Framework 8.1

T-Server for Nortel Communication Server 1000 with SCCS/MLS

Deployment Guide

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 13. 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_ncs1000_03-2012_v8.1.001.02

Table of Contents

List of Procedures

Preface

Welcome to the *Framework 8.1 T-Server for Nortel Communication Server 1000 with SCCS/MLS 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 Nortel Communication Server 1000 with SCCS/MLS Deployment Guide. 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.

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:

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on page 251.

About T-Server for Nortel Communication Server 1000 with SCCS/MLS

T-Server is the Genesys software component that provides an interface between your telephony hardware and the rest of the Genesys software components in your enterprise. It translates and keeps track of events and

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

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

- T-Server for Nortel Meridian 1
- T-Server for Nortel Symposium Call Center

The current name, *T-Server for Nortel Communication Server 1000 with SCCS/MLS*, reflects Genesys's decision to address the Meridian 1 and Symposium functionality in one T-Server.

Intended Audience

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

- If you are new to T-Server, read the *Framework 8.1 Deployment Guide* and the Release Note, and then read all of the sections of this document that apply to your software and its accompanying components. Refer back to the *Framework 8.1 Deployment Guide* as needed.
- If you are an experienced T-Server user—someone with computer expertise, who is used to installing, configuring, testing, or maintaining Genesys software—you may find it more time efficient to go to the Index to see what is new or different in T-Server release 8.1. If you take that approach, please also read Release Notes and refer to other related resources, such as the *Genesys Events and Models Reference Manual* and *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 Nortel Communication Server 1000 with SCCS/MLS Deployment Guide* has been updated with the following:

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

Genesys

Part

1 T-Server Deployment

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

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

- Chapter 1, "T-Server Fundamentals," on page 17, describes T-Server, its place in the Framework 8 architecture, T-Server redundancy, and multi-site issues. It stops short of providing configuration and installation information.
- Chapter 2, "T-Server General Deployment," on page 31, presents configuration and installation procedures for all T-Servers.
- Chapter 3, "High-Availability Deployment," on page 45, addresses high availability (HA).
- Chapter 4, "Multi-Site Support," on page 57, details the variations available for T-Server implementations across geographical locations.
- Chapter 5, "Starting and Stopping T-Server Components," on page 115, describes how, and in what order, to start up T-Server among other Framework components. It also provides possible stopping commands.

New for All T-Servers in 8.1

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

- T-Server no longer connects to applications that have disabled status in the configuration environment.
- The default value of the `background-processing` configuration option has been changed to `true`. See "background-processing" on page 234 for details.

- T-Server now supports the Unresponsive Process Detection feature. The following configuration options enable this feature:
  - "heartbeat-period" on page 229
  - "hangup-restart" on page 230

  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 Acresso FLEXNet Publisher v11.9 license manager.

Notes:
- Configuration option changes common to all T-Servers are described in "Changes from Release 8.0 to 8.1" on page 259.
- For information about the new features that are available in your T-Server in the initial 8.1 release, see Part Two of this document.

Genesys

Chapter

1 T-Server Fundamentals

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

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

- If you are an experienced user of T-Server, start with "New for All T-Servers in 8.1" on page 15, and then move to the chapters comprising Part Two of this document, where specific information about your T-Server is available.
- If you are new to T-Server, begin with "Learning About T-Server." Once you have read through that and subsequent sections, you are ready for the other chapters in Part One that go into detail about T-Server configuration and installation.

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

Learning About T-Server

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

Framework and Media Layer Architecture

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

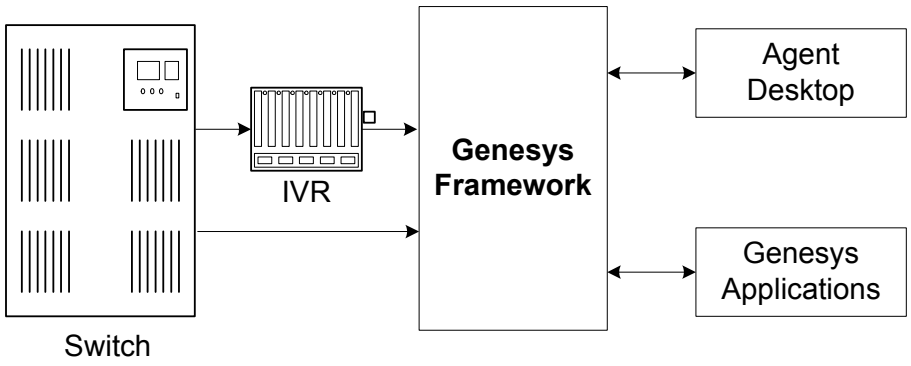

Figure 1: Framework in a Genesys Solution

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

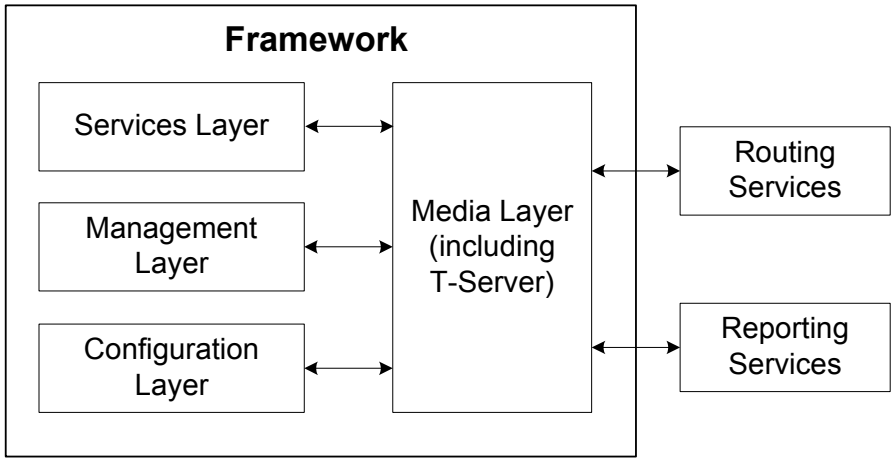

Figure 2: The Media Layer in the Framework Architecture

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

Figure 3 presents the generalized architecture of the Media Layer.

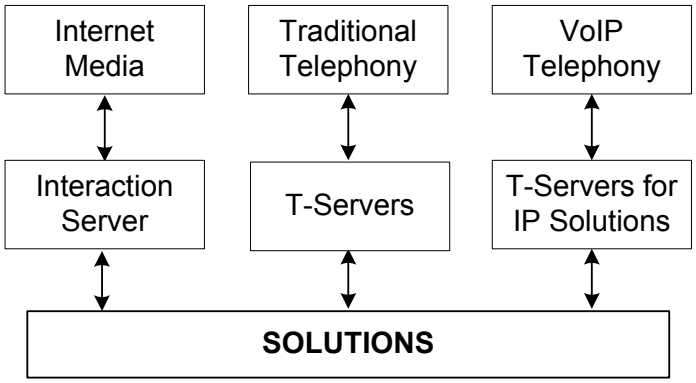

Figure 3: Media Layer Architecture

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

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

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

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

T-Server Requests and Events

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

Details of T-Server Functionality

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

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

Bridging

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

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

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

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

Messaging

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

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

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

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

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

Interaction Tracking

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

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

Difference and Likeness Across T-Servers

Although Figure 3 on page 19 (and other figures) depicts T-Server that works with telephony systems as a single product, this is a simplification. Because

almost every traditional telephony device has its own characteristics and communication protocols, Genesys makes different T-Servers for different telephony systems. (That means your T-Server will not work with another switch.) Thus, all T-Servers play a common role in the architecture, but their specific features differ from implementation to implementation, based on the media device in use.

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

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

T-Server Functional Steps During a Sample Call

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

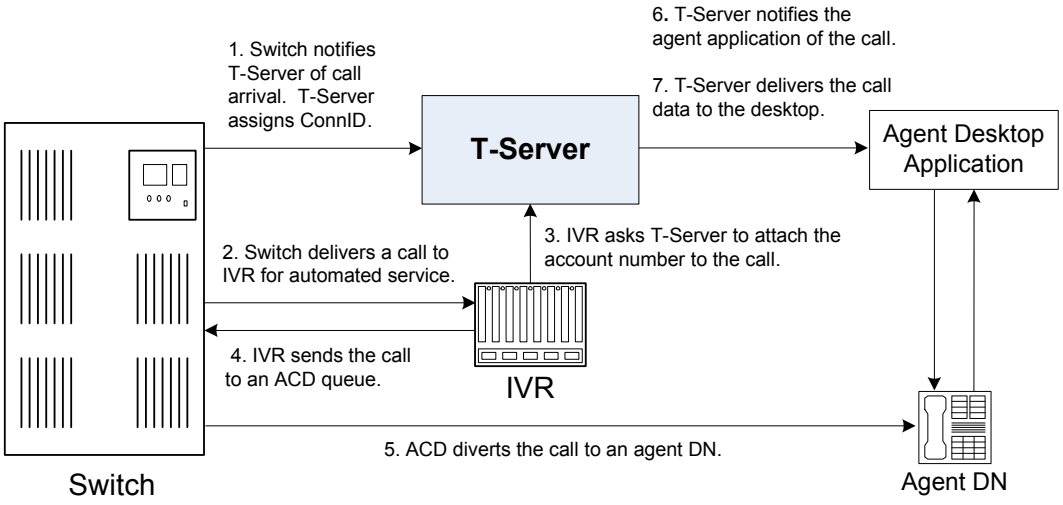

Figure 4: Functional T-Server Steps

Step 1

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

Step 2

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

Step 3

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

Step 4

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

Step 5

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

Step 6

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

Step 7

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

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

Advanced Disconnect Detection Protocol

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

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

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

ADDP applies only to connections between Genesys software components.

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

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

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

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

Redundant T-Servers

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

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

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

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

Support for Hot Standby Redundancy in Various T-Servers

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

Table 1 only summarizes `hot standby` redundancy support in various T-Servers. For detailed, up-to-date information on the subject, see the *Genesys Supported Media Interfaces Reference Manual* 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 SCAI14 or above with call-progress and noncontroller-released messages enabled. See the switch-specific information in Part 2 of this *Deployment Guide* for additional information on HA configurations.

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

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

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

Multi-Site Support

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

For instructions on installing and configuring a multi-site environment, including information on the Inter Server Call Control (ISCC) features, please see Chapter 4, "Multi-Site Support," on page 57.

Agent Reservation

T-Server provides support for clients to invoke the agent reservation function, `TReserveAgent()`. This function allows a server application that is a client of T-Server to reserve a DN along with an agent, a `Place`, or both, so that no other T-Server client can route calls to it during a specified reservation interval. Alternatively, when clients use the ISCC feature (see "ISCC Call Data Transfer Service" on page 59), they can use an agent reservation embedded in an ISCC request. (To do so, clients have to specify a certain `Extensions` attribute in an ISCC request when initiating an ISCC transaction. See page 66 for the list of ISCC requests.)

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

In addition to invoking the `TReserveAgent` function, you can customize the Agent Reservation feature by configuring options in the T-Server `Application` object. See"agent-reservation Section" on page 242 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 242).

Client Connections

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

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

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

Next Steps

Now that you have gained a general understanding of the roles and features available with T-Servers, you are ready to learn how T-Servers are installed and configured. That information is presented in the next few chapters of this *Deployment Guide.* So unless you are already familiar with T-Server deployment and operation procedures, continue with Chapter 2, "T-Server General Deployment," on page 31. Otherwise, you may want to jump to Part Two of this *Deployment Guide*, where you will find information about your specific T-Server.

Genesys

Chapter

2 T-Server General Deployment

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

This chapter contains these sections:

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

Prerequisites

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

Software Requirements

Framework Components

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

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

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

Media Layer and LCA

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

Supported Platforms

Refer to the *Genesys Supported Operating Environment Reference 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 12, "T-Server Common Configuration Options," on page 233. *T-Server-specific* configuration options are described in a separate chapter. T-Server also supports unified Genesys log options, as described in the "Common Configuration Options" chapter.

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

Deployment Sequence

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

Task Summary: T-Server Deployment Sequence

Objective	Related Procedures and Actions
1. Deploy Configuration Layer objects and ensure Configuration Manager is running.	See the *Framework 8.1 Deployment Guide* for details.
2. Deploy Network objects (such as Host objects).	See the *Framework 8.1 Deployment Guide* for details.
3. Deploy the Management Layer.	See the *Framework 8.1 Deployment Guide* for details.
4. Test your configuration and installation.	See Chapter 5, "Starting and Stopping T-Server Components," on page 115.

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

Deployment of T-Server

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

Configuration of Telephony Objects

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

Recommendations

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

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

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

Switching Offices

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

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

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

Switches

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

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

   See Chapter 4, “Multi-Site Support,” on page 57, for step-by-step instructions.

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

DNs and Agent Logins

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

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

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

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

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

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

Multi-Site Operations

See the section, “Configuring Multi-Site Support” on page 100, for information on setting up DNs for multi-site operations.

Configuration of T-Server

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

Recommendations

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

Procedure: Configuring T-Server

Start of procedure

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

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

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

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

5. In a multi-site environment, you must complete additional T-Server configuration steps to support multi-site operations; see Chapter 4, "Multi-Site Support," on page 57.

End of procedure

Next Steps

- See "Installation of T-Server" on page 40.

Procedure: Configuring multiple ports

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

Start of procedure

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

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

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

End of procedure

Installation of T-Server

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

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

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

Procedure: Installing T-Server on UNIX

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

Start of procedure

1. In the directory to which the T-Server installation package was copied, locate a shell script called install.sh.
2. Run this script from the command prompt by typing sh and the file name. For example: sh install.sh.
3. When prompted, confirm the host name of the computer on which T-Server is to be installed.
4. When prompted, specify the host and port of Configuration Server.
5. When prompted, enter the user name and password to access Configuration Server.
6. When prompted, select the T-Server application you configured in "Configuring T-Server" on page 39 from the list of applications.
7. Specify the destination directory into which T-Server is to be installed, with the full path to it.
8. If the target installation directory has files in it, do one of the following:
   - Type 1 to back up all the files in the directory (recommended).
   - Type 2 to overwrite only the files in this installation package. Use this option only if the installation being upgraded operates properly.
   - Type 3 to erase all files in this directory before continuing with the installation.

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

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

End of procedure

Next Steps

- To verify manual installation, go to "Verifying the installation of T-Server" on page 43.
- To test your configuration and installation, go to Chapter 5, "Starting and Stopping T-Server Components," on page 115, and try it out.
- To configure and install redundant T-Servers, see Chapter 3, "High-Availability Deployment," on page 45.
- To install T-Servers for a multi-site environment, proceed to Chapter 4, "Multi-Site Support," on page 57.

Procedure: Installing T-Server on Windows

Start of procedure

1. In the directory to which the T-Server installation package was copied, locate and double-click `Setup.exe` to start the installation.
2. When prompted, specify the connection parameters to the Configuration Server associated with this T-Server.
3. When prompted, select the T-Server `Application` you configured in "Configuring T-Server" on page 39 from the list of applications.
4. Specify the license information that T-Server is to use: either the full path to, and the name of, the license file, or the license server parameters.
5. Specify the destination directory into which T-Server is to be installed.
6. Click `Install` to begin the installation.
7. Click `Finish` to complete the installation.

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

End of procedure

Next Steps

- To verify manual installation, go to "Verifying the installation of T-Server" on page 43.

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

Procedure: Verifying the installation of T-Server

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

Prerequisites

- Procedure: Installing T-Server on UNIX, on page 41
- Procedure: Installing T-Server on Windows, on page 42

Start of procedure

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

End of procedure

Next Steps

At this point, you have configured and installed T-Server using Configuration Manager. If you want to test your configuration and installation, go to Chapter 5, "Starting and Stopping T-Server Components," on page 115, and try it out. Otherwise, if you want to configure and install redundant T-Servers, see Chapter 3, "High-Availability Deployment," on page 45. If you want to install T-Servers for a multi-site environment, proceed to Chapter 4, "Multi-Site Support," on page 57.

Genesys

Chapter

3 High-Availability Deployment

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

Every switch/T-Server combination offers different high-availability options. The Framework Management Layer currently supports two types of redundant configurations: `warm standby` and `hot standby`. All T-Servers offer the `warm standby` redundancy type and, starting with release 7.1, the `hot standby` redundancy type is implemented in T-Servers for most types of switches. Some T-Servers support a switch's ability to provide two CTI links to two T-Servers or even one CTI link to two T-Servers. Other T-Servers require Genesys's HA Proxy in order to support the `hot standby` redundancy type. See Table 1 on page 25 and the T-Server-specific information later in this document for details on your T-Server.

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

Warm Standby Redundancy Type

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

Warm Standby Redundancy Architecture

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

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

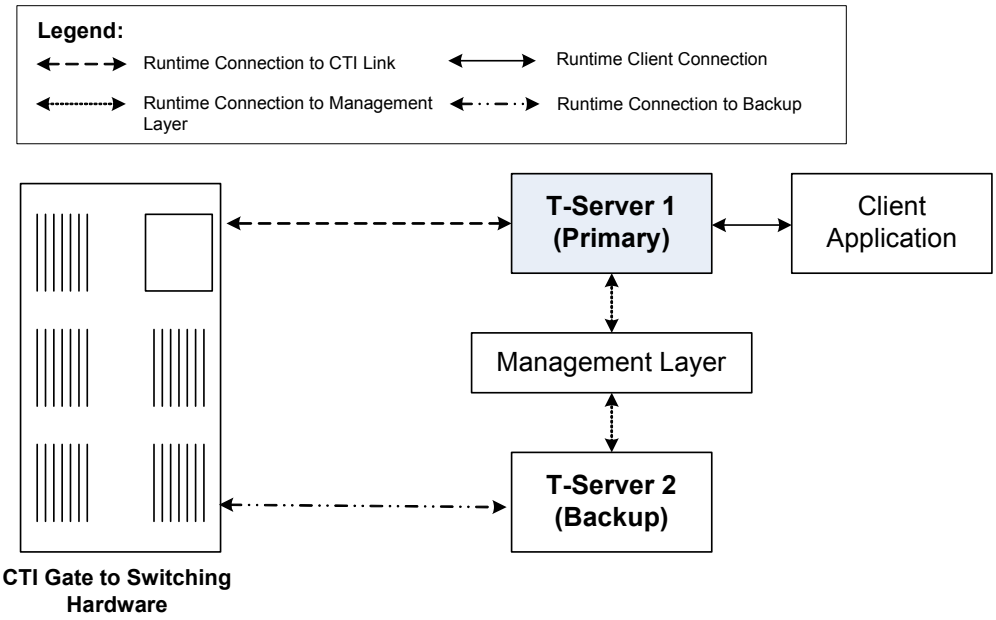

Figure 5: Warm Standby Redundancy Architecture

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

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

Hot Standby Redundancy Type

Genesys uses the expression *hot standby* to describe the redundancy type in which a backup server application remains initialized, clients connect to both the primary and backup servers at startup, and the backup server data is synchronized from the primary server. Data synchronization and existing client connections to the backup guarantee higher availability of a component. (See Figure 6 on page 48.)

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

Hot Standby Redundancy Architecture

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

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

The details of `hot standby` redundancy implementation between T-Servers and their switches vary depending on switch support for multiple CTI links. If your T-Server supports `hot standby` (see Table 1 on page 25), refer to Part Two for detailed information on the available `hot standby` schema.

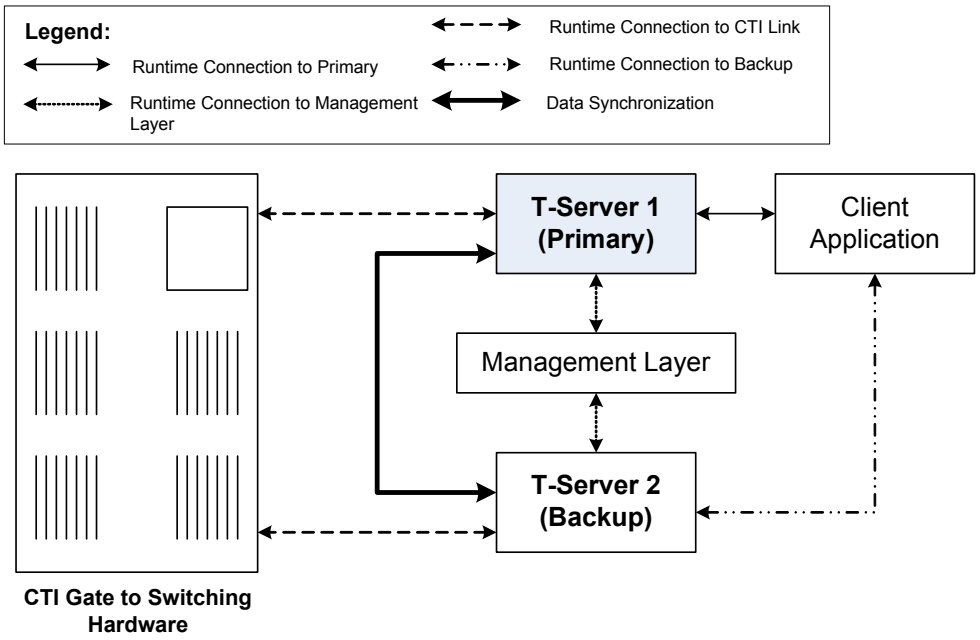

Figure 6: Hot Standby Redundancy Architecture

Benefits of Hot Standby Redundancy

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

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

  Note: Refer to "ISCC Call Data Transfer Service" on page 59 for ISCC feature descriptions.

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

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

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

Prerequisites

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

Requirements

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

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

Synchronization Between Redundant T-Servers

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

Configuration Warnings

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

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

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

Warm Standby Deployment

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

General Order of Deployment

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

1. Configure two T-Server `Application` objects as described in "Configuration of T-Server" on page 39.
2. Make sure the `Switch` object is configured for the switch these T-Servers should serve, as described in "Configuration of T-Server" on page 39.
3. Modify the configuration of the primary and backup T-Servers as instructed in the following sections.

After completing the configuration steps, ensure that both T-Servers are installed (see page 52).

Modification of T-Servers for Warm Standby

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

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

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

Start of procedure

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

End of procedure

Next Steps

- Procedure: Modifying the backup T-Server configuration for warm standby, on page 52

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

Start of procedure

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

End of procedure

Warm Standby Installation of Redundant T-Servers

The installation of a redundant T-Server is the same as that for the stand-alone T-Server. If you have not installed the primary and backup T-Servers yet, follow the instructions in "Installation of T-Server" on page 40 for both installations.

Hot Standby Deployment

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

General Order of Deployment

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

1. Configure two T-Server `Applications` objects as described in "Configuring T-Server" on page 39.

2. Make sure the Switch object is configured for the switch these T-Servers should serve, as described in "Configuration of Telephony Objects" on page 36.
3. Modify the configuration of the primary and backup T-Servers as instructed in the following sections.

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

Table 1 on page 25 summarizes hot standby redundancy support in various T-Servers. For detailed, up-to-date information on the subject, see the *Genesys Supported Media Interfaces Reference Manual* located on the 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 40.

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

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

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

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

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

End of procedure

Next Steps

- Procedure: Modifying the backup T-Server configuration for hot standby, on page 55

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

Start of procedure

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

   Note: For information on adding multiple ports, see "Configuring multiple ports" on page 40.

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

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

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

End of procedure

Hot Standby Installation of Redundant T-Servers

The installation of a redundant T-Server is the same as that for the stand-alone T-Server. If you have not installed the primary and backup T-Servers yet, follow instructions in "Installation of T-Server" on page 40 for both installations.

Next Steps

At this point, you have learned how to configure and install redundant T-Servers. Go to Chapter 5, "Starting and Stopping T-Server Components," on page 115, to test your configuration and installation, or continue with Chapter 4, "Multi-Site Support," on page 57, for more possibilities.

Genesys

Chapter

4 Multi-Site Support

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

This chapter is divided into the following sections:

Note: Each switch/T-Server combination offers different multi-site options. For details describing your specific switch/T-Server environment, refer to Chapter 12, "T-Server Common Configuration Options," on page 233.

The following instructions apply to both local and remote switches and T-Servers. Because different vendor switches can be installed at the local and remote locations, this chapter covers several, but not all, possible configurations. To help determine which sections of this chapter apply to your situation, refer to Table 3 on page 75 and Table 4 on page 80.

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

Multi-Site Fundamentals

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

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

- **Call matching—**To link instances of a call distributed across multiple sites and to re-attach essential data associated with the call (`ConnID`, `UserData`, `CallType`, and `CallHistory`). The following T-Server features support this capability:
  - ISCC Call Data Transfer Service (active external routing)—when requested by a T-Server client by specifying the desired destination in the location parameter, and also with various ISCC strategies performed by direct dial or by using the Transfer Connect Service. See "ISCC Transaction Types" on page 66 and "Transfer Connect Service Feature" on page 78.
  - Inter Server Call Control/Call Overflow (ISCC/COF) feature (passive external routing)—applicable when calls are overflowed to another site either directly or manually (see page 79).
  - Number Translation feature (see page 83).
  - Network Attended Transfer/Conference (NAT/C) feature (see page 91).

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

- **Call data synchronization between associated call instances (ISCC Event Propagation)—**To provide the most current data to call instances residing on remote T-Servers. The following T-Server features support this capability:
  - User Data propagation (see page 94)
  - Party Events propagation (see page 95)

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

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

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

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

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

ISCC Call Data Transfer Service

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

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

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

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

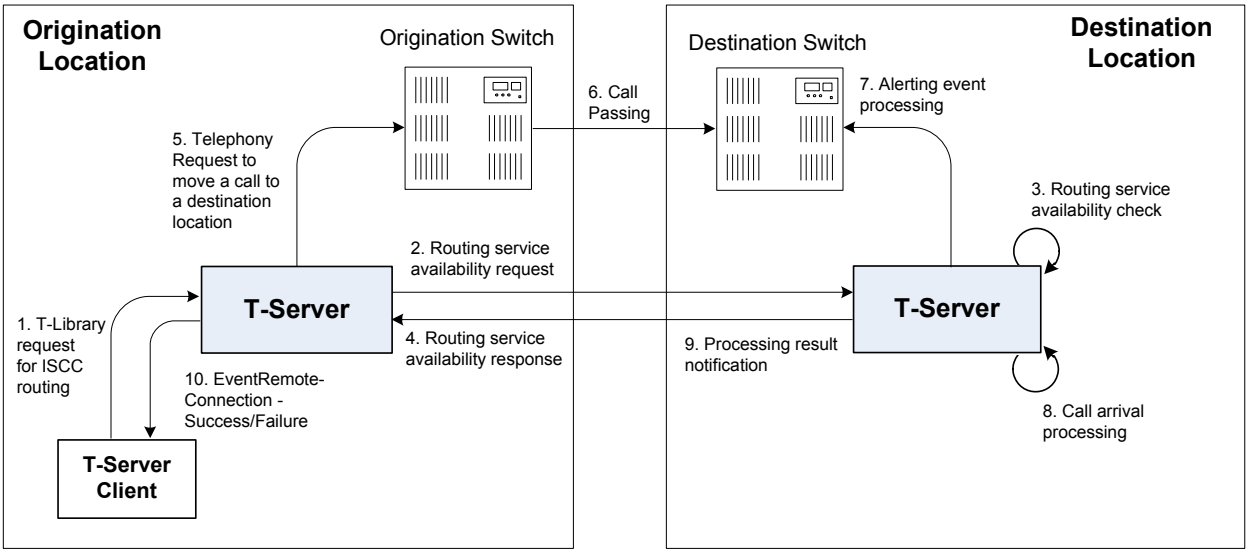

Figure 7: Steps in the ISCC Process

ISCC Call Flows

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

Step 1

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

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

Step 2

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

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

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

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

> **Note:** For more information on Access Codes and Default Access Code, see "Switches and Access Codes" on page 102.

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

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

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

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

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

Step 3

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

Note: The resource-allocation-mode and resource-load-maximum configuration options determine how resources are allocated. For option descriptions, refer to Chapter 12, "T-Server Common Configuration Options," on page 233 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 `ConnID`s are identical, the arrived call is assigned a new unique `ConnID`.

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

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

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

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

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

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

Step 9

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

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

Step 10

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

Client-Controlled ISCC Call Flow

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

Step 1

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

Step 2

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

Step 3

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

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

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

Step 4

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

Step 5

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

Step 6

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

Step 7

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

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

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

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

ISCC Transaction Types

As switches of different types provide calls with different sets of information parameters, a single mechanism for passing call data between the switches is not feasible in some cases. Therefore, the ISCC feature supports a number of mechanisms for passing call data along with calls between locations. This section describes ISCC transaction type principles, identifies which transaction types are supported for each T-Server, and defines each transaction type (beginning with "direct-ani" on page 67).

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

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

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

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

Determining and Configuring Transaction Type Support

You can find descriptions of these transaction types starting on page 67. Use Table 3 on page 75 to identify the transaction types your destination T-Server supports. A blank table cell indicates that T-Server does not support a certain transaction type.

You can configure the transaction types specific to your T-Server as values of the `cast-type` configuration option specified in the ISCC configuration section `extrouter`. Refer to Chapter 12, "T-Server Common Configuration Options," on page 233 for the option description.

ISCC Transaction Type General Principles

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

- `direct-ani`, page 67
- `direct-notoken`, page 69
- `dnis-pool`, page 70
- `pullback`, page 71
- `reroute`, page 72
- `route` (aliased as `route-notoken`), the default transaction type, page 73

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

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

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

direct-ani

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

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

Typically, the `ANI` attribute represents the original call identifier (customer phone number), which guarantees that the attribute remains unique. However, you can use the `non-unique-ani` resource type to block ISCC from matching calls based on an ANI that is known to be non-unique. (See "Configuring access resources for non-unique ANI" on page 111 for details.)

direct-callid

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

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

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

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

direct-network-callid

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

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

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

direct-uui

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

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

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

direct-notoken

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

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

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

dnis-pool

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

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

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

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

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

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

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

In Load-Balancing Mode

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

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

pullback

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

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

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

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

reroute

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

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

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

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

route

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

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

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

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

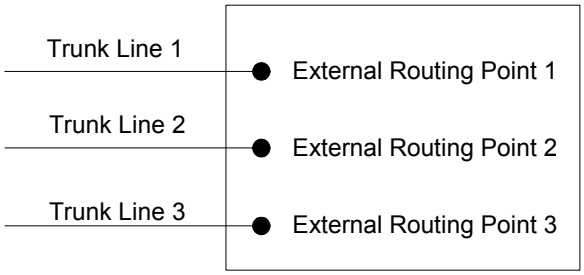

Figure 8: Point-to-Point Trunk Configuration

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

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

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

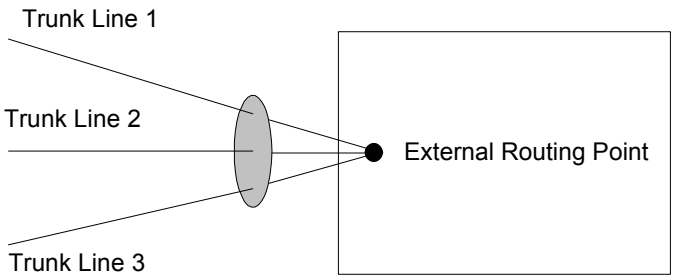

Figure 9: Multiple-to-Point Trunk Configuration

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

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

route-uui

The `route-uui` transaction type employs the dedicated External Routing Point feature of the `route` transaction type (page 73) and the UUI matching feature of the `direct-uui` transaction type (page 69). This transaction type accommodates those switches that require a designated External Routing Point even though they use UUI for tracking.

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

T-Server Transaction Type Support

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

Table 3: T-Server Support of Transaction Types

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Transfer Connect Service Feature

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

Procedure:
Activating Transfer Connect Service

Start of procedure

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

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

End of procedure

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

ISCC/Call Overflow Feature

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Note: When the ISCC/COF feature is in use, the Number Translation feature becomes active. For more information on feature configuration, see "Number Translation Feature" on page 83.

ISCC/COF Call Flow

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

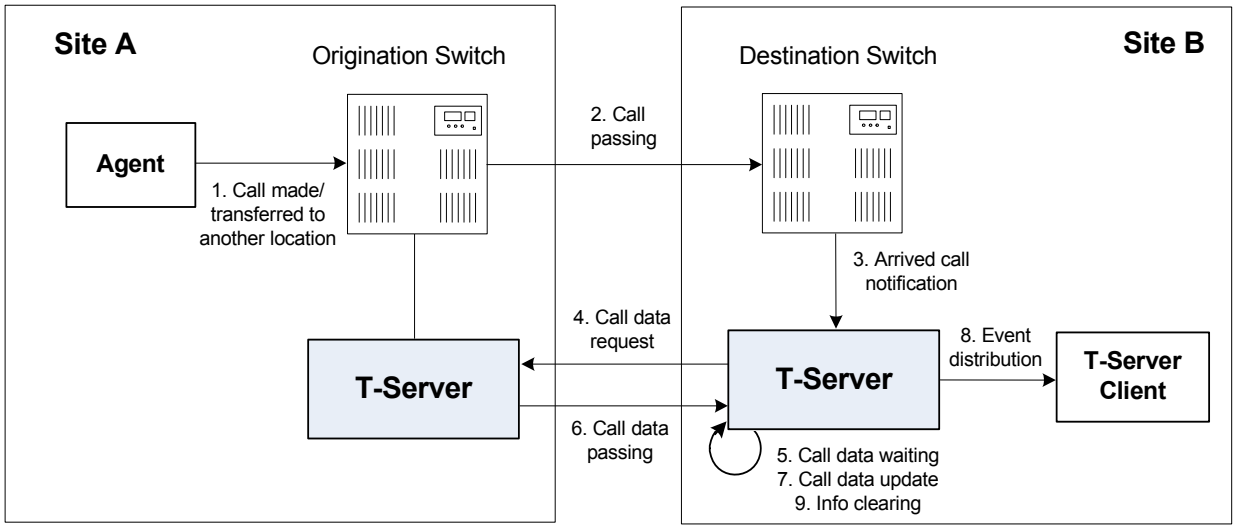

Figure 10: Steps in the ISCC/COF Process

Step 1

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

Step 2

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

Step 3

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

Step 4

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

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

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

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

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

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

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

Step 5

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

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

Step 6

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

Step 7

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

Step 8

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

Step 9

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

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

Number Translation Feature

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

The translation process involves two algorithms, one for rule selection and the other for the actual translation. Through the first algorithm, T-Server selects a rule that will be used for number translation. Through the second algorithm, T-Server translates the number according to the selected rule definition. See “Number Translation Rules” on page 84 for more information on configuring rules for your environment.

Number translation occurs as follows:

1. The switch reports a number, typically via `AttributeANI`.
2. T-Server evaluates all configured inbound rules to determine which one is the best fit for the received number. The best fit is determined by comparing the length of, and the specific digits in, the input number with the inbound pattern of each configured rule. See "Rule Examples" on page 89 for specific examples.
3. T-Server translates the number according to the selected rule.

To enable T-Server to translate numbers, you must perform specific configuration tasks that are associated with translation. See "Configuring Number Translation" on page 91.

Number Translation Rules

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

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

Using ABNF for Rules

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

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

Common Syntax Notations

Syntax notations common to many of these rules include:

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

Component Notations

Component notations include:

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

  where:

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

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

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

  where:

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

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

  where:

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

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

  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 89) is `in-pattern=1AAABBBCCCC; out-pattern=91ABC`.
  - `flexible-length-group` specifies a group composed of 0 or more digits in the group represented by the `group-identifier`. For example, in `in-pattern=1[415,650]A*B, *B` represents the flexible length group containing the remaining digits in the number.
  - `entity` represents digits defined for a specific purpose, for example, country code.

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

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

  See the earlier explanation under `abstract-group`.

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

  See the earlier explanation under `abstract-group`.

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

  where:

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

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

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

  where:

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

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

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

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

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

Recommendations for Rule Configuration

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Rule Examples

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

Rules

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

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

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

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

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

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

Examples

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Procedure:
Configuring Number Translation

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

Overview

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

Start of procedure

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

   For the option description and its valid values, see Chapter 12, "T-Server Common Configuration Options," on page 233.
6. When you are finished, click `Apply`.
7. Click `OK` to save your changes and exit the `Properties` dialog box.

End of procedure

Network Attended Transfer/Conference Feature

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

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

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

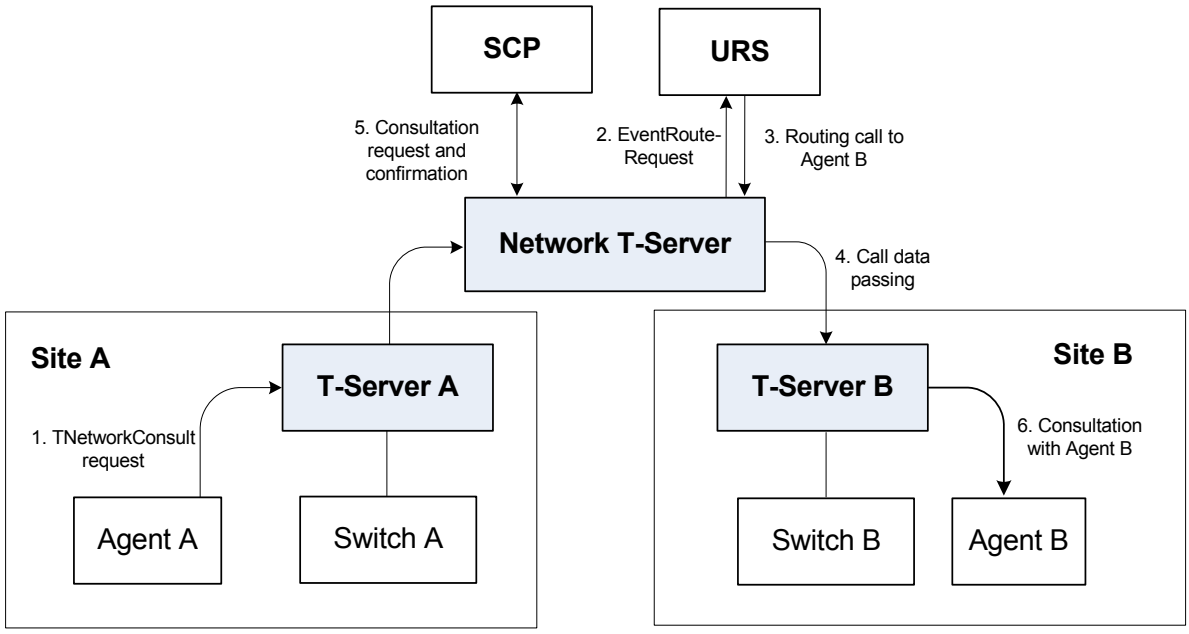

Figure 11: Steps in the NAT/C Process in URS-Controlled Mode

Step 1

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

Step 2

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

Step 3

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

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

Step 4

The Network T-Server proceeds to obtain the access number from T-Server B, and passes the call data to T-Server B. (See "ISCC Call Data Transfer Service" on page 59 for details.)

Step 5

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

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

Step 6

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

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

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

Event Propagation Feature

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

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

User Data Propagation

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

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

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

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

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

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

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

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

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

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

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

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

Party Events Propagation

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

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

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

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

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

Switch Partitioning

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

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

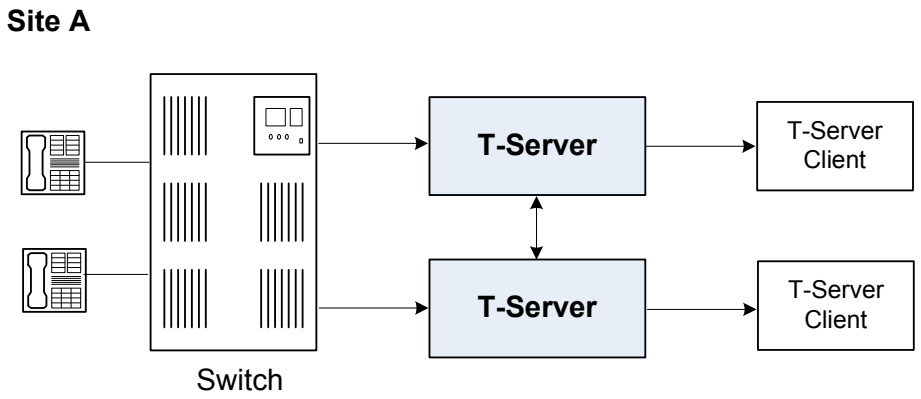

Figure 12: Switch Partitioning Architecture

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 238.

To control race conditions that may occur in the switch-partitioned environment, use the `epp-tout` configuration option (see page 253).

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

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

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

Table 5: T-Server Support for Switch Partitioning

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

Event Propagation Configuration

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

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

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

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

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

Procedure: Activating Event Propagation: basic configuration

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

Start of procedure

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

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

   This setting enables Party Events propagation.

   For the option description and its valid values, see Chapter 12, "T-Server Common Configuration Options," on page 233.
6. When you are finished, click `Apply`.
7. Click `OK` to save your changes and exit the `Properties` dialog box.

End of procedure

Next Steps

- For advanced feature configuration, do the following procedure:

  Procedure: Modifying Event Propagation: advanced configuration, on page 98

Procedure: Modifying Event Propagation: advanced configuration

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

Prerequisites

- Procedure: Activating Event Propagation: basic configuration, on page 98

Overview

You can set Event Propagation parameters using:

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

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

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

Start of procedure

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

> **Note:** If no `Default Access Code` is configured, see page 103 for instructions. If no `Access Codes` are configured, see page 104 for instructions.

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

     `propagate=yes`

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

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

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

```
propagate=party
```

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

```
propagate=no
```

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

End of procedure

ISCC Transaction Monitoring Feature

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

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

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

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

Configuring Multi-Site Support

Prior to configuring T-Server to support multi-site operation, you must read the "Licensing Requirements" on page 33, as well as previous sections of this chapter on multi-site deployment. In particular, Table 3 on page 75 shows which transaction types are supported by a specific T-Server, while Table 4 on page 80 shows whether your T-Server supports the `NetworkCallID` attribute for

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

Note: Before attempting to configure a multi-site environment, Genesys recommends that you plan the changes you want to make to your existing contact centers. You should then gather the configuration information you will need (such as the name of each T-Server application, port assignments, and switch names), and use Configuration Manager to create and partially configure each T-Server object. Review multi-site option values in the "extrouter Section" on page 243 and determine what these values need to be, based on your network topology.

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

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

You must configure these objects for origination and destination locations. Multi-site support features activate automatically at T-Server startup. See "DNs" on page 108 for details.

Applications

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

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

Procedure: Configuring T-Server Applications

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

Start of procedure

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

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

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

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

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

> **Note:** For a list of options and valid values, see "extrouter Section" on page 243, in the "T-Server Common Configuration Options" chapter in Part Two of this document.

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

End of procedure

Next Steps

- See "Switches and Access Codes."

Switches and Access Codes

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

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

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

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

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

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

Note: When migrating from previous releases of T-Servers to 8.1, or when using T-Servers of different releases (including 8.1) in the same environment, see "Compatibility Notes" on page 107.

Procedure: Configuring Default Access Codes

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

Prerequisites

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

Start of procedure

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

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

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

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

End of procedure

Next Steps

- See "Configuring Access Codes."

Procedure: Configuring Access Codes

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

Prerequisites

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

Start of procedure

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

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

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

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

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

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

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

Table 6: Target Type: ISCC Protocol Parameters

ISCC Protocol Parameters	Description
dnis-tail=<number-of-digits>	Where number-of-digits is the number of significant DNIS digits (last digits) used for call matching. 0 (zero) matches all digits.
propagate=<yes, udata, party, no>	Default is yes. For more information, see "Modifying Event Propagation: advanced configuration" on page 98.
direct-network-callid=<>	For configuration information, see Part Two of this document. (Use Table 4 on page 80 to determine if your T-Server supports the direct-network-callid transaction type.)

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

Table 7: Target Type: ISCC Call Overflow Parameters

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

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

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

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

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

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

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

End of procedure

Next Steps

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

Compatibility Notes

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

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

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

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

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

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

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

DNs

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

Procedure: Configuring access resources for the route transaction type

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

Prerequisites

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

Start of procedure

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

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

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

6. When you are finished, click Apply.

End of procedure

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

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

Start of procedure

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

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

End of procedure

Procedure: Configuring access resources for direct-* transaction types

Start of procedure

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

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

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

End of procedure

Procedure:
Configuring access resources for ISCC/COF

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

Start of procedure

Note: Use Table 4 on page 80 to determine if your T-Server supports the ISCC/COF feature.

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

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

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

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

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

End of procedure

Procedure:
Configuring access resources for non-unique ANI

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

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

Start of procedure

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

End of procedure

Procedure: Modifying DNs for isolated switch partitioning

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

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

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

Start of procedure

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

7. When you are finished, click Apply.

End of procedure

Configuration Examples

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

Multiple Transaction Types

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

In this example, you configure an origination and a destination switch for as described in "Switches and Access Codes" on page 102.

1. Among configured Switches, select the origination Switch.
2. Open the Switch Properties dialog box and click the Default Access Codes tab.
3. Click Add to open the Access Code Properties dialog box.
4. Set the Access Code field to 9.
5. When you are finished, click Apply.
6. Among configured Switches, select the destination Switch.
7. Under the destination Switch, configure a DN as described in "Configuring access resources for the route transaction type" on page 108.
8. Set the DN Number field to 5001234567.
9. Click the Advanced tab of this DN's Properties dialog box.
10. Select the Use Override check box and enter 1234567 in the Use Override field.
11. When you are finished, click Apply or Save.
12. Use a T-Server client application to register for this new DN with the destination T-Server and, therefore, with the switch.
13. Request to route a call from any DN at the origination switch to the destination DN you have just configured:
    - If you are using the route ISCC transaction type, the client requests that T-Server deliver a call to a destination location using the DN number 5001234567. ISCC requests that the switch dial one of the external routing points at the destination location, using the value either of the Access Number field or of the Access Code field, which is 9, concatenated with the external routing point at the destination location. The call is routed to the DN number 5001234567.

- If you are using the `direct-ani` ISCC transaction type, the client requests that T-Server deliver a call to a destination location using the DN number `1234567`, which is the `Use Override` value. ISCC requests that the switch dial `91234567`, which is a combination of the `Switch Access Code` value and the `Use Override` value. The destination T-Server is waiting for the call to directly arrive at DN number `5001234567`.

Call Overflow Methods

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

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

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

when configuring `Switch Access Codes` as described on page 104.

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

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

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

when configuring `Switch Access Codes` as described on page 104.

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

Next Steps

Continue with Chapter 5, "Starting and Stopping T-Server Components," on page 115 to test your configuration and installation.

Chapter

5 Starting and Stopping T-Server Components

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

Command-Line Parameters

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

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

Common command-line parameters are as follows:

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

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

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

Starting and Stopping with the Management Layer

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

Start of procedure

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

   The command-line parameters common to Framework server components are described on page 115.
6. When you are finished, click `Apply`.
7. Click `OK` to save your changes and exit the `Properties` dialog box.

End of procedure

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

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

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

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

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

Starting with Startup Files

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

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

When using a startup file, verify that the startup parameters the installation script inserted in the startup file are correct. Use the following instructions for UNIX and Windows to start those application for which startup files are created. See the appropriate sections in "Starting Manually" on page 119 to identify which applications should be running for a particular application to start.

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

Start of procedure

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

End of procedure

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

Start of procedure

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

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

Or

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

  ```
  startServer.bat
  ```

End of procedure

Starting Manually

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

The command-line parameters common to Framework server components are described on page 115.

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

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

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

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

Table 9: T-Server and HA Proxy Executable Names

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

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

T-Server Type	T-Server Executable File Name		HA Proxy Executable File Name	
	UNIX	Windows	UNIX	Windows
Nortel Communication Server 1000 with SCSS/MLS	succession_server	succession_server.exe	Not Applicable	
Philips Sopho iS3000	iS3000_server	iS3000_server.exe	ha_proxy_iS3000	ha_proxy_iS3000.exe
Radvision iContact	nts_server	nts_server.exe	Not Applicable	
Samsung IP-PCX IAP	samsung_server	samsung_server.exe	Not Applicable	
Siemens Hicom 300/HiPath 400 CSTA I	rolmcb4_server	rolmcb4_server.exe	Not Applicable	
Siemens HiPath 3000	HiPath3000_server	HiPath3000_server.exe	Not Applicable	
Siemens HiPath 4000 CSTA III	HiPath4000_server	HiPath4000_server.exe	Not Applicable	
Siemens HiPath DX	HiPathDX_server	HiPathDX_server.exe	Not Applicable	
SIP Server	sip_server	sip_server.exe	Not Applicable	
Spectrum	spectrum_server	spectrum_server.exe	Not Applicable	
Tadiran Coral	Coral_server	Coral_server.exe	Not Applicable	
Teltronics 20-20	Teltronics2020_server	Teltronics2020_server.exe	ha_proxy_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 123.

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

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

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

The command-line parameters common to Framework server components are described on page 115.

Procedure:
Starting HA Proxy on UNIX manually

Start of procedure

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

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

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

   Table 9 on page 120 lists HA Proxy executable names for supported switches.

End of procedure

Procedure:
Starting HA Proxy on Windows manually

Start of procedure

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

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

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

   Table 9 on page 120 lists HA Proxy executable names for supported switches.

End of procedure

T-Server

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

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

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

The command-line parameters common to Framework server components are described on page 115.

Procedure: Starting T-Server on UNIX manually

Start of procedure

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

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

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

   Table 9 on page 120 lists T-Server executable names for supported switches.

End of procedure

Procedure: Starting T-Server on Windows manually

Start of procedure

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

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

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

   Table 9 on page 120 lists T-Server executable names for supported switches.

End of procedure

Verifying Successful Startup

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

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

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

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

Stopping Manually

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

Procedure: Stopping T-Server on UNIX manually

Start of procedure

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

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

End of procedure

Procedure: Stopping T-Server on Windows manually

Start of procedure

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

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

End of procedure

Starting and Stopping with Windows Services Manager

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

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

where the command-line parameters common to Framework server components are described on page 115 and

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

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

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

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

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

Next Steps

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

Genesys

Part

2 T-Server Configuration

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

- Chapter 6, "Switch-Specific Configuration," on page 129, 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 139, describes which features are supported by this T-Server including T-Library functionality, and error messages.
- Chapter 11, "Common Configuration Options," on page 211, describes log configuration options common to all Genesys server applications.
- Chapter 12, "T-Server Common Configuration Options," on page 233, describes configuration options that are common to all T-Server types including options for multi-site configuration.
- Chapter 10, "T-Server-Specific and DN Configuration Options," on page 233, describes configuration options specific to this T-Server including the link-related options—those which address the interface between T-Server and the switch.

New in T-Server for Nortel Communication Server 1000 with SCCS/MLS

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 Nortel Communication Server 1000 with SCCS/MLS:

- **Support for the soft-login-support configuration option.** T-Server now supports the `soft-login-support` configuration option to process agent-related CTI messages.
- **Support for the Nortel Contact Center 6.0 Standby Server.** T-Server now supports a pre-configured standby server with a CTI-link capability. See page 159 for details.
- T-Server is now supported on the following platforms:
  - Red Hat Enterprise Linux 5 64-bit
  - HP-UX 11i v3

Notes:
- Configuration option changes that apply to T-Server for Nortel Communication Server 1000 with SCCS/MLS are described in "Changes from Release 8.0 to 8.1" on page 249.
- For a list of new features common to all T-Servers, see Part One of this document.

Chapter

6 Switch-Specific Configuration

This chapter presents switch-specific reference information for configuring T-Server for the Nortel Communication Server 1000 with SCCS/MLS switch and includes these sections:

Known Limitations

- When Nortel Communication Server 1000 with SCCS/MLS uses the Meridian CTI link, Call Supervisor and Activity Code features are not supported. See the option “link-type” on page 238 for details about specifying the Meridian CTI link.
- T-Server supports Call Supervisor and Activity Codes functionality only for Symposium link version SCCS 4.2 and higher.
- T-Server does not support use of Nortel CallPilot for call treatments while the calls are being routed by Genesys URS.
- In a high-availability (HA) environment, all calls with treatments applied to CDNs at the time of a T-Server switchover will be routed by the switch to a default destination.

Setting DN Properties

Table 10 contains information on how to set DN types and properties depending on the switch configuration in the Configuration Layer.

Table 10: Setting the DN Properties

Switch DN Type[a]	Configuration Layer DN Properties	
	DN Type	Switch-Specific Type
Regular DN (Agent Extension, Phone Set)	Extension	N/A
	Voice Treatment Port	1
	Extension	8[b]
ACD Position	ACD Position	N/A
	Voice Treatment Port	2
CDN	Routing Point	N/A
ACD	ACD Queue	2[c]
Voice Channel	Voice Mail	N/A

a. Any DN that is configured in the Configuration Layer must be configured on the switch as an AST-enabled DN, because the switch only provides messaging on DNs that are configured as AST-enabled.

b. The new switch-specific type 8 acts the same as the Voice Treatment port when the `vtport-generate-hook-events` configuration option is set to `false`.

c. If an ACD switch DN type has a switch-specific type 2, T-Server uses `EventQueued` and `EventDiverted` events even if it receives a `RouteRequest` request for this DN.

Note: • You can find a list of the switch releases that T-Server supports on the Genesys Technical Support website at: `http://genesyslab.com/support/dl/retrieve/default.asp?item=ADD31C7A5836A432A02EACD31A658AF1&view=item`

Supported Hot-Standby Configurations

The following high-availability (HA) configurations are currently supported:

- `Hot standby` redundancy type `dual` for a single CTI link with two T-Servers.

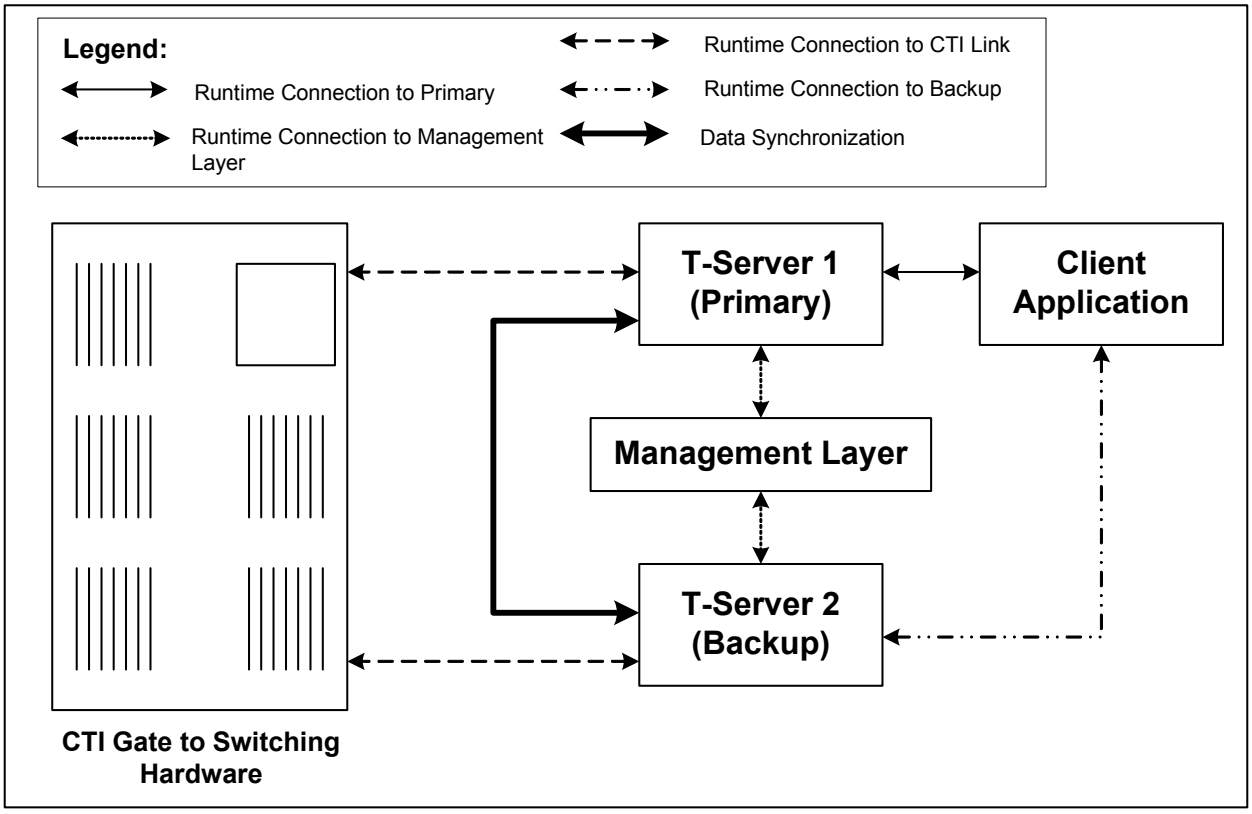

Figure 13: Hot Standby Redundancy Type Dual for a Single CTI Link with Two T-Servers

Note: Starting with release 7.1, T-Server does not support an HA configuration with the Meridian redundant link. Existing customers need to continue using the 7.0 Meridian T-Server.

Multi-Site/Multi-Switch Configuration

In a multi-switch environment (such as NACD overflow configurations), T-Server for Nortel Communication Server 1000 with SCCS/MLS requires that you assign each switch a unique Home Location Code (HLOC). The HLOC feature (available on switch release 23 and later) prefixes all switch `CallID` numbers with the unique HLOC code for that switch. `CallIDs` then become unique among the network of switches. T-Server uses this feature in tracking network call-flow scenarios (such as NACD overflows). The HLOC is defined on the switch in `LD 15, Customer Data Block`. See the Nortel Communication Server 1000 documentation for more details.

Support for direct-network-callid

T-Server for Nortel Communication Server 1000 with SCCS/MLS supports the `direct-network- callid` transaction type. With this transaction type, 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. To use this transaction type, you must complete procedure "Configuring transaction type".

Note: The `direct-network-callid` transaction type is used only in conjunction with the `TRouteCall` and `TMuteTransfer` function calls. They are applied only to the call that is in progress, and do not apply to functions that involve in the creation of a new call—for example, `TMakeCall`.

Procedure: Configuring transaction type

Purpose: To configure the `direct-network-callid` transaction type.

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 configuration option `cast-type` to `direct-network-callid`.

5. When you are finished, click Ok.
6. Click the Switches tab.
7. Select the switch from the list and open the Switch Properties dialog box. Click the Access Codes tab.
8. Click Add to open the Access Code Properties dialog box.
9. In the Target Type field, select Target ISCC.
10. In the Route Type field, select Direct Network Call ID or Default.
11. In the ISCC Protocol Parameters field, type:
    direct-network-callid=MLS
12. Click OK.

End of procedure

Overlay Configurations

This section describes overlay configurations for the following Nortel switch link types:

- Meridian Link
- Symposium Call Center Server (SCCS)

Nortel Communication Server 1000 and Meridian 1 provide messaging only for DNs configured as AST DNs. In a Nortel Communication Server 1000 or Meridian 1 contact-center environment, you must configure all DNs as AST DNs so that T-Server can accurately track the status of calls and of the agents involved in the call.

Depending on the link type, make the following switch configuration entries to facilitate messaging.

Procedure: Configuring the Meridian Link

Purpose: To configure the Meridian Link switch link type.

Start of procedure

1. On the configuration record in `LD 17` (see Table 11), define an Application Module Link (AML) port. Define a Value-Added Server (VAS) server with a unique `VAS ID` and assign it to the AML port you defined in `LD 17`.

Table 11: LD 17 Configuration Parameters for Meridian Link

AML Parameters				VAS Parameters	
ADAN	AML 8	T2	000	VAS	
CTYP	ESDI/MSDL	T3	010	VSID	00
DNUM	8	N1	128	DLOP	
DES		N2	08	AML	08
BPS	19200	K	2	SECU	YES
CLO	INT	RXMT	05	INTL	0004
IADR	003	CRC	0	MCNT	0400
RADR	001	ORUR	005	CONF	DIR
T1	04	ABOR	005		

Note: You must configure the parameters listed in Table 11 according to the switch documentation. Genesys requires no specific configuration values. Consult the Nortel reference documentation for proper configuration values.

2. Configure the `Customer for Status Change` messages in `LD 15` (required):
   - Set `VSID` to the `VAS ID` of Meridian Link as defined in the `LD 17` configuration parameters.
3. Configure 500/2500 sets in `LD 10` (required):
   a. You must set the `IAPG` prompt to `1`.
   b. You must set the `AST` prompt to `YES`.
   c. For a set of the `ACD Queue` type, you must set:
      — `Class of Service` to `AGTA`.

— AACD prompt to YES.

d. To transfer and conference from a 500/2500 port, Class of Service has to include XFA.

e. The AST, AACD, and IAPG parameters do not copy from one set to another when you make copies of 500/2500 sets. You must specify them manually for each set you configure.

f. Configure Digital sets in LD 11 (required for set to work with T-Server):

g. You must set the AST prompt to indicate which keys on the set are the AST keys (maximum of two keys).
For example—AST 00 07 (Key 00 and Key 07 will be AST Keys).

h. You must set the IAPG prompt to 1.

i. For the Transfer feature of T-Server to work, you must program a transfer key on each phone.

j. For the Conference feature of T-Server to work, you must program a conference key on each phone.

k. The AST and IAPG parameters do not copy from one set to another when you make copies of Digital sets. You must specify them manually for each set you configure.

4. Configure the ACD Queue DNs in LD 23 (required for the ACD Queue to work with T-Server):

a. You must set the ISAP prompt to YES.

b. You must set the VSID prompt to the VSID assigned to Meridian Link in LD 17.

5. Configure the controlled DNs (CDNs) in LD 23 (optional, depending on the application):

a. You must set the VSID and HSID prompts to the VAS ID LD 17 assigned to Meridian Link in LD 17.

b. Configure the DNIS Notification (required for the DNIS to be available to T-Server).

c. Configure the Customer Data Block in LD 15 (you must set the OPT prompt to DNI).

d. Configure the Trunk Route Configuration in LD 16 (you must set the DNIS prompt to YES for the Trunk Route to pass DNIS).

End of procedure

Procedure: Configuring SCCS

Purpose: To configure the SCCS switch link type.

Start of procedure

1. On the configuration record in `LD 17` (see Table 12), define an `Application Module Link` (AML) parameter.

Table 12: LD 17 Configuration Parameters for SCCS

AML Parameters	
ADAN	ELAN 16
CTYP	ELAN
DES	—
N1	512

Note: You must configure the parameters listed in Table 12 according to the switch documentation. Genesys requires no specific configuration values. Consult the Nortel reference documentation for proper configuration values

2. Configure 500/2500 sets in `LD 10` (required):
   a. You must set the `AST` prompt to `YES`.
   b. For a set of the ACD Queue type, you must set:
      — `Class of Service` to `AGTA`.
      — `AACD` prompt to `YES`.
   c. To transfer and conference from a `500/2500` port, `Class of Service` has to include `XFA`.
   d. The `AST` and `AACD` parameters do not copy from one set to another when you make copies of 500/2500 sets. You must specify them manually for each set you configure.
3. Configure Digital sets in `LD 11` (required for set to work with T-Server):
   a. You must set the `AST` prompt to indicate which keys on the set will be `AST` (maximum of two keys).
   For example—`AST 00 07` (`Key 00` and `Key 07` will be `AST Keys`).
   b. For the Transfer feature of T-Server to work, you must program a transfer key on each phone.

c. For the Conference feature of T-Server to work, you must program a conference key on each phone.

d. The AST parameter does not copy from one set to another when you make copies of Digital sets. You must specify AST manually for each set you configure.

e. Configure controlled DNs (CDNs) in LD 23 (optional, depending on application):

f. You must set the CNTL prompt to YES.

g. Configure the DNIS Notification (required for the DNIS to be available to T-Server).

h. Configure the Customer Data Block in LD 15 (you must set the OPT prompt to DNI).

i. Configure the Trunk Route Configuration in LD 16 (you must set the DNIS prompt to YES for the Trunk Route to pass DNIS).

End of procedure

Operation and Configuration of Peripheral Equipment

This section describes how to integrate peripheral equipment with your Genesys T-Server solution and how to attach the data that equipment generates to calls.

1. Requirements for attaching data from an IVR (other than a Meridian IVR):
   - You must connect the non-Meridian IVR to the PBX through 2500 station ports, which are configured for Meridian Link as described in "Overlay Configurations" on page 133.
   - An AttachData function call to T-Server must be made prior to the transfer out of the IVR. The DN or ACD-ID of the channel must be identified to T-Server so that the data can be attached to the correct call.
   - The Agent Queue and phone must be AST enabled, as described in "Overlay Configurations" on page 133.
   - The PBX software must be release 23.x or higher.
   - The Meridian Link software must be release 5.x or higher.
2. Requirements for attaching data from a Meridian IVR:
   - Each channel of the Meridian IVR must be dedicated to a port of the Meridian Mail system in the Channel Allocation Table for Meridian Mail Administration. The access application must be assigned to these ports. A different class must be assigned to each port dedicated to Meridian Access.

- One controlled DN (CDN) per Meridian IVR channel must be built in the PBX as just described in the previous point. In addition to T-Server, a CDN controller called `iroute` must be running. It can run on the same server as T-Server.
- A `user` function must be written to make an `AttachData` call to T-Server. A table must be included to convert the channel number of the IVR to the ACD-ID of the Meridian Mail port associated with that IVR channel. The DN or ACD-ID of the channel must be identified to T-Server so that the data can be attached to the correct call. Buffer 1 of the input data should contain the DN/ACD-DN of the call that it is to be transferred to.
- The call should now be transferred to a CDN dedicated to the channel of the IVR that the call is on. The CDN controller, a client of T-Server, must have acquired this CDN prior to a call being sent to this CDN. The CDN should transfer the call to the appropriate DN/ACD-DN that was passed to T-Server in Step 1.
- The PBX software must be release 23.x or higher.
- The Meridian Link software must be release 5.x or higher.

The following example is based on the configuration described in Table 13.

Example

A call rings into ACD Queue `3600` and is routed by the PBX at `ACD-ID 2901`. This port is dedicated to IVR channel 1. Therefore, channel 1 answers the call. The caller enters his or her account number. Based on this account number, the call should be transferred to an agent in Queue `5200`. A user cell then attaches data to T-Server. Buffer 1 contains `5200`. The `Attach to DN, 2901,` is selected from the table in the `user` function because the call is on channel 1 of the IVR. For the same reason, the call is then transferred to `CDN 4201`. The CDN controller should retrieve the route to the DN from the attached data, in this case `5200,` and then route the call to that DN.

Table 13: Configuration of Peripheral Equipment

Meridian Mail		IVR		
ACD-DN	**ACD-ID**	**Channel**	**Class**	**CDN**
3600	2900	0	1	4200
3600	2901	1	2	4201
3600	2902	2	3	4202

Chapter

7 Supported T-Server Features

This chapter describes the telephony functionality supported by the T-Server for Nortel Communication Server with SCCS/MLS. It includes these sections:

T-Library Functionality

The tables in this chapter present T-Library functionality supported in T-Server for Nortel Communication Server 1000 with SCCS/MLS. The table entries use these notations:

N—Not supported

Y—Supported

E—Event only is supported

I—Supported, but reserved for internal Genesys use

In Table 14, when a set of events is sent in response to a single request, the events are listed in an arbitrary order. An asterisk (*) indicates the event that contains the same `Reference ID` as the request. For more information, refer to the *Genesys Events* and *Models Reference Manual* and *Voice Platform SDK 8.x .NET (*or *Java) API Reference* for complete information on the T-Server events, call models, and requests.

Table 14 reflects only the switch functionality that is used by Genesys software and might not include the complete set of events offered by the switch.

Certain requests listed in Table 14 are reserved for internal Genesys use and are listed here merely for completeness of information.

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

Table 14: Supported T-Library Functionality

Feature Request	Request Subtype	Corresponding Event(s)	Supported
General Requests			
TOpenServer		EventServerConnected	Y
TOpenServerEx		EventServerConnected	Y
TCloseServer		EventServerDisconnected	Y
TSetInputMask		EventACK	Y
TDispatch		Not Applicable	Y
TScanServer		Not Applicable	Y
TScanServerEx		Not Applicable	Y
Registration Requests			
TRegisterAddress[a]		EventRegistered	Y
TUnregisterAddress[a]		EventUnregistered	Y
Call-Handling Requests			
TMakeCall[b]	MakeCallRegular	EventDialing	Y
	MakeCallDirectAgent		N
	MakeCallSupervisorAssist		N
	MakeCallPriority		N
	MakeCallDirectPriority		N
TAnswerCall		EventEstablished	Y

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TReleaseCall		EventReleased	Y
TClearCall		EventReleased	N
THoldCall		EventHeld	Y
TRetrieveCall		EventRetrieved	Y
TRedirectCall		EventReleased	N
TMakePredictiveCall		EventDialing*, EventQueued	N
Transfer/Conference Requests			
TInitiateTransfer[b]		EventHeld, EventDialing*	Y
TCompleteTransfer		First arriving EventReleased*, EventPartyChanged	Y
TInitiateConference[b]		EventHeld, EventDialing*	Y
TCompleteConference		EventReleased*, EventRetrieved, EventPartyChanged, EventPartyAdded	Y
TDeleteFromConference		EventPartyDeleted*, EventReleased	N
TReconnectCall[c]		EventReleased, EventRetrieved*	Y
TAlternateCall		EventHeld*, EventRetrieved	Y
TMergeCalls	MergeForTransfer	EventReleased*, EventPartyChanged	N
	MergeForConference	EventReleased*, EventRetrieved, EventPartyChanged, EventPartyAdded	N
TMuteTransfer[b]		EventHeld, EventDialing*, EventReleased, EventPartyChanged	Y

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TSingleStepTransfer[b]		EventReleased*, EventPartyChanged	N
TSingleStepConference		EventPartyAdded* or EventRinging*, EventEstablished	N
Call-Routing Requests			
TRouteCall[b]	RouteTypeUnknown	EventRouteUsed	I
	RouteTypeDefault		I
	RouteTypeLabel		N
	RouteTypeOverwriteDNIS		N
	RouteTypeDDD		N
	RouteTypeIDDD		N
	RouteTypeDirect		N
	RouteTypeReject		N
	RouteTypeAnnouncement		N
	RouteTypePostFeature		N
	RouteTypeDirectAgent		N
	RouteTypePriority		N
	RouteTypeDirectPriority		N
	RouteTypeAgentID		N
	RouteTypeCallDisconnect		N

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
Call-Treatment Requests			
TApplyTreatment	TreatmentUnknown	(EventTreatmentApplied + EventTreatmentEnd)/Event-TreatmentNotApplied	N
	TreatmentIVR		N
	TreatmentMusic		I
	TreatmentRingBack		I
	TreatmentSilence		I
	TreatmentBusy		N
	TreatmentCollectDigits		N
	TreatmentPlay-Announcement		N
	TreatmentPlay-AnnouncementAndDigits		N
	TreatmentVerifyDigits		N
	TreatmentRecordUser-Announcement		N
	TreatmentDeleteUser-Announcement		N
	TreatmentCancelCall		N
	TreatmentPlayApplication		N
	TreatmentSetDefaultRoute		N
	TreatmentTextToSpeech		N
	TreatmentTextToSpeech-AndDigits		N
	TreatmentFastBusy		N
	TreatmentRAN		N
TGiveMusicTreatment		EventTreatmentApplied	I
TGiveRingBackTreatment		EventTreatmentApplied	I
TGiveSilenceTreatment		EventTreatmentApplied	I

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
DTMF (Dual-Tone Multifrequency) Requests			
TCollectDigits		EventDigitsCollected	Y
TSendDTMF		EventDTMFSent	Y
Voice-Mail Requests			
TOpenVoiceFile		EventVoiceFileOpened	Y
TCloseVoiceFile		EventVoiceFileClosed	Y
TLoginMailBox		EventMailBoxLogin	Y
TLogoutMailBox		EventMailBoxLogout	Y
TPlayVoice		EventVoiceFileEndPlay	Y
Agent and DN Feature Requests			
TAgentLogin	(See "Support for Agent States and Workmodes" on page 148)	EventAgentLogin	Y
TAgentLogout		EventAgentLogout	Y
TAgentSetReady		EventAgentReady	Y
TAgentSetNotReady		EventAgentNotReady	Y
TMonitorNextCall	MonitorOneCall	EventMonitoringNextCall	N
	MonitorAllCalls		N
TCancelMonitoring		EventMonitoringCanceled	N
TCallSetForward	ForwardModeNone	EventForwardSet	Y
	ForwardModeUnconditional		N
	ForwardModeOnBusy		N
	ForwardModeOnNoAnswer		N
	ForwardModeOnBusyAnd-NoAnswer		N
	ForwardModeSendAllCalls		N
TCallCancelForward		EventForwardCancel	Y
TSetMuteOff		EventMuteOff	N

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TSetMuteOn		EventMuteOn	N
TListenDisconnect		EventListenDisconnected	N
TListenReconnect		EventListenReconnected	N
TSetDNDOn		EventDNDOn	Y
TSetDNDOff		EventDNDOff	Y
TSetMessageWaitingOn		EventMessageWaitingOn	Y
TSetMessageWaitingOff		EventMessageWaitingOff	Y
		EventOffHook	Y
		EventOnHook	Y
		EventDNBackInService	Y
		EventDNOutOfService	Y
Query Requests			
TQuerySwitch[a]	SwitchInfoDateTime	EventSwitchInfo	N
	SwitchInfoClassifierStat		N
TQueryCall[a]	CallInfoPartiesQuery	EventPartyInfo	N
	CallInfoStatusQuery		Y
TQueryAddress[a]	AddressInfoAddressStatus	EventAddressInfo	N
	AddressInfoMsgWaiting-Status		N
	AddressInfoAssociation-Status		N
	AddressInfoCallForwarding-Status		N
	AddressInfoAgentStatus		N
	AddressInfoNumberOf-AgentsInQueue		N
	AddressInfoNumberOf-AvailableAgentsInQueue		N

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TQueryAddress[a]	AddressInfoNumberOfCalls-InQueue	EventAddressInfo	N
	AddressInfoAddressType		N
	AddressInfoCallsQuery		N
	AddressInfoSendAllCalls-Status		N
	AddressInfoQueueLogin-Audit		Y
	AddressInfoNumberOfIdle-Trunks		N
	AddressInfoNumberOf-TrunksInUse		N
	AddressInfoDatabaseValue		N
	AddressInfoDNStatus		Y
	AddressInfoQueueStatus		Y
TQueryLocation[a]	LocationInfoAllLocations	EventLocationInfo[d]	I
	LocationInfoLocationData		I
	LocationInfoMonitor-Location		I
	LocationInfoCancelMonitor-Location		I
	LocationInfoMonitorAll-Locations		I
	LocationInfoCancelMonitor-AllLocations		I
TQueryServer[a]		EventServerInfo	Y
User-Data Requests			
TAttachUserData (Obsolete)		EventAttachedDataChanged	Y
TUpdateUserData		EventAttachedDataChanged	Y

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TDeleteUserData		EventAttachedDataChanged	Y
TDeleteAllUserData		EventAttachedDataChanged	Y
ISCC (Inter Server Call Control) Requests			
TGetAccessNumber[b]		EventAnswerAccessNumber	I
TCancelReqGetAccess-Number		EventReqGetAccess-NumberCanceled	I
Special Requests			
TReserveAgent		EventAgentReserved	Y
TSendEvent		EventACK	I
TSendEventEx		EventACK	I
TSetCallAttributes		EventCallInfoChanged	I
TSendUserEvent		EventACK	Y
TPrivateService		EventPrivateInfo	Y
Network Attended Transfer Requests[e]			
TNetworkConsult		EventNetworkCallStatus	Y
TNetworkAlternate		EventNetworkCallStatus	Y
TNetworkTransfer		EventNetworkCallStatus	Y
TNetworkMerge		EventNetworkCallStatus	Y
TNetworkReconnect		EventNetworkCallStatus	Y
TNetworkSingleStep-Transfer		EventNetworkCallStatus	Y
TNetworkPrivateService		EventNetworkPrivateInfo	Y

Table 14: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
ISCC Transaction Monitoring Requests			
TTransactionMonitoring		EventACK	Y
		EventTransactionStatus	E

a. Only the requestor will receive a notification of the event associated with this request.

b. Since this feature request may be made across locations in a multi-site environment, if the location attribute of the request contains a value relating to any location other than the local site—except when the response to this request is `EventError`—there is a second event response that contains the same reference ID as the first event. This second event is either `EventRemoteConnectionSuccess` or `EventRemoteConnectionFailed`.

c. `TReconnectCall` will not function properly if Auto Hold Allowed is enabled on the Nortel Communication Server 1000 with SCCS/MLS.

d. Two subtypes are supported by `EventLocationInfo`: `LocationInfoLocationMonitorCanceled` and `LocationInfoAllLocationsMonitorCanceled`.

e. 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.

Support for Agent States and Workmodes

This section describes supported agent states and workmodes and how they are used, including self-correcting agent states.

Agent State Descriptions

LoggedOut

In this state, the agent is logged out. Calls are not delivered to the agent's position. T-Server processes this state normally.

NotReady

In this state, the agent is logged in but not ready. Calls are not delivered to the agent's position. T-Server processes this state normally.

Ready

In this state, the agent is logged in and ready. Calls are delivered to the agent. This is the normal state in which new calls arrive to the agent. T-Server processes this state normally.

AgentWalkAway

In this state, the agent has left his or her position but is still logged in. Effectively, the agent is `NotReady` and calls are not delivered.

When an agent enters the `AgentWalkAway` state, T-Server sends an `EventAgentNotReady` message with the `TAgentWorkMode` set to `AgentWalkAway`.

During an active call, when the agent returns, he or she returns to the state previous to `AgentWalkAway`:

- If the agent state was `NotReady`, then T-Server sends `EventAgentNotReady`.
- If the agent state was `Ready`, then T-Server sends `EventAgentReady`.

In both cases, the `AttributeAgentWorkMode` is set to `AgentReturnBack`.

AfterCallWork

In this state, the agent is logged in but is conducting post-call work. Agents may enter this state before the call has ended.

`AfterCallWork` is not fully supported on the switch. T-Server provides this state by simulating this capability in the software. When an agent enters the `AfterCallWork` state, T-Server sends an `EventAgentNotReady` message with the `TAgentWorkMode` set to `AgentAfterCallWork`. To leave this state, the agent must either log out or specifically become `Ready`.

Self-Correcting Agent States

The Nortel Communication Server 1000 with SCCS/MLS switch does not provide the capability to query for agent states. Therefore, T-Server makes certain assumptions about the agents. Release 7.0 of T-Server has the added capability to self-correct when its agent states are out of synchronization because of missing or erroneous CTI link messages, software errors, a system change (such as link failure or startup), or high-availability (HA) events.

The purpose of self-correcting agent states is to resynchronize T-Server to the switch when possible. However, this does not guarantee that T-Server synchronization will be accurate in all cases.

Normally, this feature is transparent to the user. However, in a few cases, switch message limitations can prevent T-Server from correctly interpreting the agent state, as discussed below.

Agent Logged In on T-Server Startup or Reconnect

When T-Server starts up, or when it reconnects following a network problem or HA event, the agent states are unknown. Initially, T-Server assumes that all agent states are unknown, denoted by a `-1` in status indications.

If a `TRequestAgentLogin` occurs for an agent who is already logged in, but for whom T-Server indicates an unknown state, the switch replies `The set is in the target state`. T-Server cannot determine whether the agent is `Ready` or `NotReady`. If the `nrdy-after-login` configuration option is set to `on` (default), T-Server distributes `EventAgentLogin` and `EventAgentNotReady` messages. If this option is set to `off`, T-Server distributes `EventAgentLogin` and `EventAgentReady` messages.

If T-Server cannot determine whether the agent is logged in or not, and if an attempt is made to make the agent `Ready` or `NotReady` (through `TRequests` or the phone set keys) while the agent is in an unknown state, T-Server rejects the request and leaves the agent in an unknown state. T-Server does this because the switch sends indications of `ready` or `not ready` independently of whether the agent is logged in or logged out. Since the Genesys Agent State Model allows an agent to be only `Ready` or `NotReady` when logged in, T-Server cannot determine the correct state of the agent unless it has already obtained information that the agent is logged in.

If an agent observes the inability to transition from `Ready` to `NotReady` (or vice versa), then the agent must try to log in again through a `TRequestAgentLogin`, or the agent must log out and log back in again. Only then can T-Server accurately reflect the agent state.

Note: These special cases occur rarely (only on startup or as a result of some transient error condition) and should not arise in normal system operation.

If an agent is already logged in, but T-Server has the agent in an unknown state, and the agent's ACD Queue sends a call to the agent, T-Server interprets that event as meaning that the agent is logged in and ready. According to the Agent State Model, new calls arriving at an agent's position can mean nothing else, and T-Server transitions the agent state accordingly. However, switch messages do not always allow T-Server to determine `AgentID`. In such cases, T-Server will synchronize the agent state at the DN, but it will report the `AgentID` only if this is desirable according to the configuration settings. See "Agent States Configuration Options" on page 151 for details.

T-Server no longer reports multiple successful login or logout messages when the link response indicates `Set is in target state`. When T-Server receives this error message for login and logout requests, it now performs the following actions:

- If the agent state stored internally in T-Server is the same as the actual state on the switch, T-Server sends an `EventError` event to the client with an `ErrorCode 186 - Set is in target state` message.
- If the agent state stored internally in T-Server is *not* the same as the agent state on the switch, and the value of the configuration option, `update-login-on-err`, is set to `on` (default), T-Server updates the agent's internal state and distributes events that indicate a change in agent state. If the value of this configuration option is set to `off`, T-Server sends an `EventError` event to the client with an `ErrorCode 186 - Set is in target state` message.

Note: Genesys recommends setting the value of `update-login-on-err` configuration option to `off` if hotseating is in use.

Feature Configuration

The following configuration options support the Agent States and Workmode feature:

- `nrdy-after-login`
- `update-login-on-err`

Agent States Configuration Options

Starting with the 7.5 release, two configuration options were introduced to enhance the self-correcting agent states feature:

- `default-agent-id-is-position`
- `default-agent-id`

T-Server uses these configuration options to determine whether to populate `AttributeAgentID` in `EventAgentLogin`, `EventAgentReady`, and `EventAgentNotReady`, after the self-correcting agent state logic was applied to an agent.

The value of one of the configuration options, `default-agent-id-is-position`, is set at the T-Server `Application` level and affects `AgentID` determination at any agent position DN of the corresponding switch. The value of the other configuration option, `default-agent-id`, is set at the `DN object` level and allows you to define the self-correcting T-Server behavior with respect to an individual agent position.

Support for the TAlternateCall Function

T-Server supports the TAlternateCall function, which uses the Nortel Swap feature that allows agents to switch between two calls (main and consultation calls) and also to transfer them if needed.

Feature Configuration

The following configuration option supports the TAlternateCall function feature:

—enable-consult-swap

Note: Consult your Nortel representative for information about the Swap feature support.

Support for Incoming UUI Data

You can now specify the way T-Server stores the User-to-User Information (UUI) data.

Feature Configuration

The following configuration option supports the Incoming UUI Data feature:

—uudata-attach-type

Note: The switch must be correctly configured and the UUI data correctly formatted to allow the switch to propagate the UUI data to T-Server. Refer to the Nortel documentation for details.

Support for Timed After Call Work (TACW)

T-Server supports the Timed After Call Work (TACW) feature that allows an agent, after a call is released, to remain in a NotReady (ACW) state for a pre-configured amount of time.

To configure TACW, you must complete the following steps:

- Enable the feature through the soft-tacw-support configuration option (see below).
- Configure the wrap-up time either individually, at the agent level, or globally, at the T-Server level. At the agent level, the wrap-up time is configured using the Wrap-up Time property of the Person (Agent) object

in Configuration Manager. At the T-Server level, the wrap-up time is configured in the corresponding T-Server Application using the `soft-wrap-up-time` configuration option (see below).

- In order for T-Server to know which extensions are associated with a position, use either the `terminal-id` or the `set-discovery` configuration options.

TACW, when configured, will place an agent into the `NotReady` (ACW) state after the agent releases a call if the following conditions are met:

- The call is a business call. All calls released from positions are business calls. A call released from an extension is a business call only if it has passed through an ACD Queue or a Routing Point.
- The agent is in the `Ready` state.
- The agent has no calls on its associated extension.

After the configured wrap-up period, the agent will be placed into the `Ready` state if the following conditions are met:

- The agent is not engaged in a call on its position or associated extension. Otherwise the agent is placed into the `Ready` state after the call is completed.
- The agent has not manually changed the agent state.

Note: For self-correcting agent states, by default the agent-id is unknown. This makes the wrap-up time information unavailable to T-Server. This can be resolved with the `default-agent-id-is-position` configuration option.

Feature Configuration

The following configuration options support the TACW feature:

- `default-agent-id-is-position`
- `set-discovery`
- `soft-tacw-support`
- `soft-wrap-up-time`
- `terminal-id`

Support for MLS IP Call Recording

T-Server supports the MLS IP Call Recording feature.

Two new private service requests are available to start and stop recording on a specific DN:

- Start Recording (TPrivateService)
  Service ID 32005
- Stop Recording (TPrivateService)
  Service ID 32006

The start recording private service request supports the following mandatory parameters via extensions key-value pairs:

- `TxIPAddress` (string): The value is the transmit IP address (x.x.x.x).
- `TxPort` (int): The value is the transmit IP port.
- `RxIPAddress` (string): The value is the receive IP address (x.x.x.x).
- `RxPort` (int): The value is the receive IP port.

The start recording private service request supports the following optional parameters through the `Extensions` attribute key value pairs:

- `WarningTone` (int): The value can be `0` or `1`. Setting the value to `0` turns off the warning tone, and setting the value to `1` turns on the warning tone. The default value is `0`.

An `EventACK` event is generated in response to a successful start and stop record request.

Note: The following prerequisites are listed in the *Nortel Contact Center Manager Meridian Link Services Interface Specification - December 2006*:

- Succession Release 4.5.
- Symposium Call Center Server 5.0 (with specific IP Call Recording PEP) or Contact Center Manager 6.0.
- Nortel IP Client Phase 2 sets with ICRA CLS.
- Adequate AST ISMs provisioned to ensure all sets to be recorded can be associated.

Feature Configuration

Use the following configuration option for supporting automatic set discovery at registration:

—`set-discovery`

Support for Trunk Optimization

Nortel Communication Server 1000 with SCCS/MLS supports the trunk anti-tromboning optimization scenario.

Anti-tromboning occurs when Switch 1 routes a call to remote Switch 2 using an outbound trunk, and Switch 2 then routes the call back to Switch 1.

Switch 1 detects that the call is using both an outbound and an inbound trunk. This is called “tromboning” because a diagram of the call flow looks like a trombone slide. Anti-tromboning reroutes the call away from these redundant trunks.

However, use of this trunk optimization scenario depends largely on the details of the switch environment. Please contact Genesys Technical support for details on availability of this support in your environment.

Limitations to Support for Trunk Anti-Tromboning

Trunk Anti-Tromboning (TAT) support has these known limitations:

Supported

- Calls between two Nortel Communication Server 1000 with SCCS/MLS switches *only.*

Not Supported

- Calls between Nortel Communication Server 1000 with SCCS/MLS and Nortel Communication Server 2000/2100 switches.
- Call scenarios involving Tandem Nodes.
- Call Triangulation scenarios.
- Network Call Pickup scenarios.
- Network Call Forwarding scenarios.
- Virtual Network Services scenarios.
- Call scenarios in which a transfer back is completed while the call is waiting for routing on a CDN.
- Call scenarios in which the original inbound call was never presented to an agent on the first switch. One example is when an inbound call to a CDN on switch A is routed to switch B, and then transferred back to switch A.
- Manual call transfer scenarios due to absence messages from the link for the consultation leg of the call.

Known Issues

The following known issues apply to T-Server’s support for TAT:

- During a call’s initial step, it must be dialed or transferred by a CTI request to the remote switch. If the call arrives at the remote switch by some other means (for instance, if it is routed there, or is sent there because of call overflow, or is transferred to an unmonitored DN that then transfers it there), TAT is not processed when the transfer back is complete.

- When a call returns back to the initial switch, it must be answered or routed before a remote agent completes a transfer.

Support for Advanced Features

T-Server also supports the following advanced features:

- `Emergency` key
- `Call Supervisor` key
- `Activity Code` key

Emergency Key

Pressing the `Emergency` key on the agent's phone set initiates a `no-hold` conference to the agent's supervisor, and if recording hardware is installed on the switch, the call is recorded. For the call leg to be established, the supervisor must press the `Answer Emergency` key to answer the call. The call leg to the supervisor is a special switch-provided function, and the switch does not indicate any call states for the call leg. The call is therefore considered an unmonitored call.

TClient Invocation of Emergency

The Emergency feature can be invoked from a `TClient` by sending T-Server a `TPrivateService` message with the `serviceID` set to `32001` (decimal). Refer to the *Genesys Events* and *Models Reference Manual* and *Voice Platform SDK 8.x .NET (*or *Java) API Reference* for additional details. When T-Server receives the `TPrivateService` request, it sends the `SetFeature:Emergency` message to the switch on behalf of the DN specified. Further processing is handled in the `Emergency Notification` message, as discussed in "T-Server Processing of Emergency and Call Supervisor Notification."

Call Supervisor Key

Pressing the `Call Supervisor` key on the agent's phone set when the agent is not in an ACD call initiates a call to the agent's supervisor. Pressing this key when the agent is in an ACD call, places the call on hold and initiates a conference leg to the agent's supervisor. The supervisor must then press the `Answer Agent` key to complete the call leg. The agent may then conference the supervisor into the call or release the supervisor call leg and retrieve the call. The call leg to the supervisor is a special switch-provided function, and the

switch does not indicate any call states for the call leg. The call is therefore considered an unmonitored call.

Note: T-Server supports Call Supervisor functionality only for Symposium link version SCCS 4.2 and higher.

TClient Invocation of Call Supervisor

This feature may be invoked from a TClient by sending T-Server a `TPrivateService` message with the `serviceID` set to `32002` (decimal). Please refer to the *Genesys Events* and *Models Reference Manual* and *Voice Platform SDK 8.x .NET (*or *Java) API Reference* for additional details. When T-Server receives the `TPrivateService` request, it sends the `SetFeature:CallSupervisor` message to the switch on behalf of the DN specified. Further processing is handled in the `Call SupervisorNotification` message, as discussed in "T-Server Processing of Emergency and Call Supervisor Notification."

T-Server Processing of Emergency and Call Supervisor Notification

When T-Server receives the `Emergency Notification` message from the switch, it sends an `EventPrivateInfo` message with an `Event` field value of `32001` (decimal) to all TClients. The details of the previous call in-progress are included in this message. In addition to standard event information, the `AttributeExtensions` contains additional call details.

Specifically, the key `AgentAction` is set to `Emergency ON` and the keys `OrigAddress` and `DestAddress` have data from the origination and destination location of the call, as applicable. When T-Server receives notification from the switch that the emergency has ended, it sends a `TPrivateService` event with the `serviceID` value set to `32001` and populates the `Extensions` attribute with the key `AgentAction` and the string value `Emergency OFF`.

When T-Server receives the `Call Supervisor` message from the switch, it sends an `EventPrivateInfo` message with an `Event` field value of `32002` (decimal) to all TClients. The details of the previous call in-progress are included in this message. In addition to standard event information, the `AttributeExtensions` contains additional call details. Specifically, the key `AgentAction` is set to the value of `Call Supervisor` and the keys `OrigAddress` and `DestAddress` have

data from the origination and destination location of the call, as applicable. See Table 15.

Table 15: Values Used in Private Requests and Events

Service	ID	Event
Emergency	32001	32001
Call Supervisor	32002	32002

Refer to the *Genesys Events* and *Models Reference Manual* and *Voice Platform SDK 8.x .NET (*or *Java) API Reference* for additional details.

Activity Code Key

With Nortel Communication Server 1000 with SCCS/MLS, an agent may signal an `Activity Code` when changing state from `Ready` to `NotReady` (or from `NotReady` to `Ready`). On the phone set, this is programmed as a special key that flashes. If the agent presses the `Activity Code` key, he or she can enter a sequence of digits on the keypad and then press the `Activity Code` key again, which sends a special switch message that T-Server can evaluate.

Note: T-Server supports Activity Codes functionality only for Symposium link version SCCS 4.2 and higher.

TClient Invocation of Activity Codes

An agent can use `TRequestAgentReady` or `TRequestAgentNotReady` and include the Activity Code in extensions. This simulates pressing the actual `Activity Code` key on the phone set. To use this feature, the `TRequest` must include the string key `ReasonCode` in the `TAttributeExtensions` and must set the string data for that key to the desired `Activity Code`.

T-Server Processing of Activity Codes

T-Server responds to a successful `Activity Code` request by sending either an `EventAgentReady` or `EventAgentNotReady` (depending on the actual agent state) with details in the `TAttributeExtensions`. This is identical to the `TRequest` format, using the string key `ReasonCode` with the string data for that key set to the desired `Activity Code`.

Support for Emulated Agent States

T-Server provides a fully functional emulated-agent model that you can use.

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

- Login and logout (`TAgentLogin`/`TAgentLogout`)
- Agent set ready (`TAgentSetReady`)
- Agent set not ready—using various work modes (`TAgentSetNotReady`)

Feature Configuration

The following configuration option supports the Emulated Agents States feature:

—`soft-login-support`

If the value for the `soft-login-support` configuration option is set to `true`, then all agent functionality is emulated. If the value for this option is set to `false`, then all agent functionality is switch-based.

Note: The *Genesys Events and Models Reference Manual* and the *Voice Platform SDK 8.x .NET (or Java) API Reference Manual* define the agent state/agent mode transitions that are permissible.

Support for the Nortel Contact Center 6.0 Standby Server

T-Server supports a `warm standby` type of connection to the Nortel Contact Center 6.0 Standby Server. The Standby Server provides two links for a connection with T-Server.

T-Server Configuration

T-Server must have both of the Standby Server links configured in the `TServer` or CTI-Link sections with the necessary link information.

T-Server Connection Procedure

The following steps describe how T-Server attempts to connect to both of the links:

1. T-Server attempts to connect to both of the links.
2. When one link successfully connects, T-Server stops attempting the connection to the other link.

3. If the connected link ever fails or drops, T-Server goes back to Step 1.

Use of the Extensions Attribute

T-Server for the Nortel Communication Server 1000 with SCCS/MLS supports use of the `Extensions` attribute as detailed in Table 16. T-Server populates the `Extensions` attribute in `EventAddressInfo` and `EventPartyInfo` as described in the latest version of the *Genesys Events* and *Models Reference Manual*. In addition, T-Server uses the `Extensions` attribute to distribute a call origination address, origination address type, destination address, and destination address type with each message that delivers call information—for example, `ConnectionID`.

T-Server can take the call origination address and origination address type from the `Origination Address` or `Other Device` Information Element (IE) of the Mlink message that prompted delivery of the first `EventDialing/EventQueued/EventRouteRequest/EventRinging` messages. This information does not change during the call. T-Server takes the destination address and destination address type from the `Destination Address` IE of the switch message and from the `StatusChanged/CallOffered` messages. This information might change during the call.

Table 16 indicates how T-Server for Nortel Communication Server 1000 with SCCS/MLS supports the use of the `Extensions` attribute.

Table 16: Use of the Extensions Attribute

Attribute Extensions					
Request/ Event	**Key**	**Value Type**	**Q.931 Type of Number (Byte 0)**	**Mlink DN Type (Byte 1)**	**Value Description**
EventDialing EventQueued EventRouteRequest EventRinging	OrigAddress, DestAddress	string	—	—	Address digits

Table 16: Use of the Extensions Attribute (Continued)

Attribute Extensions					
Request/ Event	**Key**	**Value Type**	**Q.931 Type of Number (Byte 0)**	**Mlink DN Type (Byte 1)**	**Value Description**
EventDialing EventQueued EventRouteRequest EventRinging	OrigAddrType, DestAddrType	binary value	0x00	0x00	Unknown
			0x10	0x01	International
			0x20	0x02	National
			0x30	0x03	Special [Q.931: Network-specific] number
			0x40	0x04	Subscriber number
			0x00	—	For any other cases
TSendDTMF	ToneDuration	integer	—	—	Used to specify the duration of each tone, in a hundredth (.01) of a second increments. The valid range of values accepted by the switch for this parameter is from `0` to `560` (`0` to `5.6` seconds).
	PauseDuration	integer	—	—	Used to specify the duration of the pause between tones, in a hundredth (.01) of a second increments. The valid range of values accepted by the switch for this parameter is from `0` to `560` (`0` to `5.6` seconds).
TSendDTMF	PauseSymbol Duration	integer	—	—	Used to specify the duration of the pause created by the comma symbol (“,”) when inserted into the DTMF string, in a hundredth (.01) of a second increments. The valid range of values accepted by the switch for this parameter is from `0` to `560` (`0` to `5.6` seconds).
TAgentSetReady TAgentSetNotReady	ReasonCode	integer	—	—	Changes agent’s activity code.

Table 16: Use of the Extensions Attribute (Continued)

Attribute Extensions					
Request/ Event	**Key**	**Value Type**	**Q.931 Type of Number (Byte 0)**	**Mlink DN Type (Byte 1)**	**Value Description**
EventAgentReady EventAgentNot Ready	ReasonCode	integer	—	—	Indicates change to Agent's activity code to the nominated value.
T-Server Common Part Extensions					
EventServerInfo	sdn-licenses-in-use	integer	—	—	Specifies how many SDN licenses are currently in use.
	sdn-licenses-available	integer	—	—	Specifies how many SDN licenses are currently available.

DN out-of-service State Support

If T-Server is unable to register a DN configured in the Configuration Layer, T-Server places it into an out-of-service state, which has the following consequences:

- Any client registering this DN is shown that its status is `out-of-service`. Clients are unable to perform requests on this DN (except register/unregister).
- The T-Server periodically attempts to re-register out-of-service DNs (see the configuration option, "out-of-service-retry-interval" on page 240).
- If the re-register attempt is successful, the DN is placed into an idle (in-service) state and `EventDnBackInService` is raised for the DN.
- If the T-Server receives an unsolicited `DirectoryNumberReleaseIndication` from the switch for an in-service DN, the T-Server changes the DN state to out-of-service and raises `EventDnOutOfService` for the DN.

For more information about these events, refer to the *Genesys Events and Models Reference Manual.*

Note:
- This feature applies to the Symposium link only (not the Meridian link).
- All DNs to be used in a customer environment must be registered in the Configuration Layer.

T-Server Error Messages

The following tables present the complete set of error messages T-Server distributes with an `EventError` event.

Connection Status Error Messages

Invalid Parameters

Table 17: Invalid Parameters

T-Library Error Code	Symbolic Name	Description	Switch Error Code
51	TERR_UNSUP_OPER	Requested operation not supported by T-Server	—
53	TERR_INVALID_ATTR	Attribute in request was invalid	—
56	TERR_INV_CONNID	Conn ID in request is invalid	—
58	TERR_OUT_OF_SERVICE	Requested operation in on out-of-service DN	—
59	TERR_NOT_CONFIGURED	DN is not configured in CME	—
60	TERR_INV_CALL_TN	Invalid Calling TN	0A00
61	TERR_INV_CALL_DN	Invalid Calling DN	0A01
70	TERR_INCM_CALL_DN	Incomplete Calling DN	0A02
71	TERR_INV_CALD_DN	Invalid Called DN	0A03
72	TERR_INCM_CALD_DN	Incomplete Called DN	0A04
73	TERR_INCM_CALD_TN	Incomplete Called TN	0A05
74	TERR_INV_ORIG_MANN	Invalid Origination Manner	0A06
75	TERR_INV_DEST_MANN	Invalid Destination Manner	0A07
76	TERR_INV_ORIG_UTYPE	Invalid Origination User Type	0A08
77	TERR_INVLD_CSTM_NUM	Invalid Customer Number	0A09
78	TERR_SYS_OR_DB_ERR	System or Database Error	0A0A

Unsuccessful Call Origination

Table 18: Unsuccessful Call Origination

T-Library Error Code	Symbolic Name	Description	Switch Error Code
79	TERR_ORIG_PTY_BSY	Origination Party Busy	0B00
80	TERR_ORIG_RSR_BLK	Origination Resource Blocking	0B01
81	TERR_ORIG_SET_MTN	Origination Set Maintenance Busy	0B02
82	TERR_ON_HOOK	Is On Hook	0B03
83	TERR_ORIG_DN_BUSY	Origination DN Busy	0B04
84	TERR_ORIG_RING	Origination Ringing	0B05
85	TERR_UNABL_DISC_ORIG	Unable To Disconnect Origination	0B06
86	TERR_ORIG_ACCS_BLK	Origination Access Blocking	0B07
87	TERR_ORIG_IN_PHOLD	Origination On Permanent Hold	0B08
88	TERR_ORIG_SYS_ERR	Origination System Error	0B0A
89	TERR_ORIG_END	Origination End	0B0B
90	TERR_ORIG_PTY_IN_ACD	Origination Party in ACD	0B0C

Unsuccessful Call Termination

Table 19: Unsuccessful Call Termination

T-Library Error Code	Symbolic Name	Description	Switch Error Code
91	TERR_TERM_PTY_BUSY	Terminating Party Busy	0C00
92	TERR_DEST_RSR_BLK	Destination Resource Blocking	0C01
93	TERR_DEST_INV_STATE	Destination Invalid State	0C02
94	TERR_DEST_ACCS_BLK	Destination Access Blocking	0C07
95	TERR_DEST_SYS_ERR	Destination System Error	0D0A

Unsuccessful Conference or Transfer Operation

Table 20: Unsuccessful Conference or Transfer Operation

T-Library Error Code	Symbolic Name	Description	Switch Error Code
96	TERR_CANT_COML_TRN	Cannot Complete Transfer	0D00
97	TERR_CANT_INIT_TRN	Cannot Initiate Transfer	0D01
98	TERR_CANT_CMPL_TRN	Cannot Complete Transfer	0D02
99	TERR_CANT_RTR_ORG	Cannot Retrieve Original Call	0D03

Common Error Messages

Table 21: Common Error Messages

T-Library Error Code	Symbolic Name	Description	Switch Error Code
100	TERR_UNKNOWN	Unknown Cause	0000
101	TERR_BAD_ASSOC_ID	Bad Association ID	0002
102	TERR_MSG_TYPE	Bad Message Type	0003
103	TERR_UNEXP_ELEMENT	Unexpected Information Element	0004
104	TERR_MSG_NOT_PART	Message Not Part of Registered Service	0005
105	TERR_ELEM_MISSING	Information Element Missing	0006
106	TERR_BAD_AFF_ASSOC	Bad Affected Association ID	0007
107	TERR_BAD_MSG_LENGTH	Bad Message Length	0008
108	TERR_BAD_SEQ_NUMBER	Bad Sequence Number	0009
109	TERR_LINK_DOWN	Link Down or Bad Link Specified	000A
110	TERR_REQ_IN_PROGRESS	Request Already in Progress	000B
111	TERR_TOO_MANY_REQ	Too Many Outstanding Requests	000C
112	TERR_MSG_OUT_OF_SEQ	Message Out of Sequence (for example, application tries to open a voice file before sending a message to logon mailbox)	000D

Error Messages in Application Registration Response

Table 22: Error Messages in Application Registration Response

T-Library Error Code	Symbolic Name	Description	Switch Error Code
113	TERR_ASSOC_TAB_FULL	Association Table Is Full	0502
114	TERR_APPL_TAB_FULL	Application Table Is Full	0503
115	TERR_APPL_EXISTS	Application Already Exists	0504
116	TERR_BAD_MACH_NAME	Bad Meridian 1 Machine Name	0505
117	TERR_BAD_HOST_NAME	Bad Host Machine Name	0506
118	TERR_SERV_UNAVAIL	Requested Service Unavailable	0507
119	TERR_BAD_PASSWD	Bad Password	0508
120	TERR_POLL_TIMEOUT	Polling Timeout	0509
121	TERR_BAD_MAIL_NAME	Bad Meridian Mail Name	050A

Error Messages in DN Registration Response

Table 23: Error Messages in DN Registration Response

T-Library Error Code	Symbolic Name	Description	Switch Error Code
122	TERR_CANT_REG_DNS	Cannot Register All DNs	0702
123	TERR_DN_NOT_EXIST	DN For Association Does Not Exist	0703
124	TERR_DN_TAB_FULL	DN Table Is Full	0704
125	TERR_DN_ALRDY_REG	DN Already Registered	0705
126	TERR_CUST_REG	Customer Number Must Be Registered to Register a DN(s)	0706
127	TERR_CANT_REMOVE_DN	Cannot Remove DN	0707
128	TERR_BAD_DN_TYPE	Bad DN Type for DN Registration	0708

Link Maintenance Error Messages

Table 24: Link Maintenance Error Messages

T-Library Error Code	Symbolic Name	Description	Switch Error Code
129	TERR_PROC_NOT_EXIST	Link Process Does Not Exist	0902
130	TERR_LINK_ID_EXIST	Link ID Already Exists	0903
131	TERR_MACH_NAME_EXIST	Meridian 1 Machine Name or Host ID Already Exists	0904
132	TERR_BAD_LINK_ID	Bad Link ID	0905
133	TERR_LINK_ALRDY_EST	Link Already Established	0906
134	TERR_LINK_ALRDY_DIS	Link Already Disabled	0907
135	TERR_OPEN_CONF_FILE	Error in Opening Configuration File	0908
136	TERR_LINK_CONF_FAIL	Link Configuration Failed	0909
137	TERR_LINK_ENBL_FAIL	Enable Link Command Failed	090A
138	TERR_LINK_DIS_FAIL	Disable Link Command Failed	090B
139	TERR_LNK_CMD_NOT_SUP	Link Command Not Supported	090C
140	TERR_LNK_STS_REQ	Link Statistics Request Failed	090D
141	TERR_LNK_CONF_LARGE	Link Configuration Information Is Too Large	090E
142	TERR_LNK_CMD_FAILED	Link Command Failed Because of Reconfiguration of Associated Link	090F
143	TERR_TRACE_ALRDY_EN	Trace Already Enabled	0910
144	TERR_TRACE_ALRDY_DIS	Trace Already Disabled	0911
145	TERR_SFW_NOT_EQIPD	Link to Meridian 1 Failed Because Required Software Option Not Equipped	0912
146	TERR_ID_MISMATCH	Link to Meridian 1 Failed Because of Meridian 1 ID Mismatch	0913
147	TERR_NO_LINK_RESPND	No Link Responding	0914

Message Facility Error Messages

Table 25: Message Facility Error Messages (Recording, Monitoring, Statistics, Filtering)

T-Library Error Code	Symbolic Name	Description	Switch Error Code
148	TERR_FCL_ALRDY_ENB	Facility Already Enabled	0B02
149	TERR_FCL_ALRDY_DIS	Facility Already Disabled	0B03
150	TERR_MSG_ALRDY_SET	Message(s) Already Set	0B04
151	TERR_MSG_ALRDY_CLR	Message(s) Already Cleared	0B05
152	TERR_UNABL_REC_FILE	Unable to Open/Write/Close Recording File	0B06
153	TERR_BAD_AFFCT_MSG	Bad Affected Message	0B07
154	TERR_CANT_CLR_ALL	Cannot Clear All (filter, monitor, or record)	0B08

Voice-Processing Error Messages

Table 26: Voice-Processing Error Messages

T-Library Error Code	Symbolic Name	Description	Switch Error Code
155	TERR_ACCNT_INACT	Account Inactive (timeout expired)	0C00
156	TERR_ADMIN_SHUTDOWN	Meridian Mail Shutdown by Administrator	0C01
157	TERR_SYSTEM_ERR	Meridian Mail System Error	0C02
158	TERR_VCHAN_NOT_ANS	Incoming Voice Channel Not Answered in 15 Seconds	0C03
159	TERR_MANY_BAD_LOGIN	Too Many Bad Login Attempts	0C04

Flow-Control Error Messages

Table 27: Flow-Control Error Messages

T-Library Error Code	Symbolic Name	Description	Switch Error Code
160	TERR_FLOW_CTL_1	Flow Control—Level 1	0D02
161	TERR_FLOW_CTL_2	Flow Control—Level 2	0D03
162	TERR_FLOW_CTL_3	Flow Control—Level 3	0D04
163	TERR_FLOW_CTL_CLR	Flow-Control Condition Cleared	0D05

System Error Message

Table 28: System Error Message

T-Library Error Code	Symbolic Name	Description	Switch Error Code
164	TERR_BAD_SYSTEM_CMD	Bad System Command	0F00

Error Messages in Basic Call Management

Table 29: Error Messages in Basic Call Management

T-Library Error Code	Symbolic Name	Description	Switch Error Code
165	TERR_ACCESS_RESTICT	Access Restricted	1002
166	TERR_RES_UNAVAIL	Resource Unavailable	1003
167	TERR_INV_CUST_NUM	Invalid Customer Number	1004
168	TERR_INV_ORIG_ADDR	Invalid Origination Address	1005
169	TERR_INV_DEST_REQ	Invalid Destination Request	1006
170	TERR_INV_MANNER	Invalid Manner	1007
171	TERR_UNSUCC_RETRV	Unsuccessful Retrieve Original	1008
172	TERR_UNSUCC_TRANSFER	Unsuccessful Transfer	1009

Table 29: Error Messages in Basic Call Management (Continued)

T-Library Error Code	Symbolic Name	Description	Switch Error Code
173	TERR_UNSUCC_CONFER	Unsuccessful Conference	100A
174	TERR_UNSUCC_ANSWER	Unsuccessful Answer Request	100B
175	TERR_UNSUCC_RELEASE	Unsuccessful Release Request	100C
176	TERR_UNSUCC_REFER	Refer to Connection Status IE For Information	1070

SetFeatureInvocation Fault Messages

Table 30: SetFeatureInvocation Fault Messages

T-Library Error Code	Symbolic Name	Description	Switch Error Code
177	TERR_TARG_DN_INV	Target DN Invalid	2004
178	TERR_TARG_DN_NOT_AST	Target DN Not AST	2005
179	TERR_FTR_NOT_INVOK	Feature Could Not e Invoked	2007
180	TERR_FTR_NOT_CFG	Feature Not Configured to Set	2008
181	TERR_FTR_OUT_OF_RNG	Requested Feature Out of Valid Range	2009
182	TERR_TARG_NOT_AGENT	Target Set Not ACD Agent	200A
183	TERR_TARG_VIRT_AGENT	Target Set Is a Virtual Agent	200B
184	TERR_MAINT_BUSY	Set Is Maintenance Busy	200C
185	TERR_SET_WRONG_STATE	Set Is in Wrong State for Invocation	200D
186	TERR_SET_TARG_STATE	Set Is in Target State	200E
187	TERR_ACD_LOGOFF	No NRDY/RDY While ACD Set Is Logged Out	200F
188	TERR_CUST_NRDY	Package C Customer Cannot Use NRDY With IDN Call	2010
189	TERR_FTR_IE_INV	Feature IE Is Missing or Invalid	2011

Table 30: SetFeatureInvocation Fault Messages (Continued)

T-Library Error Code	Symbolic Name	Description	Switch Error Code
190	TERR_DN_IE_INV	DN IE Is Missing or Invalid	2012
191	TERR_AGENT_ID_IE_INV	Agent ID IE Is Missing or Invalid	2013
192	TERR_AGENT_ID_INV	Agent ID Is Invalid	2014
193	TERR_CFW_DN_IE_INV	CFW DN IE Is Invalid	2015
194	TERR_CFW_TOO_LONG	The Call Forward DN Is Too Long	2016
195	TERR_CFW_DN_INV	The Call Forward DN Is Invalid	2017
196	TERR_INVOKE_CFW	User Is Invoking Call Forward	2018
197	TERR_MSB_NOT_SUPP	MSB/MSI Not Supported for 500/2500 Sets	2019
198	TERR_5ACD_STATUS	500/2500 ACD Agent Already Changed Status	201A
199	TERR_5ACD_RUNG	500/2500 ACD Agent Set Is Being Rung	201B
200	TERR_MANUAL_LOGIN	User Is Manually Logged In o the 500/2500 ACD Set	201C
201	TERR_OPT209_NOT_EQTP	Meridian Link Server Option 209 Is Not Equipped	201D

Release/Acquire Message-Failure Messages

Table 31: Release/Acquire Message-Failure Messages

T-Library Error Code	Symbolic Name	Description	Switch Error Code
202	TERR_RES_ACQ_ANOTHER	The Resource Is Already Acquired by Another Application	2020
203	TERR_RES_ALRDY_ACQ	The Resource Is Already Acquired by his Application	2021
204	TERR_RES_NOT_RELEASD	The Resource Is Not Released	2022

Table 31: Release/Acquire Message-Failure Messages (Continued)

T-Library Error Code	Symbolic Name	Description	Switch Error Code
205	TERR_CANT_ACQ_ALL	The Application Cannot Acquire the Complete Resource	2023
206	TERR_RES_TYPE_UNKNWN	The Resource Type Is Unknown	2024
207	TERR_AML_DOWN	The AML Is Down	2025
208	TERR_RES_TAB_FULL	The Resource Table Is Full	2026
209	TERR_CDN_NOT_CNG	The CDN Is Not Configured to Operate in Controlled Mode	2027
210	TERR_POLLTMR_OF_RNG	The Poll Timer Is Out of Range	2028
211	TERR_RES_ID_LNG_LONG	The Resource ID Length Is Too Long	2029
212	TERR_ADMIN_DEV_DIS	Device Disabled by Administration	202A
213	TERR_NO_RSP_OPER_REQ	No Response to Operation Request	202B
214	TERR_LOGON_TR_EXCEED	Logon Tries Exceeded	202C
215	TERR_NOT_EQTP_FTR	This Release of the Meridian 1 Software s Not Equipped to Operate This Feature	2030
216	TERR_RES_NOT_ACQ	The Resource Is Not Acquired by the Application	2031
217	TERR_REG_NOT_SET	Registration Not Set Up for Request	2032
218	TERR_INTERNAL_ERR	Internal Error	2033
219	TERR_BAD_RES_ID	Bad Resource ID	2034
220	TERR_NO_RES_AVAIL	No Internal Resource Available	2035
221	TERR_SRV_NOT_AVAIL	Service Not Available on Device	2036
222	TERR_DEV_NOT_AVAIL	Device Not Available	2037

Voice-Processing Failure Messages

Table 32: Voice-Processing Failure Messages

T-Library Error Code	Symbolic Name	Description	Switch Error Code
223	TERR_BAD_PARAM	Bad Parameter Passed to Function	3001
224	TERR_NO_RESULT_AVAIL	No Result Available Yet	3002
225	TERR_NO_RESULT_TMOUT	No Result, the Command Timed Out	3003
226	TERR_OUT_OF_MEM	Out of Memory (local)	3004
227	TERR_INV_APPL_CLASS	Invalid Application Class	3005
228	TERR_INV_CMD_BFR_ACQ	Command Invalid Before Acquire	3006
229	TERR_MUST_REGISTER	Must Register First	3007
230	TERR_MUST_DEREGIST	Must Unregister First	3008
231	TERR_DN_BUSY	DN Is Busy	3009
232	TERR_DN_NO_ANSWER	No Answer at DN	300A
233	TERR_CALL_REJECTED	Call Has Been Rejected	300B
234	TERR_CONN_ATMPT_FAIL	Call Connection Attempt Has Failed	300C
235	TERR_CALL_COLLISION	Call Resulted in Collision	300D
236	TERR_TIMEOUT_PRF_OP	Timeout Performing Operation	300E
237	TERR_DISCON_CALL	Call Has Disconnected	300F
238	TERR_NO_QUEUE_SPACE	Message Send Failed: No Queue Space	3010
239	TERR_INV_PROC_TYPE	Invalid Process Type	3011
240	TERR_SERR_ACCS_APIQ	System Error Accessing API Queue	3012
241	TERR_SERR_ACCS_EVQ	System Error Accessing Event Queue	3013
242	TERR_MON_FUNC_INS	Monitor Function Already Installed	3014
243	TERR_CLNT_NOT_MON	Client Is Not the Monitor Process	3015
244	TERR_WRONG_ACCS_VER	API Not Usable: Wrong ACCESS Version	3016

Table 32: Voice-Processing Failure Messages (Continued)

T-Library Error Code	Symbolic Name	Description	Switch Error Code
245	TERR_ACCS_SMFR	Could Not Access/Open a Semaphore	3017
246	TERR_NO_FILE	No File at Path Specified	3018
247	TERR_CANT_FORK	Could Not Fork Process at Path	3019
248	TERR_LNK_MNGR_DEAD	Link Manager Was Already Dead	301A
249	TERR_NOT_SPWN_LPM	Did Not Spawn LMP through m_StartLink	301B
250	TERR_DEAD_CHLD	Caller Had Dead Child Besides LMP	301C
251	TERR_LMT_TOO_LONG	LMP Took Too Long to Die	301D
252	TERR_LH_MM_CMD_FAILED	LH Not Synchronized with MM Command Failed	301E
253	TERR_LH_MM_CMD_SCCD	LH Not Synchronized with MM Command Succeeded	301F
254	TERR_LH_NOT_SYNCH	LH Not Synchronized with MM	3020
255	TERR_UNEXPCTD_VAL	LH Returned an Unexpected Value	3021
256	TERR_MON_RESTRICT	API Is Restricted From Monitor	3022
257	TERR_NO_LH_CONF	No LH Configuration File Found	3023
258	TERR_OP_NOT_CUR_SUP	Operation Not Currently Supported	3063
259	TERR_INV_PASSWD	Invalid Password	3066
260	TERR_NO_MM_ACCS	No MM ACCESS Toolkit Available	3067
261	TERR_SERVER_FULL	No Free Blocks, Server Is Full	3068
262	TERR_DISK_SPACE	No Free Disk Space in User Cabinet	3069
263	TERR_MUST_BE_LOGD	Must Be Logged On to Use This Command	306A
264	TERR_ACNT_ACCS_DEND	Access to Account Denied	306D
265	TERR_COMMAND_FAILED	Command Failed, Check SEER Console	306F
266	TERR_INV_ACCNT_TYPE	Invalid Account Type	3071

Table 32: Voice-Processing Failure Messages (Continued)

T-Library Error Code	Symbolic Name	Description	Switch Error Code
267	TERR_ALRDY_ACQUIRED	Already Acquired	3073
268	TERR_MNY_FAILD_MLOG	Too Many Failed m_Logon Attempts	3075
269	TERR_API_NOT_SUP	API Function Is Not Supported	3078
270	TERR_BAD_USER_ID	Bad User ID or Mailbox Number	307A
271	TERR_INV_FLAG	Invalid Flag (0 or 1 are valid)	3080
272	TERR_WARN_LOGGED	Warning: Logged on Elsewhere	3081
273	TERR_API_NOT_SUP_MM	API Being Used Not Supported by MM	3083
274	TERR_INV_CUST_SPEC	Invalid Customer Number Specified	3085
275	TERR_CANT_ISSUE_CMD	Cannot Issue Command While Logged In	3086
276	TERR_APPL_ACK_ENS	An Application Has Already Acquired ENS	3087
277	TERR_BE_ENS_APPL	Must Be ENS Appl to Invoke APINS	3088
278	TERR_NOT_AVAIL_OPT	Option Not Available to Customer	3096
279	TERR_ACQ_LIMIT_RCH	Max. # of Acquire Requests Reached	3097
280	TERR_SESS_RELEASED	Session Already Released by System	3098
281	TERR_NO_CONNECTION	No Connection Has Been Established	30C8
282	TERR_NO_VCHAN_AVAIL	No Voice Channel Available	30C9
-	-	Invalid Voice Start Position	30CB
283	TERR_INV_PLAY_POS	Invalid Play Position	30CC
284	TERR_INV_RECORD_POS	Invalid Recording Position	30CD
285	TERR_INV_DIRECTION	Invalid Direction (parameter)	30D0
286	TERR_VCHAN_IN_USE	Voice Channel Already in Use	30D3
287	TERR_NO_VCHAN_ACQ	No Voice Channel Has Been Acquired	30D4
288	TERR_NO_INC_CALL_ANS	No Incoming Call to Answer	30D5

Table 32: Voice-Processing Failure Messages (Continued)

T-Library Error Code	Symbolic Name	Description	Switch Error Code
289	TERR_ADDONCALL_FIRST	Must Call m_AddOnCall First	30D6
290	TERR_CHAN_ALRDY_ACPT	Channel Already Accepting Calls	30D7
291	TERR_OTHER_TEL_OPER	Other Telephony Operation in Progress	30D9
292	TERR_PLAY_CMD_IN_PRG	Play Command Already in Progress	30DF
293	TERR_INV_CMD_SEQ	Invalid Command Sequence	30E0
294	TERR_REC_CMD_IN_PRG	Record Command Already in Progress	30E1
295	TERR_VOICE_OPER_FLR	Voice Operation Failure	30E3
296	TERR_NO_VOICE_SEG	No Voice Segment to Play	30E4
297	TERR_AT_END_VSEG	At End of Voice Segment	30E5
298	TERR_TOO_MUCH_SLNCE	Ended Because Too Much Silence	30E7
299	TERR_RECORD_LIMIT	Recording Limit Has Been Reached	30E8
300	TERR_BAD_NUM_OF_SEGS	Bad Number of Segments Specified	30E9
301	TERR_SEG_QUEUE_FULL	Segment Play Queue Is Full	30EB
302	TERR_INV_DTMF_STRING	Invalid DTMF String	30EC
303	TERR_BAD_CONTEXT	Context Must Be SOUND/SILENCE	30ED
304	TERR_BAD_DURATION	Duration Must Be <= 5 Minutes	30EE
305	TERR_NO_PREV_DETECT	No Previous Detect in Progress	30EF
306	TERR_SND_DETECT_PRGR	Sound Detect Already in Progress	30F0
307	TERR_INST_EV_HANDLR	Must Install Event Handler First	30FA
308	TERR_NO_SUCH_ENTRY	No Such Entry Found in Directory	3135
309	TERR_UNABLE_ACCS_CAB	Unable to Access User Cabinet	3190
310	TERR_INV_FILE_HND	Invalid File Handle Passed to Command	3191
311	TERR_UNASGN_FILE_HND	Unassigned File Handle	3192

Table 32: Voice-Processing Failure Messages (Continued)

T-Library Error Code	Symbolic Name	Description	Switch Error Code
312	TERR_INV_COMMIT_FLAG	Invalid Commit Flag (parameter)	3193
313	TERR_BEGN_OF_FILE	Reached the Beginning of File	3195
314	TERR_CANT_OPEN_RD	Cannot Open Read File in Write Mode	3196
315	TERR_END_OF_FILE	Reached the End of File	3197
316	TERR_FILE_ALRDY_OPEN	File Is Already Open	3199
317	TERR_RONLY_FILE	Read-Only File: Not Committed	319A
318	TERR_CMD_RONLY_FILE	Cannot Do Command on Read-Only File	319B
319	TERR_INV_FNAME_FMT	Invalid Filename Format	319F
320	TERR_FILE_NUM_LIMIT	Maximum Open File Limit Reached	31A0
321	TERR_MFILEPTRN_FST	Must Call m_File Pattern First	31A3
322	TERR_FILE_NOT_EXIST	File Does Not Exist	31A4
323	TERR_INV_NEW_FLAG	Invalid New Flag Passed	31A9
324	TERR_INV_FILE_ACCSS	Invalid File Access Mode Used	31AA
325	TERR_INV_DEL_PARM	Invalid Delete Parameter	31AF
326	TERR_INV_FILE_CMD	Command Invalid on This File Type	31B0
327	TERR_SEG_NOT_FOUND	Segment ID Not Found on File	31B1
328	TERR_INV_FLD_LENGTH	Invalid Length on Field	31B2
329	TERR_SEGPATTRN_FST	Must Call m_SegPattern First	31B4
330	TERR_INV_SCRIPT_LNG	Invalid Script Length Field	31B5
331	TERR_ISS_SCR_RTR	Issue Script Retrieve Command First	31B6
332	TERR_NO_VOICE_SEG_FL	No Voice Segments in the File	31B7
333	TERR_TOO_MANY_SEG	Too Many Open Segment Files for Play	31B9
334	TERR_SCRPT_TOO_LONG	Script for Voice Segment Too Long	31BA

Table 32: Voice-Processing Failure Messages (Continued)

T-Library Error Code	Symbolic Name	Description	Switch Error Code
335	TERR_SEGS_LIMIT	Reached Maximum Number of Segments Allowed in File	31BC
336	TERR_BAD_VSEG_FTYPE	Bad Voice Segment File Type	31BD
337	TERR_INV_LANG	Invalid Language Specified	31BE
338	TERR_INV_SEG_EDT_POS	Invalid Segment-Editing Position	31C0
339	TERR_INV_SEG_EDT_OPR	Invalid Segment-Editing Operator	31C1
340	TERR_INV_AMOUNT	Invalid Amount Specified	31C2
341	TERR_NOT_MSG_FILE	File Is Not a Message File	31F4
342	TERR_INV_RECEIVER	Invalid Receiver in Address List	31FC
343	TERR_MSG_RECPTS_LIM	Exceeded Max. # of Message Recipients	31FD
344	TERR_INV_SUBJ_STR	Invalid Subject String	31FF
345	TERR_CANT_SEND_EMPT	Cannot Send an Empty Message	3200
346	TERR_CALLSND_RCVD_-MSG	CallSender/Reply Only on Received Messages	3201
347	TERR_MADDPATTERN_FST	Must Call m_AddrPattern First	3203
348	TERR_CANT_RPLY	Cannot Reply to External Message	3207
349	TERR_CANT_FRWD	Cannot Forward a Private Message	3208
350	TERR_NEED_RCVRS	Need One or More Receivers to Send	320A
351	TERR_MULT_NAMES	Multiple Names Matched, Specify	320B
352	TERR_CANT_SEND_INC	Cannot Send an Incoming Message	320C
353	TERR_DELAY_DLVRY	Delay Delivery Time Too Long	320D
354	TERR_RMT_SITE	Remote Site Not Recognized	320E
355	TERR_OPER_INVALID	Operations Invalid on System Messages	320F
356	TERR_CANT_RPLY_ALL	Cannot Reply All to Broadcast Message	3210

Table 32: Voice-Processing Failure Messages (Continued)

T-Library Error Code	Symbolic Name	Description	Switch Error Code
357	TERR_CANT_RPLY_AMIS	Cannot Reply All on AMIS Message	321A
358	TERR_NOT_FND_LIST	List Number Not Found	3258
359	TERR_INV_PDL	Invalid PDL List Number	3259
360	TERR_PDL_LIMIT	Exceeded Maximum Number of Entries in PDL	325A
361	TERR_UNACCS_USER_PRF	Unable to Access User Profile	325B
362	TERR_ADMIN_ACCS_ONLY	Restricted to Administration Access Only	326E
363	TERR_INV_BOX_NUMBER	Invalid Box Number	326F
364	TERR_INV_LAST_NAME	Invalid Last Name	3271
365	TERR_INV_FST_NAME	Invalid First Name	3272
366	TERR_INV_LIST_NUM	Invalid List Number	3273
367	TERR_SHORT_PASSWD	Password Too Short	3274
368	TERR_INV_GRT_TYPE	Invalid Greeting Type	3275
369	TERR_OLD_PASSWD_LOG	Old Password and Logged On Elsewhere	3276
370	TERR_OLD_PASSWD	Old Passwords Cannot Be Reused	3277
371	TERR_PERS_VRF_OPEN	Personal Verification Already Open	3278
372	TERR_GRTN_ALRDY_OPEN	Greeting Already Open	3279
373	TERR_NON_NUMERIC	Nonnumeric in Numeric Field	327A
374	TERR_NO_MATCH_BOX	No Matching Box Address in PDL	327C
375	TERR_MPDLPATTERN_FST	Must Call m_PDLPattern First	327D
376	TERR_NOT_PDL_FILE	Not a PDL File	327E
377	TERR_INV_EXT_MSG_TYP	Invalid External Message Type	327F
378	TERR_HILEV_BFRE_API	Set HiLev Flag Before Invoking API	32BC
379	TERR_INV_DIGIT	Invalid Digit in ExitDigits	32BD

Table 32: Voice-Processing Failure Messages (Continued)

T-Library Error Code	Symbolic Name	Description	Switch Error Code
380	TERR_DIG_TIMEOUT	Interdigit Timeout Occurred	32BE
381	TERR_KBUF_OVRFLOW	Key Buffer Overflow Occurred	32BF
382	TERR_API_INTRPTD	API Interrupted MM Event	32C0
383	TERR_INV_ITM	ItemToPlay in Invalid Format	32C1
384	TERR_INV_PLAYTYPE	InvalidPlayType Specified	32C2
385	TERR_NO_PLAYEND	PLAYEND Event Not Received	32C3
386	TERR_INV_DIR_NUM	Invalid Directory Number Passed	3326
387	TERR_INV_ANSWER_FLAG	Invalid Answer Flag	3328
388	TERR_DN_RESTR_PRF	DN Has a Restricted Prefix	332B
389	TERR_LH_TAB_FULL	LH Register Table Full	3384
390	TERR_LHT_TAB_FULL	LH Trans Table Full	338E

Call Status Error Messages

Table 33: Call Status Error Messages

T-Library Error Code	Symbolic Name	Description	Switch Error Code
400	TERR_INV_PRIO	Invalid Priority	1
401	TERR_INV_MESSG_LEN	Invalid Message Length	2
402	TERR_INV_ROUTE_ADDR	Invalid Route Address	3
403	TERR_INV_APPL_TYPE	Invalid Application Type	4
404	TERR_INV_MESSG_TYPE	Invalid Message Type	5
405	TERR_INV_MESSG_REFID	Invalid Message Reference ID	6
406	TERR_INV_CUSTOM_NUM	Invalid Customer Number	7
407	TERR_UNAVL_CALL_REG	Cannot Obtain Call Register	8
408	TERR_INV_CALL_REFID	Invalid Call Reference ID	9

Table 33: Call Status Error Messages (Continued)

T-Library Error Code	Symbolic Name	Description	Switch Error Code
409	TERR_CALL_PRSNT	Call Being Presented, Request Rejected	10
410	TERR_INAPPR_TRTM	Inappropriate First Treatment, Call in Default	11
411	TERR_NOT_ACQRD_CDN	Application Has Not Acquired the CDN	12
412	TERR_INV_SUBTYPE	Invalid Subtype	32
413	TERR_INV_MUSIC_ROUTE	Invalid Music Route or Destination	33
414	TERR_MUSIC_CONN_BLKD	Music Connection Blocked	34
415	TERR_INV_DEST_DN	Invalid Destination DN	33
416	TERR_INV_TONE_TREAT	Invalid Tone Treatment	33
417	TERR_DN_MATCHES_CDN	Destination DN Is the Originating CDN	34
418	TERR_TONE_CONN_BLKD	Tone Connection Blocked	34
470	TERR_PARTY_NOT_ON_CALL	Requested operation is for party that is not active on the specified call.	—

Network Attended Transfer/Conference Error Messages

Table 34: Network Attended Transfer/Conference Error Messages

T-Library Error Code	Symbolic Name	Description
1901	TERR_NATC_UNEXP_CONSULT	Unexpected request for `TNetworkConsult`
1902	TERR_NATC_UNEXP_ALTERNATE	Unexpected request for `TNetworkAlternate`
1903	TERR_NATC_UNEXP_RECONNECT	Unexpected request for `TNetworkReconnect`
1904	TERR_NATC_UNEXP_TRANSFER	Unexpected request for `TNetworkTransfer`
1905	TERR_NATC_UNEXP_MERGE	Unexpected request for `TNetworkMerge`

Table 34: Network Attended Transfer/Conference Error Messages (Continued)

T-Library Error Code	Symbolic Name	Description
1906	TERR_NATC_UNEXP_SST	Unexpected request for `TNetworkSingleStepTransfer`
1907	TERR_NATC_UNEXP_NPS	Unexpected request for `TNetworkPrivateService`
1908	TERR_NATC_UNEXP_MSG	Unexpected message

Chapter

11 Common Configuration Options

Unless otherwise noted, the common configuration options that this chapter describes are common to all Genesys server applications and applicable to any Framework server component. This chapter includes the following sections:

Note: Some server applications also support log options that are unique to them. For descriptions of a particular application's unique log options, refer to the chapter/document about that application.

Setting Configuration Options

Unless specified otherwise, set common configuration options in the `Options` of the `Application` object, using one of the following navigation paths:

- In Genesys Administrator—`Application` object > `Options` tab > `Advanced View (Options)`
- In Configuration Manager—`Application` object > `Properties` dialog box > `Options` tab

Warning! Configuration section names, configuration option names, and predefined option values are case-sensitive. Type them in Genesys Administrator or Configuration Manager exactly as they are documented in this chapter.

Mandatory Options

You do not have to configure any common options to start Server applications.

log Section

This section must be called `log`.

verbose

Default Value: `all`
Valid Values:

`all`	All log events (that is, log events of the Standard, Trace, Interaction, and Debug levels) are generated.
`debug`	The same as `all`.
`trace`	Log events of the Trace level and higher (that is, log events of the Standard, Interaction, and Trace levels) are generated, but log events of the Debug level are not generated.
`interaction`	Log events of the Interaction level and higher (that is, log events of the Standard and Interaction levels) are generated, but log events of the Trace and Debug levels are not generated.
`standard`	Log events of the Standard level are generated, but log events of the Interaction, Trace, and Debug levels are not generated.
`none`	No output is produced.

Changes Take Effect: Immediately

Determines whether a log output is created. If it is, specifies the minimum level of log events generated. The log events levels, starting with the highest priority level, are Standard, Interaction, Trace, and Debug. See also "Log Output Options" on page 218.

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 218). 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 `<number>`	Sets the maximum segment size, in kilobytes. The minimum segment size is `100 KB`.
`<number> MB`	Sets the maximum segment size, in megabytes.
`<number> hr`	Sets the number of hours for the segment to stay open. The minimum number is `1` hour.

Changes Take Effect: Immediately

Specifies whether there is a segmentation limit for a log file. If there is, sets the mode of measurement, along with the maximum size. If the current log segment exceeds the size set by this option, the file is closed and a new one is created. This option is ignored if log output is not configured to be sent to a log file.

expire

Default Value: `false`
Valid Values:

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

Changes Take Effect: Immediately

Determines whether log files expire. If they do, sets the measurement for determining when they expire, along with the maximum number of files (segments) or days before the files are removed. This option is ignored if log output is not configured to be sent to a log file.

Note: If an option's value is set incorrectly—out of the range of valid values— it will be automatically reset to `10`.

keep-startup-file

Default Value: `false`
Valid Values:

`false`	No startup segment of the log is kept.
`true`	A startup segment of the log is kept. The size of the segment equals the value of the `segment` option.
`<number> KB`	Sets the maximum size, in kilobytes, for a startup segment of the log.
`<number> MB`	Sets the maximum size, in megabytes, for a startup segment of the log.

Changes Take Effect: After restart

Specifies whether a startup segment of the log, containing the initial T-Server configuration, is to be kept. If it is, this option can be set to `true` or to a specific size. If set to `true`, the size of the initial segment 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_format

Default Value: `short`
Valid Values:

`short`	An application uses compressed headers when writing log records in its log file.
`full`	An application uses complete headers when writing log records in its log file.

Changes Take Effect: Immediately

Specifies the format of log record headers that an application uses when writing logs in the log file. Using compressed log record headers improves 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 218). 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 218.

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 222.

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

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

all

Default Value: No default value

Valid Values (log output types):

`stdout`	Log events are sent to the Standard output (`stdout`).
`stderr`	Log events are sent to the Standard error output (`stderr`).
`network`	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. Setting the `all` log level option to the `network` output enables an application to send log events of the `Standard`, `Interaction`, and `Trace` levels to Message Server. `Debug`-level log events are neither sent to Message Server nor stored in the Log Database.
`memory`	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends all log events. The log output types must be separated by a comma when more than one output is configured. For example:

```
all = stdout, logfile
```

Note: To ease the troubleshooting process, consider using unique names for log files that different applications generate.

alarm

Default Value: No default value

Valid Values (log output types):

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

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the Alarm level. The log output types must be separated by a comma when more than one output is configured. For example:

```
standard = stderr, network
```

standard

Default Value: No default value
Valid Values (log output types):

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

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Standard` level. The log output types must be separated by a comma when more than one output is configured. For example:

```
standard = stderr, network
```

interaction

Default Value: No default value
Valid Values (log output types):

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

Changes Take Effect: Immediately

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

```
interaction = stderr, network
```

trace

Default Value: No default value

Valid Values (log output types):

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

Changes Take Effect: Immediately

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

```
trace = stderr, network
```

debug

Default Value: No default value

Valid Values (log output types):

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

Changes Take Effect: Immediately

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

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

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

Log File Extensions

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

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

  Note: Provide `*.snapshot.log` files to Genesys 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:

`0`	This method of detecting an unresponsive application is not used by this application.
`3-604800`	Length of timeout, in seconds; equivalent to 3 seconds–7 days.

Changes Take Effect: Immediately

Specifies the maximum amount of time, in seconds, in which heartbeat messages are expected from an application. If Local Control Agent (LCA) does not receive a heartbeat message from the application within this period, it assumes the application is not responding and carries out corrective action.

This option can also be used to specify the maximum heartbeat interval for threads registered with class zero (0). This thread class is reserved for use by the Management Layer only.

If this option is not configured or 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.

Chapter

12 T-Server Common Configuration Options

This chapter describes the configuration options that are generally common to all T-Server types, with some exceptions noted. It contains the following sections:

T-Server also supports common log options described in Chapter 11, "Common Configuration Options," on page 211.

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 258.
Changes Take Effect: Immediately

Specifies the time interval that T-Server waits before processing client requests in background mode. You must set the `background-processing` option to `true` in order for this option to take effect.

check-tenant-profile

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: For the next connected client

When set to `true`, T-Server only allows a client to register if the client provides the correct name and password of a T-Server Tenant. If the client provides the Tenant name concatenated with a slash (/) and the Tenant password for the Tenant to which T-Server belongs as the value of `AttributeApplicationPassword` in the `TRegisterClient` request, T-Server allows that client to register DNs that are included in the switch configuration in the Configuration Database, but it does not allow the client to register DNs that are *not* included in the switch configuration.

consult-user-data

Default Value: `separate`
Valid Values:

`separate`	Stores user data for original and consultation calls in separate structures. The data attached to the original call is available for review or changes only to the parties of that call. The data attached to the consultation call is available only to the parties of the consultation call.
`inherited`	Copies user data from an original call to a consultation call when the consultation call is created; thereafter, stores user data separately for the original and the consultation call. Changes to the original call's user data are not available to the parties of the consultation call, and vice versa.
`joint`	Stores user data for an original call and a consultation call in one structure. The user data structure is associated with the original call, but the parties of both the original and consultation calls can see and make changes to the common user data.

Changes Take Effect: For the next consultation call created

Specifies the method for handling user data in a consultation call.

Note: A T-Server client can also specify the `consult-user-data` mode in the `Extensions` attribute `ConsultUserData` key for a conference or transfer request. If it is specified, the method of handling user data is based on the value of the `ConsultUserData` key-value pair of the request and takes precedence over the T-Server `consult-user-data` option. If it is not specified in the client request, the value specified in the `consult-user-data` option applies.

customer-id

Default Value: No default value. (A value must be specified for a multi-tenant environment.)
Valid Values: Any character string
Changes Take Effect: Immediately

Identifies the T-Server customer. You must set this option to the name of the tenant that is using this T-Server. You must specify a value for this option if you are working in a multi-tenant environment.

Note: Do not configure the `customer-id` option for single-tenant environments.

dn-scope

Default Value: `undefined`
Valid Values: `undefined, switch, office, tenant`
Changes Take Effect: Immediately
Related Feature: “Switch Partitioning” on page 96

Specifies whether DNs associated with the `Switch, Switching Office,` or `Tenant` objects will be considered in the T-Server monitoring scope, enabling T-Server to report calls to or from those DNs as internal.

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

With a value of `undefined` (the default), pre-8.x T-Server behavior applies and the switch partitioning is not turned on.

Note: Setting the option to a value of `office` or `tenant,` which requires T-Server to monitor a large set of configuration data, may negatively affect T-Server performance.

log-trace-flags

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

Valid Values (in any combination):

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

Changes Take Effect: Immediately

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

management-port

Default Value: `0`

Valid Values: `0` or any valid TCP/IP port

Changes Take Effect: After T-Server is restarted

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

merged-user-data

Default Value: `main-only`

Valid Values:

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

Changes Take Effect: Immediately

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

Note: The option setting does not affect the resulting data for merging calls if the `consult-user-data` option is set to `joint`. (See "consult-user-data" on page 235.)

propagated-call-type

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Switch Partitioning" on page 96

Determines what T-Server reports as the value of the `CallType` attribute in events related to calls that have been synchronized with another site via ISCC, as follows:

- When set to `false`, T-Server reports in events related to calls that have been synchronized with another site via ISCC the same value for the `CallType` attribute as it did in pre-8.0 releases and adds the new `PropagatedCallType` attribute with the value of the `CallType` attribute at the origination site. This provides backward compatibility with existing T-Server clients.
- When set to `true`, T-Server reports in events related to calls that have been synchronized with another site via ISCC the same value for the `CallType` attribute as at the origination site, and adds the new `LocalCallType` attribute with the same value as `CallType` in pre-8.0 releases.

server-id

Default Value: An integer equal to the value `ApplicationDBID` as reported by Configuration Server
Valid Values: Any integer from `0–16383`
Changes Take Effect: Immediately

Specifies the `Server ID` that T-Server uses to generate `Connection IDs` and other unique identifiers. In a multi-site environment, you must assign each T-Server a unique `Server ID`, in order to avoid confusion in reporting applications and T-Server behavior.

Configuration of this option is necessary for Framework environments in which there are two or more instances of the Configuration Database.

Note: If you do not specify a value for this option, T-Server populates it with the `ApplicationDBID` as reported by Configuration Server. Each data object in the Configuration Database is assigned a separate DBID that maintains a unique Server ID for each T-Server configured in the database.

Warning! Genesys does not recommend using multiple instances of the Configuration Database.

user-data-limit

Default Value: `16000`
Valid Values: `0`–`65535`
Changes Take Effect: Immediately

Specifies the maximum size (in bytes) of user data in a packed format.

Note: When T-Server works in mixed 8.x/7.x/6.x environment, the value of this option must not exceed the default value of `16000` bytes; otherwise, 6.x T-Server clients might fail.

license Section

The License section contains the configuration options that are used to configure T-Server licenses. They set the upper limit of the seat-related DN licenses (`tserver_sdn`) that T-Server tries to check out from a license file. See "License Checkout" on page 240.

This section must be called `license`.

Notes:
- T-Server also supports the `license-file` option described in the *Genesys Licensing Guide*.
- The `license` section is not applicable to Network T-Server for DTAG.
- On selected platforms, the limitation of 9999 licenses may no longer apply. Use values greater than 9999 only when instructed by Genesys Technical Support.

If you use two or more T-Servers, and they share licenses, you must configure the following options in the `license` section of the T-Servers.

num-of-licenses

Default Value: `0` or `max` (all available licenses)
Valid Values: String `max` or any integer from `0`–`9999`
Changes Take Effect: Immediately

Specifies how many DN licenses T-Server checks out. T-Server treats a value of `0` (zero) the same as it treats `max`—that is, it checks out all available licenses.

The sum of all `num-of-licenses` values for all concurrently deployed T-Servers must not exceed the number of seat-related DN licenses (`tserver_sdn`) in the corresponding license file. The primary and backup

T-Servers share the same licenses, and therefore they need to be counted only once. T-Server checks out the number of licenses indicated by the value for this option, regardless of the number actually in use.

num-sdn-licenses

Default Value: `0` or `max` (all DN licenses are seat-related)
Valid Values: String `max` (equal to the value of `num-of-licenses`), or any integer from `0`–`9999`
Changes Take Effect: Immediately

Specifies how many seat-related licenses T-Server checks out. A value of `0` (zero) means that T-Server does not grant control of seat-related DNs to any client, and it does not look for seat-related DN licenses at all.

The sum of all `num-sdn-licenses` values for all concurrently deployed T-Servers must not exceed the number of seat-related DN licenses (`tserver_sdn`) in the corresponding license file. The primary and backup T-Servers share the same licenses, and therefore they need to be counted only once. T-Server checks out the number of licenses indicated by the value for this option, regardless of the number actually in use.

Notes:
- For Network T-Servers, Genesys recommends setting this option to `0`.
- Be sure to configure in the Configuration Database all the DNs that agents use (Extensions and ACD Positions) and that T-Server should control. For further information, see Chapter 7, "DNs and Agent Logins," page 38.

License Checkout

Table 24 shows how to determine the number of seat-related DN licenses that T-Server attempts to check out. See the examples on page 241.

Table 24: 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 24: 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

a. In this table, the following conventions are used: `x` and `y` - are positive integers; `max` is the maximum number of licenses that T-Server can check out; `min (y, x)` is the lesser of the two values defined by `y` and `x`, respectively.

b. The License Checkout column shows the number of licenses that T-Server attempts to check out. The actual number of licenses will depend on the licenses' availability at the time of checkout, and it is limited to 9999.

Examples

This section presents examples of option settings in the `license` section.

Example 1

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

Example 2

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

Example 3

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

Example 4

If...		Then...
Options Settings	**License File Settings**	**License Checkout**
num-of-licenses = max	tserver_sdn = 5000	1000 seat-related DNs
num-sdn-licenses = 1000		

agent-reservation Section

The `agent-reservation` section contains the configuration options that are used to customize the T-Server Agent Reservation feature. See "Agent Reservation" on page 28 section for details on this feature.

This section must be called `agent-reservation`.

Note: The Agent Reservation functionality is currently a software-only feature that is used to coordinate multiple client applications. This feature does not apply to multiple direct or ACD-distributed calls.

collect-lower-priority-requests

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether an agent reservation request is collected, depending on its priority during the time interval specified by the `request-collection-time` configuration option. When set to `false`, during the `request-collection-time` interval T-Server collects reservation requests of the highest priority only, rejecting newly submitted requests that have a lower priority or rejecting all previously submitted requests if a request with a higher priority arrives. When set to `true` (the default), agent reservation requests are collected as they were in pre-8.x releases.

reject-subsequent-request

Default Value: `true`
Valid Values:

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

Changes Take Effect: Immediately

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

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

request-collection-time

Default Value: `100 msec`
Valid Values: See "Timeout Value Format" on page 258.
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 258.
Changes Take Effect: Immediately

Specifies the default interval for which a an Agent DN is reserved. During this interval, the agent cannot be reserved again.

extrouter Section

The `extrouter` section contains the configuration options that are used to support multi-site environments with the Inter Server Call Control (ISCC) feature. The configuration options in this section of the document are grouped with related options that support the same functionality, as follows:

- ISCC Transaction Options, page 245
- Transfer Connect Service Options, page 249
- ISCC/COF Options, page 250
- Event Propagation Options, page 252
- Number Translation Option, page 253
- GVP Integration Option, page 254

This configuration section must be called `extrouter`.

For a description of the ways in which T-Server supports multi-site configurations and for an explanation of the configuration possibilities for a multi-site operation, see the "Multi-Site Support" chapter.

Note: In a multi-site environment, you must configure the `timeout`, `cast-type`, and `default-dn` options with the same value for both the primary and backup T-Servers. If you do not do this, the value specified for the backup T-Server overrides the value specified for the primary T-Server.

match-call-once

Default Value: `true`

Valid Values:

`true`	ISCC does not process (match) an inbound call that has already been processed (matched).
`false`	ISCC processes (attempts to match) a call as many times as it arrives at an ISCC resource or multi-site-transfer target.

Changes Take Effect: Immediately

Specifies how many times ISCC processes an inbound call when it arrives at an ISCC resource. When set to `false`, ISCC processes (attempts to match) the call even if it has already been processed.

Note: Genesys does not recommend changing the default value of the `match-call-once` option to `false` unless you have specific reasons. Setting this option to `false` may lead to excessive or inconsistent call data updates.

reconnect-tout

Default Value: `5 sec`

Valid Values: See "Timeout Value Format" on page 258.

Changes Take Effect: At the next reconnection attempt

Specifies the time interval after which a remote T-Server attempts to connect to this T-Server after an unsuccessful attempt or a lost connection. The number of attempts is unlimited. At startup, T-Server immediately attempts the first connection, without this timeout.

report-connid-changes

Default Value: `false`

Valid Values:

`true`	`EventPartyChanged` is generated.
`false`	`EventPartyChanged` is not generated.

Changes Take Effect: Immediately

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

use-data-from

Default Value: `current`

Valid Values:

`active`	The values of `UserData` and `ConnID` attributes are taken from the consultation call.
`original`	The values of `UserData` and `ConnID` attributes are taken from the original call.
`active-data-original-call`	The value of the `UserData` attribute is taken from the consultation call and the value of `ConnID` attribute is taken from the original call.
`current`	If the value of `current` is specified, the following occurs: • Before the transfer or conference is completed, the `UserData` and `ConnID` attributes are taken from the consultation call. • After the transfer or conference is completed, `EventPartyChanged` is generated, and the `UserData` and `ConnID` are taken from the original call.

Changes Take Effect: Immediately

Specifies the call from which the values for the `UserData` and `ConnID` attributes are taken for a consultation call that is routed or transferred to a remote location.

Note: For compatibility with the previous T-Server releases, you can use the values `consult`, `main`, and `consult-user-data` for this option. These are aliases for `active`, `original`, and `current`, respectively.

ISCC Transaction Options

cast-type

Default Value: `route, route-uui, reroute, direct-callid, direct-uui, direct-network-callid, direct-notoken, direct-digits, direct-ani, dnis-pool, pullback`

Valid Values: `route, route-uui, reroute, direct-callid, direct-uui, direct-network-callid, direct-notoken, direct-digits, direct-ani, dnis-pool, pullback`

Changes Take Effect: For the next request for the remote service

Specifies—using a space-, comma- or semicolon-separated list—the routing types that can be performed for this T-Server.

The valid values provide for a range of mechanisms that the ISCC feature can support with various T-Servers, in order to pass call data along with calls between locations.

Because switches of different types provide calls with different sets of information parameters, some values might not work with your T-Server. See Table 3 on page 75 for information about supported transaction types by a specific T-Server. The "Multi-Site Support" chapter also provides detailed descriptions of all transaction types.

Notes: For compatibility with the previous T-Server releases, you can use the `direct` value for this option. This is an alias for `direct-callid`.

An alias, `route-notoken`, has been added to the `route` value.

default-dn

Default Value: No default value
Valid Values: Any DN
Changes Take Effect: For the next request for the remote service

Specifies the DN to which a call is routed when a Destination DN (`AttributeOtherDN`) is not specified in the client's request for routing. If neither this option nor the client's request contains the destination DN, the client receives `EventError`.

Note: This option is used only for requests with route types `route`, `route-uui`, `direct-callid`, `direct-network-callid`, `direct-uui`, `direct-notoken`, `direct-digits`, and `direct-ani`.

direct-digits-key

Default Value: `CDT_Track_Num`
Valid Values: Any valid key name of a key-value pair from the `UserData` attribute
Changes Take Effect: For the next request for the remote service

Specifies the name of a key from the `UserData` attribute that contains a string of digits that are used as matching criteria for remote service requests with the `direct-digits` routing type.

Note: For compatibility with the previous T-Server releases, this configuration option has an alias value of `cdt-udata-key`.

dn-for-unexpected-calls

Default Value: No default value
Valid Values: Any DN
Changes Take Effect: Immediately

Specifies a default DN for unexpected calls arriving on an External Routing Point.

network-request-timeout

Default Value: `20 sec`
Valid Values: See "Timeout Value Format" on page 258.
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 258.
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 258.
Changes Take Effect: For the next request for remote service

Specifies the time interval that a T-Server at the origination location waits for a notification of routing service availability from the destination location. Counting starts when the T-Server sends a request for remote service to the destination site.

resource-allocation-mode

Default Value: `circular`
Valid Values:

`home` T-Server takes an alphabetized (or numerically sequential) list of configured DNs and reserves the first available DN from the top of the list for each new request. For example, if the first DN is not available, the second DN is allocated for a new request. If the first DN is freed by the time the next request comes, the first DN is allocated for this next request.

`circular` T-Server takes the same list of configured DNs, but reserves a subsequent DN for each subsequent request. For example, when the first request comes, T-Server allocates the first DN; when the second request comes, T-Server allocates the second DN; and so on. T-Server does not reuse the first DN until reaching the end of the DN list.

Changes Take Effect: Immediately

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

resource-load-maximum

Default Value: `0`
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 258.
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 258.
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 258.
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 258.
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 258.
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 258.
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 258.
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 258.
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 258.
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 258.
Changes Take Effect: Immediately

Specifies the time interval for periodic checks for stuck calls. These checks affect both notification and cleanup functionality, and are made by checking the T-Server's own call information with call information available in the switch. For performance reasons, T-Server does not verify whether the

notify-idle-tout or cleanup-idle-tout option has expired before performing this check.

Note: Setting this option to a value of less than a few seconds can affect T-Server performance.

Examples

This section presents examples of option settings in the call-cleanup section.

Example 1

```
cleanup-idle-tout = 0
notify-idle-tout = 0
periodic-check-tout = 10
```

With these settings, T-Server will not perform any checks for stuck calls.

Example 2

```
cleanup-idle-tout = 0
notify-idle-tout = 5 min
periodic-check-tout = 10 min
```

With these settings, T-Server performs checks every 10 minutes and sends notifications about all calls that have been idle for at least 5 minutes.

Example 3

```
cleanup-idle-tout = 20 min
notify-idle-tout = 5 min
periodic-check-tout = 10 min
```

With these settings, T-Server performs checks every 10 minutes, sends notifications about all calls that have been idle for at least 5 minutes, and attempts to clean up all calls that have been idle for more than 20 minutes.

Translation Rules Section

The section name is specified by the inbound-translator-<*n*> option. It contains options that define translation rules for inbound numbers.

You can choose any name for this section, provided that it matches the value of the section. Every option in this section corresponds to a rule and must conform to the format described below. You can configure as many rules as necessary to accommodate your business needs.

rule-<*n*>

Default Value: No default value
Valid Value: Any valid string in the following format:
in-pattern=<input pattern value>;out-pattern=<output pattern value>
Changes Take Effect: Immediately

Defines a rule to be applied to an inbound number. The two parts of the option value describe the input and output patterns in the rule. When configuring the

pattern values, follow the syntax defined in "Using ABNF for Rules" on page 84. See "Configuring Number Translation" on page 91 for examples of these rules as well as detailed instructions for creating rules for your installation. For example, a value for this configuration option might look like this:

```
rule-01 = in-pattern=0111#CABBB*ccD;out-pattern=ABD
```

security Section

The `security` section contains the configuration options that are used to configure secure data exchange between T-Servers and other Genesys components. Refer to the *Genesys 8.x Security Deployment Guide* for complete information on the security configuration.

Timeout Value Format

This section of the document describes the values to use for those T-Server common options that set various timeouts. The current format allows you to use fractional values and various time units for timeout settings.

For timeout-related options, you can specify any value that represents a time interval, provided that it is specified in one of the following formats:

`[[[`*hours*`:]`*minutes*`:]`*seconds*`][`*milliseconds*`]`

or

`[`*hours* `hr][`*minutes* `min][`*seconds* `sec][`*milliseconds* `msec]`

Where a time unit name in italic (such as *hours*) is to be replaced by an integer value for this time unit.

Integer values with no measuring units are still supported, for compatibility with previous releases of T-Server. When you do not specify any measuring units, the units of the default value apply. For example, if the default value equals `60 sec`, specifying the value of `30` sets the option to 30 seconds.

Example 1

The following settings result in a value of 1 second, 250 milliseconds:

```
sync-reconnect-tout = 1.25
sync-reconnect-tout = 1 sec 250 msec
```

Example 2

The following settings result in a value of 1 minute, 30 seconds:

```
timeout = 1:30
timeout = 1 min 30 sec
```

Changes from Release 8.0 to 8.1

Table 25 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 25: 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 234.

Chapter

10 T-Server-Specific and DN Configuration Options

This chapter describes the configuration options that are unique to the T-Server for Nortel Communication Server 1000 with SCCS/MLS. It includes the following sections:

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

The options common to all T-Servers are described in Chapter 11, "Common Configuration Options," on page 211 and Chapter 10, "T-Server-Specific and DN Configuration Options," on page 233.

Application-Level Options

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

Table 37 on page 234 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 that are applicable to the connection protocol that is used in your environment.

Table 37: Mandatory Options

Option Name	Default Value	Details
TServer Section		
link-*n*-name	Mandatory field. No default value.	Specifies the section name containing the configuration options assigned to that link, where *n* is a consecutive number for a CTI link. See description on page 238.
CTI-Link Section		
protocol	Mandatory field. No default value.	Specifies the connection protocol T-Server uses in communicating with the switch. See description on page 247.
hostname	Mandatory field. No default value.	Specifies the host of the link according to the switch configuration. See description on page 246.
port	Mandatory field. No default value.	Specifies the TCP/IP port of the link according to the switch configuration. See description on page 247.

TServer Section

The section must be called `TServer`.

acw-by-request-only

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

Enables agents to enter the AfterCallWork state only if the workmode in the TAgentNotReady is set to AfterCallWork. If the value of this configuration option is set to true, agents enter the AfterCallWork state only if the workmode in the TAgentNotReady is set to AfterCallWork.

If the value of this configuration option is set to false, agents will also enter AfterCallWork state if the workmode in TAgentNotReady is NoCallDisconnect, or if the option no-call-disconnect is set to a non-zero value.

callpilot-dn-range

Default Value: 0 (zero)
Valid Values: A sequence of numerical value ranges and/or single numerical values, separated by commas—for example: 6100-6200,7100-7200,7323
Changes Take Effect: Immediately

Specifies the CallPilot DNs (DNs of type Extension and ACD Position) involved in transfers; that is, transfers either from, or to, DNs controlled by T-Server.

Note:
- This option must be properly configured to support calls involving CallPilot DNs in order to prevent possible stuck calls.
- The CallPilot DNs must not be configured as DN configuration objects in Configuration Manager.

cdn-cabq-timeout

Default Value: 5000
Valid Values: Any positive integer
Changes Take Effect: Immediately

Specifies the amount of time (in milliseconds) that T-Server delays processing a CallAbandonedQueue message from the switch in scenarios involving a call transferred or conferenced to a Controlled DN (CDN). If such a transfer is completed while the call is still on the CDN (that is, prior to routing), the switch sends a CallAbandonedQueue message and a new RouteRequest (for the main call) to indicate that the consult leg of the call is gone and that the transfer is complete. Use this configuration option to designate the length of time required to determine if a transfer has been completed or if the call has been abandoned.

Note: The 5-second default value should be sufficient for most deployments. However, for installations where abandons occur, setting this option results in a delay in sending EventAbandoned, which may affect statistics. Administrators should be aware of this delay and be prepared to accommodate it in these cases.

consult-call-unverified-timeout

Default Value: `900,000` (15 minutes)
Valid Values: Any positive integer
Changes Take Effect: At the beginning of the next time interval

Specifies the maximum interval (in milliseconds) that T-Server will keep a consultation call active after the corresponding original call is released or retrieved.

Note: In many scenarios, T-Server can determine for certain that the consultation call no longer exists, and if so, T-Server will release the consultation call sooner.

create-addr-on-register

Default Value: `true`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

Registers and sends requests for DNs that do not have an entry in Configuration Manager. If the value of this configuration option is set to `true`, clients can register and send requests for DNs that do not have an entry in Configuration Manager. If the value of this configuration option is set to `false`, clients registering for DNs not found in Configuration Manager will see the following error message `DN is not configured in CME`.

default-agent-id-is-position

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: At the next applicable agent state processing
Related Feature: “Support for Agent States and Workmodes” on page 148

Specifies whether T-Server will report `AgentIDs` for agents of the switch after the self-correcting agent state logic was applied to them. When the value of this configuration option is set to `false`, T-Server will not report the `AgentIDs` in the login events unless the `default-agent-id` configuration option is set for the corresponding position DNs.

If the value of this configuration option is set to `true`, T-Server will report the `AgentIDs` assuming them to be the same as the position IDs of the corresponding DNs.

See “Self-Correcting Agent States” on page 149 for more information.

Note: Setting this option does not affect agents who log in while T-Server is running; in this case T-Server will use `AgentID` provided by the switch.

delete-external-call-timeout

Default Value: `30000` (30 seconds)
Valid Values: Any positive integer
Changes Take Effect: At the beginning of the next time interval

Specifies the interval (in milliseconds) that T-Server waits before deleting a call only with external parties or non-AST parties from the call table.

dest-busy-codes

Default Value: `0` (null)
Valid Values: Any space-delimited set of hexadecimal progress codes
Changes Take Effect: Immediately

Specifies the list of additional space-delimited progress codes (in hexadecimal form) that are translated into an `EventDestinationBusy` event. These progress codes are in addition to the five hexadecimal codes that T-Server always translates into an `EventDestinationBusy` event. These hexadecimal codes are as follows:

`0C00`	Terminating party is busy.
`0C01`	Destination resource blocking
`0C19`	Terminating party is busy.
`0C08`	Unassigned number
`0C0D`	Invalid number format

dest-busy-invalid-num-codes

Default Value: `0C08`

Valid Value: Any space-delimited set of hexadecimal progress codes

Changes Take Effect: Immediately

Specifies the list of space-delimited progress codes (in hexadecimal form) that are translated into an `EventDestinationBusy` event with an `AttributeCallState 11 (CallStateSitInvalidnum)` attribute.

enable-consult-swap

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: Immediately
Related Feature: “Support for the TAlternateCall Function” on page 152

If the value of this configuration option is set to `true`, T-Server uses the MLS enhancements for “Swap/Disconnect during Transfer or Conference call.” This switch feature allows T-Server to implement the `TAlternateCall` request that places an active call on hold and connects the held (inactive) call. When the value of this configuration option is set to `true`, the value of the `rls-consult-rtv` configuration option is ignored.

If the value of this configuration option is set to `false`, `TAlternateCall` is disabled and the option `rls-consult-rtv` is used to determine whether the active call is released after the held call is retrieved.

Note: The `enable-consult-swap` configuration option should be only set to `true` when the Nortel Swap feature is active on the switch.

link-*n*-name

Default Value: Mandatory field. No default value.
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 and `n` cannot have a value of `0` (zero).

Note: `link-`*`n`*`-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 before the connection is reestablished.

link-timeout

Default Value: `2000`
Valid Values: `0`, or any positive integer
Changes Take Effect: At the beginning of the next time interval

Specifies the interval (in milliseconds) that T-Server waits before making reconnection attempts to the switch. If the value of this configuration option is set to `0` (zero), no reconnection attempts will be made.

link-type

Default Value: `symposium`
Valid Values: `meridian`, `symposium`
Changes Take Effect: After T-Server restart

Indicates which link—`Meridian` (MLS) or `Symposium` (SCCS)—is in use.

make-call-manner

Default Value: `semi-polite`
Valid Values:

`belligerent`	A call is released.
`semi-polite`	The request is rejected if any active call is in progress.
`polite`	The request is rejected if any call is in progress, even if it is not active.

Changes Take Effect: At next call made

Determines how a `TMakeCall` request affects any existing call on the extension.

max-attempts-to-register

Default Value: `10`
Valid Values: `0`, or any positive integer
Changes Take Effect: Immediately

Sets the number of times T-Server attempts to register a DN or acquire a CDN from the switch. On any failed attempt to register or acquire, T-Server automatically tries again after a wait of 1–10 seconds. T-Server repeats this process as many times as the value set for this option. If T-Server is attempting the registration is being attempted as a result of a client request, T-Server does not return an `EventError` event until the final try has failed. When this configuration option is set to a value of `0` (zero), the registration request is sent only once.

mute-xfer-method

Default Value: `2step`
Valid Values:

`2step`	The `TInitiateTransfer` and `TCompleteTransfer` functions are used.
`fast`	The `TFastTransfer` function is used.

Changes Take Effect: Immediately

Sets the method for using the `TMuteTransfer` function.

no-call-disconnect

Default Value: `0` (All current calls are disconnected)
Valid Values:

`0`	All current calls are disconnected.
`1`	Current ACD calls are not disconnected.
`2`	Current DN calls are not disconnected.
`3`	Neither ACD nor DN calls are disconnected.

Changes Take Effect: Immediately

Specifies the way the switch handles calls, if the `workmode` parameter is set to `AgentNoCallDisconnect` in the `TAgentSetNotReady` function call. If 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, it will set the behavior for AfterCallWork.

Note: If agents use no-call-disconnect, Stat Server 7.0 or higher is required.

nrdy-after-login

Default Value: true
Valid Values: true, false
Changes Take Effect: Immediately

Distributes EventAgentNotReady after a successful login procedure. If the value of this configuration option is set to true, T-Server distributes EventAgentNotReady after a successful login procedure. If the value of this configuration option is set to false, T-Server distributes EventAgentReady.

out-of-service-retry-interval

Default Value: 900,000 (15 minutes)
Valid Values: Any positive integer
Changes Take Effect: At the beginning of the next time interval

Specifies the interval (in milliseconds) during which T-Server attempts to register out-of-service DNs. If the value of this configuration option is set to 0 (zero), T-Server does not attempt to register out-of-service DNs. If an out-of-service DN is successfully registered after this interval, an EventBackInService message is sent for the DN and it enters back-in-service (idle) state.

response-timeout

Default Value: 15000 (15 seconds)
Valid Values: Any positive integer
Changes Take Effect: At beginning of next time interval

Specifies the interval (in milliseconds) that T-Server, after sending a request, will wait for a response from the switch. For registration requests, T-Server will retry the request after this time interval. For client requests, T-Server will return EventError(TERR_TIMEOUT) after this time interval.

rls-consult-rtv

Default Value: immediately
Valid Values: immediately, upon-release-notification
Changes Take Effect: Immediately

If the value of this configuration option is set to immediately, T-Server reports a consultation call as released when the TReconnectCall function call has been successful. If the value of this configuration option is set to upon-release-notification, T-Server does not report a release of the

consultation call at retrieval of the original call, but waits for the link notification that the consultation call no longer exists.

Note: The value of the `rls-consult-rtv` configuration option is ignored when the `enable-consult-swap` configuration option is set to `true`.

routing-state-timeout

Default Value: `600000` (10 minutes)
Valid Values: Any positive integer from `0` to `14,400,000` (4 hours)
Changes Take Effect: Immediately

Specifies for how long (in milliseconds) a Routing Point can be in a routing state, where the `EventRouteRequest` event has been sent and no response has been received, before the routed call party is deleted by T-Server. This configuration option prevents a call from being stuck at a Routing Point when the switch has routed the call without notifying T-Server.

Note: Routing Points controlled by IVRs may legitimately be in routing state for an extended period until an agent is found. If this is the case, the configuration option value should be increased to prevent the premature deletion of the call.

rtp-info-password

Default Value: none
Valid Values: A password string or empty.
Changes Take Effect: Immediately

Specifies the password that must be supplied by a voice-monitoring application as the value for the `RTP-PASSWORD` key in the `Extensions` attribute in either `TRegisterAddress` or `TPrivateService`. If this configuration option is not provided, no voice-monitoring application is allowed to receive RTP stream data.

scu-emerg-type

Default Value: `0xc`
Valid Values: `0xc`, `0xd`
Changes Take Effect: Immediately

Specifies the value of the `Call Type Informational Element` that is expected in the `Status Change/Unringing` message when an agent establishes a `No-Hold` conference (using an `Emergency` key) with the supervisor.

Note: Different releases of the Meridian 1 switch software use different values to identify this message. You can identify which value your switch uses for `StatusChange/Unringing` by consulting the T-Server logs.

set-discovery

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After T-Server restart
Related Feature: “Support for Timed After Call Work (TACW)” on page 152

Enables Set Information Discovery on DN registration. If this configuration option is set to `true`, this option enables Set Information Discovery on DN registration. If this configuration option is set to `false`, T-Server is not able to gain terminal number (TN) information at registration time. TN information is required for MLS IP Call Recording and Timed After Call Work.

Note: This option requires that Set Information Discovery is supported on the switch.

set-dnis-from-dest

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Populates the DNIS with a destination number in call-related events. If the value of this configuration option is set to `true`, T-Server populates the DNIS with a destination number in call-related events when a real DNIS does not come from the switch.

soft-login-support

Default Values: `false`
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted
Related Feature: “Support for Emulated Agent States” on page 159

Turns on or off the Emulated Agent States (also known as the Soft Agent) feature. If the value of this configuration option is set to `true`, T-Server processes all agent-related feature requests (`TAgentLogin`, `TAgentLogout`, `TAgentSetReady`, `TAgentNotReady`) internally, without interacting with the CTI link. T-Server accepts all client requests, provided that they do not contradict the Agent-State diagram. See the *Genesys Events and Models Reference Manual* and *Voice Platform SDK 8.x .NET (or Java) API Reference* for more information.

The following conditions must be met:

- `AgentLogin` must be configured in the Configuration Layer.
- Only one login with any given `AgentLogin` is allowed at any time.
- Only one agent login is allowed on the DN.
- If the password is configured in the Configuration Layer, it must be supplied in `TAgentLogin` (otherwise the request fails).

- The workmode is not used in `TAgentSetReady`, but T-Server supports all `NotReady` substates that are shown on the Agent-State diagram.

Note: When the `soft-login-support` configuration option is enabled, T-Server processes agent-related CTI messages, but it does not distribute corresponding agent-related events when an agent logs in or out manually using a phone set. To avoid desynchronization between the switch and the reporting application, agents should not log in or out manually (a so-called "hard" login) when the `soft-login-support` configuration option is enabled.

soft-tacw-support

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Support for Timed After Call Work (TACW)" on page 152

Enables the Timed After Call Work (TACW) feature. If the value of this configuration option is set to `true`, this option enables the Timed After Call Work (TACW) feature and the feature-related configuration options, such as `soft-wrap-up-time` and `terminal-id`.

soft-wrap-up-time

Default Value: `0` (zero)
Valid Values: See "Timeout Value Format" on page 258.
Changes Take Effect: Immediately
Related Feature: "Support for Timed After Call Work (TACW)" on page 152

When the Timed ACW is enabled with the `soft-tacw-support` configuration option, the `soft-wrap-up-time` configuration option sets the wrap-up time globally for all agents on the corresponding switch that do not have it configured individually using the Wrap-up Time property of the `Person` (`Agent`) object in Configuration Manager. The individual wrap-up time settings take priority over the global wrap-up time set with this configuration option.

terminal-id

Default Value: empty
Valid Values: Any positive integer
Changes Take Effect: Immediately
Related Feature: "Support for Timed After Call Work (TACW)" on page 152

Sets the TN (terminal number) for the DN. T-Server uses this value to associate multiple DNs to the same terminal for use with the Timed ACW feature. If this configuration option is not set, the T-Server is still able to identify the TN at the time when a call is made to the DN. However, use of this option allows T-Server to have this information earlier. Set this option in the

TServer section of the Annex tab of a DN configuration object in Configuration Manager.

Note:
- This is a DN-specific option.
- If the value of the set-discovery configuration option is set to true, and Set Information Discovery is enabled on the switch, this option setting is not required.

update-login-on-err

Default Value: true
Valid Values: true, false
Changes Take Effect: Immediately
Related Feature: "Support for Agent States and Workmodes" on page 148

Determines the behavior of T-Server when T-Server receives the ErrorCode 0x200E (decimal 8206) - Set is in target state message for login and logout requests from the switch. If the value of this configuration option is set to true (default), T-Server updates the agent's state and distributes events that indicate the change. If the value of this configuration option is set to false, T-Server sends an EventError event to the client with a ErrorCode 186 - Set is in target state message.

uudata-attach-type

Default Value: none
Valid Values: none, binary, parsed
Changes Take Effect: Immediately
Related Feature: "Support for Incoming UUI Data" on page 152

Specifies how T-Server stores the UUI data. If the value of this configuration option is set to none, T-Server does not store the UUI data. If the value of this configuration option is set to binary, T-Server stores the UUI data in the binary format as shown in Table 38.

Table 38: UUI Data in the Binary Format

Key	Value: TKVList structure	
UU_DATA	**Key**	**Value**
	CS0_BIN	Binary data contained in codeset 0 of UUI
	CS6_BIN	Binary data contained in codeset 6 of UUI
	CS7_BIN	Binary data contained in codeset 7 of UUI

If the value of this configuration option is set to parsed, T-Server will parse the data according to the format described in ATT Toll Free Transfer Connect

Service (TR 50075), and store the results of parsing in the ASCII format shown in Table 39.

Table 39: UUI Data in the ASCII Format

Key	Value (TKVList structure)	
UU_DATA	**Key**	**Value**
	CS0_TAG_XX	String (ASCII) data contained in codeset 0 tag XX
	CS6_TAG_XX	String (ASCII) data contained in codeset 6 tag XX
	CS7_TAG_XX	String (ASCII) data contained in codeset 7 tag XX

Note:
- If no data is available for a particular codeset, the corresponding codeset key is missing from the `UU_DATA` structure.
- `XX` can be any hex tag value from `00` to `FF`.
- If the UUI is not in the format described in ATT TR50075, the parsed tag values may be missing or undefined (in this case the binary format is recommended).

CTI-Link Section

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

A link-connection protocol can only be TCP/IP. Protocol-based options are described in "hostname" on page 246.

application-id

Default Value: `TServer`
Valid Values: Any valid application ID
Changes Take Effect: Immediately (an automatic reconnection occurs)

Specifies the Meridian Link application ID.

Note: In release 5.1, this configuration option was named `mlink-application-id` in the `TServer` configuration section.

Warning! The length of the Application ID must not exceed the limit introduced by Nortel. Otherwise, an attempt to register an application may result in link failure.

customer-number

Default Value: `0` (zero)
Valid Values: Any customer group that exists on the switch
Changes Take Effect: Immediately (an automatic reconnection occurs)

Specifies the number of the Meridian customer group that functions with T-Server. This setting must match the customer number in the Meridian Link module and the switch.

Note: In release 5.1, this configuration option was named `mlink-customer-number` in the `TServer` configuration section.

host-id

Default Value: `Lanlink`
Valid Values: Any valid host ID
Changes Take Effect: Immediately (an automatic reconnection occurs)

Specifies the Meridian Link host ID. The value of this configuration option must match the value specified in the switch configuration.

Note: In release 5.1, this configuration option was named `mlink-host-id` in the `TServer` configuration section.

hostname

Default Value: Mandatory field. No default value.
Valid Values: Any valid host name
Changes Take Effect: Immediately

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

mail-name

Default Value: No default value.
Valid Values: Any valid Meridian Mail device name
Changes Take Effect: Immediately (an automatic reconnection occurs)

Specifies the name of the Meridian Mail device. The value of this configuration option must match the value specified in the switch configuration. If you do not specify a value for this configuration option, Meridian Mail is not used.

Note: In release 5.1 and earlier, this configuration option was named `mail-name` in the `TServer` configuration section.

poll-interval

Default Value: `0`
Valid Values: `0`, or any positive integer from `1-60`
Changes Take Effect: Immediately (an automatic reconnection occurs)

Specifies the interval (by groupings of tens of seconds) that T-Server waits to initiate polling from the Meridian Link. Setting this configuration option to a value of `0` turns off polling.

Note: In release 5.1, this configuration option was named `mlink-poll-interval` in the `TServer` configuration section. When T-Server registers CDNs, polling is automatically turned on.

port

Default Value: Mandatory field. No default value.
Valid Values: Any valid port address
Changes Take Effect: Immediately

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

DN-Level Options

These options are set in the `TServer` section of the `Annex` tab of a `DN` configuration object in Configuration Manager. See "Self-Correcting Agent States" on page 149 for more information.

default-agent-id

Default Value: `0` (zero)
Valid Values: Any string
Changes Take Effect: At the next applicable agent state processing
Related Feature: "Support for Agent States and Workmodes" on page 148

Specifies whether T-Server will report `AgentID` for the agent at the corresponding DN after the self-correcting agent state logic was applied to this agent. If the value of this configuration option is set to `NULL`, T-Server will not report `AgentID` unless the value of the `default-agent-id-is-position` configuration option is set to `true`. If this configuration option is specified,

T-Server will report `AgentID` as the value of this configuration option. The `default-agent-id` configuration configuration option takes precedence over the `default-agent-id-is-position` configuration option.

Note: Setting of this configuration option does not affect agents who log in while T-Server is running; in this case T-Server will use `AgentID` provided by the switch.

vtport-generate-hook-events

Default Value: `true`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

If the value of this configuration option is set to `false`, DNs configured in Configuration Manager as Voice Treatment Ports will send `EventOffHook` and `EventOnHook` events only when the switch sends these events. If the value of this configuration option is set to `true`, these DNs will behave like regular DNs and T-Server will generate an `EventOnHook` event when all parties for the DN have released, and an `EventOffHook` event when a party on the DN changes to an active state.

Multi-Site Support Section

The Multi-Site Support section contains the configuration options that are used to support multi-site environments with the Inter Server Call Control (ISCC) feature.

This section must be called `extrouter`.

For a description of the ways in which T-Server supports multi-site configurations and for an explanation of the configuration possibilities for a multi-site operation, see the "Multi-Site Support" chapter.

default-network-call-id-matching

Default Value: No default value
Valid Values: `MLS`
Changes Take Effect: Immediately

When a value for this configuration option is specified, T-Server uses the `NetworkCallID` attribute for the ISCC/COF call matching.

To activate this feature, the `cof-feature` configuration option must be set to `true`.

Changes from Release 8.0 to 8.1

Table 40 lists configuration options that changed between the 8.0 and 8.1 releases of T-Server. If a configuration option has been replaced with another that enables the same functionality, the new option name and location in this chapter are noted.

Table 40: Configuration Option Changes from 8.0 to 8.1

Option Name	Type of Change	Details
TServer Section		
soft-login-support	New	See page 242 for details.

Genesys

Supplements

Related Documentation Resources

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

T-Server for Nortel Communication Server 1000 with SCCS/MLS

- 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 will help you use Configuration Manager.
- 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

Consult these additional resources as necessary:

- The *Genesys Migration Guide,* also on the Genesys Documentation Library DVD, which contains a documented migration strategy from Genesys product releases 5.*x* and later to all Genesys 8.*x* releases. Contact Genesys Technical Support for additional information.
- The *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and CTI terminology and acronyms used in this document.
- The Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at `http://genesyslab.com/support`.

Information 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*

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:

810fr_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 41 describes and illustrates the type conventions that are used in this document.

Table 41: Type Styles

Type Style	Used For	Examples
Italic	• Document titles • Emphasis • Definitions of (or first references to) unfamiliar terms • Mathematical variables Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on page 254).	Please consult the *Genesys Migration Guide* for more information. Do *not* use this value for this option. A *customary and usual* practice is one that is widely accepted and used within a particular industry or profession. The formula, $x + 1 = 7$ where x stands for . . .
Monospace font (Looks like `teletype` or `typewriter text`)	All programming identifiers and GUI elements. This convention includes: • The *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages. • The values of options. • Logical arguments and command syntax. • Code samples. Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.	Select the `Show variables on screen` check box. In the `Operand` text box, enter your formula. Click `OK` to exit the `Properties` dialog box. T-Server distributes the error messages in `EventError` events. If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls. Enter `exit` on the command line.
Square brackets ([ ])	A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.	`smcp_server -host [/flags]`
Angle brackets (< >)	A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise. **Note:** In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.	`smcp_server -host <confighost>`

Index

Symbols

A

B

C

D

E

F

H

I

K

L

M

N

O

P

R

S

T

U

V

W

X