Functionality (Continued)

Feature Request/Subtype	Supported	Corresponding Events/Comments
TCompleteConference	Y	EventReleased*, EventRetrieved, EventPartyChanged, EventPartyAdded Only three-party conferences are supported.
TDeleteFrom Conference	Y	EventPartyDeleted*, EventReleased
TReconnectCall	Y	EventReleased, EventRetrieved*
TAlternateCall	Y	EventHeld*, EventRetrieved
TMergeCalls	N	
• For Transfer	N	
• For Conference	N	
TMuteTransfer	Y	EventReleased*, EventPartyChanged
TSingleStepTransfer	Y	EventHeld, EventDialing*
TSingleStep Conference	Y	EventReleased*, EventRetrieved, EventPartyChanged, EventPartyAdded Activates CCD supervisor monitoring of CCD agents. Use the GCTI_SUPERVISOR_STEP_IN and GCTI_PARTICIPATION_TYPE Extensions attribute keys to set the Monitoring mode. You must configure an ACD Listening button in the PBX for the supervisor to use this request.

Table 21: Supported T-Library Functionality (Continued)

Feature Request/Subtype	Supported	Corresponding Events/Comments
Call-Routing Requests		
TRouteCall	Y	<p>EventRouteUsed</p> <p>If RSI Routing Points (available on PBX version 4.2 or higher) are not used, T-Server can emulate the routing service using virtual devices (virtual Zs). In this case the switch-routing service is not used. When a call is delivered to a virtual Z in the Hunt Group, T-Server generates a RouteRequestEvent to its clients.</p> <p>See Chapter 15, “Using Alcatel A4400 Routing Services Interface” on page 267, and page 164 for more information.</p>
• Unknown	Y	
• Default	Y	<p>For RSI routing services:</p> <ol style="list-style-type: none"> 1. If destination is not specified in RouteCall: <ul style="list-style-type: none"> • If default destination is specified in PBX configuration, the PBX routes the call. • Otherwise, the call is disconnected. 2. If the destination is specified in RouteCall, then the call is routed to the specified destination.
• Label ^b	Y	
• OverwriteDNIS ^b	Y	
• DDD ^b	Y	
• IDDD ^b	Y	
• Direct ^b	Y	
• Reject	Y	<p>For RSI routing services, if destination is not specified in RouteCall:</p> <ul style="list-style-type: none"> • If default destination is specified in PBX configuration, the PBX routes the call. • Otherwise, the call is disconnected. <p>From T-Server release 7.5, T-Server does not request the release of the call from RSI, but only rejects the routing service. So, the PBX configuration must handle the rejected call (for example, by overflow or disconnect).</p>
• Announcement ^b	Y	

Table 21: Supported T-Library Functionality (Continued)

Feature Request/Subtype	Supported	Corresponding Events/Comments
• PostFeature ^b	Y	
• DirectAgent ^b	Y	
• Priority ^b	Y	
• DirectPriority ^b	Y	
• AgentID ^b	Y	
Call-Treatment Requests		
TApplyTreatment	Y	(EventTreatmentApplied + EventTreatmentEnd)/EventTreatmentNotApplied
• Unknown	N	
• IVR	N	
• Music	Y	Only for a call on RSI.
• RingBack	Y	Only for a call on RSI.
• Silence	Y	Only for a call on RSI.
• Busy	Y	Only for a call on RSI.
• CollectDigits	Y	Only for a call on RSI.
• PlayAnnouncement	Y	Only for a call on RSI.
• PlayAnnouncement AndDigits	Y	Only for a call on RSI.
• VerifyDigits	N	
• RecordUser Announcement	N	
• DeleteUser Announcement	N	
• CancelCall	Y	Only for a call on RSI.
• PlayApplication	N	
• SetDefaultRoute	N	
• TextToSpeech	N	

Table 21: Supported T-Library Functionality (Continued)

Feature Request/Subtype	Supported	Corresponding Events/Comments
• TextToSpeechAnd Digits	N	
• FastBusy	N	
• RAN	N	
TGiveMusicTreatment	N	
TGiveRingBack Treatment	N	
TGiveSilence Treatment	N	
DTMF (Dual-Tone Multifrequency) Requests		
TCollectDigits	N	
TSendDTMF	Y	EventDTMFSent
Voice-Mail Requests		
TOpenVoiceFile	N	
TCloseVoiceFile	N	
TLoginMailBox	N	
TLogoutMailBox	N	
TPlayVoice	N	
Agent & DN Feature Requests		
TAgentLogin	Y	EventAgentLogin See “Agent Login” on page 147 for information on CCD agents. See “Support for the Keep-Alive Feature” on page 196 for information about emulated agents.
• WorkModeUnknown	Y	
• ManualIn	Y	For emulated agents
• AutoIn	Y	
• AfterCallWork	N	

Table 21: Supported T-Library Functionality (Continued)

Feature Request/Subtype	Supported	Corresponding Events/Comments
• AuxWork	N	
• NoCallDisconnect	N	
TAgentLogout	Y	EventAgentLogout See Table 13 on page 145 .
TAgentSetIdleReason	N	
TAgentSetReady	Y	EventAgentReady
TAgentSetNotReady	Y	EventAgentNotReady
• WorkModeUnknown	Y	
• ManualIn	N	
• AutoIn	Y	
• AfterCallWork	Y	
• AuxWork	Y	
• NoCallDisconnect	N	
TMonitorNextCall	Y	EventMonitoringNextCall
TCancelMonitoring	Y	EventMonitoringCanceled
TCallSetForward	Y	EventForwardSet
• None	Y	
• Unconditional	Y	
• OnBusy	N	
• OnNoAnswer	N	
• OnBusyAndNo Answer	N	
• SendAllCalls	N	
TCallCancelForward	Y	EventForwardCancel
• None	Y	
• Unconditional	Y	

Table 21: Supported T-Library Functionality (Continued)

Feature Request/Subtype	Supported	Corresponding Events/Comments
• OnBusy	N	
• OnNoAnswer	N	
• OnBusyAndNo Answer	N	
• SendAllCalls	N	
TSetMuteOff	N	
TSetMuteOn	N	
TListenDisconnect	N	
TListenReconnect	N	
TSetDNDOOn	N	
TSetDNDOff	N	
TSetMessage WaitingOn	N	
TSetMessage WaitingOff	N	
Query Requests		
TQuerySwitch	N	EventSwitchInfo
• DateTime	N	
• ClassifierStat	N	
TQueryCall	Y	
• PartiesQuery	N	
• StatusQuery	Y	EventPartyInfo
TQueryAddress	Y	EventAddressInfo
• AddressStatus	Y	
• MessageWaiting Status	N	
• AssociationStatus	N	

Table 21: Supported T-Library Functionality (Continued)

Feature Request/Subtype	Supported	Corresponding Events/Comments
• CallForwarding Status	Y	
• AgentStatus	Y	
• NumberOfAgentsIn Queue	Y	When using the CCD, this information is only provided for CCD/RSI Processing Groups.
• NumberOfAvailable AgentsInQueue	Y	When using the CCD, this information is only provided for CCD/RSI Processing Groups.
• NumberOfCallsIn Queue	Y	
• AddressType	Y	<p>If the address type is ACDQ (CCD Pilot), acknowledgment provides the CCD status, which includes the following information:</p> <ul style="list-style-type: none"> • Pilot status (open/close) • Transfer possible (yes/no) • Waiting time • Pilot saturated (yes/no) <p>GCTI_PRIORITY_TRANSFER and ACR List Extensions attribute keys can be used in this case.</p>
• CallsQuery	Y	
• SendAllCallsStatus	Y	
• QueueLoginAudit	Y	
• NumberOfIdleTrunks	N	
• NumberOfTrunksIn Use	N	
• DatabaseValue	N	
• DNStatus	Y	
• QueueStatus	Y	Only for ACD/RP devices
TQueryLocation	I	EventLocationInfo ^c
• AllLocations	I	
• LocationData	I	

Table 21: Supported T-Library Functionality (Continued)

Feature Request/Subtype	Supported	Corresponding Events/Comments
• MonitorLocation	I	
• CancelMonitor Location	I	
• MonitorAllLocations	I	
• CancelMonitorAll Locations	I	
TQueryServer	Y	EventServerInfo
User-Data Requests		
TAttachUserData	Y	EventAttachedDataChanged T-Server sets the CSTA call correlator data using the CSTA Associate Data service if the GCTI_CSTA_CORRELATOR and/or the GCTI_CSTA_ACCOUNT_INFO Extensions attribute keys are present.
TUpdateUserData	Y	EventAttachedDataChanged See TAttachUserData
TDeleteUserData	Y	EventAttachedDataChanged
TDeleteAllUserData	Y	EventAttachedDataChanged
ISCC (Inter-Server Call Control) Requests		
TGetAccessNumber	I	EventAnswerAccessNumber
TCancelReqGet AccessNumber	I	EventReqGetAccessNumberCanceled
Special Requests		
TReserveAgent	I	EventAgentReserved
TSendEvent	I	EventACK
TSendEventEx	I	EventACK
TSetCallAttributes	Y	EventCallInfoChanged
TSendUserEvent	Y	EventACK
TPrivateService	Y	EventPrivateInfo

Table 21: Supported T-Library Functionality (Continued)

Feature Request/Subtype	Supported	Corresponding Events/Comments
Network Requests^d		
TNetworkConsult	Y	EventNetworkCallStatus
TNetworkAlternate	Y	EventNetworkCallStatus
TNetworkTransfer	Y	EventNetworkCallStatus
TNetworkMerge	Y	EventNetworkCallStatus
TNetworkReconnect	Y	EventNetworkCallStatus
TNetworkSingleStep-Transfer	Y	EventNetworkCallStatus
TNetworkPrivate-Service	Y	EventNetworkPrivateInfo
ISCC Transaction Monitoring Requests		
TTransaction Monitoring	Y	EventACK
	E	EventTransactionStatus

- a. Conference calls are supported only for manual operations.
- b. Treated like subtype Unknown.
- c. Two subtypes are supported by EventLocationInfo: LocationMonitorCanceled and InfoAllLocations-MonitorCanceled.
- d. All T-Servers support NAT/C requests with AttributeHomeLocation, provided that this attribute identifies a network location that is capable of processing such requests.

CTI-Supported Functionality for SIP Extensions

Table 22 lists and describes CTI-supported requests for SIP extensions.

Table 22: CTI-Supported Functionality

Feature Request/Subtype	Support	Comments
TRegisterAddress	Y	
TUnregisterAddress	Y	

Table 22: CTI-Supported Functionality (Continued)

Feature Request/Subtype	Support	Comments
TMakeCall	Y	There is no auto-originate function for SIP extensions, so TMakeCall will result in the line being engaged on the SIP extension. The line button needs to be physically selected on the SIP extension in order for the call to be initiated. Auto-answer can be enabled on the SIP extension to avoid the need to press any buttons on the set.
TAnswerCall	N	The switch will accept and acknowledge the TAnswerCall request from T-Server but do nothing else.
TReleaseCall	Y	
THoldCall	N	
TRetrieveCall	N	
TRedirectCall	N	The switch will accept and acknowledge the TRedirectCall request from T-Server but do nothing else.
TMakePredictiveCall	N	
TInitiateTransfer	N	
TCompleteTransfer	N	
TInitiateConference	N	
TCompleteConference	N	
TDeleteFromConference	Y	
TReconnectCall	N	
TAlternateCall	N	
TMuteTransfer	N	
TSingleStepTransfer	N	
TSingleStepConference	N	
TRouteCall	N	

Table 22: CTI-Supported Functionality (Continued)

Feature Request/Subtype	Support	Comments
TApplyTreatment	N	
TSendDTMF	N	
TAgentLogin	Y	Supported for emulated agents only.
TAgentLogout	Y	Supported for emulated agents only.
TAgentSetReady	Y	Supported for emulated agents only.
TAgentSetNotReady	Y	Supported for emulated agents only.
TMonitorNextCall	N	
TCancelMonitoring	N	
TCallSetForward	Y	Only modes None and Unconditional Forward are supported.
TCallCancelForward	Y	Only modes None and Unconditional Forward are supported.
TQueryCall	Y	Only status queries are supported.
TQueryAddress	Y	See Table 21 for a full list of support.
TQueryServer	Y	
TAttachUserData	Y	
TUpdateUserData	Y	
TDeleteUserData	Y	
TDeleteAllUserData	Y	
TSetCallAttributes	Y	
TSendUserEvent	Y	
TPrivateService 4 Set Device in Service	Y	
TPrivateService 6 Request Supervisor Help	N	No agent functionality is available for SIP extensions.
TPrivateService 7 Cancel Requested Supervisor Help	N	No agent functionality is available for SIP extensions.

Table 22: CTI-Supported Functionality (Continued)

Feature Request/Subtype	Support	Comments
TPrivateService 15 Activate Permanent Listening	N	No agent functionality is available for SIP extensions.
TPrivateService 2011 Start Message Recording	N	
TPrivateService 2008 Stop Message Recording	N	
TPrivateService 2020 Park Call	N	Call parking from SIP extensions is not supported.
TPrivateService 23 Force Device Reset	Y	

Support for Agent Work Modes

Table 23 indicates the types of agent work modes that T-Server for Alcatel A4400/OXE supports.

Table 23: Supported Agent Work Modes

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

Private Services and Events

Table 24 describes private services and events.

Table 24: Private Services and Events

Service	Attribute	Value	Description
Set Device In Service	Service Number	4	Allows an extension to be put out of or into service (escape service Set Device In/Out Of Service).
	This DN	Device in question	Specifies the device for which the service is requested.
	Extensions key: GCTI_SET_IN_SERVICE	0, 1	Specifies the required action for the service: <ul style="list-style-type: none"> 1 puts device back in service. 0 or any other value puts device out of service. Note: From T-Server 7.5, it is possible to disable routing on RSI points even if there are calls on the Routing Point. When routing is disabled, or when RSI is removed from Configuration Manager, and there are no clients, T-Server immediately stops RSI and overflows existing calls.
Request Supervisor Help	Service Number	6	Allows an agent to request help from a supervisor (escape service Supervisor Help Request).
	ThisDN	Device in question	Specifies a device for which service is requested.
Cancel Requested Supervisor Help	Service Number	7	Allows an agent to cancel previously requested supervisor assistance and a supervisor to reject a request for help from an agent (escape service Cancel Supervisor Help Request).
	ThisDN	Device in question	Specifies the device for which the service is requested.
	Extensions key: GCTI_OTHER_DN	Other device	If agent requests service, the extension key specifies the supervisor extension for which the agent cancels the request for help. If supervisor requests service, the extension specifies the agent device for which requested help is rejected.

Table 24: Private Services and Events (Continued)

Service	Attribute	Value	Description
Activate Permanent Listening	Service number	15	Allows a supervisor to activate agent device listening permanently (escape service Permanent Listening Activation).
	ThisDN	Device in question	Specifies the device for which the service is requested. You must configure the device as a supervisor in the switch configuration.
	Extensions key: GCTI_OTHER_DN	Other device	Specifies an agent device to which a supervisor wants to listen.
Start Message Recording	Service number	2011	Begins the recording of a message on a specified extension (CSTA 2 service Message recording start).
	ThisDN	Device in question	Specifies the device for which the service is requested.
	ConnectionID	Call in question	Specifies a call for which the service is requested.
Stop Message Recording	Service number	2008	Stops the recording of a message on a specified extension (CSTA 2 service Message recording stop).
	ThisDN	Device in question	Specifies the device for which the service is requested.
	Connection ID	Call in question	Specifies a call for which the service is requested.
Park Call	Service number	2020	Allows the parking of a call on a specified device (CSTA 2 service Park call).
	ThisDN	Device in question	Specifies the device for which the service is requested.
	Connection ID	Call in question	Specifies a call for which the service is requested.
	Extensions key: GCTI_OTHER_DN	Other device	Specifies the device on which the call should be parked.

Table 24: Private Services and Events (Continued)

Service	Attribute	Value	Description
Supervisor Assistance Request Event	Event number	6 or 7	Private event Supervisor Assist Request Event. Value 6 is reported when an agent requests assistance (other DN is empty) or a supervisor offers assistance (other DN indicates an agent device for which assistance is offered). Value 7 is reported after an agent cancels a request for assistance (other DN indicates a supervisor device) or a request has been rejected (other DN indicates whose request for assistance was rejected).
	ThisDN	Device in question	Agent or supervisor device.
	OtherDN	Other device	Supervisor or agent device, or empty.
	Call attributes		Reflects information about a call on a device that either requested assistance or had a request rejected.
Busy Device Event	Event number	1005	Private event Busy Device Event.
	ThisDN	Device in question	Device for which event is received.
	OtherDN	Other device	Dialing number of calling device.
	Extensions key: GCTI_BUSY_CAUSE	Cause	Specifies a busy cause.
CallingLine Identity	Event number	1008	Private Event Calling Party Name Event
	This DN	Device in question	A device the event was received for.
	Call attributes		Reflects information about the call on device.
	Extensions key: Value of GCTI_OTHER_DEVICE_NAME		Reflects information regarding Calling Party Name.
	Extensions key: Value of CallerID		Reflects information regarding Calling Party Name.

Table 24: Private Services and Events (Continued)

Service	Attribute	Value	Description
Transaction Code Event	Event number	2100	CSTA 2 event Call Information Event.
	ThisDN	Device in question	Agent/supervisor device.
	Extensions key: Value of <code>accode-name</code> option	Transaction code	Specifies a received transaction code.
Message Recording Started Event	Event number	2103	CSTA 2 event Message recording started.
	ThisDN	Device in question	Device on which a recording service is started.
	Call attributes		Reflects information about a call (if any) that existed on the device when recording is started.
Message Recording Stopped Event	Event number	2105	CSTA 2 event Message recording stopped.
	ThisDN	Device in question	Device on which a recording service is stopped.
	Call attributes		Reflects information about a call (if any) that existed on the device when recording is stopped.
Call Parked Event	Event number	2120	CSTA 2 event Delivered.
	ThisDN	Device in question	Device on which the call is parked.
	OtherDN	Other party in a call	Other party in the call.
	Call attributes		Reflects information about the call parked on the device.
Call Unparked Event	Event number	2121	CSTA 2 event Delivered/Diverted/Connection Cleared.
	ThisDN	Device in question	Device from which the call is unparked.
	OtherDN	Other party in a call	Other party in the call.
	Call attributes		Reflects information about the call unparked from the device.

Table 24: Private Services and Events (Continued)

Service	Attribute	Value	Description
Forced Device Reset	Service number	23	Escape service 'Forced Device Reset.'
	ThisDN	Device in question	Device for which the service is requested.

Use of the Extensions Attribute

T-Server for the Alcatel A4400 switch supports use of the `Extensions` attribute, as detailed in [Table 25](#). This attribute is populated in other events/requests, as specified in the `Used In` column.

Table 25: Use of the Extensions Attribute

Extension		Used In	Description
Key	Type		
GCTI_CSTA_CALLS_IN_QUEUE	Integer	EventQueued	The total number of calls in a queue, as reported by CCD.
GCTI_CSTA_CALLS_IN_FRONT	Integer	EventQueued	The number of calls in a queue in front of this call, as reported by CCD.
GCTI_INFO_STR	String	EventAgentNotReady	A text string providing additional description of the nature of the event.
GCTI_SOURCE_ERROR_TYPE	Integer	EventError	Original type of the error returned by the switch.
GCTI_SOURCE_ERROR_CODE	Integer	EventError	Original error code of the error returned by the switch.
GCTI_AUTO_ORIGINATE	Integer	TMakeCall	Permits automatic answering of a call on the origination device.
GCTI_SUPERVISOR_CALL	Integer	TMakeCall TInitiateTransfer TInitiateConference TMuteTransfer TSingleStepTransfer	Allows the override of the default value for the <code>supervisor-call</code> option for a particular request. When this extension is set to a nonzero value, an initiated call is treated as a supervisor call.

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
GCTI_INHIBIT_PROGRESS_TONE	Integer	TMakeCall TInitiateTransfer TInitiateConference TMuteTransfer TSingleStepTransfer	<p>Allows the override of the default value for the inhibit-progress-tone configuration option for a particular request.</p> <p>When this extension is set to a nonzero value, the progress tone is inhibited.</p> <p>Note: Only available for IVR devices.</p>
GCTI_INHIBIT_HOLD_TONE	Integer	TInitiateTransfer TInitiateConference TMuteTransfer TSingleStepTransfer	<p>Allows the override of the default value for the inhibit-hold-tone option for a particular request.</p> <p>When this extension is set to a nonzero value, the hold tone is inhibited.</p> <p>Note: Only available for IVR devices.</p>
GCTI_THIS_DEVICE_NAME	String	EventAgentLogin EventAgentLogout EventAgentReady EventAgentNotReady EventAddressInfo	The name of the DN specified in the <code>ThisDN</code> attribute.
GCTI_OTHER_DEVICE_NAME	String	EventRinging EventQueued	<p>The name of the DN specified in the <code>OtherDN</code> attribute.</p> <p><code>EventQueued</code> or <code>EventRinging</code> triggers reporting of the extension in all subsequent events for the call as long as CCD is involved in the call.</p>

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
GCTI_PRIORITY_TRANSFER	Integer	TInitiateTransfer TInitiateConference TMuteTransfer TSingleStepTransfer TQueryAddress EventAddressInfo	<p>Allows the override of the default value for the priority-transfer option for a particular request.</p> <p>When this extension is set to a nonzero value, the priority transfer flag is associated with a consultation call to a pilot.</p> <p>Pilot QueryAddress with <code>info_type 8</code> reports the availability of priority transfer on the pilot.</p> <p>A positive response in EventAddressInfo indicates that priority transfer is possible.</p> <p>Note: If both GCTI_SUPERVISOR_CALL and GCTI_PRIORITY_TRANSFER are defined in the same call-related request, GCTI_SUPERVISOR_CALL is ignored.</p> <p>Note: Only available for IVR devices.</p>
GCTI_SUPERVISOR_STEP_IN	Integer	TSingleStep-Conference	<p>Allows the override of the default value for the supervisor-step-in option for a particular request.</p> <p>When this extension is set to a nonzero value, a request for a single-step conference is treated as a supervisor step-in. See also GCTI_PARTICIPATION_TYPE.</p>
GCTI_PARTICIPATION_TYPE	Integer	TSingleStepTransfer	<p>Allows the override of the default value of the participation-type option for a particular request. See also GCTI_SUPERVISOR_STEP_IN.</p>

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
GCTI_SUPERVISED_TRANSFER	Integer	EventAddressInfo TInitiateTransfer TInitiateConference	<p>Allows the override of the default value for the supervised-transfer option for a particular request.</p> <p>When this extension is set to a nonzero value, a consultation call to a pilot is initiated in Supervised mode.</p> <p>EventQueued or Event Ringing triggers reporting of the extension in all subsequent events for the call as long as CCD is involved in the call.</p> <p>Note: Only available for IVR devices.</p>
GCTI_HEADSET_MODE	Integer	TAgentLogin	<p>Allows the override of the default value for the headset-mode option for a particular request.</p> <p>When this extension is set to a nonzero value, Headset mode is activated. You must configure the phone set where this feature is activated with a Headset button in the PBX.</p>
GCTI_BLOCKED	Integer	EventAddressInfo	<p>Indicates whether the pilot is blocked; 1 for yes or 0 for no. The extension is provided if the queried device is a queue pilot.</p> <p>EventQueued or Event Ringing triggers reporting of the extension in all subsequent events for the call as long as CCD is involved in the call.</p>

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
GCTI_TRANSFER_POSSIBLE	Integer	EventAddressInfo	Indicates whether it is possible to transfer to a pilot number; 1 for yes or 0 for no. The extension is provided if the queried device is a pilot. EventQueued or Event Ringing triggers reporting of the extension in all subsequent events for the call as long as CCD is involved in the call.
GCTI_AGENT_GROUP	String	See description	Represents the CCD Processing Group where the agent is logged in. EventQueued or Event Ringing triggers reporting of the extension in all subsequent events for the call as long as CCD is involved in the call.
GCTI_NETWORK_TIMESLOT	Integer	See description	Specifies the time slot used by the current call. EventQueued or Event Ringing triggers reporting of the extension in all subsequent events for the external call as long as CCD is involved in the call.
GCTI_GLOBAL_WAITING_TIME	Integer	See description	Represents global waiting time in CCD distribution. EventQueued or Event Ringing triggers reporting of the extension in all subsequent events for the call as long as CCD is involved in the call.

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
GCTI_WAITING_TIME	Integer	See description	Represents waiting time in CCD distribution. EventQueued or Event Ringing triggers reporting of the extension in all subsequent events for the call as long as CCD is involved in the call.
GCTI_ESTIMATED_WAITING_TIME	Integer	EventAddressInfo	Indicates the estimated time that a call waits on the pilot before the PBX distributes it to an agent. The extension is provided if the queried device is a pilot and the pilot is not blocked. EventQueued or Event Ringing triggers reporting of the extension in all subsequent events for the call as long as CCD is involved in the call.
GCTI_SATURATION	Integer	EventAddressInfo	Indicates whether the pilot is saturated. The extension is provided if the queried device is a pilot and the pilot is not blocked. EventQueued or Event Ringing triggers reporting of the extension in all subsequent events for the call as long as CCD is involved in the call.
GCTI_NOT_READY_ACTIVATION	Integer	TAgentNotReady EventAddressInfo EventAgentNotReady	Specifies the withdrawal type as configured in the CCD Processing Group. Note: If set, this extension overrides the same key in the User Data extension.
GCTI_PREASSIGNED_AGENT	Integer	EventAddressInfo EventAgentLogin	Indicates whether the agent is in the Preassigned state.

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
GCTI_REQUESTING_DEVICE	String	TMakeCall	Indicates the device (must be analog virtual device) to be used for taxation purposes for a call.
GCTI_REMAIN_RETRY	Integer	TRouteCall	Indicates the number of route attempts to be made if a call cannot be successfully routed from an RSI. Note: You can only use this extension for calls on an RSI.
GCTI_REROUTE_AUTHORISATION	Integer	TRouteCall	Informs switch of the conditions that will allow rerouting if the current routing attempt fails. Note: You can only use this extension for calls on an RSI.
GCTI_RELEASE_WITH_BUSY_CAUSE	Integer, String	TReleaseCall	Used to allow release of T0/T2 calls in Ringing state.
GCTI_PILOT_NUMBER	String	See description	Appears when a CCD call is presented to a port on an IVR-in-Queue. EventQueued or Event Ringing triggers reporting of the extension in all subsequent events for the call as long as CCD is involved in the call.
GCTI_CURRENT_GUIDE_LEVEL	Integer	See description	Appears when a CCD call is presented to a port on an IVR-in-Queue. EventQueued or Event Ringing triggers reporting of the extension in all subsequent events for the call as long as CCD is involved in the call.

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
GCTI_REROUTED_CALL_INDICATION	Integer	See description	Indicates if the call was rerouted in the network. EventQueued or Event Ringing triggers reporting of the extension in all subsequent events for the call as long as CCD is involved in the call.
GCTI_VPS_CODE	Integer	See description	Indicates the voice-processing system (VPS) protocol used in actions the switch initiates to the VPS. EventQueued or Event Ringing triggers reporting of the extension in all subsequent events for the call as long as CCD is involved in the call.
GCTI_LAST_REDIRECTION_DEVICE	String	See description	Indicates the last device that redirected the call. EventQueued or Event Ringing triggers reporting of the extension in all subsequent events for the call as long as CCD is involved in the call.
GCTI_SUPERVISED_ROUTE	Integer/string	TRouteCall	Specifies the use of Divert (value of 0) or Transfer (nonzero value) Services to route the call when the destination is unknown, unmonitored, or is a pilot. See “Supervised Routing to CCD Pilots” on page 169 .
GCTI_EMUL_WAIT_TIME	Integer	Any call-related events	Indicates the time, in seconds, that a call is or has been in the Nonestablished state since the last time the call was established. Note: Reported if the option report-emul-wait-info is set.

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
GCTI_EMUL_GLOB_WAIT_TIME	Integer	Any call-related events	Indicates the total accumulated time, in seconds, that a call is or has been in the Nonestablished state. Note: Reported if the option report-emul-wait-info is set.
GCTI_RECONNECT_GUIDE_LEVEL	Integer	TReleaseCall	Specifies to which CCD guide level a call on an IVR-in-Queue is to be connected for further distribution. When you have configured an IVR as IVR-in-Queue and a call is established on one of its ports, you can return the call to the CCD and specify to which guide level to connect the call. Note: The selected guide level must be greater than the current guide level.
GCTI_SECRET_ID_NN	String	EventRinging and all events related to the call and generated afterwards	Indicates one of the secret identities of the caller. <i>NN</i> is a two-digit number starting from 01.
GCTI_GLOB_CID	Binary string (8 bytes)	Any call-related event	Indicates the global call identifier in the switching domain.
GCTI_OLD_GLOB_CID	Binary string (8 bytes)	Any call-related events in scenario that involves consultation call	Indicates the old global call identifier in the switching domain.
GCTI_OTHER_DN	String	TPrivateService	Specifies the following: <ul style="list-style-type: none"> • Supervisor device for canceling assistance • Agent device for permanent listening activation See Table 24 on page 227 .

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
GCTI_SET_IN_SERVICE	Integer	TPrivateService	Value 1 puts the device into service. All other values put the device out of service.
GCTI_ACTIVE_MONITORING	Integer	EventCallInfo-Changed EventDialing EventRinging EventEstablished EventPartyAdded EventPartyChanged EventPartyDeleted EventReleased	See “Monitoring Mode” on page 156 .
EXPECTED_WAIT_TIME	Integer	TApplyTreatment TPlayAnnouncement TPlayAnnouncement-AndDigits	Specifies the expected waiting time, in seconds, to be announced to the caller.
POS_IN_QUEUE	Integer	TApplyTreatment TPlayAnnouncement TPlayAnnouncement-AndDigits	Specifies the position in queue to be announced to the caller.
GCTI_CC_TREATMENT_TYPE	Integer	TAgentSetNotReady EventAgentNotReady	Specifies whether Extensions key GCTI_CC_TREATMENT_TYPE is reported in call- and device-related events. The following values can appear: 0–E-mail application for postponed e-mail 1–Outbound application 2–Web chat application 3–Web callback application 4–E-mail application for transferred e-mail 5–Ending pause and staying in Withdraw state 255–Other

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
GCTI_SET_RESERVATION	Integer	TAgentSetNotReady	Allows you to reserve an agent. You can define the reason for reservation in Extensions key GCTI_CC_TREATMENT_TYPE. The Extensions key is only treated for requests with workmode WorkingAfterCall. See “CCO Agent Reservation Feature” on page 188 .
GCTI_RESET_RESERVATION	Integer	TAgentSetNotReady	Allows you to cancel reservation of an agent. You can define the reason for canceling the reservation in Extensions key GCTI_CC_TREATMENT_TYPE. The extension is only treated for requests with workmode WorkingAfterCall. See “CCO Agent Reservation Feature” on page 188 .
GCTI_NAT_INDICATION	String	EventRinging and all events related to the call and generated afterwards	Reports the National-International indication information from the switch.
GCTI_NAT_INDICATIONTYPE	Integer	EventRinging and all events related to the call and generated afterwards	Reports the type of the National-International indication number in the existing Extensions key GCTI_NAT_INDICATION.
GCTI_PARTY_NAME	String	Any call-related event	Party name of a calling DN that is reported in corresponding PBX private data. Appears only if the monitored party makes a call to an internal destination.
GCTI_CSTA_ACCOUNT_INFO	String	TPrivateEvent	Reports transaction code information if entered by agents at the end of business calls.

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
GCTI_BUSINESS_CALL	Integer	Any call-related event	Indicates that T-Server considers the call a business call and will apply business-call handling (if configured) after the call. This Extensions key is always present in outbound calls. Value 0 indicates a private call, and value 1 indicates a business call.
GCTI_SUB_THIS_DN	String	Any call-related event	T-Server provides this Extensions key if it has substituted the ThisDN value. See “Agent Substitution” on page 159 .
GCTI_SUB_OTHER_DN	String	Any call-related event	T-Server provides this Extensions key if it has substituted the OtherDN value. See “Agent Substitution” on page 159 .
GCTI_SUB_THIRD_DN	String	Any call-related event	T-Server provides this Extensions key if it has substituted the ThirdPartyDN value. See “Agent Substitution” on page 159 .
GCTI_PASSWORD	String	TAgentLogout	Enables a client to successfully issue a TRequestAgentLogout for an agent who is required to provide a password when T-Server has not been able to store the password provided during the login.

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
NO_ANSWER_TIMEOUT	String	TRouteCall	<p>If set, the value of this <code>Extensions</code> key overrides any value set in any of the following configuration options for the current call:</p> <ul style="list-style-type: none"> <code>no-answer-timeout</code> <code>agent-no-answer-timeout</code> <code>extn-no-answer-timeout</code> <code>posn-no-answer-timeout</code>
NO_ANSWER_ACTION	String	TRouteCall	<p>If set, the value of this <code>Extensions</code> key overrides any value set in any of the following configuration options for the current call:</p> <ul style="list-style-type: none"> <code>no-answer-action</code> <code>agent-no-answer-action</code>
NO_ANSWER_OVERFLOW	Comma-separated list	TRouteCall	<p>If set, the value of this <code>Extensions</code> key overrides any value set in any of the following configuration options for the current call:</p> <ul style="list-style-type: none"> <code>no-answer-overflow</code> <code>agent-no-answer-overflow</code> <code>extn-no-answer-overflow</code> <code>posn-no-answer-overflow</code>

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
ReasonCode	String	EventAddressInfo EventAgentNotReady	In TAgentNotReady and EventAgentNotReady, specifies an application-specific withdrawal type (values 1-9 as defined in Processing Group configuration). The Extensions attribute has the same value as in the Extensions key GCTI_NOT_READY_ACTIVATION, but is only taken into account if GCTI_NOT_READY_ACTIVATION is not specified in requests. ReasonCode cannot be filtered out.
	String/ integer	TAgentNotReady	
dnd	Integer	EventAddressInfo	Provides the status of the DND state on the device. Always returns value -1, because DND is not supported.
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
NumberOfOrigDNs	Integer	EventPartyChanged	Enables retrieval of the first conference party. EventPartyChanged is sent with additional extensions containing a list of all conference parties.
OrigDN-1 OrigDN-N	String		
ConvertOtherDN	Integer or string	See “Hot-Standby HA Synchronization” on page 201	0—Disables all conversions for the call. 1—Forces the relevant conversion for the call.

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
SUPERVISED_ROUTE	String	TRouteCall	<p>Enables overriding of default supervised-route-timeout option on a call-by-call basis.</p> <p>If the value is set to 0 (zero) and Extensions key GCTI_SUPERVISED_ROUTE has a nonzero value, T-Server performs routing via Transfer.</p>
CPNDigits	String	TMakeCall TMakePredictiveCall TRouteCall (only RSI)	<p>Specifies digits according to the numbering/dialing plan format that will be passed as a call line identity.</p> <p>The extension will be ignored if the value is empty or absent.</p> <p>Extensions key GCTI_REQUESTING_DEVICE is ignored in TMakeCall if the current Extensions attribute is specified (incompatible in PBX).</p>
CPNType	Integer or string	TMakeCall TMakePredictiveCall TRouteCall (only RSI)	<p>Specifies the type of number specified in the Extensions key CPNDigits. The following values are accepted by PBX (otherwise value 0 is assumed):</p> <ul style="list-style-type: none"> 0—Unknown 1—International number 2—National number 3—Network-specific number 4—Subscriber number 6—Abbreviated number 7—Reserved for extension <p>The extension will be ignored if the extension specified by key CPNDigits is empty or absent.</p>

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
CPNP L an	Integer or string	TMakeCall TMakePredictiveCall TRouteCall (only RSI)	Specifies the numbering plan identification for digits specified in the Extensions key CPND igits. The following values are accepted by the PBX (otherwise value 0 is assumed): 0—Unknown 1—ISDN/telephony numbering plan 3—Data numbering plan 4—Telex numbering plan 8—National standard numbering plan 9—Private numbering plan 15—Reserved for extension The extension is ignored if the Extensions specified by key CPND igits is empty or absent.
CPNPresentation	Integer or string	TMakeCall TMakePredictiveCall TRouteCall (only RSI)	Specifies a presentation indicator for digits specified in Extensions key CPND igits. The following values are accepted by the PBX (otherwise value 0 is assumed): 0—Presentation allowed 1—Presentation restricted 2—Number not available due to inter working 3—Reserved for extension The extension is ignored if the Extensions specified by key CPND igits is empty or absent.

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
CPNScreening	Integer or string	TMakeCall TMakePredictiveCall TRouteCall (only RSI)	Specifies a screening indicator for digits specified in Extensions key CPNDigits. The following values are accepted by the PBX (otherwise value 0 is assumed): 0—User-provided, not screened 1—User-provided, verified and passed 2—User-provided, verified and failed 3—Network provided The extension will be ignored if the extension specified by key CPNDigits is empty or absent.
SwitchSpecificType	String	TRegisterAddress	Defines the switch-specific type of a DN that is not configured in Configuration Manager.
EmulateLogin	String	TAgentLogin	With value yes, T-Server performs an emulated login. With value no, T-Server attempt 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 login's agent session. It can be overridden by the value in the WrapUpTime extension in TAgentNotReady.

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
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.
BusinessCallType	string or integer	TMakeCall TInitiateTransfer TMuteTransfer TInitateConference TMakePredictiveCall TAnswerCall TRouteCall	<p>Specifies the call business type to be used by T-Server for the new call or the answering party. Valid values are:</p> <p>0/private—Private call 1/business—Business call 2/work—Work-related call</p> <p>Once a call type is assigned, it is only possible to promote it to a business type. It is not possible to change between private and work calls. Therefore, the corresponding flag in the bsns-call-dev-types option has to be disabled on a Routing Point or an External Routing Point for this extension to be use with <code>RequestRouteCall</code>; otherwise the call will already have been classified as business without the possibility to revert it back to private.</p>

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
AgentLogoutOnUnregister	string	TAgentLogin TRegisterAddress	Specifies whether the T-Server performs an automatic logout of an agent whenever their client application unregisters its DN from the T-Server. true—the T-Server will logout emulated and native agents on unregister false—the T-Server will not logout emulated or native agents on unregister emu-only—the T-Server will logout emulated agents only on unregister.
AssociateClientWithLogin	boolean	TAgentLogin TRegisterAddress	Specifies whether the client should be associated with the agent session.
AssociateClientWithLogin	boolean	EventAgentLogin EventRegistered EventPrivateInfo	Specifies that the client has been associated with the agent session.
AgentEmuLoginOnCall	boolean	TAgentLogin TAgentLogout	Specifies whether the T-Server allows an emulated agent login or logout from a device where there is a call in progress.
LegalGuardTime	integer	TAgentLogin	Specifies the amount of emulated legal guard time allocated to an agent at the end of a business call.
SyncEmuACW	integer	TAgentLogin	Specifies whether the T-Server synchronizes emulated ACW and/or legal guard with the switch for native agents.

Table 25: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	Type		
ReleasingParty	string	EventReleased EventAbandoned	Identifies which party was the initiator of the call release. Possible values are: 1—Local 2—Remote 3—Unknown
LinkLoad	string	EventRouteRequest	Current CTI link bandwidth usage as a percentage of <code>use-link-bandwidth</code> . The feature is disabled if <code>use-link-bandwidth</code> set to zero. Possible values are: 0—Ok 1—High
PBXTimeStamps	integer	All DN-based events	Specifies the UTC time (in seconds) provided in a PBX event for the corresponding event generated by T-Server.
CallerID	string	All events related to the call	The name of the Calling Party provided by the switch either in private data of event <code>Delivered</code> or in private event <code>Calling Party Name</code>
CallPickup	string	EventReleased	Specifies the dialing number of a device that performed a pickup scenario on a ringing call or puts unknown if the information is not provided by the PBX.
T-Server Common Part Extensions			
<code>sdn-licenses-in-use</code>	integer	EventServerInfo	Specifies how many SDN licenses are currently in use.
<code>sdn-licenses-available</code>	integer	EventServerInfo	Specifies how many SDN licenses are currently available.

Extension Filtering

In some instances you will not need some of the extensions that T-Server provides. In this case you can block the sending of such extensions. “Ext-Filter Section” on [page 407](#).

User Data Keys

[Table 26](#) provides details of the User Data keys defined by option accode-name.

Table 26: User Data Keys

Key	Type	Used In	Description
GCTI_CSTA_CORRELATOR	Binary or String	EventAttachedDataChanged TAttachUserData TUpdateUserData TInitiateConference TInitiateTransfer TMakeCall TMakePredictiveCall TMuteTransfer TSingleStepConference TSingleStepTransfer	Provides mapping to CTSA correlator data.
GCTI_CSTA_ACCOUNT_INFO	String	EventAttachedDataChanged TAttachUserData TUpdateUserData TInitiateConference TInitiateTransfer TMakeCall TMuteTransfer TSingleStepConference TSingleStepTransfer	Provides mapping to CTSA Account Information data.
GCTI_SKILL_NUMBER_ (1–10)	Integer	AssociateData TInitiateConference TInitiateTransfer TMakeCall TMakePredictiveCall TMuteTransfer TQueryAddress TSingleStepTransfer	Defines the skill number in the ACR list.

Table 26: User Data Keys (Continued)

Key	Type	Used In	Description
GCTI_ACR_STATUS_(1-10)	Integer	AssociateData TInitiateConference TInitiateTransfer TMakeCall TMakePredictiveCall TMuteTransfer TQueryAddress TSingleStepTransfer	Defines the parameters of the ACR list. This Extensions key is ignored if you have not defined the corresponding GCTI_SKILL_NUMBER_(1-10) parameter.
GCTI_EXPERT_EVALUATION_LEVEL_(1-10)	Integer	AssociateData TInitiateConference TInitiateTransfer TMakeCall TMakePredictiveCall TMuteTransfer TQueryAddress TSingleStepTransfer	Defines the parameters of the ACR list. This Extensions key is ignored if you have not defined the corresponding GCTI_SKILL_NUMBER_(1-10) parameter.
GCTI_MAGIC_ID	String	EventAttachedDataChanged TAttachUserData TUpdateUserData TInitiateConference TInitiateTransfer TMakeCall TMuteTransfer TSingleStepConference TSingleStepTransfer TMakePredictiveCall	If specified, this allows use of Magic ID in correlator data context to specify a special meaning for correlator; for example, it can be used to display information on the agent phone set. The value of this Extensions key should be the ASCII (string) representation of the hex Magic ID. So, if the Magic ID is C015 in hex, then T-Server accepts GCTI_MAGIC_ID='C015'. Note: This key is taken into account only if you have also specified GCTI_CSTA_CORRELATOR.

Reasons Keys

[Table 27](#) provides details of the Reasons keys.

Table 27: Reasons Keys

Key	Type	Used In	Description
GCTI_NOT_READY_ACTIVATION	Integer/string	TAgentNotReady EventAgentNotReady	Specifies an application-specific withdrawal type (values 1-9 as defined in Processing Group configuration). Activated if the corresponding Extensions key ReasonCode is not set in extensions.
ReasonCode	String in events Integer/string in requests	TAgentNotReady EventAgentNotReady	Used to specify an application specific withdrawal type (values 1-9 as defined in PG configuration). The Extensions key has the same value as in reason GCTI_NOT_READY_ACTIVATION. This reason key is checked only if reason GCTI_NOT_READY_ACTIVATION is not specified in requests. ReasonCode cannot be filtered out.

T-Server Error Messages

[Table 28](#) presents the complete set of error messages T-Server distributes in EventError.

Table 28: T-Server Error Messages

T-Library Error Code	Description
T-Server-Defined Errors	
40	No additional licenses
41	Client has not registered for DN

Table 28: T-Server Error Messages (Continued)

T-Library Error Code	Description
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
123	DN for association does not exist

Table 28: T-Server Error Messages (Continued)

T-Library Error Code	Description
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
223	Invalid parameter passed to function

Table 28: T-Server Error Messages (Continued)

T-Library Error Code	Description
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
711	Invalid retrieve request

Table 28: T-Server Error Messages (Continued)

T-Library Error Code	Description
712	Cannot find route point in call
714	Invalid Call_ID
717	Agent not logged in
742	Invalid DN
749	Agent already logged in
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
1017	Invalid transaction type
1018	Invalid or missing transaction data

Table 28: T-Server Error Messages (Continued)

T-Library Error Code	Description
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
1144	Invalid calling device
1145	Invalid called device

Table 28: T-Server Error Messages (Continued)

T-Library Error Code	Description
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
System Resource–Availability Errors	
1170	Generic

Table 28: T-Server Error Messages (Continued)

T-Library Error Code	Description
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
1178	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
Switch-Routing Errors	
1195	Routing timer or delay ringback timer expired
1196	Caller abandoned call

Table 28: T-Server Error Messages (Continued)

T-Library Error Code	Description
1197	Call successfully routed
1198	Aborted because of RouteSelect resource problem
Network Attended Transfer/Conference Errors	
1901	Unexpected request TNetworkConsult
1902	Unexpected request TNetworkAlternate
1903	Unexpected request TNetworkReconnect
1904	Unexpected request TNetworkTransfer
1905	Unexpected request for TNetworkMerge
1906	Unexpected request TNetworkSingleStepTransfer
1907	Unexpected request TNetworkPrivateService
1908	Unexpected message

8

Supported High Availability (HA) Configurations

This chapter presents switch-specific high availability (HA) information for the T-Server for Alcatel A4400/OXE. It contains the following sections:

- [Genesys HA Configuration, page 263](#)
- [HA and PBX Licensing Issues, page 265](#)

Genesys HA Configuration

High-availability configuration for T-Server for the A4400/OXE PBX differs from the standard described in Part 1 of this document, in that both the primary and backup T-Servers connect directly to the PBX instead of to a separate CTI-link application.

The diagrams in the following two sections show this.

Hot Standby Mode

[Figure 18](#) shows the architecture of hot standby HA mode.

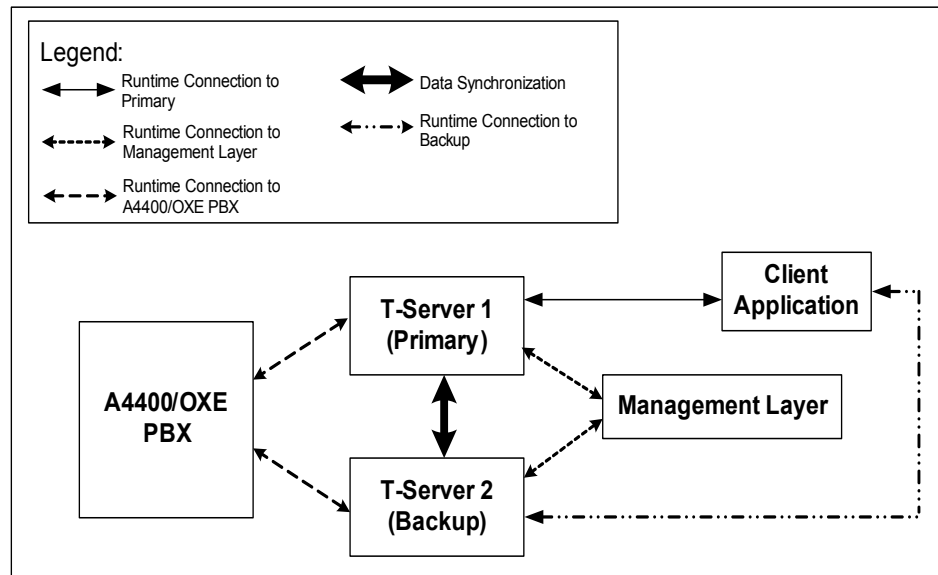


Figure 18: A4400/OXE Hot-Standby Mode Architecture

Warm Standby Mode

Figure 19 shows the architecture of warm standby HA mode.

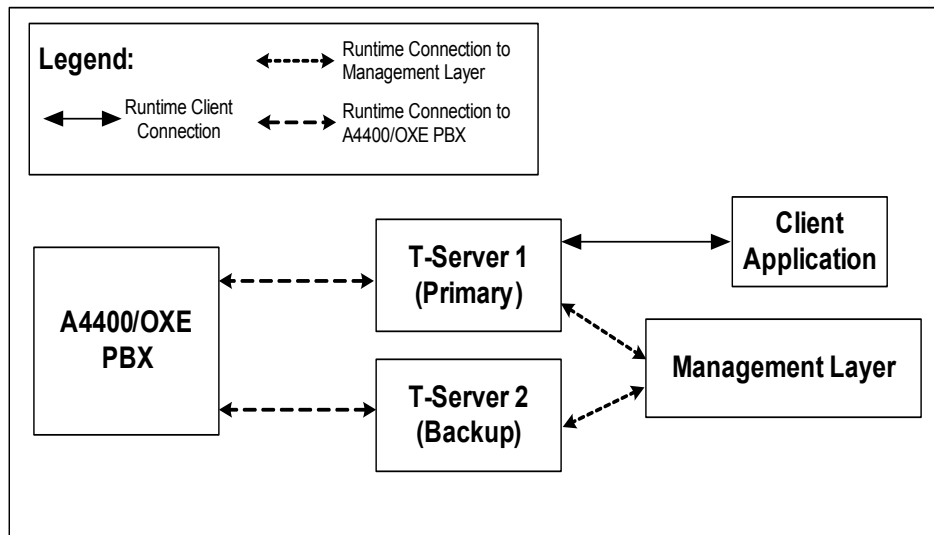


Figure 19: A4400/OXE Warm-Standby Mode Architecture

HA and PBX Licensing Issues

When running in Backup mode, the Alcatel A4400 T-Server must still start the monitoring of all devices in the PBX in order to receive call information. This process enables the backup T-Server to be activated with minimum delay and with no loss of call information. This means that in an HA environment, where two T-Servers are connected to the PBX (one active and one hot standby), the number of monitoring requests on the PBX doubles. Therefore, the required number of CSTA licenses on the PBX also doubles.

Configuration

Backup T-Server Configuration Options

To achieve the most predictable behavior, set the value of the T-Server configuration options for the backup T-Server to the same values as those configured for the primary T-Server.

If you fail to do this and a switchover activates the backup T-Server, clients may experience different T-Server behavior, even though full T-Server functionality is maintained.

Backup T-Server Switch Association

To activate the high-availability feature, add an Application object in Configuration Layer for the backup T-Server. Unlike the primary T-Server Application object, you need not associate the backup T-Server Application object with a switch on the *Switches* tab. It automatically uses the same DN information as the primary T-Server.

T-Server Start Information

On the *Start Info* tab for both the primary and the backup T-Server, set the *Redundancy Type* as *Hot Standby*.

Also, by selecting the *Auto-Restart* check box, you can define whether the Local Control Agent (LCA) application is allowed to automatically restart the T-Server application in the event of an application fault. You should normally select this check box.

Backup T-Server Connection

When the configuration of the backup T-Server has been completed, you must associate the backup application with the primary T-Server. Do this by setting the *Backup Server* field in the primary T-Server Application object as shown in [Figure 20](#).

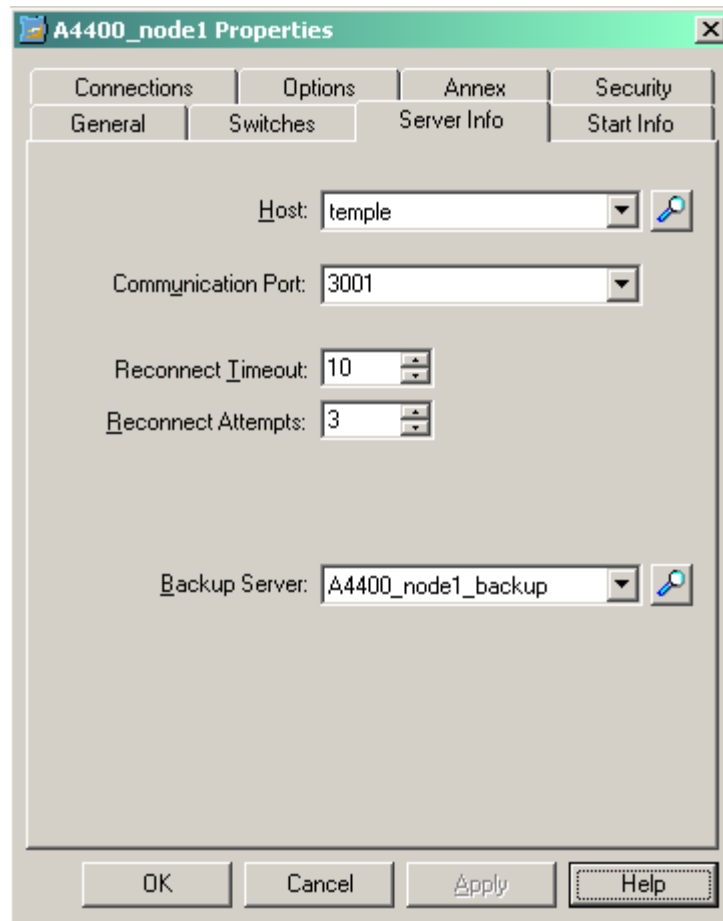


Figure 20: Setting the Backup Server Connection

Environment Startup

When the environment starts up, Solution Control Server (SCS) activates as the primary T-Server the T-Server that is started first. This means the T-Server configured as the primary in the Configuration Layer. However, since the client registration procedure includes information about backup and primary T-Server status, this difference is transparent to clients using a compatible version of the T-Server Common Library.

9

Using Alcatel A4400 Routing Services Interface

This chapter describes how to use RSI. It contains these sections:

- [RSI Description, page 267](#)
- [Configuring RSI in the Alcatel A4400, page 268](#)
- [RSI Agent Functionality, page 270](#)
- [Call Routing, page 270](#)
- [Treatments, page 274](#)
- [RSI Reroute Behavior, page 284](#)
- [Voice Guides in the Alcatel A4400 PBX, page 285](#)
- [Private Data in Route Requests on RSI, page 287](#)

RSI Description

Unlike emulated routing using Hunting Groups, the Alcatel A4400 R4.2+ provides Routing Services Interface (RSI) functionality that delivers native routing capabilities on the PBX. This functionality, specifically designed for integration with the Genesys T-Server, provides many features, including:

- Native PBX routing support.
- Ability to define an overflow destination via RSI in the PBX.
- Ability to route to any destination.
- Full call-treatment support including (but not limited to) playing announcements and collecting digits.
- Full Genesys high-availability support.
- Unlimited number of queued calls.
- Business agent functionality on the PBX without using the CCD.

This means you can create a full routing implementation that takes advantage of the call-routing features available in Genesys Universal Routing Server, as well as of the Alcatel A4400 CCD agent features.

RSI and CSTA Interfaces

T-Server provides full support for both the RSI and the CSTA link. The difference between the RSI link and the CSTA link is only in the PBX licensing model. Depending on which model is chosen, the PBX RSI routing functionality is either available or not. Other than this, there is no difference between RSI and CSTA.

Configuring RSI in the Alcatel A4400

Procedure: Creating a new RSI in the PBX

Start of procedure

1. Open the mgr application and locate mgr/Application/CCD/RSI.
2. Set the following values:
 - **Time Between Two Calls**—This option defines a pause time (in milliseconds) for agents after they hang up a call that was routed to them from this RSI.
 - **Auto WrapUp Timer**—This option defines a wrap-up time (in milliseconds) for agents after they hang up a call that was routed to them from this RSI.
 - **RSI Overflow Timer**—This option defines a time (in milliseconds) that the PBX waits for a call to be routed. If the time expires, the call is overflowed to the address specified in option **RSI Overflow Address**.
 - **RSI Overflow Address**—This option specifies an overflow address (any valid address) to which the PBX sends a call if:
 - Option **RSI Overflow Timer** expires.
 - T-Server did not enable routing on this RSI.

Note: If you do not define an overflow address, the call is canceled instead of overflowed.

- **Ringing Overflow Timeout**—This option defines a time interval (in milliseconds) during which the PBX waits for a call ringing on a CCD/RSI agent to be answered. If the agent fails to answer the call within this timeout, T-Server overflows the call to the address specified in option **Ringing Overflow Address**.
- **Ringing Overflow Address**—This option specifies an overflow address (any valid address) to which the PBX sends a call if the timeout defined in option **Ringing Overflow Timeout** expires.
- **Local Call Authorization**—This option allows or disallows local (hybrid-link) or ABC calls to this RSI.
- **RSI Supervised Transfer**—With value `true` in the RSI configuration, the PBX activates the routing dialog for consultation calls and allows consultation calls to be routed. With value `false`, the PBX does not activate the routing dialog until transfer is completed.

With value `true`, special attention should be paid to T-Server option `rsi-report-xfer`.

3. If the call is to be routed from the RSI to other A4400 PBXs, then an entry for the RSI on one PBX must be created on the other PBX(s). The exact same configuration is used for the other PBX(s), with the exception that the Node Number must be set to the PBX node where the RSI was first created. For example—if RSI 31000 is created on PBX node 1, then it is also configured on PBX node 2, but on PBX node 2, the Node Number is set to 1.

Note: This configuration is performed automatically if Broadcast is enabled on all of the PBXs.

End of procedure

Note: Only a qualified Alcatel engineer should make changes to these options.

Procedure: Configuring RSI in the Configuration Layer

Start of procedure

1. Navigate to CME\Resources\Switches\<Switch Name>\DNs\New DN
2. Set the following values:
 - **Number**—Must correspond to the RSI number configured in the PBX.
 - **Type**—Routing Point
 - **Enabled**—Checked

- Register—Checked
- Switch-Specific Type—2 (on Advanced tab of DN Properties)

End of procedure

RSI Agent Functionality

The PBX provides the same features and business-call handling for RSI agents as for normal CCD agents, including logon, logoff, wrap-up, and pause. An *RSI agent* is defined as an agent who is logged in to an RSI Processing Group (as distinct from any other type of Processing Group).

Like the CCD pilots, wrap-up and pause timer are configured per RSI and are applied by the PBX according to the RSI from which a call is routed.

Note: T-Server also supports emulated agents with RSI routing. This means that no agent is logged in on the PBX and T-Server maintains full control of agent states.

Call Routing

Genesys Universal Routing Server (URS) performs call routing on an RSI. Calls can be routed to any valid destination, including external ones.

Treatment Support

The RSI feature in T-Server for Alcatel A4400 fully supports treatments, including those listed below, which are described in subsequent sections:

- “Collect Digits Treatment” on [page 274](#)
- “Play Announcement Treatment” on [page 276](#)
- “Play Announcement and Collect Digits Treatment” on [page 281](#)
- “Music Treatment” on [page 281](#)
- “Busy Treatment” on [page 282](#)
- “Ringback Treatment” on [page 282](#)
- “Silence Treatment” on [page 282](#)
- “Cancel Call Treatment” on [page 282](#)

Mandatory Treatments

A mandatory treatment is one that the routing strategy applies to a call *before* selecting a routing destination. This treatment type is typically used, for

example, to play a message to all callers or to get the caller to enter some information (collect digits), which can then be used in the routing strategy.

To make a mandatory treatment, use the individual Treatment objects in Interaction Routing Designer (IRD). [Figure 21](#) shows the IRD treatment toolbar:



Figure 21: IRD Treatment Toolbar

With many treatments, you can choose to Wait for Treatment end (see [Figure 22](#)). If you select this option, the strategy does not proceed until the treatment finishes.

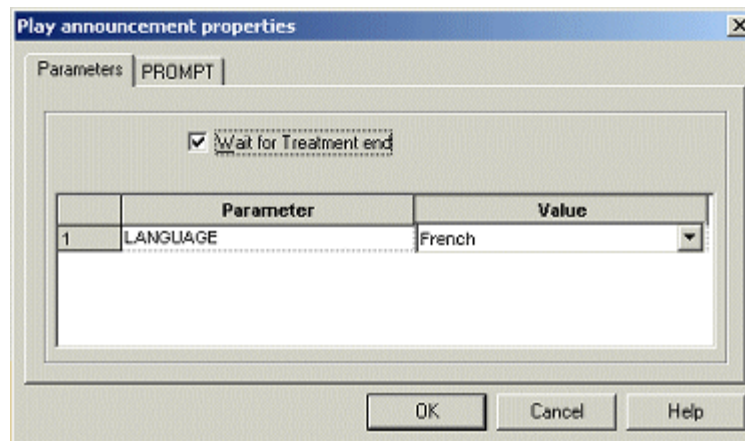


Figure 22: Wait for Treatment End Dialog Box

If this option is not selected, you can apply a treatment and then continue in the strategy; for example, play some music to the caller while a database lookup is performed. You must then re-synchronize the strategy with the treatment. You will use the `SuspendForTreatmentEnd` function to do this (see Figure 23 on [page 272](#)), which allows you to specify an optional, additional timeout.

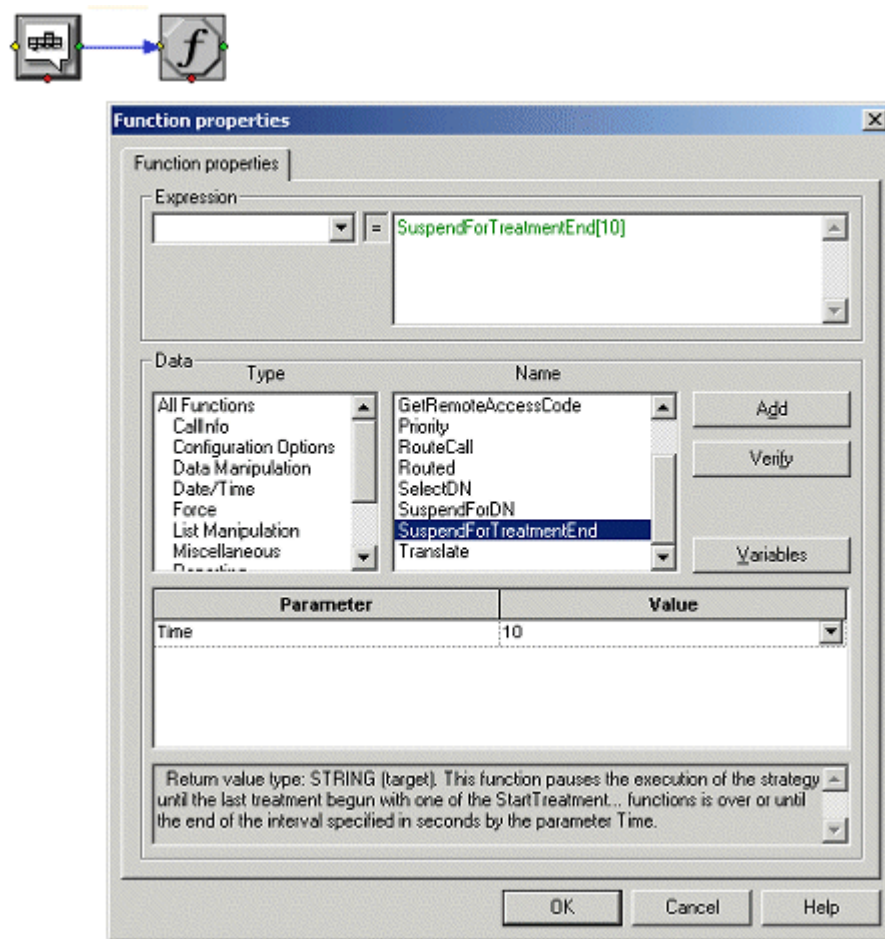


Figure 23: Suspend for Treatment End Function

In the treatment example above, the customer hears music for up to 10 seconds, after which the strategy continues. (The strategy resumes even sooner if the PBX signals that the treatment is finished before this timeout expires.)

Busy Treatments

A busy treatment is one applied only in cases where the selected routing target is not currently available. This allows the strategy to use any treatment while a call is waiting.

Busy treatments are activated in the Target Selection object in IRD as shown in the example in [Figure 24](#).

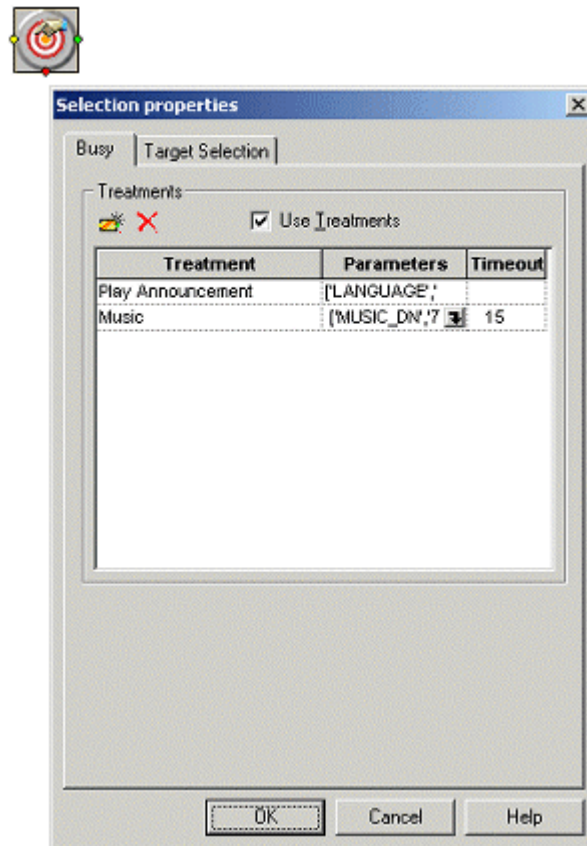


Figure 24: Busy Treatments in Target Selection Object

You can provide a list of treatments (only two are shown above) that are to be played in sequence.

You can specify a timeout (in seconds) for each treatment in the list. If you do that, the strategy applies the next treatment in the list without waiting for the end of the previous treatment when the specified time has elapsed.

Note: URS, not the PBX, manages this timeout.

Treatments

The following sections describe the different treatments the PBX and T-Server support and how they are used within IRD.

Note: The Alcatel A4400 T-Server does not support treatments in Compatibility mode, which means TGiveTreatment requests are not supported.

The following treatment types are not supported:

- Delete User Announcement
- Fast Busy
- Recorded User Announcement (RAN)
- Play Recorded Announcement
- Play Application
- Set Default Destination
- Text to Speech
- Text to Speech and Collect Digits
- Verify Digits

Collect Digits Treatment



This treatment provides functionality to collect dual-tone multi frequency (DTMF) tone digits from the user. Normally, Collect Digits is combined with an announcement telling the caller what information is required (see also “Play Announcement and Collect Digits Treatment” on [page 281](#)).

[Table 29](#) shows the IRD parameters for the Collect Digits treatment.

Table 29: Collect Digits Treatment Options

Option Name	Option Description
MAX_DIGITS	Specifies the maximum number of digits to be collected (max 20).
ABORT_DIGITS	Specifies a sequence of maximum two digits (though in practice, the switch only uses the first one). If this sequence of digits is detected, digit collection is aborted.
IGNORE_DIGITS	Specifies a sequence of maximum two digits (though in practice, the switch only uses the first one). If this sequence of digits is detected, digit collection ignores them.

Table 29: Collect Digits Treatment Options (Continued)

Option Name	Option Description
BACKSPACE_DIGITS	Specifies a sequence of maximum two digits (though in practice, the switch only uses the first one). If this sequence of digits is detected, the last correctly collected digit is deleted.
TERM_DIGITS	Specifies a sequence of maximum two digits (though in practice, the switch only uses the first one). If this sequence of digits is detected, digit collection terminates and the digits already collected are reported.
RESET_DIGITS	Specifies a sequence of maximum two digits (though in practice, the switch only uses the first one). If this sequence of digits is detected, any collected digits are discarded and collection restarts.
CLEAR_DIGITS	This option is currently not used.
START_TIMEOUT	Specifies the maximum duration (in seconds) before the caller enters the first digit.
DIGIT_TIMEOUT	Specifies the maximum interval (in seconds) between two digits.
TOTAL_TIMEOUT	Specifies the maximum duration (in seconds) of the entire digit-collection process.

Note: If a routing target becomes available during digit collection, URS waits until digit collection has completed before routing the call.

When the digits have been collected, use the Caller-Entered Digits (CED) function to assign them to a variable in the strategy, as shown in Figure 25.

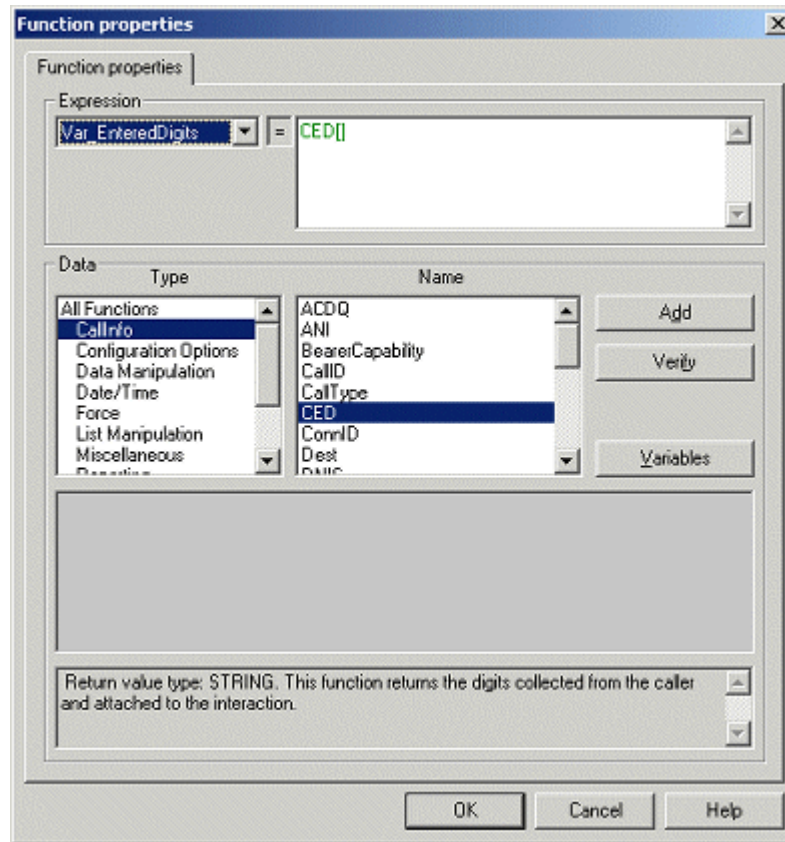


Figure 25: Caller-Entered Digits Function

Play Announcement Treatment



Use this treatment to play one or more (default maximum 10, configurable to 20 using option “[convert-othern](#)”) voice guides to the caller. The PBX performs the connection to the voice guide. IRD has two tabs in the Play Announcement Treatment object:

- Parameters
- PROMPT

Parameters Tab

You must specify a language on the Parameters tab, as shown in [Figure 26](#).

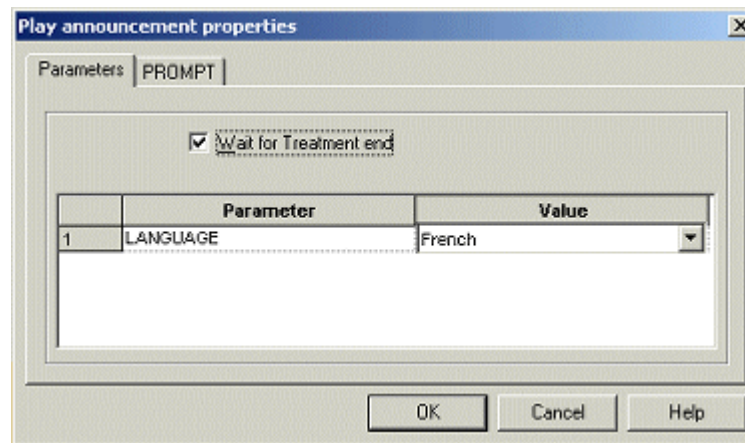


Figure 26: Setting Language in Play Announcement Treatment

The language must correspond to the correct language on the PBX and is defined in one of several ways. See “Treatment Languages” on [page 278](#) for further information on how to define treatment languages.

PROMPT Tab

Use the PROMPT tab (see [Figure 27](#)) to define a list of announcements.

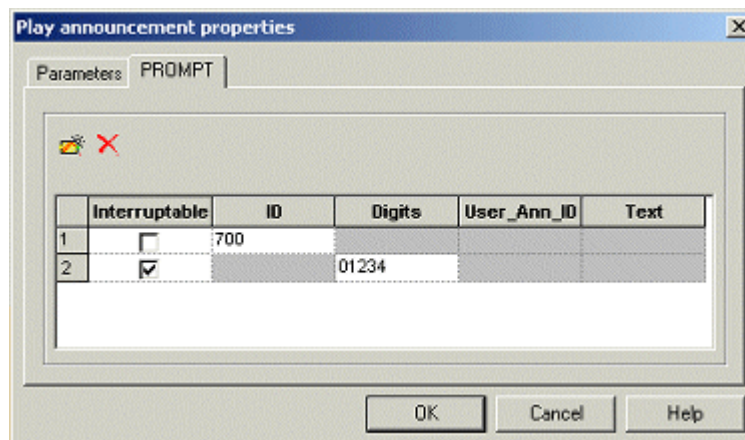


Figure 27: Defining Prompts in Play Announcement Treatment

The prompts you define here are played in succession. Table 30 on [page 278](#) describes each option in this treatment.

Table 30: Play Announcement Treatment Options

Option Name	Option Description
Interruptible	Specifies if this prompt can be interrupted. If the <code>Interruptible</code> check box is selected, the PBX will allow the announcement to be skipped by pressing a key. If it is not set, then the PBX will always play the entire treatment.
ID	Specifies an ID corresponding to the voice guide to be played for this prompt. You must configure this ID as a Voice Prompt object in Configuration Layer (CME\Resources\Voice Prompts) before you can use it in the strategy.
Digits	Specifies a string of digits to be announced to the caller. The first digit specifies how the subsequent digits are to be pronounced: 0–One at a time 1–Date 2–Time 3–Phone number 4–Money Currently, the PBX only supports value 0 (one at a time).
User_Ann_ID	This option is currently not used.
Text	This option is currently not used.

Note: You can specify either the ID field or the Digits field in each prompt, but not both.

Treatment Languages

You can use either of two methods to define the language of an announcement in the Play Announcement Treatment object:

- **Language Mapping**
Enter any string in the language field, and map this string to the required PBX language ID in T-Server.
- **PBX language ID**
Specify the PBX language ID directly in the strategy.

Language Mapping

To configure language mapping in T-Server, you must create a new section, lang-map, in the T-Server Application object in Configuration Manager. The following table shows how to define a mapping for any supported PBX language.

Table 31: T-Server Language-Mapping Options

Option Name	Option Description
lang-def	Specifies the default language (values 1-55) T-Server is to use if language mapping fails.
lang- <i><n></i> <i><n></i> = 1-55	Specifies the string to be mapped to language number <i><n></i> . If T-Server receives a Play Announcement Treatment request with this string, it uses language <i><n></i> .

The following screen shot shows the default language mapping provided in T-Server. If you do not explicitly configure language mapping, T-Server uses this mapping.

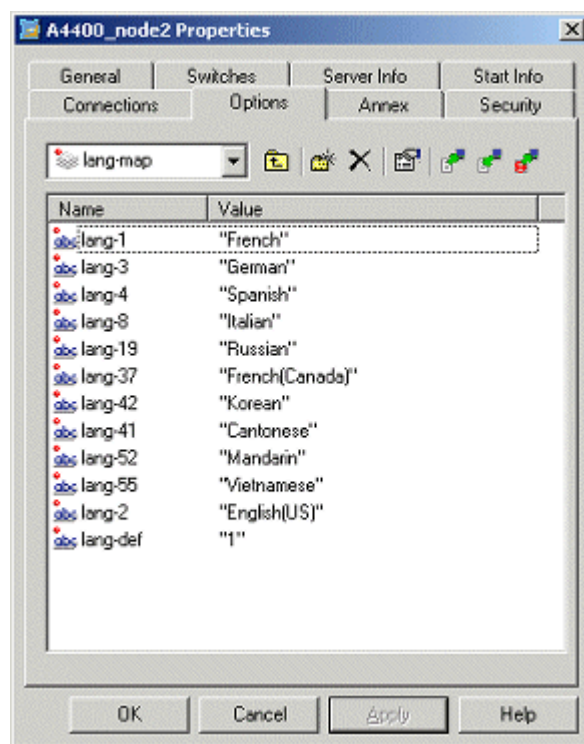


Figure 28: Default Language Mapping Dialog

Language ID

Use the PBX language ID directly in the Apply Treatment—this bypasses the language mapping in T-Server. The complete list of languages, as defined in the PBX at the time of writing, is shown in [Table 32](#).

Table 32: Language IDs

1. French	29. Slovenian
2. English	30. Greek
3. German	31. Turkish
4. Spanish	32. Yugoslavian
5. Portuguese	33. German (Swiss)
6. Flemish—Belgium	34. Italian (Swiss)
7. Danish	35. Reference
8. Italian	36. English (United States)
9. Austrian	37. French (Canada)
10. Swedish	38. Arabic (Male Voice)
11. Norwegian	39. Arabic (Female Voice)
12. Swiss	40. Bulgarian
13. Czech	41. Cantonese
14. Polish	42. Korean
15. Hungarian	43. Egyptian (Male Voice)
16. Finnish	44. Egyptian (Female Voice)
17. Brazilian	45. English (Africa)
18. Dutch	46. Hebrew
19. Ex-Russian	47. Irish
20. Roumanian	48. Icelandic
21. Slovak	49. Japanese
22. Generic English	50. Malay
23. Catalan	51. Bahasa Malay
24. Australian	52. Mandarin (China)
25. Estonian	53. Mandarin (Taiwan)
26. Lithuanian	54. Spanish (America)
27. Latvian	55. Vietnamese
28. Croatian	

Play Announcement and Collect Digits Treatment



This treatment combines the Play Announcement and the Collect Digits Treatments. In this case, the PBX stops the voice guide as soon as the caller enters the first digit to be collected. See “Collect Digits Treatment” on [page 274](#) and “Play Announcement Treatment” on [page 276](#) for more information.

Music Treatment



Use this treatment in a strategy to connect a call to a music source (PBX voice guide).

Note: Set the URS option `give_treatment` to true if you want the URS to apply a music treatment on all calls before starting any strategy.

[Figure 29](#) shows how to implement a music treatment in IRD. The example connects the caller to voice guide 700.

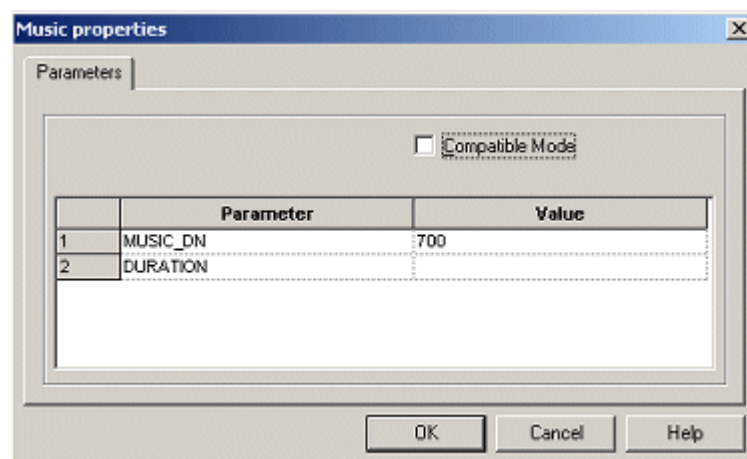


Figure 29: Music Treatment Options

[Table 33](#) describes the options for a music treatment.

Table 33: Music Treatment Options

Option Name	Option Description
MUSIC_DN	Specifies the number of the voice guide in the Alcatel A4400 to be used as the music source.
DURATION	This option is currently not used.

Because you cannot set a duration for a music treatment, you must incorporate a cancel-music treatment or apply another treatment within the strategy to end a music treatment. That requires implementing some form of synchronization; for example, using function `WaitForTreatmentEnd`.

You can use the `timeout` field in the busy treatment list to specify a duration.

Note: `Compatible` mode is not supported.

Busy Treatment



When you incorporate this treatment into a strategy, the caller hears a busy tone. Note that this treatment does not terminate the connection; only the caller can do that.

This treatment has no parameters to specify.

Note: `Compatible` mode is not supported.

Ringback Treatment



When you incorporate this treatment into a strategy, the caller hears a ringback tone. This treatment has no parameters to specify.

Note: `Compatible` mode is not supported.

Silence Treatment



When you incorporate this treatment into a strategy, the caller hears silence. This treatment has no parameters to specify.

Note: `Compatible` mode is not supported.

Cancel Call Treatment



When you incorporate this treatment into a strategy, T-Server ends the routing dialog with the PBX. This means the call is either disconnected or sent to an overflow destination, *if* you have configured one for the RSI in the PBX.

This treatment has no parameters.

IVR Treatment



Use this treatment to send calls to an IVR; for example, Genesys Voice Treatment Option. To set up an IVR that you intend to use with IVR treatments:

1. Configure all the IVR ports in Configuration Layer. For an IVR treatment to work, you *must* configure IVR ports as Voice Treatment Port objects in the Configuration Layer. If you configure the ports with DN type Extension, IVR treatments do not work correctly.
2. Create a Place Group containing all the IVR ports. Use this group as a target in the IVR treatment.

When you incorporate an IVR treatment into a strategy, URS routes the call to an available IVR port for the duration of the treatment. At the end of the treatment, URS transfers the call back to the Routing Point, where the strategy continues.

Figure 30 on [page 283](#) shows how to use IVR treatments in IRD as a mandatory treatment. The example treatment sends calls to Place Group IVR for a duration of 10 seconds. If no duration is specified, it is left to the IVR to transfer the call back to the Routing Point at the end of the treatment.

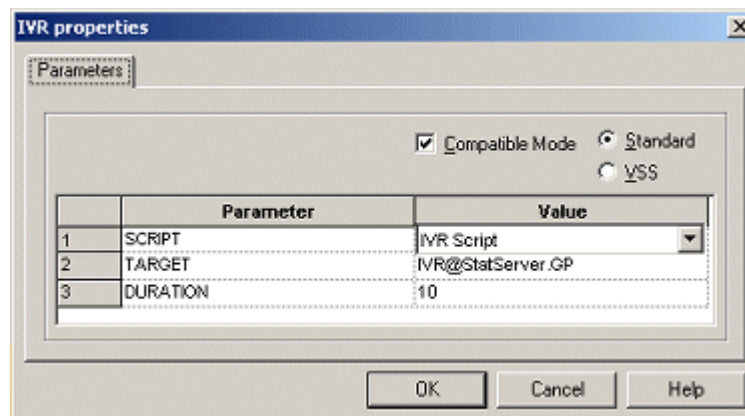


Figure 30: IVR Treatment Options

Note: IVR treatment is the only treatment type where Compatible mode is supported.

If you use an IVR treatment as a busy treatment, keep in mind that:

1. You must specify the treatment duration using the Timeout field in the busy treatment, not as a parameter.

2. You can define how URS sends the call from the IVR port to its destination once a routing target becomes available. Your strategy can direct URS either to transfer the call directly from the IVR port to the destination, or to transfer the call back to the Routing Point before routing the call to the destination. Use URS option `transfer_to_agent` to define this behavior.

RSI Reroute Behavior

If you do not define reroute behavior, the PBX sends the call to the destination defined in the overflow address of the RSI. ([“Creating a new RSI in the PBX” on page 268](#) describes how to configure this overflow.) However, your strategy can specify under what circumstances the PBX should allow a call to be rerouted. Essentially, the strategy must provide the PBX with a new target to select if routing fails.

Feature Configuration

T-Server provides two configuration options to control this behavior:

- `rsi-remain-retry`
- `rsi-reroute-auth`

If rerouting is requested, T-Server returns `EventError` to indicate that routing failed. For this event, the routing strategy should incorporate an appropriate action for T-Server to take; for example, directing T-Server to select a new target. If routing fails after the number of reroute attempts specified in option `rsi-remain-retry`, the call is overflowed to the RSI overflow destination.

RSI Reroute Authorization

[Table 34](#) shows the relationship between PBX, binary and T-Server values for configuration option `rsi-reroute-auth`.

Table 34: RSI Reroute Authorization

Condition	PBX Value (Bit String)	Binary Value	T-Server Option Value
Busy	0	0	0
DestNotObtainable	1	1	1
IncompatibleDest	2	11	3
NetworkCongestion	3	111	7
ResourceNotAvailable	4	1111	15

Table 34: RSI Reroute Authorization (Continued)

Condition	PBX Value (Bit String)	Binary Value	T-Server Option Value
TrunkBusy	5	11111	31
AllCases	255	111111	63

Voice Guides in the Alcatel A4400 PBX

Note: Only a qualified Alcatel engineer should undertake any modifications to your PBX.

Use this section as a quick guide to voice guides (VGs) in the Alcatel A4400 PBX. If you require more detailed information, please refer to the documentation provided with your Alcatel A4400 PBX.

It is important to distinguish between Recordable VGs and VGs. You can create many different Recordable VGs for the PBX. A Recordable VG can then be assigned to a VG and be used with RSI treatments.

These are the hardware prerequisites for creating and using VGs in the Alcatel A4400 PBX:

- A GPA board must be present and configured.
- A RAM card must be installed on the GPA board.

Procedure: Configuring a GPA Board in the Alcatel A4400 PBX

Start of procedure

1. Install the RAM card in the GPA board.
2. insert the GPA board in your PBX, observing the safety guidelines described in your PBX documentation.
3. Once installed, configure the GPA board as follows:
4. Configure the GPA board in the system:
 - a. Locate mgr/Shelf/Go Down Hierarchy/Create.
 - b. Select board type GPA.
5. Configure the GPA board DSP function to be used with VGs:
 - a. Locate mgr/Shelf/Go Down Hierarchy/Go Down Hierarchy/Gpa Dsp program/Create.

- b. Select a DSP function that includes the Voice Guide (VG) function.

End of procedure

Checking Voice Guides

Listening to Voice Guides

You can use a phone set to listen to all active VGs.

Procedure: **Configuring a VG listening prefix**

Start of procedure:

1. Create a Tone Test prefix in the system:
 - a. Locate mgr//Translator/Prefix Plan/Create.
 - b. Select Prefix Meaning = Local Features.
 - c. Select Local Feature = Tone Test.
2. Enable VG listening in the category of the phone set where this feature will be used:
 - a. Locate mgr/Categories/Phone Facilities Categories/Consult Modify.
 - b. In the Set Features section, make sure that option Voice Guide Listening is set to 1.

End of procedure

By dialing the VG Listening prefix, you can listen to all voice guides in the system. You can navigate to the required VG in either of two ways:

- Use the Volume +/- button on the phone set to scroll through the VGs.
- Skip directly to the wanted VG by entering the number of the VG on the phone set. You must enter four digits; so, for example, to listen to VG 705, enter 0705.

Procedure: **Checking GPA Voice Guide status**

Purpose: To check the voice guide status on each GPA board in the system.

Start of procedure

At the command prompt enter:

```
vgstat <crystal (0-19)> <cpl (0-28)>
```

where `crystal` and `cpl` are the name and location, respectively, of the GPA board in question.

End of procedure

The output from this command provides detailed information about the VGs configured on the specified GPA board.

Private Data in Route Requests on RSI

Starting with release R9.0 of the PBX, new private data in `ROUTE_SELECT` allows additional information to be sent to OXE. When the strategy selects a non-agent device, the RSI can receive a `ROUTE_SELECT` service with this private data which can include the following distribution features:

- Call forwarding is bypassed (immediate forwarding, on busy forwarding, no response forwarding, overflow on an associated set).
- Do Not Disturb is bypassed.

This private data is optional, and can only be used in the `ROUTE_SELECT` service.

Feature Configuration

The `rsi-bypass-fwd-dnd` option is used in conjunction with this feature.

-
- Notes:**
- By setting the option `rsi-bypass-fwd-dnd` on the Annex tab of a specific RSI, it is possible to override this value for that RSI.
 - `Extensions` attribute `RSIBypassFwdDND` in `TRouteCall` can be used to override the value of this configuration option on a call by call basis.
-

10

Alcatel A4400 Call Flows

This chapter presents switch-specific call-model information for Alcatel A4400 scenarios for which a greater level of detail may be required than is supplied in the *Genesys Events and Models Reference Manual*. It contains the following sections:

- [Call Queued with Automatic Camp-On, page 289](#)
- [Call Parking/Unparking, page 293](#)

Call Queued with Automatic Camp-On

This section describes T-Server reporting for a call queued with automatic camp-on.

Note: Automatic camp-on may sometimes be referred to as *executive override* in the PBX documentation.

Generally, changes in a client application that conforms to the Genesys Call Model should not be required.

Call Queued with Automatic Camp-On

[Figure 31](#) illustrates a scenario with a call queued and automatic camp-on.

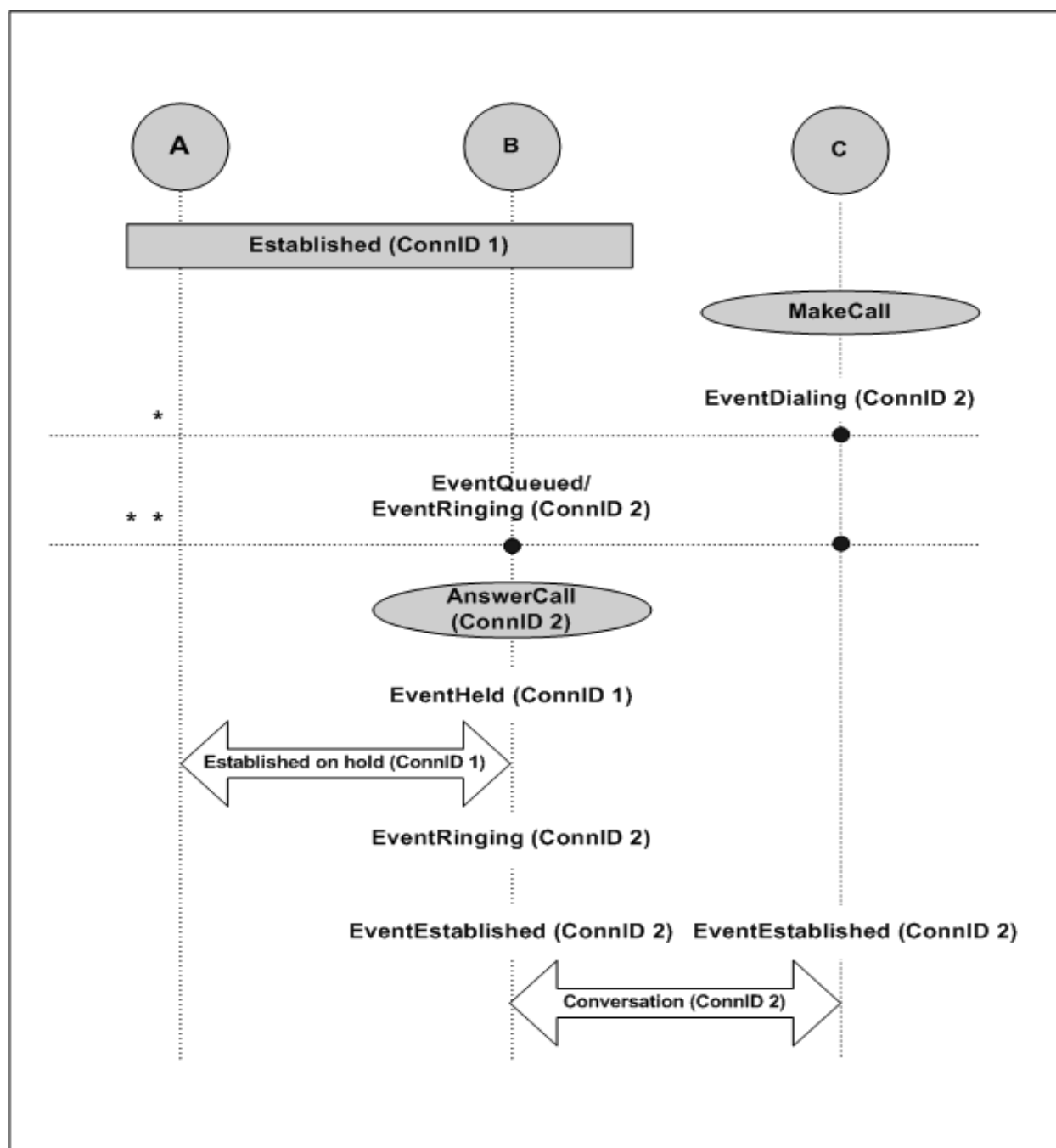


Figure 31: Call Queued With Automatic Camp-On

Table 35 gives detail about the call flow for this scenario for each party.

Table 35: Scenario on Multiline Phoneset or Automatic Camp-On

Party A (Customer)	Party B (Agent)	Party C
Established ConnID 1		
		Make Call to B (TMakeCall)
		EventDialing ConnID 2 ThisDN C ThisDNRole Origination OtherDN B OtherDNRole Destination
	EventQueued ^a ConnID 2 ThisDN B ThisDNRole Destination OtherDN C OtherDNRole Origination CallState OK	
	Answer (Manually or TAnswerCall or TAlternateCall)	
	EventHeld ConnID 1 ThisDN B OtherDN A	

Table 35: Scenario on Multiline Phoneset or Automatic Camp-On (Continued)

Party A (Customer)	Party B (Agent)	Party C
	EventRinging ^b ConnID 2 ThisDN B ThisDNRole Destination OtherDN C OtherDNRole Origination CallState OK	
	EventEstablished ConnID 2 ThisDN B OtherDN C	EventEstablished ConnID 2 ThisDN C OtherDN B

- a. T-Server may generate EventRinging instead of EventQueued depending on PBX reporting, if call 2 can be answered on Agent or T-Server is not sure if it is possible.
- b. If T-Server reports EventRinging (instead of EventQueued) at the start of scenario then no EventRinging is generated after the call (ConnID 2) is answered.

Table 36 shows abnormal call flow in this scenario.

Table 36: Abnormal Call Flow

Interruption Point	Party B (Agent)	Party C
*		EventReleased ConnID 2 ThisDN C ThisDNRole Origination CallState OK
* *	EventAbandoned ConnID 2 ThisDN B ThisDNRole Destination OtherDN C OtherDNRole Origination CallState OK	EventReleased ConnID 2 ThisDN C ThisDNRole Origination OtherDN B OtherDNRole Destination CallState OK

Call Parking/Unparking

Description

T-Server for Alcatel A4400/OXE provides one of two different mechanisms to define behavior and support handling of scenarios involving parking:

- Where a call blocks a device after being parked, or where no other calls can be distributed to such a device, T-Server uses `EventQueued` on the same device to report the state of the call.
- Where a call frees a device after being parked on the same or any unknown (internal to PBX) device, T-Server provides virtual devices that are used to report the state of the call within T-Server. The same handling is used if the PBX provides specific devices for handling call parking. In this case devices are not fictive, but correspond to physical equipment.

These principles conform to the Genesys Call Model (GCM) as described in the *Genesys Events and Models Reference Manual* and the *Voice Platform SDK 8.x. .NET (or Java) API Reference*.

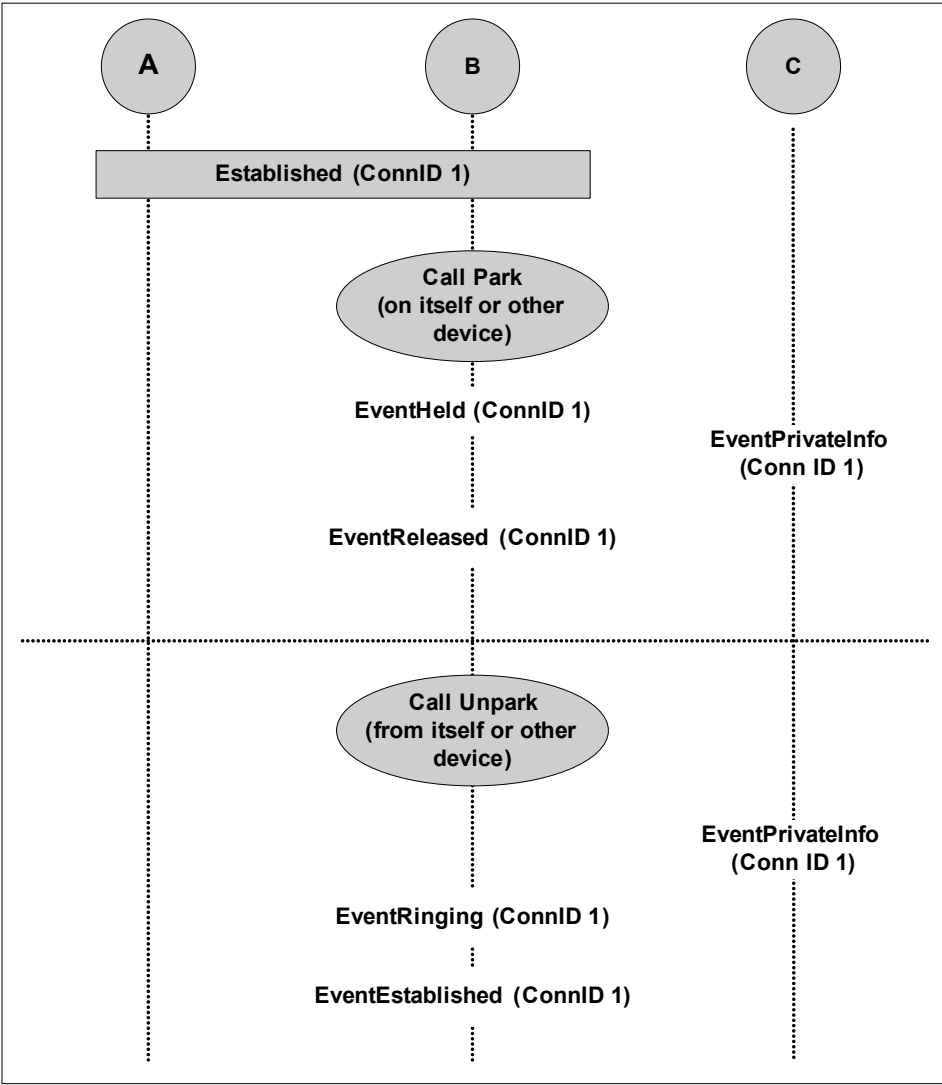


Figure 32: Call Parking/Unparking

Table 37 gives details of this call flow for each party.

Table 37: Call Parking/Unparking

Party A	Party B	Party C
Call Park		
	EventHeld ThisDN DN2 OtherDN DN2 CallState ok	
		EventPrivateInfo AttributePrivateMsgID 2120, ThisDN DN3 OtherDN DN2 CallState transferred
	EventRelease ThisDN DN2 OtherDN DN 1 ThirdPartyDN DN3 CallState ok ThisDNRole destination OtherDNRole origination ThirdPartyDNRole destination	

Table 37: Call Parking/Unparking (Continued)

Party A	Party B	Party C
Retrieve/Unpark Call		
		EventPrivateInfo AttributePrivateMsgID2121 ThisDN DN3 OtherDN DN1 ThirdPartyDN DN2 CallState redirected ThisDNRole destination OtherDNRole origination ThirdPartyDNRole destination
	EventRinging ThisDN DN2 OtherDN DN1 CallState redirected	
	EventEstablished ThisDN DN2 OtherDN DN1 CallState ok	

11

Connecting GVP:EE 6.5.5 to Alcatel A4400/OXE

This chapter describes how to set up and configure Dialogic to be connected to the Alcatel A4400 PBX using the OPS-FX protocol. It includes the following sections:

- [PBX Configuration, page 297](#)
- [Dialogic Configuration for DMV Cards, page 299](#)
- [Dialogic Configuration for JCT-E1 Cards, page 305](#)

PBX Configuration

This section describes board configuration in the PBX.

Interface Settings

The board is actually a PCM2 card which must be configured as an IVR Z30 interface. This is also the interface type that is used to connect the Alcatel IVR (4625 CCIVR). Note that the protocol setting is OPS-FX. There are only two possible values for this setting:

- None
- OPS-FX

With value None, there is no protocol on this trunk at all.

Defining Interface Protocols

It is possible to create the board with no protocol, then later activate the protocol. However, in this case, channels 16–30 will not work. This means that the OPS-FX protocol must be defined when the card is first created in the PBX.

The fields that you must configure are described in [Figure 33](#).

On the PBX, this dialog is located at:

mgr/Shelf/Descend hierarchy/Review_Modify

```
mgr/Shelf/Descend hierarchy/Review_Modify
+-Review/Modify: Board-----+
|                               |
|   Node Number (reserved) : 2   |
|   Shelf Address : 0           |
|   Board Address : 5           |
|                               |
|   Interface Type + IVR 230     |
|   Operational State + Enabled  |
|   Usage State + Busy          |
|   Main/Standby State + Main (Master) |
|   Number Of Sets Connected : 30 |
|   Country Protocol Type + Default |
|   Send Init Dynamic Msg + False |
|   Default Param + True        |
|   Incidents Teleservice + YES  |
|   IVR Protocol + OPS-FX Protocol |
|   Network Recording Use + False |
|                               |
+-----+

```

Figure 33: PCM2 Card Configuration

Device Configuration

Figure 34 shows configuration of one of the devices associated with one of the 30 channels on the trunk. This is the DN that is monitored by T-Server, and that enables this trunk to be used line-side. This is the only trunk type for which this configuration is possible.

-
- Notes:**
- The shelf and board addresses correspond to the PCM board.
 - The Equipment Address is 1, so this device is associated with the first of the 30 channels on this trunk.
 - Option ACD Station should be set to:
 - No if using Hunting Groups or URS to distribute calls to GVP ports.
 - IvR if using the ACD to distribute calls to GVP ports (IVR Processing Group).
 - Option Z IVR is set to true.
-

Other than these options, leave everything else with its default value.

```

mg=/Users/Review_Modify
+-Review/Modify: Users-----
Node Number (reserved) : 2
Directory Number : 1501
Directory name : PCM21501
Directory First Name : -----
Location Node : 2
Shelf Address : 0
Board Address : 5
Equipment Address : 1
Set Type : ANALOG
Entity Number : 1
Set Function : Default
Profile Name : -----
Key Profiles : None
Domain Identifier : 0
Language ID : 1
Secret Code : ****
Confirm : ****
Associated Set No. : 1501
Cost Center Id : 255
Cost Center Name : -----
Charging COS : Justified
Public Network COS : 2
External Forwarding COS : 255
Tel. Features COS ID : 0
Connection COS ID : 0
Hunt Group Dir No. : -----
ACD Group Directory No. : -----
Pickup Group Name : -----
Reserved Time Slot : False
Voice Mail Dir.No. : -----
Voice Mail Type : No Voice Mail
Paging Trunk Group : 255
Paging Bearer : ----
Tele-Marketing Agent : False
ISDN User
External : False
Internal : False
Display ext. calling number : True
ISDN Teleservice : Phone
Hotel-Set Operation : Administrative
Use Type Of Dir. No. : Normal
Number Of Set Users : 1
Multi-line station : NO
Dialed number masked : NO
Routing Table : 0
Associated Videophone : False
VIP (Very Important Pers.) : False
Assistant Directory Number : 1501
Calls Priority : 0
PCBT Associated : NO
Urgent Call : NO
PIN (Personal Ident.No.)
PIN No. : -----
PIN With Secret Code : True
Type of control : By COS
PIN group number : 1
Can be Called/Dialed By Name : YES
Displayed Name : PCM21501
Resides on Secret Code Counter : 0
ACD station : Ivr
Incidents Teleservice : NO
Voice Guide listening Class : 7
Caller COS : 4
VSI Transparency : True
Type of Keyboard : Default keyboard
Resides on Business Code Counter : 0
Stop : Off-hook
Use Personal Calling Number : False
Group PIN control : No group
CCA Operations : False
A4980 : No 4980
Z IVR : True
NOMADIC : False
TAPI Premium Server : NO
Conference group : -1
Announcement group : -1
Call Restriction COS : 0
Applicable Restriction COS : 0
Implicit Priority
Activation mode : 0
Priority Level : 0
Explicit Priority
Activation mode : 0
Priority Level : 0
Pre-emptable Primary Inc. Line : YES
Pre-emptable Secondary Inc. Line : YES
Priority Presentation : YES
Rch Service type : Not Valid
CUG List Number : -1
CUG Preferential : -1
CUG Outgoing Access : False
CUG Incoming Access : False
Automatical reconfiguration : CTQ forbidden - Connect TO
Called Associated Descr set : -----
DATA Connection COS ID : 0
Message Led : False
Ext.Alarm Equipment : Alarm On Opened Loop
Phone book Name (Dial by name) : PCM21501
Phone book First Name : -----
Modem Trunk Group Info
Trunk Group Id : 255
Trunk Number : 255
Ghost Z : False
Ghost Z Feature : Without
VAD use for Ghost Z : False
CTA routing : False
Cmf 4600 (DMP frequencies) : False

```

Figure 34: Device Configuration Dialog

Dialogic Configuration for DMV Cards

This section describes how to modify configuration files correctly.

The FCD File

File Location The FCD file is found at:
 C:\Program Files\Dialogic\data\ml2_dsa_r2mf.config
 (dsa indicates dual span cards, and qsa indicates quad span cards).

File Configuration In the [LineAdmin.1] section, set the following parameters, and repeat for [LineAdmin.2] (and 3 and 4 in the case of a quad span card).

```
SetParm=0x1601,2      ! LineType (dsx1_E1=2, dsx1_E1_CRC=3)
SetParm=0x1602,4      ! SignalingType (CAS=4, CCS=5, Clear=6)
SetParm=0x1603,9      ! Coding (AMI=8, HDB3=9)
```

Save the file, then, at the command line, change directory to C:\Program Files\Dialogic\data\ and run the following command:
fcdgen ml2_dsa_r2mf.config.

File Output The output should look like the following:

```
fcdgen Version 1.00 Beta 7 Build: 01
Copyright (c) Dialogic Corporation 2000
Building ml2_dsa_r2mf.fcd from ml2_dsa_r2mf.config
This generates the file ml2_dsa_r2mf.fcd
```

The CDP File

File Location The CDP file is found at:

C:\Program Files\Dialogic\cfg\pdk_sw_e1_ac4400_io.cdp

File Configuration This is the protocol variant definition. Do not use the original file that is delivered with the Dialogic Global Call Package, because it is not suitable without the following modifications:

The CAS Line Signals section should look as follows:

```
ALL CAS_SIGNAL_TRANS_t CAS_OFFHOOK = 0101,1101,50,50,0,80
ALL CAS_SIGNAL_TRANS_t CAS_ONHOOK = 1101,0101,50,250,0,300
ALL CAS_SIGNAL_TRANS_t CAS_RING_APPLIED = 0101,0001,50,50,0,80
ALL CAS_SIGNAL_TRANS_t CAS_RING_STOPPED = 0001,0101,50,50,0,80
ALL CAS_SIGNAL_PULSE_t CAS_HOOKFLASH =
1101,0101,20,20,50,50,180,200,220
ALL CAS_SIGNAL_PULSE_t CAS_DISC = 0101,1101,20,20,50,50,100,500,1000
```

The line SYS_FEATURES definition should look as follows (feature_DNIS must not be present):

```
ALL CHARSTRING_t
SYS_FEATURES="feature_hold,feature_transfer,feature_outbound,feature_inbound"
```

The PDK.CFG File

File Location The CDP file is found at:

C:\Program Files\Dialogic\cfg\pdk.cfg

If the file does not exist, you must create it.

File Configuration This file should contain one line per board with the following text:

```
board 0 fcdfile ml2_dsa_r2mf.fcd pcdfile ml2_dsa_r2mf.pcd variant  
pdk_sw_e1_ac4400_io.cdp
```

Save the file and then run the following command from the command line:

```
pdkmanagerregsetup add.
```

Dialogic Configuration Manager

Open the Dialogic Configuration Manager GUI, and make sure that the following parameters are set correctly on the Misc. tab:

```
FCDFilename = ml2_dsa_r2mf.fcd
```

```
PCDFilename = ml2_dsa_r2mf.pcd
```

If you need to change these values, stop the Dialogic card first.

Start the Dialogic card

Troubleshooting

The following sections describe how to diagnose and solve common problems.

Using the Line Admin Tool

The Line Admin tool allows you to see the status of the trunk from the Dialogic side. The Line Admin tool has the following parameters:

`-board <n>` : Board number (required). Use the `Listboards` utility to obtain the board number.

For example, to see the status of board 0, use command `lineadmin -board 0`. A screen like the one shown in [Figure 35](#) will appear. (Only line 1 is active in this example.)

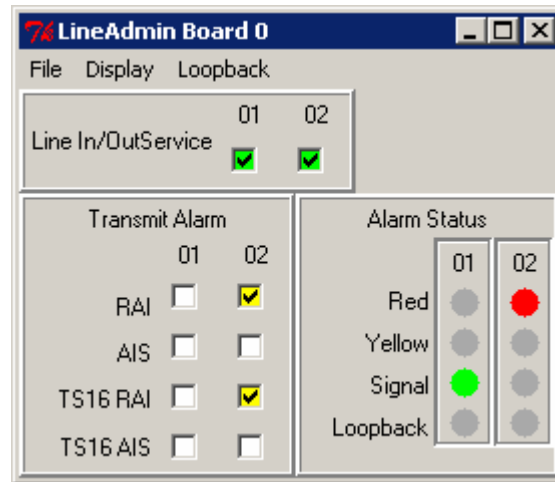


Figure 35: LineAdmin Board Dialog

Creating Traces on the PBX

Log in as user `mtch` (this will not work if you use `mtcl`).

Run the following commands:

`tuner km` Kills the mtracer process if it is already running

`mtracer &` Starts the mtracer process in the background

`tuner at` Activates traces

`debug_cpl <Shelf> <board> "entrscpu";` Activates traces; `<board>` and `<shelf>` must correspond to your PCM2 board.

When you have finished tracing, use `tuner km` to stop the trace program again.

A typical trace will look as follows:

```
OPS_FX : Outcall detect· ...
```

```
Z30 : AUTCALL ( 37) Term 1 Etat 0004->0006 RON 0005 TRON 0001 EVT
00F1
```

```
--- cut ---
```

```
OPS_FX : FIN_SONNE detecte ...
```

```
Z30 : AUTCALL ( 37) Term 1 Etat 0006->0004 RON 0005 TRON 0005 EVT
00F2
```

You can see that when a call arrives, the state changes from `0004` to `0006`, and changes back when the call is dropped (`0006` to `0004`).

Using the Telera TMTstApp Application

You can use the Telera TMTstApp application to test your configuration, and it enables you to make and receive calls. Note that it should not be run in parallel with WatchDog.

In the \cn\config directory, open the file TestApp.ini and make sure that the configuration matches what you have in the pop.ini file (POP Gateway ini file). The important sections in the pop.ini file are:

- [Popgateway1]
- [popgateway1_route1]
- [popgateway1_route2]

To run the application use the following command line:

```
cd \cn\bin
tmtstapp dialogicmgrsr511.dll
```

Using the Dialogic tspttrace Utility

The Dialogic tspttrace diagnostic utility shows the CAS signaling that is seen by Dialogic on the trunk.

The tspttrace utility has the following parameters:

- board <n> Board number (required). Use the Listboards utility to obtain the board number.
- line <n> Line number (optional, default is 1).
- chan <n> Channel number (optional, default is 1).

Example tspttrace -board 0

You can change the line and channel once the program has been started.

If this utility is started when no other application is connected to Dialogic (such as Watchdog, G_Call, or TMTstApp, nothing will be seen in the trace because tspttrace does not open the channels (it does not call gc_open). Therefore, you should use tspttrace in parallel with another application.

Trunk Signaling Problem—Trunk Active But Nothing Happens

Symptom The trunk seems to be active (the Lineadmin utility shows a green light, and it is possible to call the devices associated with the trunk channels), but Dialogic does not detect any call arriving. tspttrace does not show any state change when a call arrives. Also, the PBX trace shows a call, but the state is not changed (the new state is same as old state). For example:

```
OPS_FX : Outcall detect· ...
```

```
Z30 : AUTCALL ( 229) Term    1 Etat 0005->0005 RON 000D TRON 0005 EVT
00F1
```

```
--- cut ---
```

```
OPS_FX : FIN_SONNE detecte ...
```

```
Z30 : AUTCALL ( 229) Term    1 Etat 0005->0005 RON 000D TRON 0005 EVT
00F2
```

Solution In this case, the problem is that Dialogic is configured with the wrong variant. Make sure that the `pdk.cfg` file is configured correctly, and that you run the `pdkmanagerregsetup add` command. Then restart the Dialogic card.

Trunk Signaling Problem—Channels 16–30 Do Not Work

- Symptom**
- Channels 1–15 work but channels 16–30 do not.
 - When using the dialogic Lineadmin tool, there is a blue alarm on TS16 (see [Figure 36](#)).

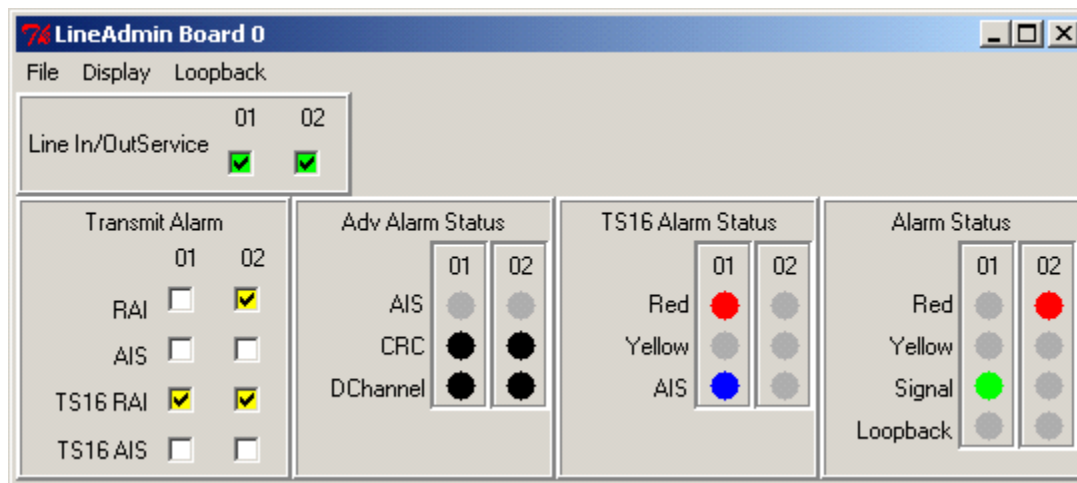


Figure 36: LineAdmin Board—TS16 Alarm Status

Solution The board was initially created with option IVR Protocol set to None, but then was subsequently changed to OPS-FX Protocol. To solve this problem, delete all the devices associated with the trunk channels and then delete the board. Then, recreate the board from scratch, making sure that option IVR Protocol is set to value OPS-FX Protocol before the configuration is applied the first time.

Dialogic Configuration for JCT-E1 Cards

This section describes the additional steps you must take to configure Dialogic and GVP to connect to the Alcatel A4400/OXE PBX using Dialogic JCT cards, after the standard Dialogic drivers have been installed.

Upgrade to Global Call Protocols Package 4.1

The standard Dialogic installation from Dialogic CD is not able to work correctly with the A4400 PBX. Therefore, after completing the Dialogic CD install, you must uninstall the Global Calls protocols Package 4.0 that is delivered as standard, and replace it with Global Calls protocols Package 4.1 which can be downloaded from the Intel website:

<http://resource.intel.com/telecom/support/releases/protocols/GCProtocols41/index.htm>

Install PTR 27176

In some cases, the server freezes when Dialogic Configuration Manager is started the first time when using JCT Cards. To prevent this from happening, you must install Dialogic PTR 27176, which can be downloaded from the Intel website:

<http://membersresource.intel.com/search/ptrs/display.asp?27176>

Dialogic Configuration Manager Settings

You must configure the following board settings in the Dialogic Configuration Manager:

- `FirmwareFile`—Use the default value
- `FirmwareFile2`—Use the default value
- `ParameterFile`—`fr_300.prm`
- `ParameterFile2`—`fr_300.prm`

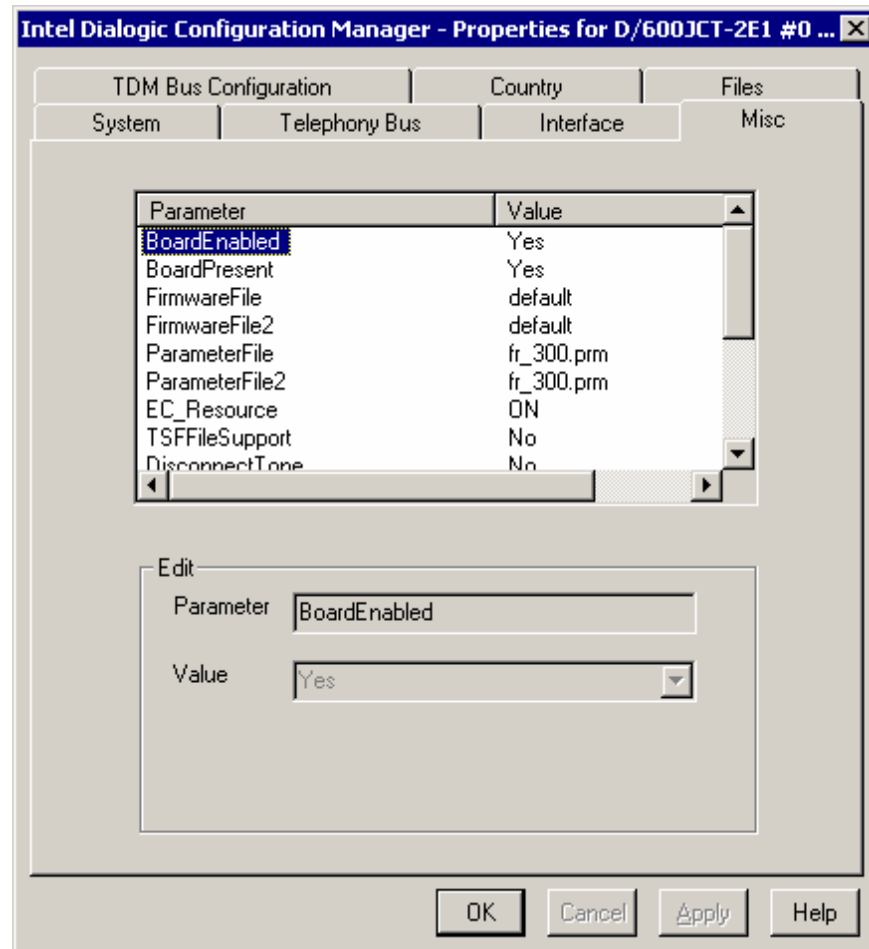


Figure 37: Dialogic Configuration Manager Settings

To change the board from Mu-Law (default) to A-Law, you must make a change in the TDM Bus configuration. This is **not** done on the TDM Bus Configuration tab of the board configuration. It must be done in the TDM Bus / Bus 0 branch of the DCM main window. See [Figure 38](#).

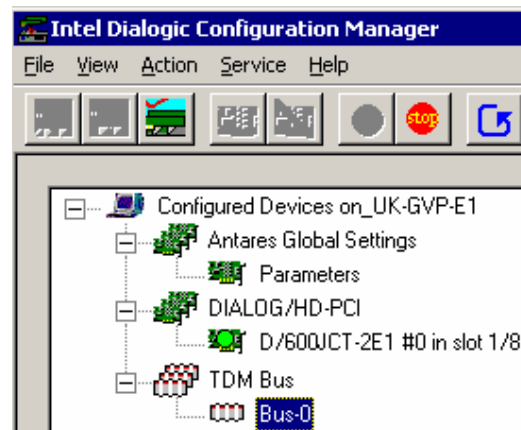


Figure 38: TDM Bus Configuration

Select Bus-0 from this tree. The dialog shown in [Figure 39](#) is displayed.

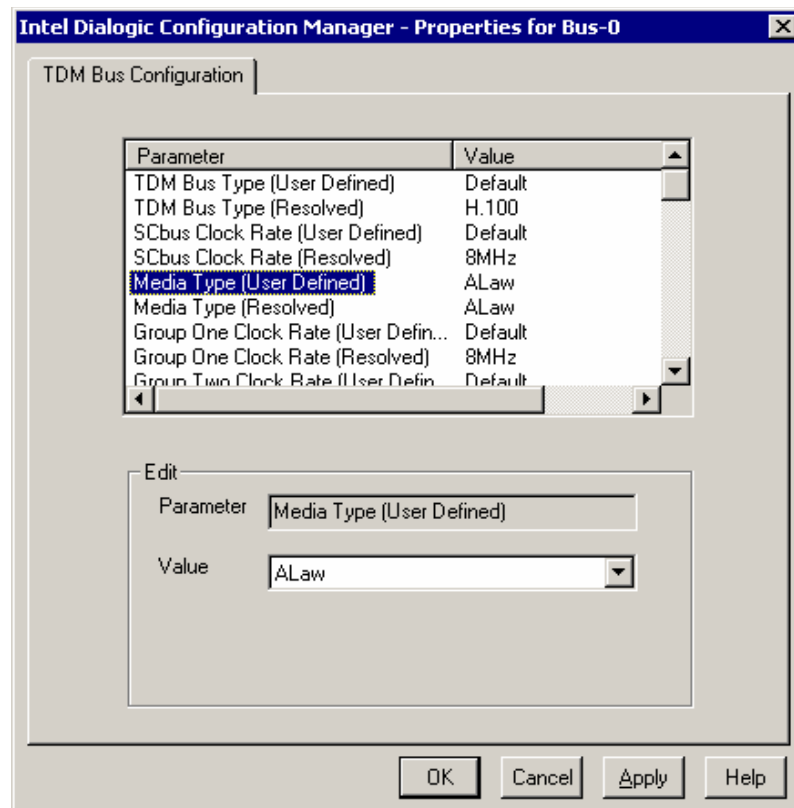


Figure 39: Properties for Bus-0

Ensure that Media Type (User Defined) has the value ALaw.

Adapt the Protocol File to Work with A4400

File Location The protocol file is located at:
C:\Program Files\Dialogic\cfg\pdk_sw_e1_ac4400_io.cdp

File Modifications This is the protocol variant definition.

Do not use the original file that is delivered with the Dialogic Global Call Package, because it is not suitable without the following modifications:

The CAS Line Signals section should look as follows:

```
ALL CAS_SIGNAL_TRANS_t CAS_OFFHOOK = 0101,1101,50,50,0,80
ALL CAS_SIGNAL_TRANS_t CAS_ONHOOK = 1101,0101,50,250,0,300
ALL CAS_SIGNAL_TRANS_t CAS_RING_APPLIED = 0101,0001,50,50,0,80
ALL CAS_SIGNAL_TRANS_t CAS_RING_STOPPED = 0001,0101,50,50,0,80
ALL CAS_SIGNAL_PULSE_t CAS_HOOKFLASH =
1101,0101,20,20,50,50,180,200,220
ALL CAS_SIGNAL_PULSE_t CAS_DISC = 0101,1101,20,20,50,50,100,500,1000
```

Note: The timing information for some signals has been modified compared to the definition given previously for GVP 6.5.3/4, so make sure that this information is entered correctly.

Defining the Protocol File To Be Used In GVP

When using JCT cards the protocol file to be used is defined in VPM.

In the menu system on the VPM main page select:

VCS Configuration->{hostname}->ServerConfiguration.

On this page set the following parameters:

[popgateway1 /route1] Signalling Type—E1-CAS

[popgateway1 /route1] T1rb Protocol File—pdk_sw_e1_ac4400_io



Chapter

12 Troubleshooting

This chapter describes how to resolve some common problems. It contains the following section:

- [Troubleshooting Procedures, page 309](#)

Troubleshooting Procedures

Procedures for troubleshooting CSTA licenses, CPU usage problems, CSTA protocol errors, network issues and timer issue are described in this chapter.

CSTA Licenses

Procedure: **Checking the Number of CSTA Licenses Available**

Purpose: To check the number of available CSTA licenses.

Start of procedure

1. Telnet to the PBX and type `spadmin`.
2. In the menu, select option 1—Display current counters to display all the licences available.

End of procedure

Procedure: Checking the licensing feature descriptions

Purpose: To check the licensing feature descriptions.

Start of procedure

1. Telnet to the PBX and type `spadmin`.
2. In the menu, select option 2-Check active file to display the licensing feature descriptions:

100 - CSTA profile1 or 2	1: business sets can be monitored. 2: business sets and CCD/RSI agents can be monitored.
101 - CSTA monitoring requests	The maximum number of CSTA monitor requests that can made to the PBX.
113 - CSTA pilots monitoring 0 or 1	1: CCD Pilots are monitorable by CSTA.
169 Voice detection channels	The number of VAD channels that are used as a virtual Z device for outbound dialing.
190 RSI call center agents	The number of RSI Processing Group agents that are available on the PBX.

Note: The OPS license file must be present on all switch nodes for the RSI ring overflow functionality to be available across switch nodes.

End of procedure

Procedure: Checking the Number of CSTA Licenses Used

Start of procedure

1. Telnet to the PBX and login as user `mtch`.
2. Type `cstainfo`. The output shows each server connected to the CSTA link, and information about which devices are monitored by each. The last line of the output shows the total number of CSTA licenses used; for example, `number of monitoring request = 176`.

- Alternatively a browser can be connected to `http://[oxe_switch_IP_address]:2555` in order to see all the monitoring requests issued by T-Server(s).

End of procedure

CPU Usage

Procedure: Analyzing 100 Percent CPU Usage

Purpose: To discover what process is causing 100% CPU usage.

Start of procedure

- Telnet to the Alcatel A4400 and type `ps -edf`.
- Wait 5 seconds and run the command again. You can identify the process that is causing the high CPU usage by comparing the two outputs. The process that usually causes the CPU usage is `main_afe`.
- Pass this information to the switch administrator for further action.

Note: The `ps -edf` command shows the accumulated amount of CPU time that the process has used since it was started.

End of procedure

Procedure: Monitoring CPU Load

Purpose: To monitor the actual amount of CPU time load.

Start of procedure

- Telnet to the PBX.
- Type `pidle 3 5`. (This will show the %CPU load five times at 3-second intervals. You can change the parameters to suit your needs.)

The output should resemble the following:

Count	Time	CPU Load	CPU Free
1	3.103	3.319 %	96.681 %
2	3.086	2.786 %	97.214 %
3	3.095	3.069 %	96.931 %

4	3.135	4.306 %	95.694 %
5	3.108	3.474 %	96.526 %

3. Monitor the CPU usage to make sure that it is not overloaded.
4. If you are worried about CPU usage, then run the `ps -edf` command several times.

End of procedure

CSTA Protocol Errors

There are numerous causes of CSTA protocol errors.

Note: One possible cause for a protocol error is that the network interface on the PBX has not been isolated. See “[Network Issues](#)” for more information.

Procedure: Identifying CSTA protocol errors

Purpose: To capture information that will help you identify the causes of CSTA protocol errors.

Start of procedure

1. Telnet to the Alcatel A4400.
2. Type the command `excvisu`.

End of procedure

Network Issues

It is an Alcatel requirement that the CTI link be isolated on the network. If you fail to do this, the PBX network interface receives network broadcast messages, which can cause problems for the CSTA interface.

Procedure: Identifying whether CTI link is isolated

Start of procedure

1. Telnet to the Alcatel A4400 and type `incvisu`.

Note: The Alcatel A4400 keeps information for up to three restarts (or CPU switchovers in a multi-CPU environment). To output this information, use the commands `incvisu -1`, `incvisu -2`, `incvisu -3`. Note that not all versions of the Alcatel A4400 support three restarts; some only support two.

2. Check the output for the following message by searching with the keyword *Ethernet*:

```
25/01/01 12:01:02 000001M|--/--/-/---|=3:1566=Desactivation  
reception broadcast  
Ethernet ; trafic broadcast trop dense
```
3. If this message is present, your network administrator *must* isolate the CTI link; for example, by using two Ethernet cards in the machine where T-Server is running. In this case, one Ethernet card is used for T-Server client connections and the other for the CTI link to the Alcatel A4400.

End of procedure

Timer Issues

Timer 42 controls how long after a call is established that a consultation call can be made. If transfers or conferences are failing, this timer may be set for too long a period. See “Configuring Switch Timers” on [page 145](#) for more information on how to set this timer.

13

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 315](#)
- [Mandatory Options, page 316](#)
- [log Section, page 316](#)
- [log-extended Section, page 330](#)
- [log-filter Section, page 332](#)
- [log-filter-data Section, page 332](#)
- [security Section, page 333](#)
- [sml Section, page 333](#)
- [common Section, page 335](#)
- [Changes from 8.0 to 8.1, page 335](#)

Note: Some server applications also support log options that are unique to them. For descriptions of a particular application's unique log options, refer to the chapter/document about that application.

Setting Configuration Options

Unless specified otherwise, set common configuration options in the Options of the Application object, using one of the following navigation paths:

- In Genesys Administrator—Application object > Options tab > Advanced View (Options)
- In Configuration Manager—Application object > Properties dialog box > Options tab

Warning! Configuration section names, configuration option names, and predefined option values are case-sensitive. Type them in Genesys Administrator or Configuration Manager exactly as they are documented in this chapter.

Mandatory Options

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

log Section

This section must be called `log`.

verbose

Default Value: `all`

Valid Values:

<code>all</code>	All log events (that is, log events of the Standard, Trace, Interaction, and Debug levels) are generated.
<code>debug</code>	The same as <code>all</code> .
<code>trace</code>	Log events of the Trace level and higher (that is, log events of the Standard, Interaction, and Trace levels) are generated, but log events of the Debug level are not generated.
<code>interaction</code>	Log events of the Interaction level and higher (that is, log events of the Standard and Interaction levels) are generated, but log events of the Trace and Debug levels are not generated.
<code>standard</code>	Log events of the Standard level are generated, but log events of the Interaction, Trace, and Debug levels are not generated.
<code>none</code>	No output is produced.

Changes Take Effect: Immediately

Determines whether a log output is created. If it is, specifies the minimum level of log events generated. The log events levels, starting with the highest priority level, are Standard, Interaction, Trace, and Debug. See also “Log Output Options” on [page 322](#).

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

buffering

Default Value: `true`

Valid Values:

`true` Enables buffering.
`false` 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 322](#)). 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:

`false` No segmentation is allowed.
`<number> KB` or Sets the maximum segment size, in kilobytes. The minimum
`<number>` segment size is `100 KB`.
`<number> MB` Sets the maximum segment size, in megabytes.
`<number> hr` Sets the number of hours for the segment to stay open. The
 minimum number is 1 hour.

Changes Take Effect: Immediately

Specifies whether there is a segmentation limit for a log file. If there is, sets the mode of measurement, along with the maximum size. If the current log segment exceeds the size set by this option, the file is closed and a new one is created. This option is ignored if log output is not configured to be sent to a log file.

expire

Default Value: `false`

Valid Values:

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

Changes Take Effect: Immediately

Determines whether log files expire. If they do, sets the measurement for determining when they expire, along with the maximum number of files (segments) or days before the files are removed. This option is ignored if log output is not configured to be sent to a log file.

Note: If an option's value is set incorrectly—out of the range of valid values— it will be automatically reset to `10`.

keep-startup-fileDefault Value: `false`

Valid Values:

<code>false</code>	No startup segment of the log is kept.
<code>true</code>	A startup segment of the log is kept. The size of the segment equals the value of the <code>segment</code> option.
<code><number> KB</code>	Sets the maximum size, in kilobytes, for a startup segment of the log.
<code><number> MB</code>	Sets the maximum size, in megabytes, for a startup segment of the log.

Changes Take Effect: After restart

Specifies whether a startup segment of the log, containing the initial T-Server configuration, is to be kept. If it is, this option can be set to `true` or to a specific size. If set to `true`, the size of the initial segment will be equal to the size of the regular log segment defined by the `segment` option. The value of this option will be ignored if segmentation is turned off (that is, if the `segment` option set to `false`).

Note: This option applies only to T-Servers.

messagefile

Default Value: As specified by a particular application

Valid Values: `<string>.lms` (message file name)Changes Take Effect: Immediately, if an application cannot find its `*.lms` file at startup

Specifies the file name for application-specific log events. The name must be valid for the operating system on which the application is running. The option value can also contain the absolute path to the application-specific `*.lms` file. Otherwise, an application looks for the file in its working directory.

Warning! An application that does not find its `*.lms` file at startup cannot generate application-specific log events and send them to Message Server.

message_formatDefault Value: `short`

Valid Values:

<code>short</code>	An application uses compressed headers when writing log records in its log file.
<code>full</code>	An application uses complete headers when writing log records in its log file.

Changes Take Effect: Immediately

Specifies the format of log record headers that an application uses when writing logs in the log file. Using compressed log record headers improves application performance and reduces the log file's size.

With the value set to short:

- A header of the log file or the log file segment contains information about the application (such as the application name, application type, host type, and time zone), whereas single log records within the file or segment omit this information.
- A log message priority is abbreviated to Std, Int, Trc, or Dbg, for Standard, Interaction, Trace, or Debug messages, respectively.
- The message ID does not contain the prefix GCTI or the application type ID.

A log record in the full format looks like this:

```
2002-05-07T18:11:38.196 Standard localhost cfg_dbserver GCTI-00-05060
Application started
```

A log record in the short format looks like this:

```
2002-05-07T18:15:33.952 Std 05060 Application started
```

Note: Whether the full or short format is used, time is printed in the format specified by the `time_format` option.

time_convert

Default Value: Local

Valid Values:

- | | |
|-------|--|
| local | The time of log record generation is expressed as a local time, based on the time zone and any seasonal adjustments. Time zone information of the application's host computer is used. |
| utc | The time of log record generation is expressed as Coordinated Universal Time (UTC). |

Changes Take Effect: Immediately

Specifies the system in which an application calculates the log record time when generating a log file. The time is converted from the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970).

time_format

Default Value: time

Valid Values:

- | | |
|---------|--|
| time | The time string is formatted according to the HH:MM:SS.sss (hours, minutes, seconds, and milliseconds) format. |
| locale | The time string is formatted according to the system's locale. |
| ISO8601 | The date in the time string is formatted according to the ISO 8601 format. Fractional seconds are given in milliseconds. |

Changes Take Effect: Immediately

Specifies how to represent, in a log file, the time when an application generates log records.

A log record's time field in the ISO 8601 format looks like this:

2001-07-24T04:58:10.123

print-attributes

Default Value: `false`

Valid Values:

`true` Attaches extended attributes, if any exist, to a log event sent to log output.

`false` Does not attach extended attributes to a log event sent to log output.

Changes Take Effect: Immediately

Specifies whether the application attaches extended attributes, if any exist, to a log event that it sends to log output. Typically, log events of the Interaction log level and Audit-related log events contain extended attributes. Setting this option to `true` enables audit capabilities, but negatively affects performance. Genesys recommends enabling this option for Solution Control Server and Configuration Server when using audit tracking. For other applications, refer to *Genesys 8.0 Combined Log Events Help* to find out whether an application generates Interaction-level and Audit-related log events; if it does, enable the option only when testing new interaction scenarios.

check-point

Default Value: 1

Valid Values: 0–24

Changes Take Effect: Immediately

Specifies, in hours, how often the application generates a check point log event, to divide the log into sections of equal time. By default, the application generates this log event every hour. Setting the option to 0 prevents the generation of check-point events.

memory

Default Value: No default value

Valid Values: `<string>` (memory file name)

Changes Take Effect: Immediately

Specifies the name of the file to which the application regularly prints a snapshot of the memory output, if it is configured to do this (see “Log Output Options” on [page 322](#)). 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 322](#).

spool

Default Value: The application’s working directory

Valid Values: <path> (the folder, with the full path to it)

Changes Take Effect: Immediately

Specifies the folder, including full path to it, in which an application creates temporary files related to network log output. If you change the option value while the application is running, the change does not affect the currently open network output.

compatible-output-priority

Default Value: false

Valid Values:

true The log of the level specified by “Log Output Options” is sent to the specified output.

false The log of the level specified by “Log Output Options” and higher levels is sent to the specified output.

Changes Take Effect: Immediately

Specifies whether the application uses 6.x output logic. For example, you configure the following options in the log section for a 6.x application and for a 7.x application:

```
[log]
verbose = all
debug = file1
standard = file2
```

The log file content of a 6.x application is as follows:

- `file1` contains Debug messages only.
- `file2` contains Standard messages only.

The log file content of a 7.x application is as follows:

- `file1` contains Debug, Trace, Interaction, and Standard messages.
- `file2` contains Standard messages only.

If you set `compatible-output-priority` to `true` in the 7.x application, its log file content will be the same as for the 6.x application.

Warning! Genesys does not recommend changing the default value of this option unless you have specific reasons to use the 6.x log output logic—that is, to mimic the output priority as implemented in releases 6.x. Setting this option to `true` affects log consistency.

Log Output Options

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

You can use:

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

You must separate the log output types by a comma when you are configuring more than one output for the same log level. See “Examples” on [page 326](#).

Warnings!

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

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

all

Default Value: No default value

Valid Values (log output types):

<code>stdout</code>	Log events are sent to the Standard output (<code>stdout</code>).
<code>stderr</code>	Log events are sent to the Standard error output (<code>stderr</code>).
<code>network</code>	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. Setting the <code>all</code> log level option to the <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.

alarm

Default Value: No default value

Valid Values (log output types):

<code>stdout</code>	Log events are sent to the Standard output (<code>stdout</code>).
<code>stderr</code>	Log events are sent to the Standard error output (<code>stderr</code>).
<code>network</code>	Log events are sent to Message Server, which resides anywhere on the network, and Message Server stores the log events in the Log Database.
<code>memory</code>	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
<code>[filename]</code>	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Alarm` level. The log output types must be separated by a comma when more than one output is configured. For example:

```
standard = stderr, network
```

standard

Default Value: No default value

Valid Values (log output types):

<code>stdout</code>	Log events are sent to the Standard output (<code>stdout</code>).
<code>stderr</code>	Log events are sent to the Standard error output (<code>stderr</code>).
<code>network</code>	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
<code>memory</code>	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
<code>[filename]</code>	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the Standard level. The log output types must be separated by a comma when more than one output is configured. For example:

```
standard = stderr, network
```

interaction

Default Value: No default value

Valid Values (log output types):

<code>stdout</code>	Log events are sent to the Standard output (<code>stdout</code>).
<code>stderr</code>	Log events are sent to the Standard error output (<code>stderr</code>).
<code>network</code>	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
<code>memory</code>	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
<code>[filename]</code>	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

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

```
interaction = stderr, network
```

trace

Default Value: No default value

Valid Values (log output types):

<code>stdout</code>	Log events are sent to the Standard output (<code>stdout</code>).
<code>stderr</code>	Log events are sent to the Standard error output (<code>stderr</code>).
<code>network</code>	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
<code>memory</code>	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
<code>[filename]</code>	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

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

```
trace = stderr, network
```

debug

Default Value: No default value

Valid Values (log output types):

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

Changes Take Effect: Immediately

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

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

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

Log File Extensions

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

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

Note: Provide `*.snapshot.log` files to Genesys Technical Support when reporting a problem.

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

Examples

This section presents examples of a log section that you might configure for an application when that application is operating in production mode and in two lab modes, debugging and troubleshooting.

Production Mode Log Section

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

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

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

Lab Mode Log Section

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

With this configuration, an application generates log events of the Standard, Interaction, Trace, and Debug levels, and sends them to the standard output and to a file named `logfile`, which the application creates in the `/usr/local/genesys/` directory. In addition, the application sends log events of the Standard, Interaction, and Trace levels to Message Server. Use this configuration to test new interaction scenarios in a lab environment.

Failure-Troubleshooting Log Section

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

With this configuration, an application generates log events of the Standard level and sends them to Message Server. It also generates log events of the Standard, Interaction, Trace, and Debug levels, and sends them to the memory output. The most current log is stored to a file named `logfile`, which the application creates in its working directory. Increased memory storage allows an application to save more of the log information generated before a failure.

Note: If you are running an application on UNIX, and you do not specify any files in which to store the memory output snapshot, a core file that the application produces before terminating contains the most current application log. Provide the application's core file to Genesys Technical Support when reporting a problem.

Debug Log Options

The options in this section enable you to generate Debug logs containing information about specific operations of an application.

x-conn-debug-open

Default Value: 0

Valid Values:

- 0 Log records are not generated.
- 1 Log records are generated.

Changes Take Effect: After restart

Generates Debug log records about “open connection” operations of the application.

Warning! Use this option only when requested by Genesys Technical Support.

x-conn-debug-select

Default Value: 0

Valid Values:

- 0 Log records are not generated.
- 1 Log records are generated.

Changes Take Effect: After restart

Generates Debug log records about “socket select” operations of the application.

Warning! Use this option only when requested by Genesys Technical Support.

x-conn-debug-timers

Default Value: 0

Valid Values:

- 0 Log records are not generated.
- 1 Log records are generated.

Changes Take Effect: After restart

Generates Debug log records about the timer creation and deletion operations of the application.

Warning! Use this option only when requested by Genesys Technical Support.

x-conn-debug-write

Default Value: 0

Valid Values:

- 0 Log records are not generated.
- 1 Log records are generated.

Changes Take Effect: After restart

Generates Debug log records about “write” operations of the application.

Warning! Use this option only when requested by Genesys Technical Support.

x-conn-debug-security

Default Value: 0

Valid Values:

0 Log records are not generated.

1 Log records are generated.

Changes Take Effect: After restart

Generates Debug log records about security-related operations, such as Transport Layer Security and security certificates.

Warning! Use this option only when requested by Genesys Technical Support.

x-conn-debug-api

Default Value: 0

Valid Values:

0 Log records are not generated.

1 Log records are generated.

Changes Take Effect: After restart

Generates Debug log records about connection library function calls.

Warning! Use this option only when requested by Genesys Technical Support.

x-conn-debug-dns

Default Value: 0

Valid Values:

0 Log records are not generated.

1 Log records are generated.

Changes Take Effect: After restart

Generates Debug log records about DNS operations.

Warning! Use this option only when requested by Genesys Technical Support.

x-conn-debug-all

Default Value: 0

Valid Values:

- 0 Log records are not generated.
- 1 Log records are generated.

Changes Take Effect: After restart

Generates Debug log records about open connection, socket select, timer creation and deletion, write, security-related, and DNS operations, and connection library function calls. This option is the same as enabling or disabling all of the previous x-conn-debug-`<op type>` options.

Warning! Use this option only when requested by Genesys Technical Support.

log-extended Section

This section must be called log-extended.

level-reassign-`<eventID>`Default Value: Default value of log event `<eventID>`

Valid Values:

- alarm The log level of log event `<eventID>` is set to Alarm.
- standard The log level of log event `<eventID>` is set to Standard.
- interaction The log level of log event `<eventID>` is set to Interaction.
- trace The log level of log event `<eventID>` is set to Trace.
- debug The log level of log event `<eventID>` is set to Debug.
- none Log event `<eventID>` is not recorded in a log.

Changes Take Effect: Immediately

Specifies a log level for log event `<eventID>` that is different than its default level, or disables log event `<eventID>` completely. If no value is specified, the log event retains its default level. This option is useful when you want to customize the log level for selected log events.

These options can be deactivated with the option [level-reassign-disable](#).

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

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

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

In addition to the preceding warning, take note of the following:

- Logs can be customized only by release 7.6 or later applications.
- When the log level of a log event is changed to any level except none, it is subject to the other settings in the [log] section at its new level. If set to none, it is not logged and is therefore not subject to any log configuration.
- Using this feature to change the log level of a log changes only its priority; it does not change how that log is treated by the system. For example, increasing the priority of a log to Alarm level does not mean that an alarm will be associated with it.
- Each application in a High Availability (HA) pair can define its own unique set of log customizations, but the two sets are not synchronized with each other. This can result in different log behavior depending on which application is currently in primary mode.
- This feature is not the same as a similar feature in Universal Routing Server (URS) release 7.2 or later. In this Framework feature, the priority of log events are customized. In the URS feature, the priority of debug messages only are customized. Refer to the *Universal Routing Reference Manual* for more information about the URS feature.
- You cannot customize any log event that is not in the unified log record format. Log events of the Alarm, Standard, Interaction, and Trace levels feature the same unified log record format.

Example

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

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

Before the log levels of the log are changed:

- Log event 1020, with default level standard, is output to `stderr` and `log_file`, and sent to Message Server.
- Log event 2020, with default level standard, is output to `stderr` and `log_file`, and sent to Message Server.
- Log event 3020, with default level trace, is output to `stderr`.
- Log event 4020, with default level debug, is output to `stderr`.

Extended log configuration section:

```
[log-extended]
level-reassign-1020=none
level-reassign-2020=interaction
level-reassign-3020=interaction
level-reassign-4020=standard
```

After the log levels are changed:

- Log event 1020 is disabled and not logged.
- Log event 2020 is output to `stderr` and `log_file`.
- Log event 3020 is output to `stderr` and `log_file`.
- Log event 4020 is output to `stderr` and `log_file`, and sent to Message Server.

level-reassign-disable

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

When this option is set to `true`, the original (default) log level of all log events in the `[log-extended]` section are restored. This option is useful when you want to use the default levels, but not delete the customization statements.

log-filter Section

The `log-filter` section contains configuration options used to define the default treatment of filtering data in log output. This section contains one configuration option, `default-filter-type`. Refer to the chapter “Hide Selected Data in Logs” in the *Genesys 8.x Security Deployment Guide* for complete information about this option.

log-filter-data Section

The `log-filter-data` section contains configuration options used to define the treatment of filtering data in log output on a key-by-key basis. This section contains one configuration option in the form of `<key name>`. Refer to the

chapter “Hide Selected Data in Logs” in the *Genesys 8.0 Security Deployment Guide* for complete information about this option.

security Section

The `security` section contains configuration options used to specify security elements for your system. In addition to other options that may be required by your application, this section contains the configuration option `disable-rbac`, which is used to enable or disable Role-Based Access Control for an application. Refer to the chapter “Role-Based Access Control” in the *Genesys 8.x Security Deployment Guide* for complete information about this option.

sml Section

This section must be called `sml`.

Options in this section are defined in the Annex of the `Application` object, as follows:

- in Genesys Administrator—`Application` object > `Options` tab > `Advanced View` (Annex)
- in Configuration Manager—`Application` object > `Properties` dialog box > `Annex` tab

Warning! Use the first three options in this section (`heartbeat-period`, `heartbeat-period-thread-class-<n>`, and `hangup-restart`) with great care, and only with those applications of which support for this functionality has been announced. Failure to use these options properly could result in unexpected behavior, from ignoring the options to an unexpected restart of the application.

heartbeat-period

Default Value: None

Valid Values:

- | | |
|-----------------------|---|
| <code>0</code> | This method of detecting an unresponsive application is not used by this application. |
| <code>3-604800</code> | Length of timeout, in seconds; equivalent to 3 seconds–7 days. |

Changes Take Effect: Immediately

Specifies the maximum amount of time, in seconds, in which heartbeat messages are expected from an application. If Local Control Agent (LCA) does not receive a heartbeat message from the application within this period, it assumes the application is not responding and carries out corrective action.

This option can also be used to specify the maximum heartbeat interval for threads registered with class zero (0). This thread class is reserved for use by the Management Layer only.

If this option is not configured or is set to zero (0), heartbeat detection is not used by this application.

heartbeat-period-thread-class-<n>

Default Value: None

Valid Values:

- 0 Value specified by `heartbeat-period` in application is used.
- 3-604800 Length of timeout, in seconds; equivalent to 3 seconds–7 days.

Changes Take Effect: Immediately

Specifies the maximum amount of time, in seconds, in which heartbeat messages are expected from a thread of class <n> registered by an application. If a heartbeat message from the thread is not received within this period, the thread is assumed to be not responding, and therefore, the application is unable to provide service.

If this option is not configured or is set to zero (0), but the application has registered one or more threads of class <n>, the value specified by the value of `heartbeat-period` for the application will also be applied to these threads.

Refer to application-specific documentation to determine what thread classes, if any, are used.

hangup-restart

Default Value: true

Valid Values: true, false

Changes Take Effect: Immediately

If set to true (the default), specifies that LCA is to restart the unresponsive application immediately, without any further interaction from Solution Control Server.

If set to false, specifies that LCA is only to generate a notification that the application has stopped responding.

suspending-wait-timeout

Default Value: 10

Valid Values: 5-600

Changes Take Effect: Immediately

Specifies a timeout (in seconds) after the Stop Graceful command is issued to an application during which the status of the application should change to `Suspending` if the application supports graceful shutdown. If the status of the application does not change to `Suspending` before the timeout expires, it is assumed that the application does not support graceful shutdown, and it is stopped ungracefully.

Use this option if you are unsure whether the Application supports graceful shutdown.

Note: Genesys recommends that you do not set this option for any Management Layer component (Configuration Server, Message Server, Solution Control Server, or SNMP Master Agent) or any DB Server. These components by definition do not support graceful shutdown, so this option is not required.

common Section

This section must be called `common`.

enable-async-dns

Default Value: `off`

Valid Values:

`off` Disables asynchronous processing of DNS requests.
`on` Enables asynchronous processing of DNS requests.

Changes Take Effect: Immediately

Enables the asynchronous processing of DNS requests such as, for example, host-name resolution.

Warnings! • Use this option only when requested by Genesys Technical Support.
• Use this option only with T-Servers.

rebind-delay

Default Value: `10`

Valid Values: `0–600`

Changes Take Effect: After restart

Specifies the delay, in seconds, between socket-bind operations that are being executed by the server. Use this option if the server has not been able to successfully occupy a configured port.

Warning! Use this option only when requested by Genesys Technical Support.

Changes from 8.0 to 8.1

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

14

T-Server Common Configuration Options

This chapter describes the configuration options that are generally common to all T-Server types, with some exceptions noted. It contains the following sections:

- [Setting Configuration Options, page 337](#)
- [Mandatory Options, page 338](#)
- [TServer Section, page 338](#)
- [license Section, page 343](#)
- [agent-reservation Section, page 346](#)
- [extrouter Section, page 347](#)
- [backup-sync Section, page 358](#)
- [call-cleanup Section, page 360](#)
- [Translation Rules Section, page 361](#)
- [security Section, page 362](#)
- [Timeout Value Format, page 362](#)
- [Changes from Release 8.0 to 8.1, page 363](#)

T-Server also supports common log options described in Chapter 13, “Common Configuration Options,” on [page 315](#).

Setting Configuration Options

Unless specified otherwise, set T-Server common configuration options in the Options of the Application object, using one of the following navigation paths:

- In Genesys Administrator—Application object > Options tab > Advanced View (Options)
- In Configuration Manager—Application object > Properties dialog box > Options tab

Mandatory Options

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

TServer Section

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

This section must be called TServer.

ani-distribution

Default Value: inbound-calls-only

Valid Values: inbound-calls-only, all-calls, suppressed

Changes Take Effect: Immediately

Controls the distribution of the ANI information in TEvent messages. When this option is set to all-calls, the ANI attribute will be reported for all calls for which it is available. When this option is set to suppressed, the ANI attribute will not be reported for any calls. When this option is set to inbound-calls-only, the ANI attribute will be reported for inbound calls only.

background-processing

Default Value: true

Valid Values: true, false

Changes Take Effect: Immediately

When set to true, T-Server processes all client requests in the background, giving higher priority to the rest of the messages. This ensures that it processes these messages without any significant delay.

With Background Processing functionality enabled, T-Server processes all switch messages immediately and waits until there are no switch messages before processing the message queue associated with T-Server client requests. T-Server reads all connection sockets immediately and places client requests in the input buffer, which prevents T-Server clients from disconnecting because of configured timeouts.

When T-Server processes client requests from the message queue, requests are processed in the order in which T-Server received them.

When set to false, T-Server processes multiple requests from one T-Server client before proceeding to the requests from another T-Server client, and so on.

background-timeout

Default Value: 60 msec

Valid Values: See “Timeout Value Format” on [page 362](#).

Changes Take Effect: Immediately

Specifies the time interval that T-Server waits before processing client requests in background mode. You must set the `background-processing` option to `true` in order for this option to take effect.

check-tenant-profile

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: For the next connected client

When set to `true`, T-Server only allows a client to register if the client provides the correct name and password of a T-Server Tenant. If the client provides the Tenant name concatenated with a slash (/) and the Tenant password for the Tenant to which T-Server belongs as the value of `AttributeApplicationPassword` in the `TRegisterClient` request, T-Server allows that client to register DNs that are included in the switch configuration in the Configuration Database, but it does not allow the client to register DNs that are *not* included in the switch configuration.

consult-user-data

Default Value: `separate`

Valid Values:

<code>separate</code>	Stores user data for original and consultation calls in separate structures. The data attached to the original call is available for review or changes only to the parties of that call. The data attached to the consultation call is available only to the parties of the consultation call.
<code>inherited</code>	Copies user data from an original call to a consultation call when the consultation call is created; thereafter, stores user data separately for the original and the consultation call. Changes to the original call's user data are not available to the parties of the consultation call, and vice versa.
<code>joint</code>	Stores user data for an original call and a consultation call in one structure. The user data structure is associated with the original call, but the parties of both the original and consultation calls can see and make changes to the common user data.

Changes Take Effect: For the next consultation call created

Specifies the method for handling user data in a consultation call.

Note: A T-Server client can also specify the `consult-user-data` mode in the `Extensions` attribute `ConsultUserData` key for a conference or transfer request. If it is specified, the method of handling user data is based on the value of the `ConsultUserData` key-value pair of the request and takes precedence over the T-Server `consult-user-data` option. If it is not specified in the client request, the value specified in the `consult-user-data` option applies.

customer-id

Default Value: No default value. (A value must be specified for a multi-tenant environment.)

Valid Values: Any character string

Changes Take Effect: Immediately

Identifies the T-Server customer. You must set this option to the name of the tenant that is using this T-Server. You must specify a value for this option if you are working in a multi-tenant environment.

Note: Do not configure the `customer-id` option for single-tenant environments.

dn-scope

Default Value: `undefined`

Valid Values: `undefined`, `switch`, `office`, `tenant`

Changes Take Effect: Immediately

Related Feature: “Switch Partitioning” on [page 100](#)

Specifies whether DNs associated with the `Switch`, `Switching Office`, or `Tenant` objects will be considered in the T-Server monitoring scope, enabling T-Server to report calls to or from those DNs as internal.

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

With a value of `undefined` (the default), pre-8.x T-Server behavior applies and the switch partitioning is not turned on.

Note: Setting the option to a value of `office` or `tenant`, which requires T-Server to monitor a large set of configuration data, may negatively affect T-Server performance.

log-trace-flags

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

Valid Values (in any combination):

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

Changes Take Effect: Immediately

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

management-port

Default Value: `0`

Valid Values: `0` or any valid TCP/IP port

Changes Take Effect: After T-Server is restarted

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

merged-user-data

Default Value: `main-only`

Valid Values:

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

Changes Take Effect: Immediately

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

Note: The option setting does not affect the resulting data for merging calls if the `consult-user-data` option is set to `joint`. (See “consult-user-data” on [page 339](#).)

propagated-call-type

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Switch Partitioning” on [page 100](#)

Determines what T-Server reports as the value of the `CallType` attribute in events related to calls that have been synchronized with another site via ISCC, as follows:

- When set to `false`, T-Server reports in events related to calls that have been synchronized with another site via ISCC the same value for the `CallType` attribute as it did in pre-8.0 releases and adds the new `PropagatedCallType` attribute with the value of the `CallType` attribute at the origination site. This provides backward compatibility with existing T-Server clients.
- When set to `true`, T-Server reports in events related to calls that have been synchronized with another site via ISCC the same value for the `CallType` attribute as at the origination site, and adds the new `LocalCallType` attribute with the same value as `CallType` in pre-8.0 releases.

server-id

Default Value: An integer equal to the value `ApplicationDBID` as reported by Configuration Server

Valid Values: Any integer from 0–16383

Changes Take Effect: Immediately

Specifies the `Server ID` that T-Server uses to generate `Connection IDs` and other unique identifiers. In a multi-site environment, you must assign each T-Server a unique `Server ID`, in order to avoid confusion in reporting applications and T-Server behavior.

Configuration of this option is necessary for Framework environments in which there are two or more instances of the Configuration Database.

Note: If you do not specify a value for this option, T-Server populates it with the `ApplicationDBID` as reported by Configuration Server. Each data object in the Configuration Database is assigned a separate DBID that maintains a unique `Server ID` for each T-Server configured in the database.

Warning! Genesys does not recommend using multiple instances of the Configuration Database.

user-data-limit

Default Value: 16000

Valid Values: 0–65535

Changes Take Effect: Immediately

Specifies the maximum size (in bytes) of user data in a packed format.

Note: When T-Server works in mixed 8.x/7.x/6.x environment, the value of this option must not exceed the default value of 16000 bytes; otherwise, 6.x T-Server clients might fail.

license Section

The License section contains the configuration options that are used to configure T-Server licenses. They set the upper limit of the seat-related DN licenses (`tserver_sdn`) that T-Server tries to check out from a license file. See “License Checkout” on [page 344](#).

This section must be called `license`.

Notes:

- T-Server also supports the `license-file` option described in the *Genesys Licensing Guide*.
- The `license` section is not applicable to Network T-Server for DTAG.
- On selected platforms, the limitation of 9999 licenses may no longer apply. Use values greater than 9999 only when instructed by Genesys Technical Support.

If you use two or more T-Servers, and they share licenses, you must configure the following options in the `license` section of the T-Servers.

num-of-licenses

Default Value: 0 or `max` (all available licenses)

Valid Values: String `max` or any integer from 0–9999

Changes Take Effect: Immediately

Specifies how many DN licenses T-Server checks out. T-Server treats a value of 0 (zero) the same as it treats `max`—that is, it checks out all available licenses.

The sum of all `num-of-licenses` values for all concurrently deployed T-Servers must not exceed the number of seat-related DN licenses (`tserver_sdn`) in the corresponding license file. The primary and backup

T-Servers share the same licenses, and therefore they need to be counted only once. T-Server checks out the number of licenses indicated by the value for this option, regardless of the number actually in use.

num-sdn-licenses

Default Value: 0 or max (all DN licenses are seat-related)

Valid Values: String max (equal to the value of num-of-licenses), or any integer from 0–9999

Changes Take Effect: Immediately

Specifies how many seat-related licenses T-Server checks out. A value of 0 (zero) means that T-Server does not grant control of seat-related DNs to any client, and it does not look for seat-related DN licenses at all.

The sum of all num-sdn-licenses values for all concurrently deployed T-Servers must not exceed the number of seat-related DN licenses (tserver_sdn) in the corresponding license file. The primary and backup T-Servers share the same licenses, and therefore they need to be counted only once. T-Server checks out the number of licenses indicated by the value for this option, regardless of the number actually in use.

-
- Notes:**
- For Network T-Servers, Genesys recommends setting this option to 0.
 - Be sure to configure in the Configuration Database all the DNs that agents use (Extensions and ACD Positions) and that T-Server should control. For further information, see Chapter 7, “DNs and Agent Logins,” [page 42](#).
-

License Checkout

[Table 38](#) shows how to determine the number of seat-related DN licenses that T-Server attempts to check out. See the examples on [page 345](#).

Table 38: License Checkout Rules

Options Settings ^a		License Checkout ^b
num-of-licenses	num-sdn-licenses	Seat-related DN licenses
max (or 0)	max	all available
max (or 0)	x	x
max (or 0)	0	0
x	max	x

Table 38: License Checkout Rules (Continued)

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

- In this table, the following conventions are used: x and y - are positive integers; max is the maximum number of licenses that T-Server can check out; min (y, x) is the lesser of the two values defined by y and x, respectively.
- The License Checkout column shows the number of licenses that T-Server attempts to check out. The actual number of licenses will depend on the licenses' availability at the time of checkout, and it is limited to 9999.

Examples

This section presents examples of option settings in the license section.

Example 1

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

Example 2

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

Example 3

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

Example 4

If...		Then...
Options Settings	License File Settings	License Checkout
num-of-licenses = max	tserver_sdn = 5000	1000 seat-related DNs
num-sdn-licenses = 1000		

agent-reservation Section

The `agent-reservation` section contains the configuration options that are used to customize the T-Server Agent Reservation feature. See “Agent Reservation” on [page 32](#) section for details on this feature.

This section must be called `agent-reservation`.

Note: The Agent Reservation functionality is currently a software-only feature that is used to coordinate multiple client applications. This feature does not apply to multiple direct or ACD-distributed calls.

collect-lower-priority-requests

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether an agent reservation request is collected, depending on its priority during the time interval specified by the `request-collection-time` configuration option. When set to `false`, during the `request-collection-time` interval T-Server collects reservation requests of the highest priority only, rejecting newly submitted requests that have a lower priority or rejecting all previously submitted requests if a request with a higher priority arrives. When set to `true` (the default), agent reservation requests are collected as they were in pre-8.x releases.

reject-subsequent-request

Default Value: `true`

Valid Values:

- | | |
|--------------------|---|
| <code>true</code> | T-Server rejects subsequent requests. |
| <code>false</code> | A subsequent request prolongs the current reservation made by the same client application for the same agent. |

Changes Take Effect: Immediately

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

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

request-collection-time

Default Value: `100 msec`

Valid Values: See “Timeout Value Format” on [page 362](#).

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 362](#).

Changes Take Effect: Immediately

Specifies the default interval for which an Agent DN is reserved. During this interval, the agent cannot be reserved again.

extrouter Section

The `extrouter` section contains the configuration options that are used to support multi-site environments with the Inter Server Call Control (ISCC) feature. The configuration options in this section of the document are grouped with related options that support the same functionality, as follows:

- [ISCC Transaction Options, page 349](#)
- [Transfer Connect Service Options, page 353](#)
- [ISCC/COF Options, page 354](#)
- [Event Propagation Options, page 356](#)
- [Number Translation Option, page 357](#)
- [GVP Integration Option, page 358](#)

This configuration section must be called `extrouter`.

For a description of the ways in which T-Server supports multi-site configurations and for an explanation of the configuration possibilities for a multi-site operation, see the “[Multi-Site Support](#)” chapter.

Note: In a multi-site environment, you must configure the `timeout`, `cast-type`, and `default-dn` options with the same value for both the primary and backup T-Servers. If you do not do this, the value specified for the backup T-Server overrides the value specified for the primary T-Server.

match-call-once

Default Value: `true`

Valid Values:

- | | |
|--------------------|--|
| <code>true</code> | ISCC does not process (match) an inbound call that has already been processed (matched). |
| <code>false</code> | ISCC processes (attempts to match) a call as many times as it arrives at an ISCC resource or multi-site-transfer target. |

Changes Take Effect: Immediately

Specifies how many times ISCC processes an inbound call when it arrives at an ISCC resource. When set to `false`, ISCC processes (attempts to match) the call even if it has already been processed.

Note: Genesys does not recommend changing the default value of the `match-call-once` option to `false` unless you have specific reasons. Setting this option to `false` may lead to excessive or inconsistent call data updates.

reconnect-tout

Default Value: `5 sec`

Valid Values: See “Timeout Value Format” on [page 362](#).

Changes Take Effect: At the next reconnection attempt

Specifies the time interval after which a remote T-Server attempts to connect to this T-Server after an unsuccessful attempt or a lost connection. The number of attempts is unlimited. At startup, T-Server immediately attempts the first connection, without this timeout.

report-connid-changes

Default Value: `false`

Valid Values:

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

Changes Take Effect: Immediately

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

use-data-from

Default Value: `current`

Valid Values:

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

Changes Take Effect: Immediately

Specifies the call from which the values for the `UserData` and `ConnID` attributes are taken for a consultation call that is routed or transferred to a remote location.

Note: For compatibility with the previous T-Server releases, you can use the values `consult`, `main`, and `consult-user-data` for this option. These are aliases for `active`, `original`, and `current`, respectively.

ISCC Transaction Options

cast-type

Default Value: `route`, `route-uu`, `reroute`, `direct-callid`, `direct-uu`, `direct-network-callid`, `direct-notoken`, `direct-digits`, `direct-ani`, `dnis-pool`, `pullback`

Valid Values: `route`, `route-uu`, `reroute`, `direct-callid`, `direct-uu`, `direct-network-callid`, `direct-notoken`, `direct-digits`, `direct-ani`, `dnis-pool`, `pullback`

Changes Take Effect: For the next request for the remote service

Specifies—using a space-, comma- or semicolon-separated list—the routing types that can be performed for this T-Server.

The valid values provide for a range of mechanisms that the ISCC feature can support with various T-Servers, in order to pass call data along with calls between locations.

Because switches of different types provide calls with different sets of information parameters, some values might not work with your T-Server. See Table 3 on [page 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-uui`, `direct-callid`, `direct-network-callid`, `direct-uui`, `direct-notoken`, `direct-digits`, and `direct-ani`.

direct-digits-key

Default Value: `CDT_Track_Num`

Valid Values: Any valid key name of a key-value pair from the `UserData` attribute

Changes Take Effect: For the next request for the remote service

Specifies the name of a key from the `UserData` attribute that contains a string of digits that are used as matching criteria for remote service requests with the `direct-digits` routing type.

Note: For compatibility with the previous T-Server releases, this configuration option has an alias value of `cdt-udata-key`.

dn-for-unexpected-calls

Default Value: No default value

Valid Values: Any DN

Changes Take Effect: Immediately

Specifies a default DN for unexpected calls arriving on an External Routing Point.

network-request-timeout

Default Value: 20 sec

Valid Values: See “Timeout Value Format” on [page 362](#).

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 362](#).

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 362](#).

Changes Take Effect: For the next request for remote service

Specifies the time interval that a T-Server at the origination location waits for a notification of routing service availability from the destination location. Counting starts when the T-Server sends a request for remote service to the destination site.

resource-allocation-modeDefault Value: `circular`

Valid Values:

- `home` T-Server takes an alphabetized (or numerically sequential) list of configured DN's 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 DN's, 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, DN's of the External Routing Point type and Access Resources with the Resource Type set to dn's) for multi-site transaction requests.

resource-load-maximumDefault Value: `0`

Valid Values: Any positive integer

Changes Take Effect: Immediately

Specifies the maximum number of ISCC routing transactions that can be concurrently processed at a single DN of the External Routing Point route type. After a number of outstanding transactions at a particular DN of the External Routing Point type reaches the specified number, T-Server considers the DN not available. Any subsequent request for this DN is queued until the number of outstanding transactions decreases. A value of `0` (zero) means that no limitation is set to the number of concurrent transactions at a single External Routing Point. In addition, the `0` value enables T-Server to perform load balancing of all incoming requests among all available External Routing Points, in order to minimize the load on each DN.

route-dn

Default Value: No default value

Valid Values: Any DN

Changes Take Effect: Immediately

Specifies the DN that serves as a Routing Point for the `route` transaction type in the multiple-to-one access mode.

timeout

Default Value: 60 sec

Valid Values: See “Timeout Value Format” on [page 362](#).

Changes Take Effect: For the next request for remote service

Specifies the time interval that the destination T-Server waits for a call routed from the origination location. Counting starts when this T-Server notifies the requesting T-Server about routing service availability. The timeout must be long enough to account for possible network delays in call arrival.

use-implicit-access-numbers

Default Value: false

Valid Values: true, false

Changes Take Effect: After T-Server is restarted

Determines whether an External Routing Point in which at least one access number is specified is eligible for use as a resource for calls coming from switches for which an access number is not specified in the External Routing Point. If this option is set to false, the External Routing Point is not eligible for use as a resource for calls coming from such switches. If this option is set to true, an implicit access number for the External Routing Point, composed of the switch access code and the DN number of the External Routing Point, will be used.

Note: If an External Routing Point does not have an access number specified, this option will not affect its use.

Transfer Connect Service Options

tcs-queue

Default Value: No default value

Valid Values: Any valid DN number

Changes Take Effect: For the next request for the remote service

Specifies the TCS DN number to which a call, processed by the TCS feature, is dialed after the originating external router obtains an access number. This option applies only if the [tcs-use](#) option is activated.

tcs-use

Default Value: never

Valid Values:

never	The TCS feature is not used.
always	The TCS feature is used for every call.
app-defined	In order to use the TCS feature for a multi-site call transfer request, a client application must add a key-value pair with a TC-type key and a nonempty string value to the UserData attribute of the request.

Changes Take Effect: Immediately

Specifies whether the Transfer Connect Service (TCS) feature is used.

Note: For compatibility with the previous T-Server releases, you can use the value `up-app-depended` for this option. This is an alias for `app-defined`.

ISCC/COF Options

cof-ci-defer-create

Default Value: 0

Valid Values: See “Timeout Value Format” on [page 362](#).

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 362](#).

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 362](#).

Changes Take Effect: For the next COF operation

Specifies the time interval during which T-Server will wait for call data requested with respect to a call originated at another site. After T-Server sends the call data request to remote T-Servers, all events related to this call will be

suspended until either the requested call data is received or the specified timeout expires. This option applies only if the `cof-feature` option is set to `true`.

cof-ci-wait-all

Default Value: `false`

Valid Values:

- `true` T-Server waits for responses from all T-Servers that might have the requested call data before updating the call data with the latest information.
- `false` T-Server updates the call data with the information received from the first positive response.

Changes Take Effect: Immediately

Specifies whether T-Server, after sending a request for matching call data, waits for responses from other T-Servers before updating the call data (such as `CallHistory`, `ConnID`, and `UserData`) for a potentially overflowed call. The waiting period is specified by the `cof-ci-req-tout` and `cof-rci-tout` options. This option applies only if the `cof-feature` option is set to `true`.

cof-feature

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Enables or disables the Inter Server Call Control/Call Overflow (ISCC/COF) feature.

cof-rci-tout

Default Value: `10 sec`

Valid Values: See “Timeout Value Format” on [page 362](#).

Changes Take Effect: For the next COF operation

Specifies the time interval that T-Server waits for call data from other T-Servers’ transactions. Counting starts when `cof-ci-req-tout` expires. This option applies only if the `cof-feature` option is set to `true`.

local-node-id

Default Value: `0`

Valid Values: `0` or any positive integer

Changes Take Effect: Immediately

This option, if enabled, checks all networked calls against the specified `NetworkNodeID` (the identity of the switch to which the call initially arrived). If the `NetworkNodeID` is the same as the value of this option, the request for call information is *not* sent. The default value of `0` disables the functionality of this

option. To establish an appropriate `NetworkNodeID`, specify a value other than the default. This option applies only if the `cof-feature` option is set to `true`.

Note: This option applies only to T-Server for Nortel Communication Server 2000/2100.

default-network-call-id-matching

Default Value: No default value

Valid Values: See the “T-Server-Specific Configuration Options” chapter for an option description for your T-Server

Changes Take Effect: Immediately

When a value for this option is specified, T-Server uses the `NetworkCallID` attribute for the ISCC/COF call matching.

To activate this feature, the `cof-feature` option must be set to `true`.

Note: SIP Server and several T-Servers support the `NetworkCallID` attribute for the ISCC/COF call matching in a way that requires setting this option to a specific value. For information about the option value that is specific for your T-Server, see the “T-Server-Specific Configuration Options” chapter of your *T-Server Deployment Guide*.

Event Propagation Options

compound-dn-representation

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies which format T-Server uses to represent a DN when reporting an `OtherDN` or `ThirdPartyDN` attribute in event propagation messages.

When set to `true`, the `<switch>:DN` (compound) format is used. This option value supports backward compatibility for pre-8.x T-Server ISCC/EPP functionality and is provided for multi-site deployments where the same DNs are configured under several switches.

When set to `false`, the DN (non-compound) format is used. This option value ensures more transparent reporting of `OtherDN` or `ThirdPartyDN` attributes and is recommended for all single-site deployments, as well as for multi-site deployments that do not have the same DNs configured under several switches. This option applies only if the `event-propagation` option is set to `list`.

Note: Local DNs are always represented in the non-compound (DN) form.

epp-tout

Default Value: 0

Valid Values: See “Timeout Value Format” on [page 362](#).

Changes Take Effect: Immediately

Specifies the time interval during which T-Server attempts to resolve race conditions that may occur in deployments that use switch partitioning or intelligent trunks. This option applies only if the [event-propagation](#) option is set to `list`.

Note: If the time interval is not long enough to account for possible network switching delays, T-Server may produce duplicated events, such as events that are propagated by the ISCC and generated locally.

event-propagation

Default Value: `list`

Valid Values:

- `list` Changes in user data and party events are propagated to remote locations through call distribution topology.
- `off` The feature is disabled. Changes in user data and party events are not propagated to remote locations.

Changes Take Effect: Immediately

Specifies whether the Event Propagation feature is enabled.

Number Translation Option

inbound-translator-<n>

Default Value: No default value

Valid Value: Any valid name

Changes Take Effect: Immediately

Specifies the name of another configuration section as the value for the `inbound-translator` option. For example,

`inbound-translator-1 = ani-translator`

where `ani-translator` is the name of the configuration that describes the translation rules for inbound numbers.

GVP Integration Option

handle-vsp

Default Value: no

Valid Values:

requests	ISCC will process and adjust requests related to this DN and containing a Location attribute before submitting them to the service provider.
events	ISCC will process and adjust each event received from the service provider in response to a request containing a Location attribute before distributing the event to T-Server clients.
all	ISCC will process and adjust both events and requests.
no	No ISCC processing of such requests and events takes place.

Changes Take Effect: Immediately

Specifies if multi-site Call Data synchronization of virtual calls or simulated call flows is performed by T-Server or is left to an external application (Service Provider) that has registered for a DN with the AddressType attribute set to VSP (Virtual Service Provider).

backup-sync Section

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

This section must be called backup-sync.

Note: These options apply only to T-Servers that support the hot standby redundancy type.

addp-remote-timeout

Default Value: 0

Valid Values: Any integer from 0–3600

Changes Take Effect: Immediately

Specifies the time interval that the redundant T-Server waits for a response from this T-Server after sending a polling signal. The default value of 0 (zero) disables the functionality of this option. To establish an appropriate timeout, specify a value other than the default. This option applies only if the [protocol](#) option is set to addp.

addp-timeout

Default Value: 0

Valid Values: Any integer from 0–3600

Changes Take Effect: Immediately

Specifies the time interval that this T-Server waits for a response from another T-Server after sending a polling signal. The default value of 0 (zero) disables the functionality of this option. To establish an appropriate timeout, specify a value other than the default. This option applies only if the [protocol](#) option is set to `addp`.

addp-trace

Default Value: off

Valid Values:

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

Changes Take Effect: Immediately

Specifies whether `addp` messages are traced in a log file, to what level the trace is performed, and in which direction. This option applies only if the [protocol](#) option is set to `addp`.

protocol

Default Value: default

Valid Values:

`default` The feature is not active.`addp` Activates the Advanced Disconnect Detection Protocol.

Changes Take Effect: When the next connection is established

Specifies the name of the method used to detect connection failures. If you specify the `addp` value, you must also specify a value for the [addp-timeout](#), [addp-remote-timeout](#), and [addp-trace](#) options.

sync-reconnect-tout

Default Value: 20 sec

Valid Values: See “Timeout Value Format” on [page 362](#).

Changes Take Effect: Immediately

Specifies the time interval after which the backup T-Server attempts to reconnect to the primary server (for a synchronized link).

call-cleanup Section

The call-cleanup section contains the configuration options that are used to control detection and cleanup of stuck calls in T-Server. For more information on stuck call handling, refer to the “Stuck Call Management” chapter in the *Framework 8.0 Management Layer User’s Guide*.

This section must be called `call-cleanup`.

cleanup-idle-tout

Default Value: 0

Valid Values: See “Timeout Value Format” on [page 362](#).

Changes Take Effect: Immediately

Specifies the time interval that T-Server waits for a call to be updated from its last update. After this time elapses, if no new events about the call are received, T-Server clears this call as a stuck call, either by querying the switch (if a CTI link provides such capabilities) or by deleting the call information from memory unconditionally. The default value of 0 disables the stuck calls cleanup.

Note: If the call-cleanup functionality is enabled in T-Server for Avaya Communication Manager, the UCID (Universal Call ID) feature must be enabled on the switch as well. This allows the UCID to be generated and passed to T-Server.

notify-idle-tout

Default Value: 0

Valid Values: See “Timeout Value Format” on [page 362](#).

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 362](#).

Changes Take Effect: Immediately

Specifies the time interval for periodic checks for stuck calls. These checks affect both notification and cleanup functionality, and are made by checking the T-Server’s own call information with call information available in the switch. For performance reasons, T-Server does not verify whether the

`notify-idle-tout` or `cleanup-idle-tout` option has expired before performing this check.

Note: Setting this option to a value of less than a few seconds can affect T-Server performance.

Examples

This section presents examples of option settings in the `call-cleanup` section.

Example 1 `cleanup-idle-tout = 0`
`notify-idle-tout = 0`
`periodic-check-tout = 10`

With these settings, T-Server will not perform any checks for stuck calls.

Example 2 `cleanup-idle-tout = 0`
`notify-idle-tout = 5 min`
`periodic-check-tout = 10 min`

With these settings, T-Server performs checks every 10 minutes and sends notifications about all calls that have been idle for at least 5 minutes.

Example 3 `cleanup-idle-tout = 20 min`
`notify-idle-tout = 5 min`
`periodic-check-tout = 10 min`

With these settings, T-Server performs checks every 10 minutes, sends notifications about all calls that have been idle for at least 5 minutes, and attempts to clean up all calls that have been idle for more than 20 minutes.

Translation Rules Section

The section name is specified by the `inbound-translator-<n>` option. It contains options that define translation rules for inbound numbers.

You can choose any name for this section, provided that it matches the value of the section. Every option in this section corresponds to a rule and must conform to the format described below. You can configure as many rules as necessary to accommodate your business needs.

rule-<n>

Default Value: No default value

Valid Value: Any valid string in the following format:

`in-pattern=<input pattern value>;out-pattern=<output pattern value>`

Changes Take Effect: Immediately

Defines a rule to be applied to an inbound number. The two parts of the option value describe the input and output patterns in the rule. When configuring the

pattern values, follow the syntax defined in “Using ABNF for Rules” on [page 88](#). See “Configuring Number Translation” on [page 95](#) for examples of these rules as well as detailed instructions for creating rules for your installation. For example, a value for this configuration option might look like this:

```
rule-01 = in-pattern=0111#CABBB*ccD; out-pattern=ABD
```

security Section

The `security` section contains the configuration options that are used to configure secure data exchange between T-Servers and other Genesys components. Refer to the *Genesys 8.x Security Deployment Guide* for complete information on the security configuration.

Timeout Value Format

This section of the document describes the values to use for those T-Server common options that set various timeouts. The current format allows you to use fractional values and various time units for timeout settings.

For timeout-related options, you can specify any value that represents a time interval, provided that it is specified in one of the following formats:

```
[[hours:]minutes:]seconds][milliseconds]
```

or

```
[hours hr][minutes min][seconds sec][milliseconds msec]
```

Where a time unit name in *italic* (such as *hours*) is to be replaced by an integer value for this time unit.

Integer values with no measuring units are still supported, for compatibility with previous releases of T-Server. When you do not specify any measuring units, the units of the default value apply. For example, if the default value equals `60 sec`, specifying the value of `30` sets the option to 30 seconds.

Example 1

The following settings result in a value of 1 second, 250 milliseconds:

```
sync-reconnect-tout = 1.25
```

```
sync-reconnect-tout = 1 sec 250 msec
```

Example 2

The following settings result in a value of 1 minute, 30 seconds:

```
timeout = 1:30
```

```
timeout = 1 min 30 sec
```

Changes from Release 8.0 to 8.1

[Table 39](#) lists the configuration options that:

- Are new or changed in the 8.1 release of T-Server
- Have been added or changed since the most recent 8.0 release of this document

If a configuration option has been replaced with another that enables the same functionality, the new option name and its location in this chapter are noted.

Table 39: Option Changes from Release 8.0 to 8.1

Option Name	Option Values	Type of Change	Details
TServer Section			
background-processing	true, false	See Details	Default value changed to true. See the option description on page 338 .

15

Configuration Options in T-Server for Alcatel A4400/OXE

This chapter describes configuration options unique to the T-Server for Alcatel A4400/OXE and includes these sections:

- [Application-Level Options, page 365](#)
- [DN-Level Options, page 408](#)
- [Changes from 8.0 to 8.1, page 411](#)

The configuration options common to all T-Servers are described in Chapter 13, “Common Configuration Options,” on [page 315](#) and Chapter 14, “T-Server Common Configuration Options,” on [page 337](#).

Application-Level Options

The configuration options specific to T-Server functionality are set in Configuration Manager, in the corresponding sections on the `Options` tab of the T-Server `Application` object.

For ease of reference, the options have been arranged in alphabetical order within their corresponding sections:

- [Mandatory Options, page 366](#)
- [TServer Section, page 366](#)
- [SwitchSpecificType Section, page 400](#)
- [Link-tcp Section, page 401](#)
- [Link-Control Section, page 401](#)
- [Lang-Map Section, page 406](#)
- [Ext-Filter Section, page 407](#)

Mandatory Options

Table 40 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 40: Mandatory Options

Option Name	Default Value	Details
T-Server Section		
link- <i>n</i> -name	No default value	See description on page 385 .
CTI-Link Section		
protocol	tcp	See description on page 401 .
hostname	No default value	See description on page 401 .
port	2555	See description on page 401 .

TServer Section

This section must be called TServer.

accept-dn-type

Default Value: +extension +position +acdqueue +routedn

Valid Values:

+/-extension	Accepts or rejects registration on DN of type Extension (AddressTypeDN)
+/-position	Accepts or rejects registration on DN of type ACD 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 (AddressTypeRoutedDN)
+/-trunk	Accepts or rejects registration on DN of type Trunk or Tie Line (AddressTypeTrunk)
+/-voicemail	Accepts or rejects registration on DN of type Voice Mail (AddressTypeVoiceChannel)
+/-data	Accepts or rejects registration on DN of type modem (AddressTypeDataChannel)

`+/-announcement` Accepts or rejects registration on DN of type Music port
(`AddressTypeAnnouncement`)

`+/-routequeue` Accepts or rejects registration on DN of type Routing Queue
(`AddressTypeRouteQueue`)

Changes Take Effect: Immediately

Defines the supported set of device types that are not configured in the Configuration Layer but that T-Server can register.

accode-data

Default Value: none

Valid Values: none, udata, ext

none	T-Server does not map switch account codes.
udata	T-Server attaches reported account codes as user data, using configured <code>Extensions</code> attribute keys such as <code>GCTI_ACCOUNT_CODE_<N></code> . T-Server then sends requests to set account codes to the switch, when such user data keys are used in client requests <code>AttachUserData</code> or <code>UpdateUserData</code> .
ext	T-Server attaches user data as extensions to all call events and does not intercept user data update requests with the reserved keys.

Changes Take Effect: Immediately

Related Feature: “Support for Account Codes” on [page 209](#)

Specifies whether T-Server has to map the switch account codes to call user data (`udata`), to extensions (`ext`) or does not map switch account codes (`none`).

Note: T-Server always uses the reserved keys sent in any call-related client request’s `Extensions` attribute, irrespective of the value of this option.

accode-name

Default Value: `AccountCode`

Valid Values: Any valid key name

Changes Take Effect: Immediately

Related Feature: “Support for Account Codes” on [page 209](#)

Specifies the data key name under which T-Server attaches the account code to the call, as either user data or extensions.

accode-privateservice

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Support for Account Codes” on [page 209](#)

Enables or disables the use of `TPrivateService` and `EventPrivateInfo` for handling the Account Code feature.

ack-on-noevt

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether client requests should be acknowledged by `EventACK` or `EventError` when requests have been acknowledged by the switch but no corresponding events have been received (if the configuration option value is set to `true`). The corresponding timeout is specified in the [agent-state-evt-tout](#) configuration option.

If the value of the configuration option is `false`, `EventError` is used.

acw-in-idle-force-ready

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Support for Emulated Agents” on [page 175](#)

Specifies whether, after timed manual wrap-up (when you have set the value of the configuration option, [timed-acw-in-idle](#), to `true`), T-Server forces the agent to the Ready state. If the value of the configuration option is set to `false`, T-Server returns the agents to the state that they were in prior to wrap-up.

acw-retain-call

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Support for Account Codes” on [page 209](#)

Instructs T-Server to retain the last agent business call while the agent is in an After-Call Work (ACW) period and allows user data updates to be added after the call has been released. This allows the account code to be attached to the previous call. When set to `true`, T-Server sends the account code in the `EventUserEvent` message, rather than the `EventAttachedDataChanged` message, to avoid confusing existing desktop applications.

Note: To enable the Standard Account Code functionality in T-Server, both this option and [acw-retain-lock](#) must be enabled.

acw-retain-lock

Default Value: 0

Valid Values: Any positive integer

Changes Take Effect: Immediately

Related Feature: “Support for Account Codes” on [page 209](#)

Specifies the period of time, in seconds, for the call to remain locked in Call Concentrator after the call has been released. When this option is enabled,

T-Server sends an `EventUserEvent` event with the key `LockCall` in `AttributeExtensions` and the value set by the option. Typically, this value should be greater than the ACW value.

Note: To enable the Standard Account Code functionality in T-Server, both this option and `acw-retain-call` must be enabled.

In addition, the `acw-retain-lock` option must be set to a value equal to, or greater than, the value set on the PBX for the Account Code timeout.

agent-emu-login-on-call

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Support for Emulated Agents” on [page 175](#)

Specifies whether the T-Server allows an emulated agent login or logout on a device where there is a call in progress.

The option can be set in the Configuration Layer in the following places in order of precedence (highest to lowest):-

1. The `TServer` section in the Annex tab of an agent.
2. The `TServer` section in the Annex tab of a device.
3. The `TServer` section of the application.

The value can also be set by using the `AgentEmuLoginOnCall` extension in the `TAgentLogin` or `TAgentLogout` requests. The value specified by the extension, where present, takes precedence over the settings configured in the Configuration Layer.

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 to 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-logout-on-unregDefault Value: `false`

Valid Values:

<code>true</code>	T-Server will log out emulated and native agents on unregister.
<code>false</code>	T-Server will not log out emulated or native agents on unregister.
<code>emu-only</code>	T-Server will log out only emulated agents on unregister.

Changes Take Effect: After agent logs out and then logs in again

Related Feature: “Support for Emulated Agents” on [page 175](#)

Specifies whether T-Server performs an automatic logout of an agent whenever their client application unregisters the DN from the T-Server. This happens whenever a client application disconnects from the T-Server.

The option can be set in the Configuration Layer in the following places in order of precedence (highest to lowest):

1. The `TServer` section in the Annex tab of the device representing the agent’s group (such as an ACD Queue).
2. The `TServer` section in the Annex tab of an agent.
3. The `TServer` section in the Annex tab of a device.
4. The `TServer` section of the Application level.

The Configuration Layer configuration setting may be overridden by adding the extension `AgentLogoutOnUnregister` to the `TAgentLogin` request.

Any subsequent self-transition `TAgentLogin` request can override the current agent association by adding the `AgentLogoutOnUnregister` key of the `Extensions` attribute with a value of `true`.

Similarly a `TRegisterAddress` request can override the current agent association by adding the `AgentLogoutOnUnregister` key of the `Extensions` attribute with a value of `true`.

agent-logout-reassocDefault Value: `false`

Valid Values:

<code>true</code>	T-Server automatically associates a new client application with the agent.
<code>false</code>	T-Server does not automatically associate a new client application with the agent.

Changes Take Effect: After agent logs out and then logs in again

Related Feature: “Support for Emulated Agents” on [page 175](#)

Specifies whether the T-Server automatically associates a new client application with the agent, when the application either:

- Registers on the agent DN, or;

- Sends a login request while the T-Server is currently waiting to log the agent out due to the previously associated client disconnecting.

Note: The new client application must have the same application name as the previously disconnected client.

agent-no-answer-action

Default Value: none

Valid Values:

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

Changes Take Effect: Immediately

Related Feature: “Support for No-Answer Supervision” on [page 191](#)

When defined in the main TServer options section, this option defines the default action that T-Server takes if a logged-in agent (real or emulated) fails to answer a call within the time defined in [agent-no-answer-timeout](#). See also the Extensions attribute key NO_ANSWER_ACTION on [page 243](#).

Note: If you define the [no-answer-action](#) configuration option on the Annex tab of an Agent ID object in Configuration Manager, that value overrides the value of agent-no-answer-action for that agent ID

agent-no-answer-overflow

Default Value: No default value

Valid Values:

none	T-Server does not attempt to overflow a call on an agent 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 drops the call.
Any valid overflow destination in a comma-separated list	T-Server returns the call to the specified destination when agent-no-answer-timeout expires.

Changes Take Effect: Immediately

Related Feature: “Support for No-Answer Supervision” on [page 191](#)

When defined in the main TServer options section, this configuration option 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 the `Extensions` attribute key `NO_ANSWER_OVERFLOW` on [page 243](#). 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 define option `no-answer-overflow` on the `Annex` tab of an `Agent Login` object in Configuration Manager, that value overrides the value of `agent-no-answer-overflow`.

agent-no-answer-timeout

Default Value: 15

Valid Value: Any integer from 0–600

Changes Take Effect: Immediately

Related Feature: “Support for No-Answer Supervision” on [page 191](#)

Defines the default time (in seconds) that T-Server waits 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`. A value of 0 (zero) disables the Agent No-Answer Supervision feature. See also the `Extensions` attribute key `NO_ANSWER_TIMEOUT` on [page 243](#).

Note: If you define the configuration option, `no-answer-timeout`, on the `Annex` tab of an `Agent ID` object in Configuration Manager, that value overrides the value of `agent-no-answer-timeout` for that agent.

Because T-Server supports emulated routing, the option `supervised-route-timeout` overrides this option 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-only-private-calls

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Support for Business-Call Handling” on [page 172](#)

Specifies whether T-Server blocks the classification of a call’s business type as `private` when there is no agent on the call.

If the value of the configuration option is set to `false`, calls with no agents present are classified with a business type of `private`, enabling No-Answer Supervision (NAS) to be applied for private calls.

If the value of the configuration option is set to `true`, calls remain classified with a business type of `unknown`.

agent-smart-monitorDefault Value: `false`Valid Values: `true`, `false`, `strict`

Changes Take Effect: Immediately

If the value of the configuration option is set to `true` reduces the number of device licenses used on the switch by monitoring only logged-in agents and extensions where no agent is logged in.

If the value of the configuration option is set to `strict` means T-Server only monitors agents (after login) configured as objects of type `Position` in the Configuration Layer.

Note: A valid value of `strict` is only effective if you have set the value of the configuration option, `agent-substitute`, to `true`. Otherwise, the valid value `strict` is treated the same as the valid value `true`.

agent-state-evt-toutDefault Value: `500`Valid Value: Any integer from `500–10000`

Changes Take Effect: Immediately

Specifies the timeout (in milliseconds) after which the action specified in the configuration option, `ack-on-noevt`, is performed on agent requests acknowledged by the switch but without corresponding events.

agent-state-trans-typeDefault Value: `acw`

Valid Values:

<code>none</code>	No transition
<code>notready</code>	Transition to <code>NotReady</code> state
<code>acw</code>	Transition to <code>WorkingAfterCall</code> state

Changes Take Effect: Immediately

Specifies the type of agent-state transition after a business call is released on a CCD agent.

agent-strict-idDefault Value: `false`Valid Values: `true`, `false`, `passwd`

Changes Take Effect: Immediately

Related Feature: “Support for Emulated Agents” on [page 175](#)

Specifies whether, for emulated agents, T-Server allows:

- Any agent ID to be used during login (`false`)
- Only agent IDs configured in Configuration Layer to be used during login (`true`)

- Only agent IDs that match an agent ID configured in Configuration Layer and that also have a matching password (passwd).

agent-substitute

Default Value: true

Valid Values: true, false

Changes Take Effect: After T-Server is restarted

Related Feature: “Agent Substitution” on [page 159](#)

A valid value of true means that T-Server generates reporting and accepts requests for CCD/RSI agents on the pro_ACD device where the agent has logged in.

agent-trans-nra-code

Default Value: 0

Valid Value: Any integer from 0–9

Changes Take Effect: Immediately

Specifies a default not-ready activation code for a forced transition to the NotReady state.

allow-20-announ

Default Value: false

Valid Value: true, false

Changes Take Effect: Immediately

If the value of the configuration option is set to true, T-Server increases the number of allowed announcements in one TApplyTreatment from 10 to 20.

auto-originate

Default Value: true

Valid Values: true, false

Changes Take Effect: Immediately

If the value of the configuration option is set to true, all calls are made in Auto-Origination mode.

Note: From release 7.0.2, auto-origination is not applicable for devices with switch-specific type 7 (PCM ports), whether configured or not.

auto-originate-enable

Default Value: false

Valid Values: true, false

Changes Take Effect: Immediately

A valid value of true enables the [auto-originate](#) configuration option and enables use of the Extensions attribute key GCTI_AUTO_ORIGINATE for TMakeCall.

auto-transfer-to-route

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

If the value of the configuration option is set to `true`, T-Server immediately transfers consultation calls made to a Routing Point after the timeout defined in [route-handover-timeout](#) expires.

backwds-compat-acw-behavior

Default Value: `false`

Valid Value: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Support for Emulated Agents” on [page 175](#)

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

If the value of the configuration option is set to `false` and 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.

If the value of the configuration option is set to `true`, 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.

bsns-call-dev-types

Default Values: `+acdq +rp +rpq +xrp`

Valid Values: A set of space separated flags.

- `+/-acdq` Turns on/off the classification of the call type as business on an ACD Queue.
- `+/-rp` Turns on/off the classification of the call type as business on a Routing Point.
- `+/-rpq` Turns on/off the classification of the call type as business on a Routing Queue.
- `+/-xrp` Turns on/off the classification of the call type as business on an External Routing Point.

Changes Take Effect: Immediately

Related Feature: “Support for Business-Call Handling” on [page 172](#)

Specifies which types of distribution devices will be exempt from default business-call handling. By default, T-Server classifies any call arriving at a distribution device (ACD Queue, Routing Point, Routing Queue, External Routing Point) as a business call. Using this option, you can disable automatic classification for calls to a particular type of distribution device. For example, if the value for this option is set to `-rp`, calls to Routing Point DN's will not be automatically classified as `business`, allowing the routing strategy to use the `BusinessCallType` Extension.

This option does not affect the application of the DN-level `bsns-call-type` option.

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: The value for this option should not be included in any PBX dialing plan, nor should any DN with a name of this value be configured in the Configuration Layer.

call-type-by-dn

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Support for the Keep-Alive Feature” on [page 196](#)

Enables or disables the setting of call type based on dialing plan analysis (when configured) and on the DN configuration in the Configuration Layer.

clean-failed-consult

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

If the value of the configuration option is set to `true`, T-Server clears consultation calls that fail for any reason, and reconnects the primary call.

clean-failed-to-pilot

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

A valid value of `true` forces T-Server to retrieve automatically any call to a blocked pilot (when `pilotInfo::transferPossible` is reported as `false`).

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.

convert-otherdn

Default Value: `+agentid +reserveddn +fwd`

Valid Values:

`+/-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).

`+/-reserveddn` Turns on/off the conversion of `otherDN` for reserved DNs.

`+/-fwd` Turns on/off conversion of `otherDN` in request `TSetCallForward`.

Changes Take Effect: Immediately

Related Feature: “Support for Smart OtherDN Handling” on [page 194](#)

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

correct-connid

Default Value: `true`

Valid Value: `true`, `false`

Changes Take Effect: Immediately

If the value of the configuration option is set to `true`, T-Server corrects the value of the wrong connection IDs provided by the application in CTI requests. A valid value of `false` disables this feature.

correct-rqidDefault Value: `true`Valid Value: `true`, `false`

Changes Take Effect: Immediately

If the value of the configuration option is set to `true`, T-Server mutates CTI requests provided by the application if the value of the request does not correspond semantically to the actual state of calls on the device. A valid value of `false` disables this feature.

Note: The value of this option must be set to `false` when using a multiline or MEA device, otherwise T-Server rejects the release request for the active call if there are two or more held calls on the device.

def-acr-eval-level

Default Value: 5

Valid Value: Any integer from 0–10

Changes Take Effect: Immediately

Specifies the default value for Extensions attribute key `GCTI_EXPERT_EVALUATION_LEVEL_ (1–10)`.

def-acr-statusDefault Value: `true`Valid Values: `true`, `false`

Changes Take Effect: Immediately

Defines the default value of Extensions attributed key `GCTI_ACR_STATUS_ (1–10)`.

default-dn-typeDefault Value: `none`

Valid Values:

<code>none</code>	T-Server assigns DN type using PBX-provided information
<code>extension</code>	T-Server uses value <code>AddressTypeDN</code>
<code>position</code>	T-Server uses value <code>AddressTypePosition</code>
<code>acdqueue</code>	T-Server uses value <code>AddressTypeQueue</code>
<code>routedn</code>	T-Server uses value <code>AddressTypeRouteDN</code>
<code>trunk</code>	T-Server uses value <code>AddressTypeTrunk</code>
<code>voicemail</code>	T-Server uses value <code>AddressTypeVoiceChannel</code>
<code>data</code>	T-Server uses value <code>AddressTypeDataChannel</code>
<code>announcement</code>	T-Server uses value <code>AddressTypeAnnouncement</code>
<code>routequeue</code>	T-Server uses value <code>AddressTypeRouteQueue</code>

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`.

dn-del-mode

Default Value: `idle`

Valid Values: See “Timeout Value Format” on [page 362](#)

<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 existed on that DN.
Timeout Value Format	T-Server applies a defined delay before unregistering the DN after the last call has left that DN. The valid value <code>idle</code> is equivalent to setting the Timeout Value to 0 (zero).

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.

Note: DNs not configured in the Configuration Layer feature cannot be used with Switch Partitioning feature. When using Switch Partitioning, all DNs must be configured in the Configuration Layer.

dn-for-undesired-calls

Default Value: No default value

Valid Values: Any valid switch DN

Changes Take Effect: Immediately

Related Feature: See “Support for the Keep-Alive Feature” on [page 196](#)

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.

emulate-login

Default Value: on-RP

Valid Values:

<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 Application level, and is available for backwards compatibility.

Changes Take Effect: Immediately

Related Feature: “Support for Emulated Agents” on [page 175](#)

Specifies whether T-Server performs an emulated agent login when the login device is configured in the Configuration Layer as a device of type `Extension`. 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 `TAgentLogin`, using the `Extensions` attribute key `EmulateLogin`.
2. In the `Agent Login` 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-stateDefault Value: `ready`

Valid Values:

<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>

Changes Take Effect: Immediately

Related Feature: “Support for Emulated Agents” on [page 175](#)

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.

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 `TAgentLogin`, using the `Extensions` attribute key `EmulateLogin`.
2. In the `Agent Login` object on the `Annex` tab.
3. In the agent login device on the `Annex` tab.

4. In the login device representing an Agent Group object during login, 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.

emu-redir-accode

Default Value: 0

Valid Values: Any string that could be used as an account code

Changes Take Effect: Immediately

Specifies the account code that is entered when an emulated redirect in progress requires one.

Note: When using no-answer supervision and the no-answer action none, T-Server drops the call if emulated redirection is required to move the call (that is, if the call was from a pilot or RSI).

emu-redir-enable

Default Value: true

Valid Values: true, false

Changes Take Effect: Immediately

Enables (if the value is set to true) or disables (if the value is set to false) support for the extended redirection feature (through a transfer).

emu-redir-handover-tout

Default Value: 0

Valid Values: 0–30, off

Changes Take Effect: Immediately

Specifies an interval (in seconds) for an originating device to complete a transfer of a consultation call to a current device, when the call is redirected to a destination by the transfer function of T-Server. If the transfer is not completed within the period specified, T-Server automatically completes the transfer to the current device.

extn-no-answer-overflow

Default Value: No default value.

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

none	T-Server does not attempt to overflow a call on an extension when <code>extn-no-answer-timeout</code> expires.
recall	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.

`release` T-Server drops the call.

Any valid overflow destination, in a comma-separated list

T-Server returns the call to the specified destination when `extn-no-answer-timeout` expires.

Changes Take Effect: Immediately

Related Feature: “Support for No-Answer Supervision” on [page 191](#)

Specifies a sequence of overflow destinations that T-Server attempts to overflow to when the time specified in the `extn-no-answer-timeout` configuration option expires.

T-Server attempts to overflow in the order specified in the list. If all overflow attempts fail, T-Server abandons overflow. See also `Extensions` attribute key `NO_ANSWER_OVERFLOW` on [page 243](#).

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 define option `no-answer-overflow` on the `Annex` tab of any individual DN of type `Extension` object in Configuration Manager, that value overrides the value of `extn-no-answer-overflow` for that `Extension` object only.

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`.

A value of 0 deactivates the no-answer supervision feature for devices of type `Extension`.

See also `Extensions` attribute key `NO_ANSWER_TIMEOUT` on [page 243](#).

Note: If you define option `no-answer-timeout` on the `Annex` tab of an `Extension` object in Configuration Manager, that value overrides the value of `extn-no-answer-timeout` for that `Extension` object.

failed-call-rls-dly

Default Value: 0

Valid Value: Any integer from 0–30

Changes Take Effect: Immediately

Defines the delay (in seconds) that T-Server applies to the execution of the recovery service in scenarios where the PBX reports event `ConnectionCleared`

with cause `Destination Not Obtainable`. This applies to regular devices only (extensions, agents, voice ports, and so on.)

headset-mode

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Controls the Headset mode of the phone set. If the value of the configuration option is set to `true`, the Headset mode is activated by default when an agent logs in. Use `Extensions` attribute key `GCTI_HEADSET_MODE` to override this value for an individual request. See also “Configuring Headset mode for agent handsets” on [page 152](#).

imm-trf-route-external

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

If the value of the configuration option is set to `true`, T-Server immediately transfers a primary call on a Routing Point after delivering the consultation call to an external destination.

Note: Use of this option affects [supervised-route-timeout](#).

inbound-bsns-calls

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Support for Business-Call Handling” on [page 172](#)

Specifies whether T-Server should consider all established inbound calls to be business calls (and consequently apply emulated wrap-up). This applies to both local and real (CCD or RSI) agents from release 7.1 of T-Server.

inherit-bsns-type

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Support for Business-Call Handling” on [page 172](#)

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

inhibit-hold-tone

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Sets a default value that controls the generation of the hold tone toward the call when transferring from an IVR device. If the value of the configuration option is set to `true`, the hold tone is inhibited. Use `Extensions` attribute key `GCTI_INHIBIT_HOLD_TONE` to override this value for an individual request.

inhibit-progress-tone

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Sets a default value that controls generation of the call progress tone toward the calling device when transferring from an IVR device. If the value of the configuration option is set to `true`, the call progress tone is inhibited. Use `Extensions` attribute key `GCTI_INHIBIT_PROGRESS_TONE` to override this value for an individual request.

internal-bsns-calls

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Support for Business-Call Handling” on [page 172](#)

Determines whether T-Server considers internal calls 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

Related Feature: “Support for Emulated Agents” on [page 175](#)

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.

link-*n*-name

Default Value: link-tcp

Valid Value: Any valid name

Changes Take Effect: Immediately

Specifies the section name containing the configuration options assigned to that link, where *n* is a consecutive number for a CTI link. You must specify a value for this option.

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

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

make-call-morph

Default Value: default

Valid Values:

conference	Initiate conference
default	Currently defined default behavior
makecall	Sends a Make Call request to the switch (no morphing)
reject	Reject the request immediately without sending it to the switch
transfer	Initiate transfer

Changes Take Effect: Immediately

Allows T-Server to send a Make Call request correction type to the switch when a device is already on a call.

max-pred-req-delay

Default Value: 3

Valid Value: Any integer from 0–10

Changes Take Effect: Immediately

Related Feature: “Support for Emulated Predictive Dialing” on [page 189](#)

Defines the maximum time (in seconds) that T-Server waits for a free dialing resource to become available before rejecting a TMakePredictiveCall request.

min-route-dly

Default Value: 0

Valid Value: Any integer from 0–1500

Changes Take Effect: Immediately

Specifies the delay (in milliseconds) after `EventRouteRequest` that T-Server applies to the first received `TRouteCall`, before rejecting all other `TRouteCalls`. After the delay, T-Server executes the `TRouteCall`. When this option is set to a nonzero value, T-Server does not retry the failed service (only one attempt is made).

Note: This option is only valid for emulated Routing Points (those configured in Configuration Manager as switch-specific type 1).

min-xfer-complete-dly

Default Value: 0

Valid Value: Any integer from 0–1000

Changes Take Place: Immediately

Specifies the length of the delay in milliseconds (initiated on receiving the `Delivered` event from the switch) applied to the execution of `TCompleteTransfer` or `TCompleteConference`.

min-xfer-init-dly

Default Value: 0

Valid Value: Any integer from 0–1000

Changes Take Place: Immediately

Specifies the length of the delay in milliseconds (initiated on receiving the `Established` event from the switch) applied to the execution of any service that will initiate a consultation call.

nas-indication

Default Value: none

Valid Values: none, ext, rsn

none	No reason or extension is provided in <code>EventReleased</code>
------	--

ext	Extensions attribute key <code>NO_ANSWER_TIMEOUT</code> is supplied in <code>EventReleased</code>
-----	---

rsn	Reason <code>NO_ANSWER_TIMEOUT</code> is supplied in <code>EventReleased</code>
-----	---

Changes Take Effect: Immediately

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

nas-private

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Place: Immediately

Related Feature: “Support for No-Answer Supervision” on [page 191](#)

Specifies whether No-Answer Supervision is enabled for 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 DNs of type Extension or Agent ID in a section called TServer. When set there, this value overrides the default value for the specific DN.

outbound-bsns-calls

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Support for Business-Call Handling” on [page 172](#)

Specifies whether T-Server should consider all established outbound calls to be business calls (and consequently apply emulated wrap-up). This applies to both local and real (CCD or RSI) agents from release 7.1 of T-Server.

override-switch-acw

Default Value: `false`

Valid Value: `true`, `false`

Changes Take Effect: Immediately

Reserved for Genesys Engineering.

participation-type

Default Value: `silent`

Valid Values: `active`, `silent`

Changes Take Effect: Immediately

Defines the type of supervisor participation during a conference created by the TSingleStepConference service. Use extension GCTI_PARTICIPATION_TYPE to override this value for an individual request.

password-separator

Default Value: No default value

Valid Value: Any nonnumeric character

Changes Take Effect: Immediately

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

pcm-port-rls-dly

Default Value: 0

Valid Value: Any integer from 0–10000

Changes Take Effect: Immediately

Defines a delay (in milliseconds) that T-Server applies before releasing a failed call on a PCM port.

posn-no-answer-overflow

Default Value: No default value

Valid Values:

<code>none</code>	T-Server does not attempt to overflow a call on a position when the timeout value specified for the <code>posn-no-answer-timeout</code> configuration option expires.
<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 drops the call.
Any valid overflow destination, in a comma-separated list	T-Server returns the call back to the specified destination when the timeout value specified for the <code>posn-no-answer-timeout</code> configuration option expires.

Changes Take Effect: Immediately

Related Feature: “Support for No-Answer Supervision” on [page 191](#)

Specifies a sequence of overflow destinations that T-Server attempts to overflow to when the timeout value specified in the `posn-no-answer-timeout` configuration option expires.

T-Server attempts to overflow in the order specified in the list. If all overflow attempts fail, T-Server abandons overflow. See also `Extensions` attribute key `NO_ANSWER_OVERFLOW` on [page 243](#).

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 define the option, `no-answer-overflow`, on the Annex tab of any individual ACD `Position` object in Configuration Manager, that value overrides the value of `posn-no-answer-overflow` for that position only.

posn-no-answer-timeout

Default Value: 15

Valid Value: Any integer from 0–600

Changes Take Effect: Immediately

Related Feature: “Support for No-Answer Supervision” on [page 191](#)

Defines the default no-answer timeout (in seconds) that T-Server applies to any device of type `ACD Position`. When the timeout ends, T-Server executes the actions defined in the `posn-no-answer-overf low` configuration option.

See also `Extensions` attribute key `NO_ANSWER_TIMEOUT` on [page 243](#).

Note: If you define the option, `no-answer-overf low`, on the Annex tab of a `ACD Position` object in Configuration Manager, that value overrides the value of `posn-no-answer-timeout` for that position.

prd-dist-call-ans-time

Default Value: 0

Valid Value: Any integer from 0–10

Changes Take Effect: Immediately

Related Feature: “Support for Emulated Predictive Dialing” on [page 189](#)

Specifies the duration of a timer (in seconds) which starts after a customer answers a predictive call. If the call has not been answered by an agent when the timer expires, T-Server abandons the call. With value 0 (zero), T-Server does not automatically abandon the call, which then rings on the agent until it is answered.

Notes:

- When set in the `TServer` section, this option defines the default value to be applied for all agents. However, you can also set this option on the Annex tab of DNs of type `Agent ID`, in a section called `TServer`. When set in this way, this value overrides the default value for the specific agent ID.
When an emulated predictive dial is made from an emulated Routing Point, and configuration options, `prd-dist-call-ans-time` and `supervised-route-timeout`, are set, the value in `prd-dist-call-ans-time` takes precedence.
- When using T-Server 8.0 with Outbound Contact Server (OCS) 7.6 or lower, this option must be set to 0 (zero).

preassign-agent-compat

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether T-Server provides backward-compatible reporting for preassigned agents. Please refer to “Preassigned and Supervisor Agents” on [page 148](#).

predictive-delay-time

Default Value: `0`

Valid Value: Any integer from `0–1000`

Changes Take Effect: From the next predictive call

Specifies the delay (in milliseconds) between initiating and completing transfer to the distribution device (Pilot or Routing Point) after a predictive call has been answered.

prioritary-transfer

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Sets a default the value that controls priority transfer when transferring from an IVR device to a pilot. If the value of the configuration option is set to `true`, all consultation calls are made in `Priority` mode without being queued in the ACD Queue. Use `Extensions` attribute key `GCTI_PRIORITY_TRANSFER` to override this value for an individual request.

real-agent-pause-time

Default Value: `0`

Valid Value: Any integer from `0–1000`

Changes Take Effect: Immediately

Related Feature: “Support for Emulated Agents” on [page 175](#)

Specifies the emulated pause time (in seconds) for CCD agents after T-Server releases a business call.

rel-cons-reconnect

Default Values: `false`

Valid Value: `true`, `false`

Changes Take Effect: After T-Server is restarted

If a request to release a consultation call is received by the client, and T-Server sends it transparently to the switch, the switch software drops the primary call as well as the consultation call.

If the configuration option is set to a value of `true`, T-Server replaces the `Release` request with a `Reconnect` request, so that the primary call is retrieved. Also if the configuration option is set to a value of `true`, there is no way to

cancel the consultation call without retrieving the primary call, and release of the primary call is forbidden.

release-alerted-calls

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

If the configuration option is set to a value of `true` enables use of Extensions attribute key `GCTI_RELEASE_WITH_BUSY_CAUSE` for `TReleaseCall` with alerting inbound T0/T2 calls.

releasing-party-report

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether T-Server reports the Extensions attribute key `ReleasingParty` in events `EventReleased` and `EventAbandoned` to indicate which party initiated the call release.

report-emul-wait-info

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Enables or disables the reporting of emulated waiting time information in Extensions attribute keys `GCTI_EMUL_WAIT_TIME` and `GCTI_EMUL_GLOB_WAIT_TIME`.

retain-call-tout

Default Value: 15

Valid Value: Any integer from 0–3600

Changes Take Effect: Immediately

Specifies the interval (in seconds) that T-Server waits before cleaning up either a released call, or a call that is connected to an unmonitored device, where the call is still in T-Server internal structures.

route-failure-alarm-high-wm

Default Value: 10

Valid Values: Positive integer for absolute value or floating point number followed by % (percent) symbol. For example—10%, 2.25%, 5E-2%.

Changes Take Effect: Immediately

Related Feature: See “Support for Failed Route Notification” on [page 198](#)

Defines the high water mark which must be reached in order for a route failure alarm to be triggered, within the period configured in the [route-failure-alarm-period](#) configuration option.

route-failure-alarm-low-wm

Default Value: 1

Valid Values: Positive integer for absolute value or floating point number followed by % (percent) symbol. For example—10%, 2.25%, 5E-2%.

Changes Take Effect: Immediately

Related Feature: See “Support for Failed Route Notification” on [page 198](#)

Defines the low water mark which must be reached, while under the route failure alarm condition, within the period configured in the [route-failure-alarm-period](#) configuration option.

route-failure-alarm-period

Default Value: 0

Valid Values: Positive integer

Changes Take Effect: Immediately

Related Feature: See “Support for Failed Route Notification” on [page 198](#)

Defines the interval (in seconds) in which the number of failed route requests is totalled, in order to determine either a possible route failure alarm or the cancelation of an alarm, based on the failed route counter reaching the relevant high or low water mark.

Note: This option also specifies the minimum time between alarm setting and alarm clearing.

route-handover-timeout

Default Value: 3

Valid Value: Any integer from 0–600

Changes Take Effect: Immediately

Specifies the interval (in seconds) for an agent to complete a transfer of a consultation call to a Routing Point when the call is routed to an external destination by the T-Server Transfer function. If the transfer is not completed within the period specified, T-Server either completes the transfer to the Routing Point automatically, or reconnects the primary call to the agent, depending on the value of the [auto-transfer-to-route](#) configuration option.

Note: Only valid for emulated Routing Points.

route-request-attempts

Default Value: 3

Valid Value: Any integer from 0–10

Changes Take Effect: Immediately

Specifies the number of times T-Server attempts to route calls before sending TEventError to T-Server clients.

Note: This option is ignored when option `min-route-dly` is set to a nonzero value.

rsi-bypass-fwd-dnd

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Specifies whether the switch will bypass (`true`), or not bypass (`false`) activated forwarding, or DND on an extension device without a real agent logged in when the switch routes a call from RSI, if the configuration option is set to `true`.

Notes:

- By setting the option `rsi-bypass-fwd-dnd` on the Annex tab of a specific RSI, it is possible to override this value for that RSI.
- Extensions attribute key `RSIBypassFwdDND` in `TRouteCall` can be used to override the value of this configuration option on a call-by-call basis.

rsi-remain-retry

Default Value: 0

Valid Value: Any integer from 0–10

Changes Take Effect: Immediately

Indicates the number of routing attempts to be made if a call cannot be successfully routed from an RSI (same as Extensions attribute key `GCTI_REMAIN_RETRY`).

rsi-reroute-auth

Default Value: 63

Valid Value: Any integer from 0–63

Changes Take Effect: Immediately

Informs the switch of the conditions that allow rerouting if the current routing attempt fails (same as Extensions attribute key `GCTI_REROUTE_AUTHORIZATION`). [Table 41](#) shows the relationship between PBX values, binary values and the T-Server option values. Reroute authorisation is equal to the superposition of corresponding values for particular cases that are allowed to initiate the Reroute request in the PBX.

Table 41: RSI Reroute Authorization

Condition	PBX Value (Bit String)	Binary Value	T-Server Option Value
Busy	1	1	1
DestNotObtainable	2	10	2
IncompatibleDest	4	100	4
NetworkCongestion	8	1000	8
ResourceNotAvailable	16	10000	16
TrunkBusy	32	100000	32
AllCases	63	111111	63

rsi-report-xfer

Default Value: true

Valid Values: true, false

Changes Take Effect: Immediately

If the value of the configuration option is set to true, T-Server reports a normal transfer scenario for calls that are transferred to an RSI Routing Point. This means that routing strategies must be written to handle such scenarios.

If the value of the configuration option is set to false, T-Server ends and restarts the routing dialog with URS when a call is transferred to an RSI Routing Point.

The call flow is different when the value of this configuration option is set to false, and is as follows:

```

EventRouteUsed (consult ConnID)
EventPartyChanged (primary ConnID, consult PreviousConnID)
EventRouteRequest (primary ConnID)

```

rsi-xfer-ftout

Default Value: 250

Valid Values: Any integer from 250–2000

Changes Take Effect: Immediately

Defines the time (in milliseconds) that T-Server waits for the PBX to complete transfer of calls to RSI Routing Points. If the PBX fails to report successful transfer completion within the configured timeout the call is reported as abandoned.

snapshot-interval

Default Value: 5

Valid Value: Any integer from 1–60

Changes Take Effect: Immediately

Specifies the interval (in seconds) between T-Server sending Snapshot Device requests to the PBX for a device with calls unknown to the system (made before startup of T-Server). T-Server continues to send snapshots until such calls are disconnected or released.

When the value of the existing snapshot-on-startup option is set, T-Server queries any new registered device (either on startup or on reconnect). If any agent/routing/extension device has a call that is unknown to the system, T-Server starts polling the device using the interval specified in option snapshot-interval.

snapshot-on-start

Default Value: false

Valid Values: true, false

Changes Take Effect: Immediately

If the value of this configuration option is set to true, T-Server attempts to recognize out-of-service and busy devices at monitor start.

Note: Set the value of this configuration option to true if you are using predictive dialing. Enabling this option slows down T-Server startup.

supervised-route

Default Value: false

Valid Value: true, false

Changes Take Effect: Immediately

Defines the default method that T-Server uses when performing supervised routing. If the value of this configuration option is set to true, T-Server uses the Transfer service. If the value of this configuration option is set to false, T-Server uses the Divert service.

Use the Extensions attribute key GCTI_SUPERVISED_ROUTE to override this option on a call-by-call basis.

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. Setting this configuration option to a

value of 0 (zero) deactivates this feature, but if the value of the `supervised-route` option is also set to true, T-Server will perform routing through Transfer.

See also `agent-no-answer-timeout`. For predictive dialing to work, you must set values greater than 0 (zero) for both this option and the `prd-dist-call-ans-time` option.

This timeout should be set to a value higher than the system latency.

Note: When set at the Application-level, this option defines the default value for all Routing Points. However, you can also set a value for this option at the DN level for a DN 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 the Extensions attribute key `SUPERVISED_ROUTE` to override the value of this configuration option on a call-by-call basis. When a value is set in the Extensions attribute, this takes precedence over the DN-level setting, followed by the Application-level setting.

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.

If a combination of extensions and option values forces T-Server to use routing through Transfer, but the timeout is set to 0 (zero), T-Server does not consider a routing as supervised, and completes transfer immediately.

supervised-transfer

Default Value: false

Valid Values: true, false

Changes Take Effect: Immediately

Sets a default value that controls supervised transfer when transferring from an IVR device to a pilot. With value true, all consultation calls are made in Supervised Transfer mode, without being queued in the ACD Queue. Use Extensions attribute key `GCTI_SUPERVISED_TRANSFER` to override this value for an individual request.

supervisor-call

Default Value: false

Valid Values: true, false

Changes Take Effect: Immediately

If the value of the configuration option is set to true, T-Server sends all consultation calls made by an agent to a CCD supervisor. Use Extensions attribute key `GCTI_SUPERVISOR_CALL` to override this value for an individual request.

supervisor-call-enable

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Setting the value of the configuration option to `true` enables the use of the Extensions attribute key `GCTI_SUPERVISOR_CALL` and `supervisor-call` option.

supervisor-step-in

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Sets a default value that controls behavior of a single-step conference call. If the value of the configuration option is set to `true`, all single-step conference calls are made in Step-In mode. Use Extensions attribute key `GCTI_SUPERVISOR_STEP_IN` to override this value for an individual request.

sync-emu-acw

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Support for Emulated Agents” on [page 175](#)

Specifies whether T-Server synchronizes emulated ACW for emulated agents. The `TAgentLogin` Extensions attribute key `SyncEmuAgentACW` overrides the value configured for this option.

sync-emu-agent

Default Value: `off`

Valid Values: `on`, `off`

Changes Take Effect: Immediately

Reserved for Genesys Engineering.

timed-acw-in-idle

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Support for Emulated Agents” on [page 175](#)

Specifies whether T-Server applies the automatic wrap-up timer (using the `wrap-up-time` parameter) when an agent sends `TAgentNotReady`. If the value of the configuration option is set to `false`, T-Server does not automatically end manual wrap-up—the agent must return manually from ACW.

unknown-bsns-calls

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Related Feature: “Support for Business-Call Handling” on [page 172](#)

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

untimed-wrap-up-value

Default Value: `1000`

Valid Value: Any nonzero positive integer

Changes Take Effect: Immediately

Related Feature: “Support for Emulated Agents” on [page 175](#)

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

unknown-xfer-merge-udata

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Place: Immediately

If the value of the configuration option is set to `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 previously unknown, normal user data inheritance mechanisms cannot be used.

use-rsi-consult

Valid Values: `true`, `false`

Default Value: `false`

Changes Take Effect: Immediately

Defines whether T-Server creates a new call on RSI with type `Consult` if private data `Supervised transfer` is present (if the value is set to `true`).

Note: The value of this option must be set to `false` if the Call Overflow feature is being used and consult calls are being made to RSI.

In order to propagate attached user data to a new call, option [unknown-xfer-merge-udata](#) must be enabled.

wait-delivered

Default Value: `60000`

Valid Values: `0-60000`

Changes Take Effect: Immediately

Specifies the length of the time in milliseconds (initiated upon receiving the `Originated` or `Network Reached` event from the switch) that T-Server waits

before proceeding to the next step in execution of emulated service that involves transfer completion.

wrap-up-threshold

Default Value: 0

Valid Values: Any positive integer

Changes Take Effect: Immediately

Related Feature: “Support for Emulated Agents” on [page 175](#)

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

wrap-up-time

Default Value: 0

Valid Value:

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

Changes Take Place: Immediately

Related Feature: “Support for Emulated Agents” on [page 175](#)

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

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 TAgentPendingACW, in Extensions attribute key WrapUpTime (applies to this agent only).
2. In TACWInIdle, in Extensions attribute key 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 TAgentLogin, in Extensions attribute key WrapUpTime (applies to this agent only).

6. In the Agent Login object, on 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.

SwitchSpecificType Section

This section must be called SwitchSpecificType.

acd-position

Default Value: 0

Valid Value: Switch-specific types for DN of type ACD Position supported by T-Server

Changes Take Effect: Immediately

Defines the switch-specific type that T-Server uses for registration of DNs of type ACD Position (AddressTypePosition) that are not configured in the Configuration Layer.

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 DNs of type Extension (AddressTypeDN) that are not configured in the Configuration Layer.

routing-point

Default Value: 0

Valid Value: Switch-specific types for DN of type Routing Point supported by T-Server

Changes Take Effect: Immediately

Defines the switch-specific type that T-Server uses for registration of DNs of type Routing Point (AddressTypeRouteDN) that are not configured in the Configuration Layer.

routing-queue

Default Value: 0

Valid Value: Switch-specific types for DN of type Routing Queue supported by T-Server

Changes Take Effect: Immediately

Defines the switch-specific type that T-Server uses for registration of DNSs of type `Routing Queue (AddressTypeRouteQueue)` that are not configured in the Configuration Layer.

Link-tcp Section

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

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: 2555

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.

protocol

Default Value: Mandatory field. No default value.

Valid Value: `tcp`

Changes Take Effect: Immediately

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

Link-Control Section

This section must be called `link-control`.

call-rq-gap

Default Value: 250

Valid Value: Any integer from 0-1000

Changes Take Place: Immediately

Related Feature: “Request Handling Enhancements” on [page 200](#)

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.

device-rq-gap

Default Value: 0

Valid Value: Any integer from 0-1000

Changes Take Place: Immediately

Related Feature: “Request Handling Enhancements” on [page 200](#)

Specifies (in milliseconds) the minimum time gap between sequential requests relating to a device. Note that this setting only affects the gap between requests in a device queue.

Note: You can also set a value for this option at a DN-level in the TServer section on the Annex tab with option `rq-gap`.

force-long-eqid

Default Value: false

Valid Values:

true	T-Server reports <code>AttributeThisTrunk</code> (and similar) in messages.
false	T-Server either provides reporting compatible with the normal edition of PBX, or automatically switches to support the extended edition of the PBX, based on the first call event from the PBX.

Changes Take Effect: At T-Server start/restart

Specifies whether T-Server supports the PBX's Long Equipment ID when T-Server is started.

As Long Equipment ID occupies more than two bytes, T-Server provides reporting for the XXL edition of PBXs. Previously, T-Server reported `Attribute[XXXX]Trunk` with the combined value of the equipment ID and equipment type, each occupying 16 bytes of integer value. T-Server reports `Attribute[XXXX]Trunk` with the combined value of the equipment ID occupying 28 bits, and the equipment type occupying the four most significant bits of integer value.

Note: In the PBX, the long equipment ID is referred to as the large equipment ID. To configure the long/large equipment ID on the PBX, do the following:

1. 0xe>cd /usr3/mao
2. 0xe>cfgUpdate
3. Option 53 "Large equipment capacity" controls this feature.
 - Set a value of 0 for short/small equipment ID.
 - Set a value of 1 for long/large equipment ID.

ha-sync-dly-lnk-conn

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: At T-Server start/restart

Related Feature: “Hot-Standby HA Synchronization” on [page 201](#)

Determines whether the backup T-Server delays sending of `EventLinkConnected` until it has been notified that T-Server synchronization has completed. If the value for the configuration option is set to `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.

link-alarm-high

Default Value: `0`

Valid Values: `0-100`

Changes Take Effect: Immediately

Related Feature: “Support for Link Bandwidth Monitoring” on [page 199](#)

Specifies the percentage of the `use-link-bandwidth` option when LMS message `LINK_ALARM_HIGH` is triggered.

A value of `0` (zero) disables the feature.

link-alarm-low

Default Value: `0`

Valid Values: `0-100`

Changes Take Effect: Immediately

Related Feature: “Support for Link Bandwidth Monitoring” on [page 199](#)

Specifies the percentage of `use-link-bandwidth` option when the LMS message `LINK_ALARM_LOW` is triggered.

max-outstanding

Default Value: `64`

Valid Value: Any integer from `8-64`

Changes Take Effect: Immediately

Related Feature: “Request Handling Enhancements” on [page 200](#)

Specifies the maximum number of outstanding sent requests awaiting the response from the link.

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. If the value of the configuration option is set to `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. If the value of the configuration option is set to `true`, clients should query the T-Server after the `EventLinkConnected`.

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: `true`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

If the value of the configuration option is set to `true`, T-Server reports `EventRegistered` for “on-demand” registration with the PBX when the procedure is completed.

If the value of the configuration option is set to `false`, T-Server reports `EventRegistered` as early as possible during the PBX registration procedure.

restart-cleanup-dly

Default Value: `10`

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 the their existence using snapshot. Value 0 (zero) means any nonverified calls are cleared up immediately after completion of link startup.

restart-cleanup-limit

Default Value: 10

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. A value of 0 zero means all the calls are deleted immediately after the link failure. See also option [restart-period](#).

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. A value of 0 (zero) means T-Server does not try to reconnect unless the link configuration is changed.

rq-conflict-check

Default Value: true

Valid Value: true, false

Changes Take Place: Immediately

Related Feature: “Request Handling Enhancements” on [page 200](#)

Specifies whether the request conflict checking feature is enabled. This feature intelligently resolves conflicting client requests.

rq-expire-tout

Default Value: 10000

Valid Value: Any integer from 1000–120000

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.

rq-gap

Default Value: 0

Valid Value: Any integer from 0–250

Changes Take Effect: Immediately

Related Feature: “Request Handling Enhancements” on [page 200](#)

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.

use-link-bandwidth

Default Value: auto

Valid Values: 0–999, auto

Changes Take Effect: Immediately

Related Feature: “Support for Link Bandwidth Monitoring” on [page 199](#)

Specifies the maximum number of requests per second throughput to be used by T-Server to calculate link alarm messages.

A value of 0 (zero) disables the feature.

Lang-Map Section

This section defines language mapping parameters.

lang-def

Default Value: 1

Valid Value: Any positive integer

Changes Take Effect: Immediately

Specifies the default language to be used for treatment services.

lang-*nn*

Default Value: No default value

Valid Value: Any string

Changes Take Effect: Immediately

Specifies the mapping between Alcatel A4400 PBX language numbers and the language name used in routing strategies created with Interaction Routing Designer.

The following values are predefined:

1	French
2	U.S. English
3	German
4	Spanish
8	Italian
19	Russian

37	French (Canadian)
41	Cantonese
42	Korean
49	Japanese
52	Mandarin
55	Vietnamese

Note: No mapping for Ukrainian is provided because the switch does not support this language.

Ext-Filter Section

The following options all specify whether the corresponding Extensions attribute key is reported in call- and device-related events. For all the options except GCTI_NOTREADY_ACTIVATION, the following values apply:

Default Value: true

Valid Values: true, false

Changes Take Effect: When the option is activated, changes take effect after T-Server receives a new event with the relevant extension from the switch.

When deactivated, reporting of the extension is suppressed only for new calls.

The following are valid option names:

```
GCTI_ACTIVE_MONITORING
GCTI_THIS_DEVICE_NAME
GCTI_OTHER_DEVICE_NAME
GCTI_WAITING_TIME
GCTI_GLOBAL_WAITING_TIME
GCTI_ESTIMATED_WAITING_TIME
GCTI_EMUL_WAIT_TIME
GCTI_EMUL_GLOB_WAIT_TIME
GCTI_SATURATION
GCTI_NETWORK_TIMESLOT
GCTI_TRANSFER_POSSIBLE
GCTI_BLOCKED
GCTI_PILOT_NUMBER
GCTI_AGENT_GROUP
GCTI_CURRENT_GUIDE_LEVEL
GCTI_REROUTED_CALL_INDICATION
GCTI_SUPERVISED_TRANSFER
GCTI_VPS_CODE
GCTI_GLOB_CID
```

```

GCTI_OLD_GLOB_CID
GCTI_LAST_REDIRECTION_DEVICE
GCTI_SECRET_ID_NN
GCTI_CC_TREATMENT_TYPE
GCTI_INFO_STR
GCTI_PREASSIGNED_AGENT
GCTI_SOURCE_ERROR_TYPE
GCTI_SOURCE_ERROR_CODE
GCTI_CSTA_CALLS_IN_QUEUE
GCTI_CSTA_CALLS_IN_FRONT
GCTI_SET_RESERVATION
GCTI_RESET_RESERVATION
GCTI_NAT_INDICATION
GCTI_BUSINESS_CALL
GCTI_PARTY_NAME
GCTI_SUB_THIS_DN
GCTI_SUB_OTHER_DN
GCTI_SUB_THIRD_DN
ReasonCode
GCTI_NOT_READY_ACTIVATION

```

DN-Level Options

You can only set the configuration options described in this section in the TServer section of the Annex tab of the relevant configuration object in Configuration Manager.

TServer Section

bsns-call-type

Default Value: none

Valid Values:

<code>business</code>	The call is classified as a business call.
<code>private</code>	The call is classified as a private call.
<code>ignore</code>	The distribution point has no effect on business call classification.

Changes Take Effect: Immediately

Related Feature: “Support for Business-Call Handling” on [page 172](#)

The option can be configured in the TServer section in the Annex tab of a distribution device. This option takes precedence over the global options

inbound-bsns-call, internal-bsns-call and outbound-bsns-call and is applied on calls arriving on the device in question.

Distribution devices include ACD Queue, Routing Point, Routing Queue and External Routing Point. This option has no effect on any other device types.

The value of the option overrides any of the following T-Server configuration options set at the Application level for the object where it has been set (depending on the type of configuration object). It may be over-ridden by the BusinessCallTypeExtensions attribute.

no-answer-action

Default Value: none

Valid Values:

none	T-Server takes no action on agents when business calls are not answered.
notready	T-Server sets agents NotReady when business calls are not answered.
logout	T-Server automatically logs out agents when business calls are not answered.

Changes Take Effect: Immediately

Related Feature: “Support for No-Answer Supervision” on [page 191](#)

This option is defined in a section called TServer on the Annex tab of any Agent ID object in Configuration Manager. The value set in the Annex tab is specific to this instance of an agent, and overrides the value for the same option set in the Options tab.

If an emulated or real PBX 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](#), option [no-answer-action](#) determines the action T-Server performs on this agent.

Note: If a call is abandoned before one of the timeouts specified for [agent-no-answer-timeout](#), [no-answer-timeout](#), or [supervised-route-timeout](#) expires (depending on which timer is applicable), T-Server performs no action on this agent.

no-answer-overflow

Default Value: No default value

Valid Values:

<code>none</code>	T-Server does not attempt to overflow a call on an agent when <code>agent-no-answer-timeout</code> expires.
<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 drops the call.
<code>default</code>	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 <code>TServer</code> section.
Any valid overflow destination, in a comma-separated list	T-Server returns the call to the specified destination when <code>agent-no-answer-timeout</code> expires.

Changes Take Effect: Immediately

Related Feature: “Support for No-Answer Supervision” on [page 191](#)

Defined in a section called `TServer` on the `Annex` tab of any of the following configuration object types in Configuration Manager:

- `AgentID`
- `Extension`
- `Position`
- `Voice Treatment Port`

The value of this option overrides any of the following T-Server configuration options set at the `Application` level for the object where it has been set (depending on configuration object type):

- `agent-no-answer-overflow` if defined for an `Agent ID` object
- `extn-no-answer-overflow` if defined for an `Extension` object or a `Voice Treatment Port` object
- `posn-no-answer-overflow` if defined for an `ACD 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.

no-answer-timeout

Default Value: 15

Valid Value: Any integer from 0–600

Changes Take Effect: Immediately

Related Feature: “Support for No-Answer Supervision” on [page 191](#)

Defined in a section called TServer on the Annex tab of any of the following types of configuration object in Configuration Manager:

- Extension
- ACD Position
- Agent Login
- Voice Treatment Port

This option 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 not-ready action.

A value of 0 deactivates no-answer supervision for this device.

The value of this option overrides any of the following T-Server configuration options set at the Application level for the object where it has been set (depending on configuration object type):

- [agent-no-answer-timeout](#) if defined for an Agent Login object
- [extn-no-answer-timeout](#) if defined for an Extension object or a Voice Treatment Port object
- [posn-no-answer-timeout](#) if defined for an ACD Position object

Changes from 8.0 to 8.1

For reference, [Table 42](#) lists the configuration options that:

- Are new or changed in the 8.1 release of T-Server
- Have been added or changed since the most recent 8.0 release of this document

If a configuration option has been replaced with another that enables the same functionality, the new option name and its location in this chapter are noted.

Table 42: Changes from 8.0 to 8.1

Option Name	Details
TServer Section	
bsns-call-dev-types	New in 8.1 See description on page 376 .

Table 42: Changes from 8.0 to 8.1 (Continued)

Option Name	Details
acw-retain-call	New since 8.0.100.07 See description on page 368 .
acw-retain-lock	New since 8.0.100.07 See description on page 368 .
routing-queue	New since 8.0.100.07 See description on page 400 .
make-call-morph	New since 8.0.002.17 See description on page 385 .
wait-delivered	New since 8.0.002.18 See description on page 398 .
link-control Section	
kpl-interval	Removed in 8.1
kpl-loss-rate	Removed in 8.1
kpl-tolerance	Removed in 8.1
DN-Level -> TServer Section	
bsns-call-type	New in 8.1 See description on page 408 .

Related Documentation Resources

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

T-Server for Alcatel 4400/OXE

- The Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at <http://genesyslab.com/support>.

Management Framework

Consult these additional resources as necessary:

- The *Framework 8.1 Deployment Guide*, which will help you configure, install, start, and stop Framework components.
- The *Framework 8.1 Configuration Manager Help*, which describes how to use Configuration Manager in either an enterprise or multi-tenant environment.
- The *Framework 8.1 Genesys Administrator Help*, which describes how to use Genesys Administrator in either an enterprise or multi-tenant environment.
- The *Framework 8.0 Configuration Options Reference Manual*, which will provide you with descriptions of configuration options for other Framework components.

Platform SDK

- The *Genesys Events and Models Reference Manual*, which contains an extensive collection of events and call models describing core interaction processing in Genesys environments.

- The *Voice Platform SDK 8.x .NET (or Java) API Reference*, which contains technical details of T-Library functions.

Genesys

- *Genesys Technical Publications Glossary*, which ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms used in this document.
- *Genesys Migration Guide*, which ships on the Genesys Documentation Library DVD, and which provides documented migration strategies for Genesys product releases. Contact Genesys Technical Support for more information.

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

- [*Genesys Supported Operating Environment Reference Manual*](#)
- [*Genesys Supported Media Interfaces Reference Manual*](#)

Consult these additional resources as necessary:

- *Genesys Hardware Sizing Guide*, which provides information about Genesys hardware sizing guidelines for the Genesys 7.x and 8.x releases.
- *Genesys Interoperability Guide*, which provides information on the compatibility of Genesys products with various Configuration Layer Environments; Interoperability of Reporting Templates and Solutions; and Gplus Adapters Interoperability.
- *Genesys Licensing Guide*, which introduces you to the concepts, terminology, and procedures relevant to the Genesys licensing system.
- *Genesys Database Sizing Estimator 8.0 Worksheets*, which provides a range of expected database sizes for various Genesys products.

For additional system-wide planning tools and information, see the release-specific listings of System Level Documents on the Genesys Technical Support website, accessible from the [system level documents by release](#) tab in the Knowledge Base Browse Documents Section.

Genesys product documentation is available on the:

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

Document Conventions

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

Document Version Number

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

81fr_ref_06-2011_v8.1.001.00

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

Screen Captures Used in This Document

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

Type Styles

[Table 43](#) describes and illustrates the type conventions that are used in this document.

Table 43: Type Styles

Type Style	Used For	Examples
Italic	<ul style="list-style-type: none"> Document titles Emphasis Definitions of (or first references to) unfamiliar terms Mathematical variables <p>Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on page 416).</p>	<p>Please consult the <i>Genesys Migration Guide</i> for more information.</p> <p>Do <i>not</i> use this value for this option.</p> <p>A <i>customary and usual</i> practice is one that is widely accepted and used within a particular industry or profession.</p> <p>The formula, $x + 1 = 7$ where x stands for . . .</p>
Monospace font (Looks like teletype or typewriter text)	<p>All programming identifiers and GUI elements. This convention includes:</p> <ul style="list-style-type: none"> The <i>names</i> of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages. The values of options. Logical arguments and command syntax. Code samples. <p>Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.</p>	<p>Select the Show variables on screen check box.</p> <p>In the Operand text box, enter your formula.</p> <p>Click OK to exit the Properties dialog box.</p> <p>T-Server distributes the error messages in EventError events.</p> <p>If you select true for the inbound-bsns-calls option, all established inbound calls on a local agent are considered business calls.</p> <p>Enter exit on the command line.</p>
Square brackets ([])	A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.	<code>smcp_server -host [/flags]</code>
Angle brackets (< >)	<p>A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise.</p> <p>Note: In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.</p>	<code>smcp_server -host <confighost></code>



Index

Symbols

[] (square brackets)	416
< > (angle brackets)	416
<key name> common log option	332

A

accept-dn-type configuration option	366
Access Code configuration	108
defined	41, 106
accode-data configuration option	367
accode-name configuration option	367
account	209
account code	209
ACD Listening button	155
acd-position configuration option	400
ack-on-noevt configuration option	368
acw-in-idle-force-ready configuration option	368
ADDP	58
addp-remote-timeout configuration option	358
addp-timeout configuration option	359
addp-trace configuration option	359
Advanced Disconnect Detection Protocol	27
agent answer supervision	243
agent features agent substitution	159
smart monitoring	158
agent login	147
Agent Login objects	42
agent no-answer supervision	191
agent reservation CCO functionality	188
defined	32
agent substitution	159
agent-emu-login-on-call configuration option	369
agent-group configuration option	369
agent-logout-on-unreg configuration option	370
agent-logout-reassoc configuration option	370
agent-no-answer-action configuration option	372
agent-no-answer-overflow configuration option	372
agent-no-answer-timeout configuration option	382
agent-only-private-calls configuration option	372
agent-reservation section configuration options	346–347
agents configuring	147
PBX	151
preassigned and supervisor	148
agent-smart-monitor configuration option	373
agent-state-evt-tout configuration option	373
agent-state-trans-type configuration option	373
agent-strict-id configuration option	397
agent-substitute configuration option	374
agent-trans-nra-code configuration option	373
alarm common log option	323

- all
 - common log option 322
- allow-20-announ
 - configuration option 374
- angle brackets 416
- ANI 71
- ani-distribution
 - configuration option 338
- annex tab
 - configuration options 193
- announcements 276
- app
 - command line parameter 119
- Application objects
 - multi-site operation 105
- Application-level options
 - configuration options 365
- audience, for document 16
- auto-originate
 - configuration option 374
- auto-originate-enable
 - configuration option 374
- auto-transfer-to-route
 - configuration option 375

B

- background-processing
 - configuration option 338
- background-timeout
 - configuration option 339
- backup servers 49
- backup-sync section
 - configuration options 358–359
 - configuring hot standby 58
- brackets
 - angle 416
 - square 416
- bsns-call-dev-types
 - configuration options 376
- bsns-call-type
 - DN-level option 408
- buffering
 - common log option 316
- business call handling 172
- business calls
 - interrupting agent wrap-up time 182
 - wrap-up time for CCD agents 180

C

- call
 - business 172
- call parking 293

- call progress detection
 - CCO functionality 183
- call unparking 293
- callback-dn
 - configuration options 376
- call-cleanup section
 - configuration options 360–361
- call-rq-gap
 - configuration option 401
- call-type-by-dn
 - configuration option 376
- camp-on 289
- cast-type
 - configuration option 70, 349
- CCO functionality
 - agent reservation 188
 - call progress detection 183
 - Voice Activity Detection 183
- CDN 77
- changes from 8.0 to 8.1
 - common configuration options 335
 - configuration options 411
 - T-Server common configuration options 363
- checking Voice Guides. 286
- check-point
 - common log option 320
- check-tenant-profile
 - configuration option 339
- clean-failed-consult
 - configuration option 377
- clean-failed-to-pilot
 - configuration option 377
- cleanup-idle-tout
 - configuration option 360
- clid-withheld-name
 - configuration option 377
- code
 - account 209
 - Code property 108, 109
- cof-ci-defer-create
 - configuration option 207, 354
- cof-ci-defer-delete
 - configuration option 207, 354
- cof-ci-req-tout
 - configuration option 86, 207, 354
- cof-ci-wait-all
 - configuration option 355
 - configuration options 207
- cof-feature
 - configuration option 207, 355
- cof-rci-tout
 - configuration option 207, 355
- collect DTMF digits 274
- collect-lower-priority-requests
 - configuration option 346
- command line parameters 119

app	119	trace	324
host	119	verbose	316
l	120	x-conn-debug-all	330
lmspath	120	x-conn-debug-api	329
nco X/Y	120	x-conn-debug-dns	329
port	119	x-conn-debug-open	328
V	120	x-conn-debug-security	329
commenting on this document	16	x-conn-debug-select	328
common configuration options	316–336	x-conn-debug-timers	328
changes from 8.0 to 8.1	335	x-conn-debug-write	328
common section	335	common options	
disable-rbac	333	common log options	316–332
enable-async-dns	335	common section	335
hangup-restart	334	mandatory options	316
heartbeat-period	333	sml section	333–335
heartbeat-period-thread-class-<n>	334	common section	
log section	316–330	common options	335
log-extended section	330–332	compatible-output-priority	
log-filter section	332	common log option	321
log-filter-data section	332–333	compound-dn-representation	
mandatory	316	configuration option	356
rebind-delay	335	Configuration Manager	
security section	333	configuring T-Server	43
setting	315	multiple ports	44
sml section	333–335	configuration option	391
suspending-wait-timeout	334	agent-strict-id	397
common log options	316–332	bsns-call-dev-types	376
<key name>	332	callback-dn	376
alarm	323	cof-ci-wait-all	207
all	322	min-route-dly	386
buffering	316	routing-point	400
check-point	320	routing-queue	400
compatible-output-priority	321	sync-emu-acw	397
debug	325	configuration options	
default-filter-type	332	accept-dn-type	366
expire	317	accode-data	367
interaction	324	accode-name	367
keep-startup-file	318	acd-position	400
level-reassign-<eventID>	330	ack-on-noevt	368
level-reassign-disable	332	acw-in-idle-force-ready	368
log section	316–330	addp-remote-timeout	358
log-extended section	330–332	addp-timeout	359
log-filter section	332	addp-trace	359
log-filter-data section	332–333	agent-emu-login-on-call	369
mandatory options	316	agent-group	369
memory	320	agent-logout-on-unreg	370
memory-storage-size	321	agent-logout-reassoc	370
message_format	318	agent-no-answer-action	372
messagefile	318	agent-no-answer-overflow	372
print-attributes	320	agent-no-answer-timeout	382
segment	317	agent-only-private-calls	372
setting	315	agent-reservation section	346–347
spool	321	agent-smart-monitor	373
standard	324	agent-state-evt-tout	373
time_convert	319	agent-state-trans-type	373
time_format	319	agent-substitute	374

agent-trans-nra-code	373
allow-20-announ	374
ani-distribution	338
auto-originate	374
auto-originate-enable	374
auto-transfer-to-route	375
background-processing	338
background-timeout	339
backup T-Server	265
backup-sync section	358–359
call-cleanup section	360–361
call-rq-gap	401
call-type-by-dn	376
cast-type	349
changes from 8.0 to 8.1	363
check-tenant-profile	339
clean-failed-consult	377
clean-failed-to-pilot	377
cleanup-idle-tout	360
clid-withheld-name	377
cof-ci-defer-create	207, 354
cof-ci-defer-delete	207, 354
cof-ci-req-tout	207, 354
cof-ci-wait-all	355
cof-feature	207, 355
cof-rci-tout	207, 355
collect-lower-priority-requests	346
common log options	316–332
common options	316–336
compound-dn-representation	356
consult-user-data	339
convert-otherdn	377
correct-connid	377
correct-rqid	378
customer-id	340
def-acr-eval-level	378
def-acr-status	378
default-dn	350
default-dn-type	378
default-network-call-id-matching	356
device-rq-gap	402
direct-digits-key	350
dn-del-mode	379
dn-for-undesired-calls	379
dn-for-unexpected-calls	351
dn-scope	100, 340
emulated-login-state	380
emulate-login	380
emu-redir-accode	381
emu-redir-enable	381
emu-redir-handover-tout	381
epp-tout	101, 357
event-propagation	357
extension	400
Ext-Filter section	407–408
extn-no-answer-overflow	381
extrouter section	347–358
failed-call-rls-dly	382
force-long-eqid	402
handle-vsp	358
ha-sync-dly-lnk-conn	403
imm-trf-route-external	383
inbound-bsns-calls	383, 391
inbound-translator-<n>	357
inherit-bsns-type	383
inhibit-hold-tone	384
inhibit-progress-tone	384
internal-bsns-calls	384
lang-def	406
Lang-Map section	406–407
lang-nn	406
legal-guard-time	180, 384
license section	343–346
link-alarm-high	403
link-alarm-low	403
local-node-id	355
log-trace-flags	341
management-port	341
mandatory options	316
match-call-once	348
max-outstanding	403
max-pred-req-delay	385
merged-user-data	341
min-xfer-complete-dly	386
min-xfer-init-dly	386
nas-indication	386
nas-private	387
network-request-timeout	351
no-answer supervision	381, 388
notify-idle-tout	360
num-of-licenses	343
num-sdn-licenses	344
outbound-bsns-calls	391
override-switch-acw	387
password-separator	387
pcm-port-rls-dly	388
periodic-check-tout	360
port	401
posn-no-answer-overflow	388
posn-no-answer-timeout	389
prd-dist-call-ans-time	389
preassign-agent-compat	390
predictive-delay-time	390
prioritary-transfer	374
propagated-call-type	100, 342
protocol	359, 401
quiet-cleanup	404
quiet-startup	404
real-agent-pause-time	390
reconnect-tout	348
reg-delay	404
register-attempts	351

- register-tout 351
- reg-silent 404
- reject-subsequent-request 347
- rel-cons-reconnect 390
- release-alerted-calls 391
- releasing-party-report 391
- report-connid-changes 348
- report-emul-wait-info 391
- request-collection-time 347
- request-tout 351
- reservation-time 347
- resource-allocation-mode 352
- resource-load-maximum 352
- restart-cleanup-dly 404
- restart-cleanup-limit 405
- restart-period 405
- retain-call-tout 391
- route-dn 352
- route-failure-alarm-high-wm 391
- route-failure-alarm-low-wm 392
- route-failure-alarm-period 392
- route-handover-timeout 392
- route-request-attempts 373
- rq-conflict-check 405
- rq-expire-tout 405
- rq-gap 406
- rsi-bypass-fwd-dnd 393
- rsi-remain-retry 393
- rsi-report-xfer 394
- rsi-reroute-auth 393
- rule-<n> 361
- security section 362
- server-id 342
- setting 337
 - common 315
- snapshot-interval 395
- snapshot-on-start 395
- supervised-route 395
- supervised-route-timeout 395
- supervisor-call-enable 397
- supervisor-step-in 397
- Switch-Specific Type section 400–401
- sync-emu-agent 397
- sync-reconnect-tout 359
- tcs-queue 353
- tcs-use 354
- timed-acw-in-idle 397
- timeout 353
- timeout value format 362
- Translation Rules section 361
- TServer section 338–343, 366–400, 401–406
- unknown-bsns-calls 398
- unknown-xfer-merge-udata 398
- untimed-wrap-up-value 398
- use-data-from 349
- use-implicit-access-numbers 353

- use-link-bandwidth 406
- user-data-limit 343
- wrap-up-threshold 399
- wrap-up-time 399
- configuring 151
 - GPA board 285
 - high availability
 - T-Server 57–59
 - multi-site operation 105–118
 - steps 105
 - T-Server 43
 - multiple ports 44
- Configuring agents 147
- Configuring extensions 146
- consult-user-data
 - configuration option 339
- conventions
 - in document 415
 - type styles 416
- convert-otherdn
 - configuration option 377
- correct-connid
 - configuration option 377
- correct-rqid
 - configuration option 378
- correlator data 204, 214, 222, 251
- CPU usage
 - monitoring load 311
- CSTA protocol errors 312
- CTI-Supported Functionality for SIP
 - Extensions 223
- customer-id
 - configuration option 340

D

- debug
 - common log option 325
- def-acr-eval-level
 - configuration option 378
- def-acr-status
 - configuration option 378
- Default Access Code
 - configuration 107
 - defined 106
- default-dn
 - configuration option 350
- default-dn-type
 - configuration option 378
- default-filter-type
 - common log option 332
- default-network-call-id-matching
 - configuration option 356
- destination location 64
- destination T-Server 70

- device-rq-gap
 - configuration option 402
- direct-ani
 - ISCC transaction type 71, 79
- direct-callid
 - ISCC transaction type 72, 79
- direct-digits
 - transaction type 79
- direct-digits-key
 - configuration option 350
- direct-network-callid
 - ISCC transaction type 72, 79, 206
- direct-notoken
 - ISCC transaction type 73, 79
- direct-uui
 - ISCC transaction type 73, 79
- disable-rbac
 - common configuration option 333
- DN objects 42
- dn-del-mode
 - configuration option 379
- dn-for-undesired-calls
 - configuration option 379
- dn-for-unexpected-calls
 - configuration option 351
- dnis-pool
 - in load-balancing mode 75
 - ISCC transaction type 66, 74, 79
- DN-level options 408
 - bsns-call-type 408
 - no-answer-action 409
 - no-answer-overflow 410
 - no-answer-timeout 411
 - TServer section 408–411
- DNs
 - configuring for multi-sites 112
- dn-scope
 - configuration option 100, 340
- document
 - audience 16
 - change history 17
 - conventions 415
 - errors, commenting on 16
 - version number 415
- DTMF digits 274

E

- emulated agents 175–182
 - legal-guard-time 180
 - wrap-up-time 399
- emulated predictive dialing 189
- emulated RPs
 - failure scenarios 167
 - routing consultation calls 168
 - routing to external destinations 167

- supervised routing to CCD pilots 169
- emulated-login-state
 - configuration option 380
- emulate-login
 - configuration options 380
- emu-redir-accode
 - configuration options 381
- emu-redir-enable
 - configuration option 381
- emu-redir-handover-tout
 - configuration option 381
- enable-async-dns
 - common configuration option 335
- epp-tout
 - configuration option 101, 357
- error messages
 - T-Server for Alcatel A4400 253
- Event Propagation
 - defined 97
- EventAttachedDataChanged 98
- event-propagation
 - configuration option 357
- expire
 - common log option 317
- extension
 - configuration option 400
- extension filtering 251
- extension no-answer supervision 192
- extensions 193
 - configuring 146
- Extensions in requests
 - withdrawal type 157
- External Call Handling 206
 - activating 206
- external routing 204
 - strategies 204
- Ext-Filter section
 - configuration options 407–408
- extrn-no-answer-overflow
 - configuration option 381
- extrouter section
 - configuration options 347–358
 - configuring for multi-site operation 106
 - configuring party events propagation 102
 - configuring the Number Translation feature 95

F

- failed-call-rls-dly
 - configuration option 382
- figures
 - hot standby redundancy 52
 - Multiple-to-Point mode 78
 - Point-to-Point mode 77
 - steps in ISCC/Call Overflow 85

font styles
 italic 416
 monospace 416
 force-long-eqid
 configuration option 402

G

Genesys Call Model 289
 GPA board 183
 configuring 285

H

HA
 See also high availability
 See hot standby
 HA configuration 49–59
 HA Proxy
 starting 126, 127
 handle-vsp
 configuration option 358
 hangup-restart
 common configuration option 334
 ha-sync-dly-lnk-conn
 configuration option 403
 heartbeat-period
 common configuration option 333
 heartbeat-period-thread-class-<n>
 common configuration option 334
 high-availability
 environment 265
 high-availability configuration 49–59
 host
 command line parameter 119
 hot standby 28, 49
 defined 29
 figure 52
 T-Server configuration 56
 hot standby HA synchronization 201

I

imm-trf-route-external
 configuration option 383
 inbound-bsns-calls
 configuration option 383, 391
 inbound-translator-<n>
 configuration option 357
 inherit-bsns-type
 configuration option 383
 inhibit-hold-tone
 configuration option 384

inhibit-progress-tone
 configuration option 384
 integrating routing points in the CCD 169
 intended audience 16
 Inter Server Call Control 64–83, 204
 Inter Server Call Control/Call Overflow 83–87
 interaction
 common log option 324
 internal-bsns-calls
 configuration option 384
 ISCC 204
 destination T-Server 70
 origination T-Server 70
 ISCC strategies 204
 ISCC transaction types 65, 70
 direct-ani 71, 79
 direct-callid 72, 79
 direct-digits 79
 direct-network-callid 72, 79, 206
 direct-notoken 73, 79
 direct-uui 73, 79
 dnis-pool 74, 79
 in load-balancing mode 75
 pullback 75, 79
 reroute 76, 79
 route 77, 79
 route-uui 78
 supported 79
 ISCC/Call Overflow 206
 ISCC/COF
 supported 84
 iscc-xaction-type 65
 italics 416

K

keep-startup-file
 common log option 318
 known limitations 263

L

l
 command line parameter 120
 lang-def
 configuration option 406
 Lang-Map section
 configuration options 406–407
 lang-nn
 configuration option 406
 language mapping 278
 languages
 treatment 278
 legal-guard-time
 configuration option 180, 384

- emulated agents 180
- level-reassign-<eventID>
common log option 330
- level-reassign-disable
common log option 332
- license section
configuration options 343–346
- link-alarm-high
configuration option 403
- link-alarm-low
configuration option 403
- Link-Control section
configuration options 401–406
- Link-tcp Section
configuration options 401
- LMS messages
messages, LMS 199
- lmspath
command line parameter 120
- local-node-id
configuration option 355
- location parameter 64
- log configuration options 316–322
- log section
common log options 316–330
- log-extended section
common log options 330–332
- log-filter section
common log options 332
- log-filter-data section
common log options 332–333
- log-trace-flags
configuration option 341

M

- magic ID 252
- Management Layer 40
- management-port
configuration option 341
- Mandatory Options
configuration options 366
- mandatory options
configuration options 338
- match-call-once
configuration option 348
- max-outstanding
configuration option 403
- max-pred-req-delay
configuration option 385
- memory
common log option 320
- memory-storage-size
common log option 321
- merged-user-data
configuration option 341

- message_format
common log option 318
- messagefile
common log option 318
- min-route-dly
configuration options 386
- min-xfer-complete-dly
configuration option 386
- min-xfer-init-dly
configuration option 386
- monospace font 416
- Multiple-to-One mode 77
- Multiple-to-Point mode 77, 78
- mutual aid queue 170

N

- nas-indication
configuration option 386
- nas-private
configuration option 387
- NAT/C feature 95
- nco X/Y
command line parameter 120
- network attended transfer/conference 95
- Network issues 312
- network objects 40
- network-request-timeout
configuration option 351
- no-answer supervision 191, 381, 388
 - agents 191
 - device-specific overrides 193
 - extensions 192
 - overrides for individual calls 193
 - positions 192
- no-answer-action
DN-level option 409
- no-answer-overflow
DN-level option 410
- no-answer-timeout
DN-level option 411
- notify-idle-tout
configuration option 360
- Number Translation feature 87–95
- number translation rules 88
- num-of-licenses
configuration option 343
- num-sdn-licenses
configuration option 344

O

- objects
 - Agent Logins 42
 - DNs 42

network	40
Switches	41
Switching Offices	41
One-to-One mode	77
origination location	64
origination T-Server	70
outbound-bsns-calls	
configuration option	391
override-switch-acw	
configuration option	387

P

park/unpark	293
password-separator	
configuration option	387
pcm-port-rls-dly	
configuration option	388
periodic-check-tout	
configuration option	360
play announcements	276
Point-to-Point mode	77
port	
command line parameter	119
configuration option	401
position no-answer supervision	192
posn-no-answer-overflow	
configuration option	388
posn-no-answer-timeout	
configuration option	389
prd-dist-call-ans-time	
configuration option	389
preassign-agent-compat	
configuration option	390
predictive dialing	189, 214
call distribution time	187
configuring	
initiating and transferring calls	175
configuring OCS	188
predictive-delay-time	
configuration option	390
predictive-time-delay	186
primary servers	49
print-attributes	
common log option	320
prioritary-transfer	
configuration option	374
private services and events	227
propagated-call-type	
configuration option	100, 342
protocol	
configuration option	359, 401
pullback	
ISCC transaction type	75, 79

Q

quiet-cleanup	
configuration option	404
quiet-startup	
configuration option	404

R

real-agent-pause-time	
configuration option	390
reasons keys	253
rebind-delay	
common configuration option	335
reconnect-tout	
configuration option	348
recordable Voice Guides	285
redundancy	
hot standby	28, 49
warm standby	28, 49
redundancy types	53, 54, 56
hot standby	29
reg-delay	
configuration option	404
register-attempts	
configuration option	351
register-tout	
configuration option	351
reg-silent	
configuration option	404
reject-subsequent-request	
configuration option	347
rel-cons-reconnect	
configuration option	390
release-alerted-calls	
configuration option	391
releasing-party-report	391
report-connid-changes	
configuration option	348
report-emul-wait-info	
configuration option	391
configuration options	391
request-collection-time	
configuration option	347
request-tout	
configuration option	66, 351
reroute	
ISCC transaction type	76, 79
reservation-time	
configuration option	347
resource-allocation-mode	
configuration option	352
resource-load-maximum	
configuration option	352
restart-cleanup-dly	
configuration option	404

- restart-cleanup-limit
 - configuration option 405
- restart-period
 - configuration option 405
- retain-call-tout
 - configuration option 391
- route
 - ISCC transaction type 66, 77, 79, 112
- route-dn
 - configuration option 352
- route-failure-alarm-high-wm
 - configuration option 391
- route-failure-alarm-low-wm
 - configuration option 392
- route-failure-alarm-period
 - configuration option 392
- route-handover-timeout
 - configuration option 392
- route-request-attempts
 - configuration option 373
- route-uui
 - ISCC transaction type 78
- routing
 - Inter Server Call Control 70–83
 - virtual routing point 144
- routing failure 167
- routing scenarios
 - configuration options
 - consultation calls 168
 - external destinations 167
- Routing Services Interface 267
- routing-point
 - configuration options 400
- routing-queue
 - configuration options 400
- rq-conflict-check
 - configuration option 405
- rq-expire-tout
 - configuration option 405
- rq-gap
 - configuration option 406
- RSI 267
 - configuration options 268
 - in Configuration Layer 269
 - reroute 284
 - treatments
 - busy 282
 - cancel call 282, 270, 274
 - IVR 283
 - music 281
 - play announcement 276, 281
 - ringback 282
 - silence 282
 - Voice Guides 285
- RSI routing 216

- rsi-bypass-fwd-dnd
 - configuration option 393
- rsi-remain-retry 284
 - configuration option 393
- rsi-report-xfer
 - configuration option 394
- rsi-reroute-auth 284
 - configuration option 393
- rule-<n>
 - configuration option 361
- run.bat 123
- run.sh 122

S

- security section
 - common configuration options 333, 362
- segment
 - common log option 317
- server-id
 - configuration option 342
- setting configuration options
 - common 315
- setting the DN properties
 - T-Server for Alcatel A4400 141
- SIP extensions 223
- smart monitoring 158
- sml section
 - common options 333–335
- snapshot-interval
 - configuration option 395
- snapshot-on-start
 - configuration option 395
- specific call models 289
- spool
 - common log option 321
- square brackets 416
- standard
 - common log option 324
- starting
 - HA Proxy 126
 - T-Server 127
- supervised routing
 - to CCD pilots 169
- supervised-route
 - configuration option 395
- supervised-route-timeout
 - configuration option 395
- supervisor monitoring 155, 215
- supervisor-call-enable
 - configuration option 397
- supervisor-step-in
 - configuration option 397
- supported agent work modes
 - supported functionality 226

- supported functionality
 - supported agent work modes 226
- supported T-Server features
 - T-Server for Alcatel A4400 171
- suspending-wait-timeout
 - common configuration option 334
- Switch objects 41
 - multi-site operation 105
- switch partitioning
 - defined. 100
 - T-Server support. 101
- switch/CTI environments 141
- Switching Office objects 41
 - multi-site operation 106, 107, 108, 112
- Switch-Specific Configuration
 - T-Server for Alcatel A4400 135
- Switch-Specific Type section
 - configuration options 400–401
- sync-emu-acw
 - configuration option 397
- sync-emu-agent
 - configuration option 397
- sync-reconnect-tout
 - configuration option 359

T

- Target ISCC
 - Access Code configuration 109
 - Default Access Code configuration 108
- tcs-queue
 - configuration option 353
- tcs-use
 - configuration option 354
- time_convert
 - common log option 319
- time_format
 - common log option 319
- timed-acw-in-idle
 - configuration option 397
- timeout
 - configuration option 66, 353
- timeout value format
 - configuration options 362
- timer issues 313
- TInitiateConference 64
- TInitiateTransfer 64
- T-Library functionality
 - supported 212
- TMakeCall 64
- TMuteTransfer 64
- trace
 - common log option 324
- transaction types (ISCC) 65, 70
 - supported 79
- transfer connect service 82

- Translation Rules section
 - configuration option 361
- treatment languages 278
- troubleshooting 309
- TRouteCall 64, 144
- trunk lines 77
- T-Server
 - configuring Application objects 43
 - for multi-sites 105
 - configuring redundancy 54
 - HA. 56
 - high availability 56
 - hot standby 56
 - multi-site operation 105–118
 - redundancy 53, 54, 56
 - starting 127, 128
 - using Configuration Manager 43
 - multiple ports 44
 - warm standby 54
- T-Server for Alcatel A4400
 - Application-level options
 - configuration options 365
 - changes from 8.0 to 8.1
 - configuration options 411
 - DN-level options 408
 - error messages 253
 - Ext-Filter Section
 - configuration options 407
 - Lang-Map Section
 - configuration options 406
 - Link-Control Section
 - configuration options 401
 - Link-tcp Section
 - configuration options 401
 - Mandatory Options
 - configuration options 366
 - supported functionality 171
 - Switch-Specific Configuration 135
 - T-Library functionality 212
 - Use of the Extensions Attribute 231
 - User Data Keys 251, 253
- TServer section
 - configuration options 338–343, 366–400
 - DN-level options 408–411
- TSingleStepTransfer 64
- TXRouteType 65
- type styles
 - conventions 416
 - italic 416
 - monospace 416
- typographical styles 415, 416

U**UNIX**

installing T-Server	45
starting applications	123
starting HA Proxy	127
starting T-Server	128
starting with run.sh	122
unknown-bsns-calls	
configuration option	398
unknown-xfer-merge-udata	
configuration option	398
untimed-wrap-up-value	
configuration option	398
Use of the Extensions Attribute	
T-Server for Alcatel A4400	231
use-data-from	
configuration option	349
use-implicit-access-numbers	
configuration option	353
use-link-bandwidth	
configuration option	406
User Data Keys	
T-Server for Alcatel A4400	251, 253
user data keys	251
user data propagation	98
user-data-limit	
configuration option	343

V**V**

command line parameters	120
VDN	77
verbose	
common log option	316
version numbering, document	415
virtual devices	
configuration layer	166
PBX	165
virtual Z devices	216
Voice Activity Detection	
CCO functionality	183
Voice Guide	
status	286
Voice Guides	276, 285

W

warm standby	28, 49
figure	50
T-Server configuration	54
Windows	
installing T-Server	46
starting applications	123

starting HA Proxy	127
starting T-Server	128
starting with run.bat	123
withdrawal type	
Extensions in requests	157
wrap-up-threshold	
configuration option	399
wrap-up-time	
configuration option	399

X

x-conn-debug-all	
common log option	330
x-conn-debug-api	
common log option	329
x-conn-debug-dns	
common log option	329
x-conn-debug-open	
common log option	328
x-conn-debug-security	
common log option	329
x-conn-debug-select	
common log option	328
x-conn-debug-timers	
common log option	328
x-conn-debug-write	
common log option	328