Framework 8.1

T-Server and HA Proxy for Nortel Communication Server 2000/2100

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

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_ncs2000_03-2012_v8.1.001.02

Table of Contents

Chapter 10

Chapter 11

Appendix

Supplements

Index

List of Procedures

Preface

Welcome to the *Framework 8.1 T-Server and HA Proxy for Nortel Communication Server 2000/2100 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 and HA Proxy for Nortel Communication Server 2000/2100. 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 269.

About T-Server and HA Proxy for Nortel Communication Server 2000/2100

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 DMS-100.
- HA Proxy for Nortel DMS-100.

The current name is T-Server and HA Proxy for Nortel Communication Server 2000/2100.

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.

Reading Prerequisites

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

Making Comments on This Document

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

Genesys Technical Support Contact Information (Continued)

Region	Telephone	E-Mail
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 and HA Proxy for Nortel Communication Server 2000/2100 Deployment Guide* has been updated with the following:

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

Part

1 T-Server Deployment

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

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

- Chapter 1, "T-Server Fundamentals," on page 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 61, details the variations available for T-Server implementations across geographical locations.
- Chapter 5, "Starting and Stopping T-Server Components," on page 119, describes how, and in what order, to start up T-Server among other Framework components. It also provides possible stopping commands.

New for All T-Servers in 8.1

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

- T-Server no longer connects to applications that have disabled status in the configuration environment.
- The default value of the `background-processing` configuration option has been changed to `true`. See "background-processing" on page 200 for details.

- T-Server now supports the Unresponsive Process Detection feature. The following configuration options enable this feature:
  - "heartbeat-period" on page 195
  - "hangup-restart" on page 196

  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 225.
- For information about the new features that are available in your T-Server in the initial 8.1 release, see Part Two of this document.

Chapter

1

T-Server Fundamentals

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

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

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

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

Learning About T-Server

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.

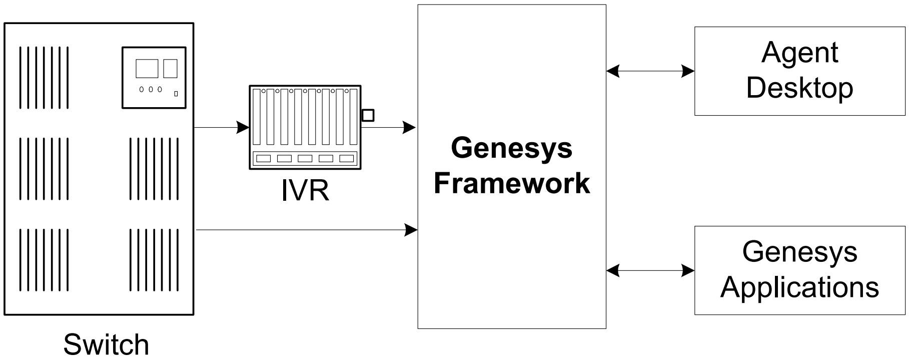

Figure 1: Framework in a Genesys Solution

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

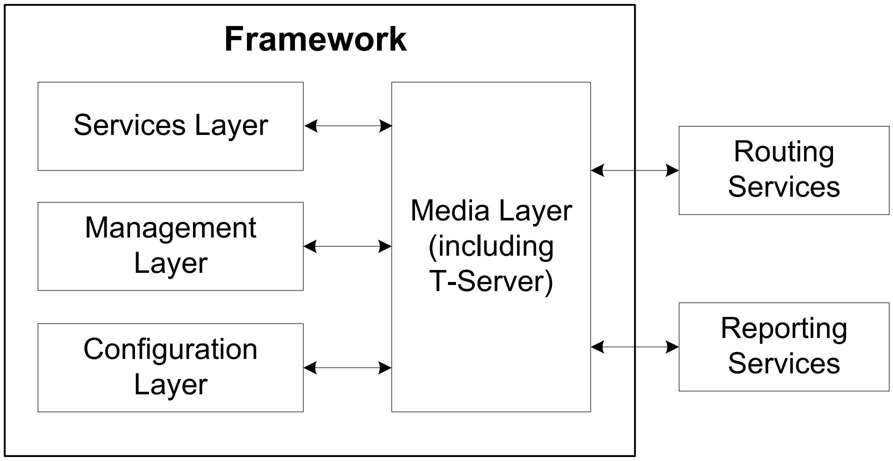

Figure 2: The Media Layer in the Framework Architecture

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.

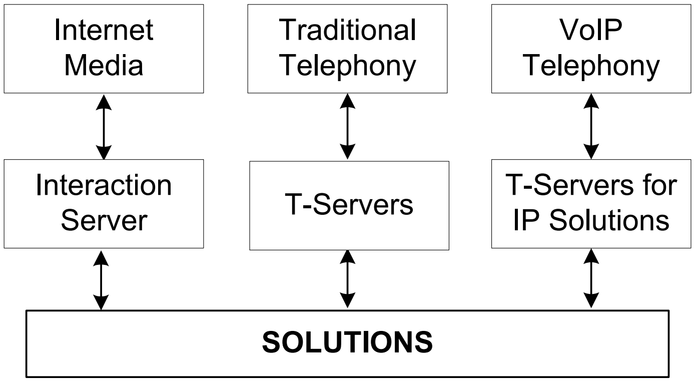

Figure 3: Media Layer Architecture

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.

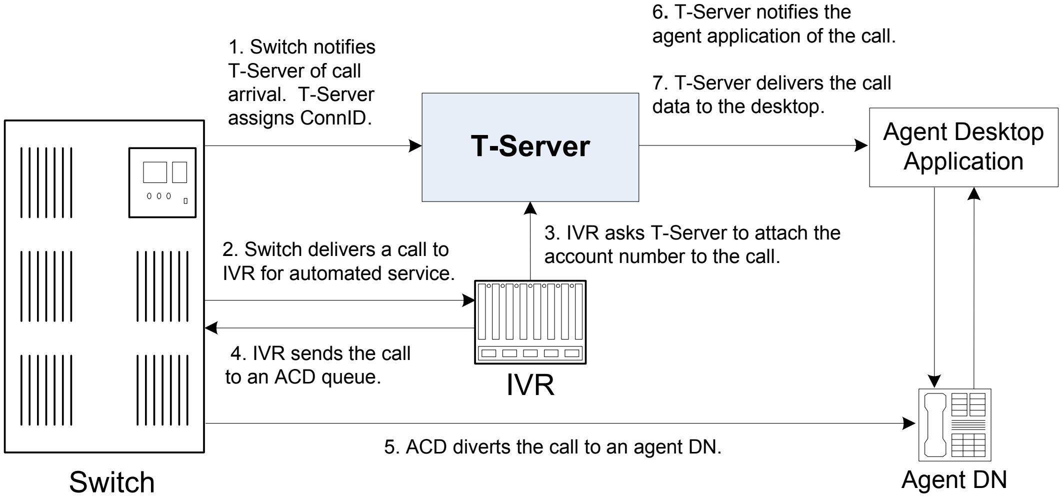
1. Switch notifies T-Server of call arrival. T-Server assigns ConnID.
6. T-Server notifies the agent application of the call.
7. T-Server delivers the call data to the desktop.
T-Server
Agent Desktop Application
3. IVR asks T-Server to attach the account number to the call.
2. Switch delivers a call to IVR for automated service.
4. IVR sends the call to an ACD queue.
IVR
5. ACD diverts the call to an agent DN.
Switch
Agent DN

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 61.

Agent Reservation

T-Server provides support for clients to invoke the agent reservation function, `TReserveAgent()`. This function allows a server application that is a client of T-Server to reserve a DN along with an agent, a `Place`, or both, so that no other T-Server client can route calls to it during a specified reservation interval. Alternatively, when clients use the ISCC feature (see "ISCC Call Data Transfer Service" on page 63), they can use an agent reservation embedded in an ISCC request. (To do so, clients have to specify a certain `Extensions` attribute in an ISCC request when initiating an ISCC transaction. See page 70 for the list of ISCC requests.)

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

In addition to invoking the `TReserveAgent` function, you can customize the Agent Reservation feature by configuring options in the T-Server `Application` object. See"agent-reservation Section" on page 208 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 208).

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 9, "T-Server Common Configuration Options," on page 199. *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 119.

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

Deployment of T-Server

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

Configuration of Telephony Objects

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

Recommendations

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

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

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

Switching Offices

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

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

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

Switches

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

   Two types of access codes exist in a Genesys configuration:

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

   See Chapter 4, “Multi-Site Support,” on page 61, for step-by-step instructions.

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

DNs and Agent Logins

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

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

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

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

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

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

Multi-Site Operations

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

Configuration of T-Server

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

Recommendations

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

Procedure: Configuring T-Server

Start of procedure

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

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

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

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

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

End of procedure

Next Steps

- See “Installation of T-Server” on page 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 119, 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 61.

Procedure: Installing T-Server on Windows

Start of procedure

1. In the directory to which the T-Server installation package was copied, locate and double-click `Setup.exe` to start the installation.
2. When prompted, specify the connection parameters to the Configuration Server associated with this T-Server.
3. When prompted, select the T-Server `Application` you configured in "Configuring T-Server" on page 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 119, 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 61.

Procedure:
Verifying the installation of T-Server

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

Prerequisites

- Procedure: Installing T-Server on UNIX, on page 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 119, 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 61.

Chapter

3

High-Availability Deployment with HA Proxy

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. It also describes how to configure and install HA Proxy, and how to modify the T-Server configuration to operate with HA Proxy. 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.

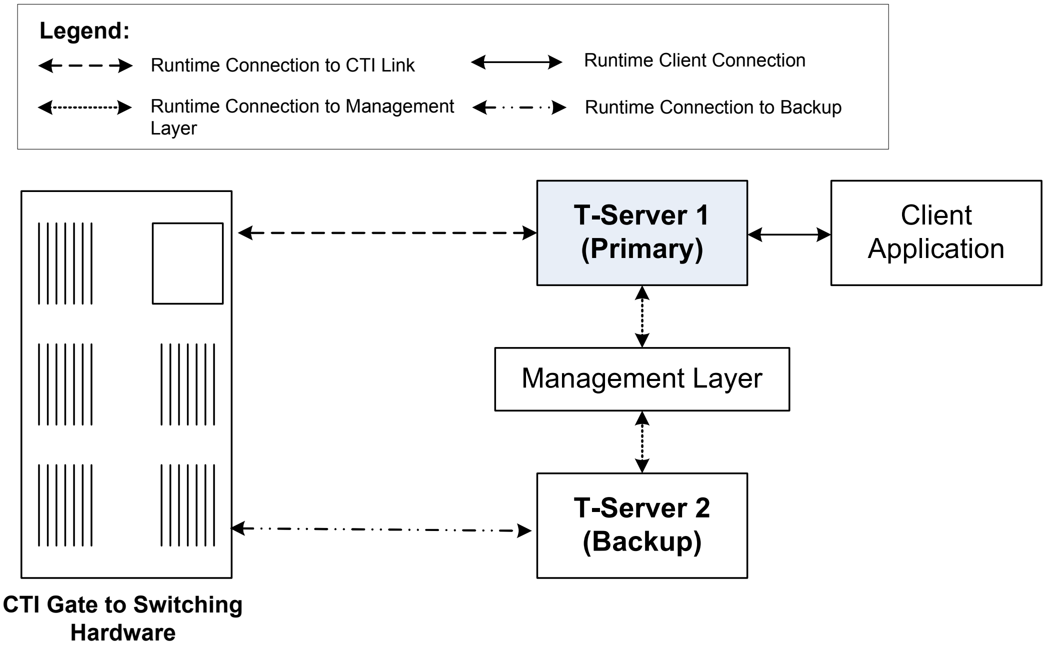
Legend:
Runtime Connection to CTI Link
Runtime Client Connection
Runtime Connection to Management Layer
Runtime Connection to Backup
T-Server 1 (Primary)
Client Application
Management Layer
T-Server 2 (Backup)
CTI Gate to Switching Hardware

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.*

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

Hot Standby Redundancy Architecture

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

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

The details of `hot standby` redundancy implementation between T-Servers and their switches vary depending on switch support for multiple CTI links. If your

T-Server supports `hot standby` (see Table 1 on page 25), refer to Part Two for detailed information on the available `hot standby` schema.

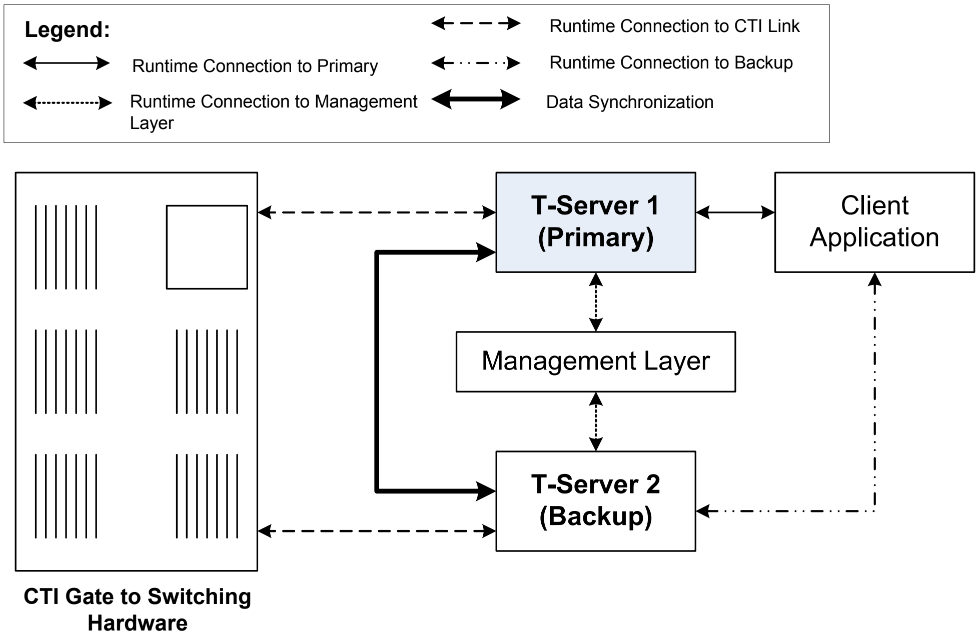
Legend:
Runtime Connection to Primary
Runtime Connection to Management Layer
Runtime Connection to CTI Link
Runtime Connection to Backup
Data Synchronization
T-Server 1 (Primary)
Client Application
Management Layer
T-Server 2 (Backup)
CTI Gate to Switching Hardware

Figure 6: Hot Standby Redundancy Architecture (T-Client Side)

Benefits of Hot Standby Redundancy

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

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

Note: Refer to "ISCC Call Data Transfer Service" on page 63 for ISCC feature descriptions.

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

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

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

HA Proxy Redundancy Architecture

Figure 7 illustrates the switch-independent side of an implementation that includes HA Proxy. Similar to the redundant architecture that does not include HA Proxy (see “Hot Standby Redundancy Architecture” on page 47), the Management Layer assigns the role of the primary server to T-Server 1 and the role of backup to T-Server 2. HA Proxy serves both T-Servers. T-Server clients register with both T-Servers, but only the primary T-Server handles client requests other than the registration requests.

The presence of HA Proxy in your implementation does not change the failover scenario. If T-Server 1 fails, the Management Layer attempts to make T-Server 2 the new primary server, and it starts processing client requests. The Management Layer attempts to restart T-Server 1, and if successful, makes T-Server 1 the new backup server.

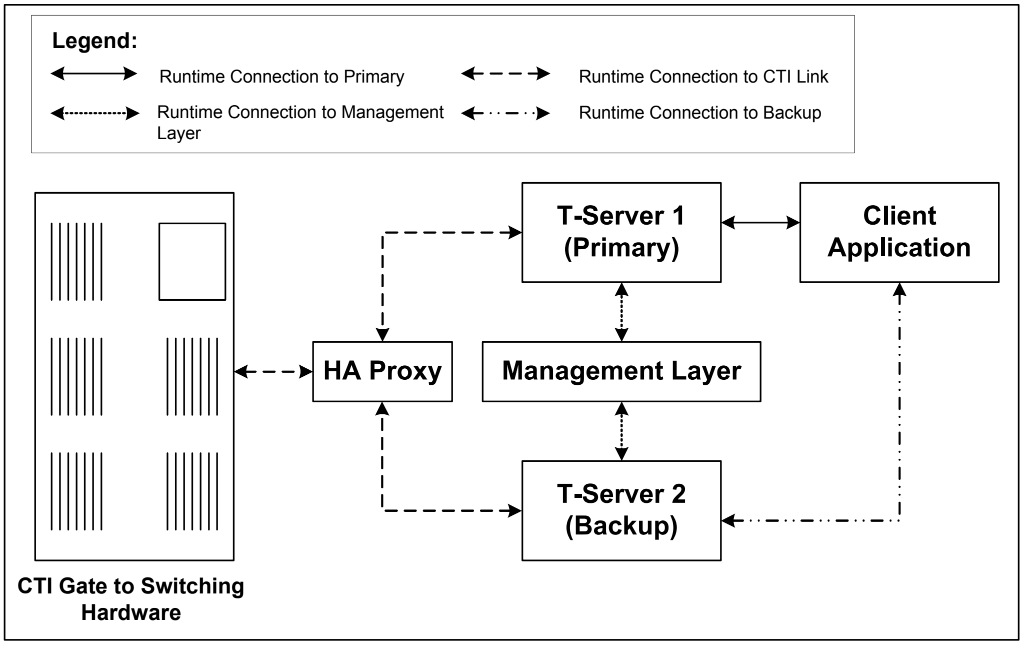
Legend:
Runtime Connection to Primary
Runtime Connection to CTI Link
Runtime Connection to Management Layer
Runtime Connection to Backup
T-Server 1 (Primary)
Client Application
HA Proxy
Management Layer
T-Server 2 (Backup)
CTI Gate to Switching Hardware

Figure 7: Hot Standby Redundancy Architecture with HA Proxy

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 and HA Proxy.

Warning! Genesys strongly recommends that you install the backup and primary T-Servers on the 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 Chapter 9, "T-Server Common Configuration Options," on page 199 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. Manually configure two T-Server Application 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 configurations 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 53).

Modification of T-Servers for Warm Standby

Modify the configuration of both the primary and backup T-Server Application objects for `warm 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 `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. It also gives deployment instructions for HA Proxy components.

The type of HA Proxy component corresponds to the type of the switch. The HA Proxy components are shipped on a separate product CD and must be purchased separately.

General Order of Deployment

Table 1 on page 25 indicates whether HA Proxy components are required for your specific T-Server to support `hot standby` redundancy, and, if so, how many HA Proxy components are required per pair of redundant T-Servers (or per link if so noted).

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

1. Manually 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. Manually configure and install the required number of HA Proxy components as described in “Deployment of HA Proxy” on page 54.
4. Modify the configuration of the backup and primary T-Servers as instructed in “Modification of T-Servers for Hot Standby” on page 56.

After completing the configuration steps, ensure that both T-Servers are installed (see page 59).

Table 1 on page 25 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`.

Deployment of HA Proxy

Procedure: Configuring HA Proxy

Follow the standard procedure to configure an HA Proxy Application object. (Refer to the *Framework 8.1Deployment Guide* for instructions.) Note that Application Templates for HA Proxy for various switches are located on the HA Proxy 7 product DVD.

In addition to the standard configuration steps, go to the Properties dialog box of the HA Proxy object and do the following:

Start of procedure

1. In a multi-tenant environment, on the General tab, specify the same Tenant name you used for the corresponding T-Server.
2. On the Switches tab, specify the Switch that HA Proxy must connect to. This must be the same Switch as specified in the configuration of the corresponding T-Server.
3. On the Server Info tab, specify the host on which HA Proxy is to be installed and the communication port that the T-Servers must use to connect to this HA Proxy.
4. On the Options tab, change the values of the configuration options according to the switch configuration.

   For option descriptions, see Part Two of this guide. Note that in addition to the configuration options described in the "HA Proxy Configuration Options" chapter, HA Proxy also supports common log options, as described in the "Common Configuration Options" chapter.

End of procedure

Next Steps

- See "Installation of HA Proxy" on page 54.

Installation of HA Proxy

After you have configured the HA Proxy Application object using Configuration Manager, install HA Proxy component on its host computer.

The following directories on the Genesys 7.5 HA Proxy product CD contain HA Proxy installation packages:

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

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

Procedure: Installing HA Proxy on UNIX

Prerequisites

- Procedure: Configuring HA Proxy, on page 54

Start of procedure

1. In the directory in which the HA Proxy installation package is located, 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, specify the host and port of Configuration Server.
4. When prompted, specify the user name and password you use to log in to the Configuration Layer.
5. When prompted, specify the host name of the computer on which HA Proxy is to be installed.
6. When prompted, select the HA Proxy `Application` object you configured in "Configuring HA Proxy" on page 54 from the list of applications.
7. Specify the destination directory into which HA Proxy is to be installed.
8. If asked which version of the product to install, the 32-bit or the 64-bit, choose the one appropriate to your environment.

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

End of procedure

Procedure: Installing HA Proxy on Windows

Prerequisites

- Procedure: Configuring HA Proxy, on page 54

Start of procedure

1. In the directory in which the HA Proxy installation package is located, locate and double-click Setup.exe to start the installation.
2. When prompted, specify the host and port of Configuration Server.
3. When prompted, specify the user name and password you use to log in to the Configuration Layer.
4. When prompted, specify the host name of the computer on which HA Proxy is to be installed.
5. When prompted, select the HA Proxy Application object you configured in “Configuring HA Proxy” on page 54 from the list of applications.
6. Specify the destination directory into which HA Proxy is to be installed.
7. Specify the program folder to which HA Proxy is to be added.
8. When icons for HA Proxy appear, click Finish to complete the installation.

End of procedure

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. Click the Connections tab.
16. To enable the T-Server connection to the switch through HA Proxy components, add each HA Proxy Application object that provides the connection to the switch.

Note: ADDP protocol is not supported for a connection between T-Server and HA Proxy. Instead, a switch-vendor proprietary protocol is used to ensure the connection. For relevant configuration parameters, see Part Two of this document.

17. Click the Options tab. Remove or modify the section(s) containing link(s) configuration with a direct connection to the switch for this T-Server. This removes any duplicate connections to the switch which may cause undetermined results. HA Proxy connects to the switch and T-Server connects to the HA Proxy. Refer to Part Two of this guide for relevant configuration options.

18. 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-sync Section” on page 220, in the “T-Server Common Configuration Options” chapter in Part Two of this document.

19. 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 58

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

Start of procedure

1. Make sure the primary and backup 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 119, to test your configuration and installation, or continue with Chapter 4, "Multi-Site Support," on page 61, for more possibilities.

Chapter

4 Multi-Site Support

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

This chapter is divided into the following sections:

Note: Each switch/T-Server combination offers different multi-site options. For details describing your specific switch/T-Server environment, refer to Chapter 9, “T-Server Common Configuration Options,” on page 199.

The following instructions apply to both local and remote switches and T-Servers. Because different vendor switches can be installed at the local and remote locations, this chapter covers several, but not all, possible configurations. To help determine which sections of this chapter apply to your situation, refer to Table 3 on page 79 and Table 4 on page 84.

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

Multi-Site Fundamentals

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

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

- **Call matching**—To link instances of a call distributed across multiple sites and to re-attach essential data associated with the call (`ConnID`, `UserData`, `CallType`, and `CallHistory`). The following T-Server features support this capability:
  - ISCC Call Data Transfer Service (active external routing)—when requested by a T-Server client by specifying the desired destination in the location parameter, and also with various ISCC strategies performed by direct dial or by using the Transfer Connect Service. See "ISCC Transaction Types" on page 70 and "Transfer Connect Service Feature" on page 82.
  - Inter Server Call Control/Call Overflow (ISCC/COF) feature (passive external routing)—applicable when calls are overflowed to another site either directly or manually (see page 83).
  - Number Translation feature (see page 87).
  - Network Attended Transfer/Conference (NAT/C) feature (see page 95).

  Note: When ISCC detects call instance reappearance on a given site, the call is assigned a unique `ConnID` and the user data is synchronized with the previous call instances. This ensures that `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 98)
  - Party Events propagation (see page 99)

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

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

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

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

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

ISCC Call Data Transfer Service

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

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

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

Figure 8 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.*

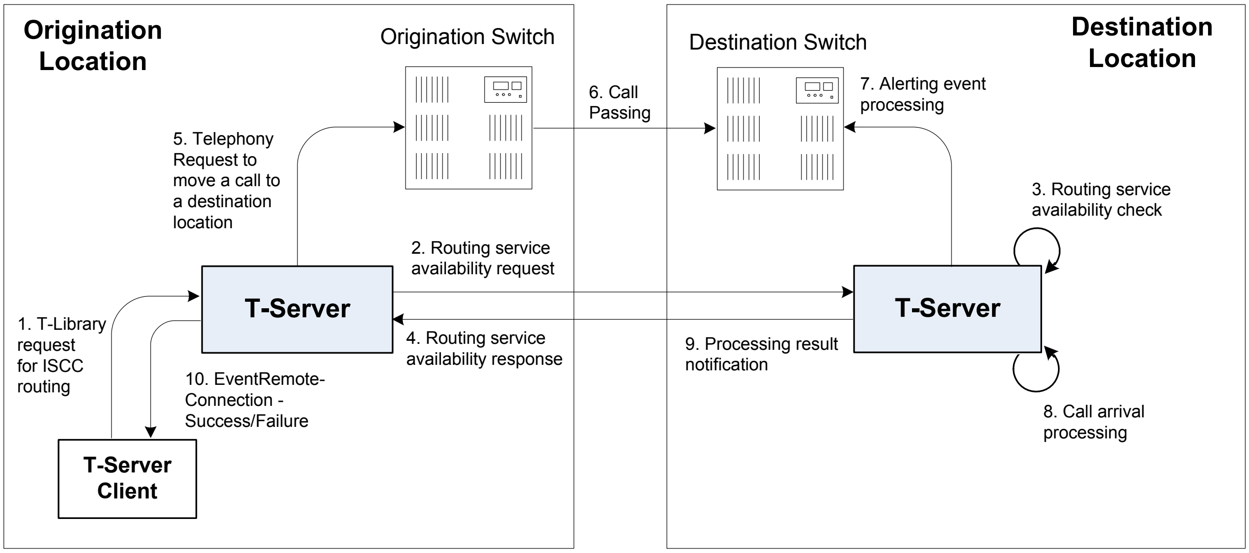
Origination Location
Origination Switch
6. Call Passing
Destination Switch
Destination Location
7. Alerting event processing
5. Telephony Request to move a call to a destination location
3. Routing service availability check
2. Routing service availability request
T-Server
T-Server
1. T-Library request for ISCC routing
4. Routing service availability response
9. Processing result notification
10. EventRemote-Connection - Success/Failure
8. Call arrival processing
T-Server Client

Figure 8: Steps in the ISCC Process

ISCC Call Flows

The following section identifies the steps (shown in Figure 8) 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 106.

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

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

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

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

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

Step 3

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

Note: The `resource-allocation-mode` and `resource-load-maximum` configuration options determine how resources are allocated. For option descriptions, refer to Chapter 9, "T-Server Common Configuration Options," on page 199 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 71).

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

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

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

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

Determining and Configuring Transaction Type Support

You can find descriptions of these transaction types starting on page 71. Use Table 3 on page 79 to identify the transaction types your destination T-Server supports. A blank table cell indicates that T-Server does not support a certain transaction type.

You can configure the transaction types specific to your T-Server as values of the cast-type configuration option specified in the ISCC configuration section extrouter. Refer to Chapter 9, "T-Server Common Configuration Options," on page 199 for the option description.

ISCC Transaction Type General Principles

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

- `direct-ani`, page 71
- `direct-notoken`, page 73
- `dnis-pool`, page 74
- `pullback`, page 75
- `reroute`, page 76
- `route` (aliased as `route-notoken`), the default transaction type, page 77

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

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

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

direct-ani

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

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

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

direct-callid

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

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

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

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

direct-network-callid

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

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

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

direct-uui

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

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

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

direct-notoken

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

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

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

dnis-pool

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

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

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

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

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

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

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

In Load-Balancing Mode

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

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

pullback

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

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

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

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

reroute

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

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

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

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

route

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

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

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

In the Point-to-Point access mode, only one trunk line is used to access an External Routing Point (for example, VDN, CDN) at the destination site. See Figure 9.

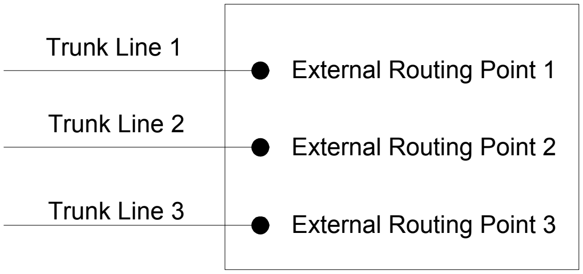

Figure 9: Point-to-Point Trunk Configuration

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

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

In the Multiple-to-Point access mode, trunk lines are assigned to the destination switch’s trunk group, from which calls are routed to the final destination. See Figure 10.

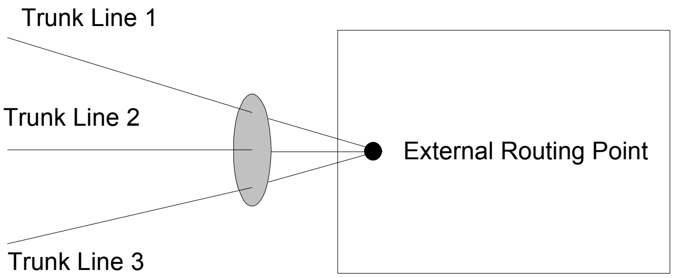

Figure 10: Multiple-to-Point Trunk Configuration

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

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

route-uui

The `route-uui` transaction type employs the dedicated External Routing Point feature of the `route` transaction type (page 77) and the UUI matching feature of the `direct-uui` transaction type (page 73). This transaction type accommodates those switches that require a designated External Routing Point even though they use UUI for tracking.

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

T-Server Transaction Type Support

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

Table 3: T-Server Support of Transaction Types

T-Server Type	Transaction Type										
	route		re-route	direct-callid	direct-uui / route-uui	direct-no-token	direct-ani	direct-digits	direct-network-callid	dnis-pool	pull-back
	one-to-one	multiple-to-one									
Aastra MXONE CSTA I	Yes			Yes[a]		Yes	Yes[a]				
Alcatel A4200/OXO	Yes			Yes		Yes	Yes				
Alcatel A4400/OXE	Yes			Yes[a,b,c]	Yes[d]	Yes	Yes[a]		Yes[e]		
Aspect ACD	Yes	Yes		Yes[c]		Yes[f]	Yes[f]				
Avaya 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 87.

ISCC/COF Call Flow

Figure 11 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.

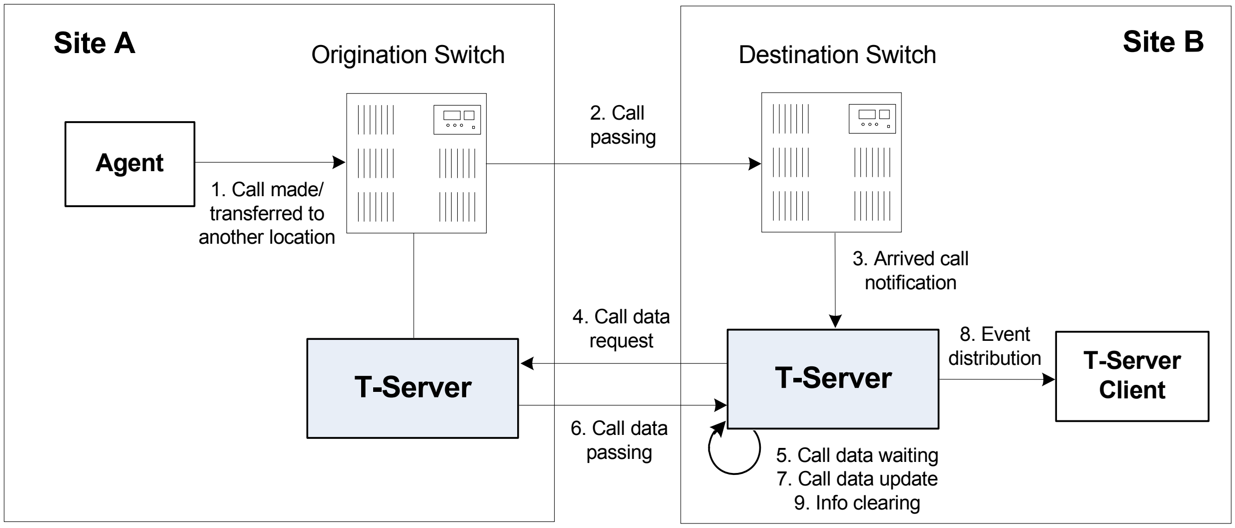

Figure 11: Steps in the ISCC/COF Process

Step 1

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

Step 2

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

Step 3

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

Step 4

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

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

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

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

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

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

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

Step 5

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

Step 6

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

Step 7

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

Step 8

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

Step 9

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

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

Number Translation Feature

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

The translation process involves two algorithms, one for rule selection and the other for the actual translation. Through the first algorithm, T-Server selects a rule that will be used for number translation. Through the second algorithm, T-Server translates the number according to the selected rule definition. See “Number Translation Rules” on page 88 for more information on configuring rules for your environment.

Number translation occurs as follows:

1. The switch reports a number, typically via `AttributeANI`.
2. T-Server evaluates all configured inbound rules to determine which one is the best fit for the received number. The best fit is determined by comparing the length of, and the specific digits in, the input number with the inbound pattern of each configured rule. See "Rule Examples" on page 93 for specific examples.
3. T-Server translates the number according to the selected rule.

To enable T-Server to translate numbers, you must perform specific configuration tasks that are associated with translation. See "Configuring Number Translation" on page 95.

Number Translation Rules

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

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

Using ABNF for Rules

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

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

Common Syntax Notations

Syntax notations common to many of these rules include:

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

Component Notations

Component notations include:

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

  where:

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

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

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

  where:

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

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

  where:

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

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

  where:

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

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

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

  where:

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

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

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

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

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

  where:

  - `digits` are numbers 0 through 9.
  - `range` is a series of digits, for example, 1-3.
  - `sequence` is a set of digits.

- `symbol-part = digits / symbols`

  where:

  - `digits` are numbers 0 through 9.
  - `symbols` include such characters as `+, -,` and so on.

- `range = "[" digits "-" digits "]" group-identifier`

  where:

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

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

- `sequence = "[" 1*(digits [","] ) "]" group-identifier`

  where:

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

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

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

  where:

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

    When you create an `out-pattern`, you include the group identifier only once because the `in-pattern` tells T-Server how many digits belong in that group. For example, `rule-04` (see page 93) is `in-pattern=1AAABBBCCCC; out-pattern=91ABC`.
  - `flexible-length-group` specifies a group composed of 0 or more digits in the group represented by the `group-identifier`. For example, in `in-pattern=1[415,650]A*B`, `*B` represents the flexible length group containing the remaining digits in the number.
  - `entity` represents digits defined for a specific purpose, for example, country code.

  The component `abstract-group` is used only for the `in-pattern`.
- `fixed-length-group = 1*group-identifier`

  See the earlier explanation under `abstract-group`.
- `flexible-length-group = "*" group-identifier`

  See the earlier explanation under `abstract-group`.
- `entity = "#" entity-identifier group-identifier`

  where:

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

  The `entity` component is a special group that assumes some kind of predefined processing, such as the Country Code detection.
- `param-part = ";" param-name "=" param-value`

  where:

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

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

  where:

  - `"ext"` refers to extension.
  - `"phone-context"` represents the value of the `phone-context` option configured on the switch.
  - `"dn"` represents the directory number.

- `param-value = 1*ANYSYMBOL`

  where:

  - `ANYSYMBOL` represents any number, letter, or symbol with no restrictions.

- `group-identifier = ALPHA`
- `entity-identifier = ALPHA`
- `digits = 1*DIGIT`
- `symbols = 1*("-" / "+" / ")" / "("/ ".")`

Recommendations for Rule Configuration

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

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

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

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

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

   ```
   name=rule-01;in-pattern=[1-9]ABBB;out-pattern=AB
   name=rule-02;in-pattern=[1-9]ABBBB;out-pattern=AB
   ```

2. A rule to transform local area code numbers (in `333-1234` format in this example):

   ```
   name=rule-03;in-pattern=[1-9]ABBBBBB;out-pattern=+1222AB
   ```

3. A rule to transform U.S. numbers (in `+1(222)333-4444` format):

   ```
   name=rule-04;in-pattern=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 9, "T-Server Common Configuration Options," on page 199.
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 12 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).

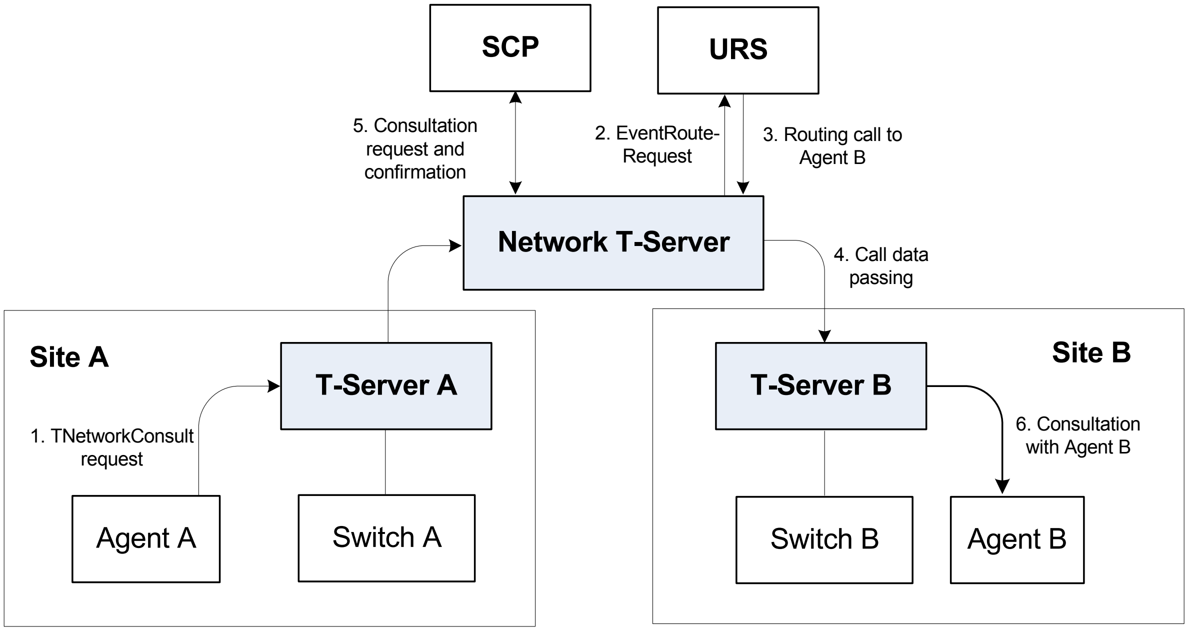

Figure 12: Steps in the NAT/C Process in URS-Controlled Mode

Step 1

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

Step 2

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

Step 3

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

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

Step 4

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

Step 5

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

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

Step 6

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

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

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

Event Propagation Feature

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

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

User Data Propagation

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

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

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

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

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

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

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

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

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

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

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

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

Party Events Propagation

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

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

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

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

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

Switch Partitioning

A multi-site environment with switch partitioning or intelligent trunks can be defined as a configuration of multiple virtual switches (or Switch objects) that are defined in Configuration Manager under a single Switching Office object representing a physical switch. Each Switch object has its own instance of a T-Server application. All T-Server applications connect to the switch via the same or different CTI link or a gateway. (See Figure 13.)

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.

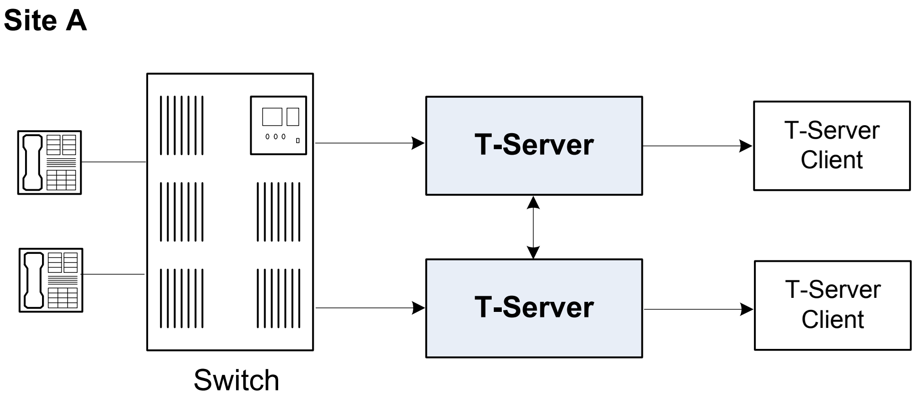

Figure 13: Switch Partitioning Architecture

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 204.

To control race conditions that may occur in the switch-partitioned environment, use the epp-tout configuration option (see page 219).

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

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

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

Table 5: T-Server Support for Switch Partitioning

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

Event Propagation Configuration

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

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

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

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

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

Procedure: Activating Event Propagation: basic configuration

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

Start of procedure

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

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

   This setting enables Party Events propagation.

   For the option description and its valid values, see Chapter 9, "T-Server Common Configuration Options," on page 199.
6. When you are finished, click `Apply`.
7. Click `OK` to save your changes and exit the `Properties` dialog box.

End of procedure

Next Steps

- For advanced feature configuration, do the following procedure:

  Procedure: Modifying Event Propagation: advanced configuration, on page 102

Procedure: Modifying Event Propagation: advanced configuration

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

Prerequisites

- Procedure: Activating Event Propagation: basic configuration, on page 102

Overview

You can set Event Propagation parameters using:

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

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

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

Start of procedure

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

   Note: If no Default Access Code is configured, see page 107 for instructions. If no Access Codes are configured, see page 108 for instructions.

4. In the Switch Access Code Properties dialog box that opens, specify a value for the ISCC Protocol Parameters field as follows:
   - To enable distribution of both user data associated with the call and call-party–associated 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 79 shows which transaction types are supported by a specific T-Server, while Table 4 on page 84 shows whether your T-Server supports the NetworkCallID attribute for the ISCC/COF feature. Use this information as you follow the instructions in this chapter.

Note: Before attempting to configure a multi-site environment, Genesys recommends that you plan the changes you want to make to your existing contact centers. You should then gather the configuration information you will need (such as the name of each T-Server application, port assignments, and switch names), and use Configuration Manager to create and partially configure each T-Server object. Review multi-site option values in the “extrouter Section” on page 209 and determine what these values need to be, based on your network topology.

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

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

You must configure these objects for origination and destination locations. Multi-site support features activate automatically at T-Server startup. See “DNs” on page 112 for details.

Applications

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

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

Procedure: Configuring T-Server Applications

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

Start of procedure

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

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

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

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

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

Note: For a list of options and valid values, see "extrouter Section" on page 209, in the "T-Server Common Configuration Options" chapter in Part Two of this document.

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

End of procedure

Next Steps

- See "Switches and Access Codes."

Switches and Access Codes

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

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

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

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

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

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

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

Procedure:
Configuring Default Access Codes

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

Prerequisites

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

Start of procedure

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

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

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

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

End of procedure

Next Steps

- See "Configuring Access Codes."

Procedure: Configuring Access Codes

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

Prerequisites

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

Start of procedure

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

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

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

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

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

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

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

Table 6: Target Type: ISCC Protocol Parameters

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

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

Table 7: Target Type: ISCC Call Overflow Parameters

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

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

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

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

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

Route Type Field Value	ISCC Transaction Type
Direct Network Call ID	direct-network-callid
Direct No Token	direct-notoken
Direct 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 84 to determine if your T-Server supports the ISCC/COF feature.

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

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

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

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

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

End of procedure

Procedure: Configuring access resources for non-unique ANI

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

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

Start of procedure

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

End of procedure

Procedure: Modifying DNs for isolated switch partitioning

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

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

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

Start of procedure

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

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

End of procedure

Configuration Examples

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

Multiple Transaction Types

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

In this example, you configure an origination and a destination switch for as described in "Switches and Access Codes" on page 106.

1. Among configured `Switches`, select the origination `Switch`.
2. Open the `Switch Properties` dialog box and click the `Default Access Codes` tab.
3. Click `Add` to open the `Access Code Properties` dialog box.
4. Set the `Access Code` field to `9`.
5. When you are finished, click `Apply`.
6. Among configured `Switches`, select the destination `Switch`.
7. Under the destination `Switch`, configure a DN as described in "Configuring access resources for the route transaction type" on page 112.
8. Set the DN `Number` field to `5001234567`.
9. Click the `Advanced` tab of this DN's `Properties` dialog box.
10. Select the `Use Override` check box and enter `1234567` in the `Use Override` field.
11. When you are finished, click `Apply` or `Save`.
12. Use a T-Server client application to register for this new DN with the destination T-Server and, therefore, with the switch.
13. Request to route a call from any DN at the origination switch to the destination DN you have just configured:
    - If you are using the `route` ISCC transaction type, the client requests that T-Server deliver a call to a destination location using the DN number `5001234567`. ISCC requests that the switch dial one of the external routing points at the destination location, using the value either of the `Access Number` field or of the `Access Code` field, which is `9`, concatenated with the external routing point at the destination location. The call is routed to the DN number `5001234567`.

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

Call Overflow Methods

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

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

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

when configuring `Switch Access Codes` as described on page 108.

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

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

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

when configuring `Switch Access Codes` as described on page 108.

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

Next Steps

Continue with Chapter 5, "Starting and Stopping T-Server Components," on page 119 to test your configuration and installation.

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 119.
6. When you are finished, click `Apply`.
7. Click `OK` to save your changes and exit the `Properties` dialog box.

End of procedure

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

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

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

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

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

Starting with Startup Files

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

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

When using a startup file, verify that the startup parameters the installation script inserted in the startup file are correct. Use the following instructions for UNIX and Windows to start those application for which startup files are created. See the appropriate sections in "Starting Manually" on page 123 to identify which applications should be running for a particular application to start.

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

Start of procedure

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

```
sh run.sh
```

End of procedure

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

Start of procedure

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

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

Or

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

  ```
  startServer.bat
  ```

End of procedure

Starting Manually

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

The command-line parameters common to Framework server components are described on page 119.

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

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

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

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

Table 9: T-Server and HA Proxy Executable Names

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

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

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

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

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

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

HA Proxy

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

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

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

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

The command-line parameters common to Framework server components are described on page 119.

Procedure: Starting HA Proxy on UNIX manually

Start of procedure

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

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

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

   Table 9 on page 124 lists HA Proxy executable names for supported switches.

End of procedure

Procedure: Starting HA Proxy on Windows manually

Start of procedure

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

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

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

   Table 9 on page 124 lists HA Proxy executable names for supported switches.

End of procedure

T-Server

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

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

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

The command-line parameters common to Framework server components are described on page 119.

Procedure: Starting T-Server on UNIX manually

Start of procedure

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

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

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

   Table 9 on page 124 lists T-Server executable names for supported switches.

End of procedure

Procedure: Starting T-Server on Windows manually

Start of procedure

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

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

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

   Table 9 on page 124 lists T-Server executable names for supported switches.

End of procedure

Verifying Successful Startup

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

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

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

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

Stopping Manually

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

Procedure:
Stopping T-Server on UNIX manually

Start of procedure

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

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

End of procedure

Procedure:
Stopping T-Server on Windows manually

Start of procedure

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

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

End of procedure

Starting and Stopping with Windows Services Manager

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

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

where the command-line parameters common to Framework server components are described on page 119 and

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

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

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

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

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

Next Steps

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

Part

2 T-Server Configuration

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

- Chapter 6, “Switch-Specific Configuration,” on page 133, describes compatibility and configuration information specific to the Sample, including how to set the DN properties and recommendations for the switch configuration.
- Chapter 7, “Supported T-Server Features,” on page 145, describes which features are supported by this Sample T-Server including T-Library functionality, use of the Extensions attribute, and error messages.
- Chapter 8, “Common Configuration Options,” on page 177, describes log configuration options common to all Genesys server applications.
- Chapter 9, “T-Server Common Configuration Options,” on page 199, describes configuration options that are common to all T-Server types including options for multi-site configuration.
- Chapter 11, “HA Proxy Configuration Options,” on page 255, describes configuration options specific to HA Proxy for the Sample switch.
- Chapter 10, “T-Server-Specific Configuration Options,” on page 227, describes configuration options specific to the Sample T-Server including the link-related options—those which address the interface between T-Server and the switch.
- Appendix , “Using LinkPlexer with T-Server,” on page 265, describes LinkPlexer functionality supported by T-Server for Nortel Communication Server 2000/2100 and provides guidelines for using this product with T-Server.

New in T-Server and HA Proxy for Nortel Communication Server 2000/2100

The following new features are available in the initial 8.1 release or have been added in the most recent 8.0 release of Nortel Communication Server 2000/2100:

- **Support for Supplementary Queue DNs and Routing Points.** T-Server now supports `EventDNBackInService` and `EventDNOutOfService` events for Supplementary Queue DNs and Routing Points. Two new configuration options have been introduced to turn these events on and off:
  - `primary-qrp`
  - `relate-primary-supp-qrp`
- **Support for operating systems.** T-Server is now supported on the following platforms:
  - AIX 7.1 64-bit
  - HP-UX Itanium (version 11i v3)
  - Red Hat Enterprise Linux 5 64-bit

Notes:
- Configuration option changes that apply to T-Server for Nortel Communication Server 2000/2100 are described in "Changes from 8.0 to 8.1" on page 254.
- For a list of new features common to all T-Servers, see "T-Server Deployment" on page 15.

Genesys

Chapter

6 Switch-Specific Configuration

This chapter presents switch-specific reference information for configuring T-Server and HA Proxy for the Nortel Communication Server 2000/2100 switch and includes these sections:

Known Limitations

Several known limitations result from the current T-Server/Nortel Communication Server 2000/2100 interface:

1. The Nortel Communication Server 2000/2100 switch supports many types of lines and devices. Only some telephones support `CTI ANSWER_CALL` or `RELEASE_CALL` requests. If getting the `ANSWER_CALL` or another function to work presents a problem, consult Nortel to find out which set types are supported.
2. The Nortel Communication Server 2000/2100 switch limits the number of ACD groups and extensions that a single T-Server (related to one Switch to Computer Application Interface—SCAI—session and one link set) can associate with the switch. T-Server must associate a DN to receive events regarding the DN. The numbers vary with different Nortel Communication Server 2000/2100 loads, but may be as low as 20 ACD groups (ACD Queues) and 2,000 extensions. If it is relevant to the contact center CTI installation, verify the limitation for the Nortel Communication Server 2000/2100 software load.

3. A call can only be redirected twice in succession through a Nortel Communication Server 2000/2100 Routing Point before the switch routes the call to the default destination. However, if a call is redirected twice, then answered and again transferred to a routing point, it can be redirected two more times. The limitation only applies to consecutive redirects. Therefore, contact center CTI designers must take this limitation into account when designing routing strategies.
4. Solution Control Interface cannot be used to shut down a T-Server if an associated T-Server, connected through External Routing functionality, has disconnected. LCA requests for shutdown will be ignored as the T-Server attempts to reconnect to the associated T-Server.
5. The Genesys call model involves creating a Connection ID for each leg of a call. T-Server may not recognize the latest state of all parties in a complex recursive call involving three, four, or more legs, which may affect reporting. Some new messages are expected in future SCAI protocol releases to allow for more accurate processing of recursive calls.

One known limitation results from the current HA Proxy/Nortel Communication Server 2000/2100 interface:

- If a TCP link is used with the Nortel Communication Server 2000/2100 switch, only one HA Proxy can be used due to a limitation of the switch. If the X.25 link protocol is used, configure one HA Proxy per X.25 link, which allows every component to be redundant if at least two links are configured for the same T-Server (for example, the Nortel Communication Server 2000/2100 link set).

Note: For T-Server in High Availability (hot standby) configuration Genesys recommends that you use link version SCAI14 or later with messages enabled.

6. Starting with T-Server version 8.0, the Nortel Communication Server 2000/2100 feature Call Pickup is no longer supported.

Switch Configuration

This section contains information related to the Nortel Communication Server 2000/2100 switch configuration, which will help you determine if the switch is correctly and efficiently operating with Genesys software.

Note: Genesys recommends that the DV- CTI messages, DV-CALL-HELD CTI messages and DV-NONCONTROLLER-RELEASED CTI messages are enabled on the switch. This will provide T-Server with a more accurate view of the switch state. The ncr-enabled and call-held-enabled configuration options must be configured to inform T-Server of these settings.

Service Version

When T-Server connects to the switch, it provides a password, a business group ID, and other parameters, including the Service Version (the SCAI protocol version). The version available on each Nortel Communication Server 2000/2100 depends on the software loaded on the switch. For example—a switch with load SCAI09 has SCAI protocol versions available up to and including SCAI11. Consult Nortel to find out which SCAI protocol is available.

Note: The Nortel Communication Server 2000/2100 is backward compatible. For example, a switch with load NA009 can be logged in with Service Version SCAI09 and SCAI10. However, the switch provides the specified level of protocol to the application. Refer to the Nortel documentation for more information regarding this functionality.

TCP Link Set Name

When a TCP link is established with the switch, the TCP link set name is required by the T-Server application to establish the connection. The TCP link set name can be found in the Nortel Communication Server 2000/2100 SCAICOMS table—for example, LINKSET_TCP1.

ACD Queues Usage

The Nortel Communication Server 2000/2100 ACD Queues can be used as three different DN Types in the Genesys environment: ACD Queue, Routing Point, and External Routing Point (see “Setting the DN Properties” on page 142).

An ACD Queue contains agents; customer calls are queued at the ACD Queue while waiting for available agents on the queue. The supplementary DNs are reported to Genesys applications as DNIS numbers.

To be used as a Routing Point or an External Routing Point, an ACD Queue must be set up in the switch with no agents configured for this queue. If Supplementary DNs are used, they can each be defined as an individual

Routing Point or an External Routing Point (or a combination) as long as they can share a single switch-default destination. Since supplementary DNs are configured under one physical ACD Queue, only one default is available in the switch for all DNs instead of one per DN. This type of queue usage counts only as one default destination per session, which overcomes the Nortel Communication Server 2000/2100 limitation of 20.

For clearer reporting, do not send calls to the Primary DN if supplementary DNs are configured in a queue used as a Routing Point or an External Routing Point.

If the ACD Queue is configured with either the `CDN` or `SCAIREDIR` option, and this ACD Queue also utilizes both primary and supplementary DNs, then the primary DN should be configured in the Configuration Layer as a Routing Point on which no strategies are to be loaded (that is, a Routing Point that is not used for routing).

For an ACD Queue configured as a Routing Point or an External Routing Point, the ACD Queue datafill on the Nortel Communication Server 2000/2100 switch must contain a CDN or a `SCAIREDIR` option to allow routing. A CDN allows a call to receive a treatment (RAN, Music, ringing, and so forth) while waiting for a target, whereas a call must be routed promptly from a `SCAIREDIR` option to a destination within a timeout configured in the switch.

CDN-Supported Treatments

The CDN Routing Points support the following treatment types:

- Announcement (RAN)
- CancelCall
- Music
- Busy
- Silence
- FastBusy
- Ringback

Note: If CDNs are not available and a call treatment is required, use an IVR.

ACD Position Configuration

The Nortel Communication Server 2000/2100 ACD Positions include a directory number and a four (or five) digit PositionID as unique identifiers. You must specify both for the Genesys environment. When configuring a DN, enter the full directory number in the DN Number Properties field and the four or five digit PositionID in the `Association Properties` field.

On the Nortel Communication Server 2000/2100 switch, calls are distributed (from an ACD Queue) or routed (from a Routing Point) to a PositionID and not to the corresponding Directory Number (DN). To allow routing to a DN, the PositionID that corresponds to the DN must be specified in the Association Properties field in the Configuration Layer.

For example—if Universal Routing Server requests that T-Server route a call to DN 1236991123, T-Server must actually route the call to the corresponding PositionID (such as, 12345).

If the Nortel Communication Server 2000/2100 switch is configured to use passwords with the agent logins, the agents must enter these passwords in a desktop application window. The Configuration Layer requires no additional configuration.

Routing to ACD Positions

To use skill-based or direct-to-agent routing, the Nortel Communication Server 2000/2100 has a feature that allows calls to be routed directly to an ACD Position (Incalls key) from a Routing Point (CDN, SCAIREDIR). The switch also allows calls to be transferred directly from a DN of an ACD Position type to a DN of the same type. For this feature, the Nortel Communication Server 2000/2100 must have the ACDXFER option configured on the ACD Group to which the agents belong.

Extension Configuration

The Nortel Communication Server 2000/2100 switch treats both a secondary DN of an ACD Agent and a stand-alone Centrex line as a Centrex Extension. To enable either within the CTI environment, the ECM option with the proper attributes enabled must be added to those lines in the switch.

Agent Events

You can use the Nortel Communication Server 2000/2100 switch to turn on or off any CTI event that is sent via the associated CTI link using the datafill in the SCAIPROF and SCAISSRV Nortel Communication Server 2000/2100 tables. The Agent Events (DV_AGENT_LOGGED_IN, DV_AGENT_READY) must be enabled, so that T-Server can properly track the agent states. These messages are enabled through the proper ACDEVENTS<XX>$ entry in the SCAIPROF table.

Logging in or out as well as switching to the Ready or NotReady state can be performed using a soft phone or a phoneset. In both cases, the switch provides complete information for T-Server to track agents states.

Messages

To ensure that the CTI link supports the desired switch functionality, all message categories can be enabled on a given link set in the Nortel Communication Server 2000/2100 `SCAIPROF` table. In some instances, the message volume needs to be lowered on the CTI links. The following table gives basic information on the relationship between a Nortel Communication Server 2000/2100 message category and the functionality enabled by the category. The most current listing of Nortel Communication Server 2000/2100 message categories with detailed information on the functionality is contained in the related Nortel documentation.

Table 10: Functionality Enabled by Message Categories

Message Category	Functionality Enabled
CTXEVENT	Events regarding Extensions and Centrex lines
ACDEVENT	Events regarding ACD positions (Incalls key)
DNQUERY	An extension status query (used by Genesys in tracking an extension status for certain call types)
RESOURCE	An ACD Queue status query
ROUTING	Routing enabled with the `SCAIREDIR` option
TPQC & ICCM	Routing and treatments of calls on CDNs
TPCC	Desktop functionality: • Initiation of a transfer and conference for an ACD position (Incalls) • Call answer, release, and holding
TPAC	Desktop functionality: • Agent logging in and out • Agent changing state to `Ready` and `NotReady`
CALLINIT	Desktop functionality: • Initiation of an outbound call
SCAI3WC	Desktop functionality: • Initiation of a transfer, conference, and so forth, for an extension or a Centrex line

Table 10: Functionality Enabled by Message Categories (Continued)

Message Category	Functionality Enabled
SCAICC	Desktop functionality: • Call answer, release, and holding
CPGEVENT	Call progress functionality: • Indicates the progress of calls and therefore can be used to replace DNQUERY. This functionality requires Service Version SCA114 or higher.

Network Node ID

The network-node-id configuration option specifies the switch that the host uses for communication to the link. See the descriptions for this option on page 241 and for the local-node-id option on .

T-Server is required to supply the Nortel Communication Server 2000/2100 Network Node ID (network-node-id) as part of establishing a link to the switch. After the T-Server has successfully connected to the switch, the switch includes the Network Node ID in all call-related messages that are sent to T-Server. The Network Node ID forms a part of the Network Call ID, which consists of the Network Node ID and the Local Call ID. The Local Call ID uniquely identifies a call on one switch, while the Network Call ID uniquely identifies a call among many switches.

In addition, the Local Call ID is used in conjunction with the Network Node ID to identify and track calls on the same switch and among different switches. However, in SCAI10 and earlier versions of CompuCall, the tracking of calls between switches was not always correct because the Local Call ID changes when the call is transferred (or otherwise moved) from one switch to another.

In version NA010 (SCAI12), this switch functionality was improved by causing the Local Call ID to remain the same when a call is moved from one switch to another. In addition, a new Network Node ID (NA010 Network Node ID) was introduced that more accurately and uniquely identifies a specific switch.

For version SCAI12 and later, T-Server requires the specification of both the original Network Node ID (network-node-id) and the new ID (na010-network-node-id). This feature is called Network ICM and must be enabled on the switch in order to use this functionality.

The value of the network-node-id configuration option must be the same as the entry in the Nortel Communication Server 2000/2100 SCAIGRP table and the na010-network-node-id configuration option must be the same as the originating point code (OPC) in the switch. In every instance, T-Server uses the network-node-id to connect to the first link. In a multilink environment

T-Server uses the na010-network-node-id for the second and subsequent links. In either case (single or multilink environment), when T-Server connects to the switch with SCAI12 or later and the Network ICM is enabled on the switch, the switch always sends the NA010 Network Node ID in call-related messages.

Note: See Chapter 15, "T-Server Common Configuration Options," page 199, for detailed descriptions of these Nortel Communication Server 2000/2100 specific options.

InvokeIDs

All switch messages (such as CALL-OFFERED) contain a unique InvokeID to distinguish each message from one another. Two sets of InvokeIDs are used with each using a series of sequential numbers as the unique identifiers. The first set is used by T-Server to send messages to the switch while the second set is used by the switch to send messages to T-Server. Once a response has been sent for the message, the unique InvokeID number is available to be reassigned.

Table 11: Sequential InvokeID Numbers

Set of InvokeIDs	Series of Numbers
Range used by T-Server	0-511 (hex 0-0x1ff)
Range used by the switch	512-1023 (hex 0x200-0x3ff)

Multilink Configuration

The Nortel Communication Server 2000/2100 switch can support multiple X.25 links defined as a link set. Up to eight Switched Virtual Circuits (SVC) can be specified per link set as T-Server supports multilink functionality. When T-Server connects to the switch with more than one SVC, a Redundancy or Load-Sharing connection is established.

Note: On the Windows platform, T-Server supports the use of the Eicon X.25 card to establish an X.25 link to the switch. Other X.25 cards may not function with T-Server unless the card is compatible with the Eicon API. Check with your hardware vendor for details on compatibility.

For each link, the DV_APPL_LOGON message is sent to open the connection and the DV_APPL_LOGOFF is sent to close it. Only one DV_DN_ASSOCIATE needs to be sent per DN, regardless of the number of links established. The Nortel Communication Server 2000/2100 switch shares the message load across the links and, therefore, utilizes all available links (that is, more than a single link is used). One T-Server must correspond to one link set, as defined and configured in the switch. For more information on the load-sharing capability of the Nortel Communication Server 2000/2100 SCAI links, refer to the *Nortel NT NIS Q218* document.

When configuring T-Server for multi-site support:

- Contact Nortel for link and Nortel Communication Server 2000/2100 capacity information.
- The Nortel Communication Server 2000/2100 switch supports one TCP connection per segment to each T-Server. The Nortel Communication Server 2000/2100 switch supports from one to a maximum of eight X.25 connections per segment to each T-Server.

Switch Error Messages

Some errors found in T-Server logs are related to switch error messages. Consult the *Nortel NT NIS Q218* specification for an interpretation of the switch error condition if the T-Server log for the Nortel Communication Server 2000/2100 contains an error of the following format:

```
12/10/xx@05:41:46 [ 05:41:46.4488 ]Link1: return error
   invoke id = 0x1a0
   error value = 0x2


03/01/00@11:47:46 [ 11:47:46.5423 ]link-dbg: reject
   invoke id = 0x82
   0x81 0x1 0x2# ?
```

T-Server error messages are described in “Error Messages” on page 171.

Setting the DN Properties

Table 12: Setting the DN Properties for the Nortel Communication Server 2000/2100 Switch

Switch DN Type	DN Type in the Configuration Layer	Switch-Specific Type	Register	Comments
Primary DN	ACD Queue	1	true	The primary DN of an ACD Queue
Primary DN	Routing Point, External Routing Point	1	true	Routing Point (option `SCAIREDIR` in Nortel Communication Server 2000/2100). An ACD Queue should not contain agents.
Primary DN	Routing Point, External Routing Point	3	true	For SCAI10+ (NA008+), the primary DN of an ACD Queue or a Routing Point (option CDN on Nortel Communication Server 2000/2100).
Supplementary DN	Routing Point, External Routing Point, ACD Queue	2	any	One of 16 possible Supplementary DNs of a Routing Point.

Table 12: Setting the DN Properties for the Nortel Communication Server 2000/2100 Switch (Continued)

Switch DN Type	DN Type in the Configuration Layer	Switch-Specific Type	Register	Comments
Incalls Key DN	ACD Position	any	any	The first key on an ACD set. **Note 1:** `PosID` can be specified in the Configuration Layer. In the Configuration Layer, enter the `PosID` parameter into the `Association` field on the `General Tab` of the DN Properties window. **Note 2:** To add a wrap-up time for an agent in the Configuration Layer, use the `Login IDs` list on the `Agent Info tab` of the Persons Properties window.
SDN (Secondary DN)	Extension	any	true	The second (or third or fourth, and so forth) key on an ACD set.
IBN Local (Centrex DN)	Extension, Voice Treatment Port, Call Processing Port, IVR Port	any	true	Any customer DN to which the `ECM` option has been added.

Genesys-Specific DN Types

The following DN types exist only in the Genesys environment and are used by an internal product to support functionality such as sending e-mail and chat messages to the router or agent desktops. The following DN types do not represent functionality specific to the Nortel Communication Server 2000/2100 switch:

- Virtual Routing Point
- Communication DN
- CoBrowse
- Chat

- E-mail Address
- Voice over IP Port

Genesys

Chapter

7 Supported T-Server Features

This chapter describes the telephony functionality supported by the T-Server for Nortel Communication Server 2000/2100. It includes these sections:

T-Library Functionality

The tables in this chapter present T-Library functionality supported in T-Server for Nortel Communication Server 2000/2100. The table entries use these notations:

N—Not supported

Y—Supported

E—Event only is supported

I—Supported, but reserved for Genesys Engineering

In Table 13, 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*.

Table 13 reflects only that switch functionality used by Genesys software and might not include the complete set of events offered by the switch.

Certain requests listed in Table 13 are reserved for internal use and are listed here merely for completeness of information.

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

Table 13: 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,b]		EventRegistered	Y
TUnregisterAddress[a]		EventUnregistered	Y
Call-Handling Requests			
TMakeCall[cd]	Regular	EventDialing	Y
	DirectAgent[e]		N
	SupervisorAssist[d]		N
	Priority[d]		N
	DirectPriority[d]		N
TAnswerCall[f]		EventEstablished	Y
TReleaseCall[e]		EventReleased	Y
TClearCall		EventReleased	N

Table 13: Supported T-Library Functionality (Continued)

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

Table 13: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
Call-Routing Requests			
TRouteCall[c,h]	Unknown	EventRouteUsed	I
	Default		I
	Label		N
	OverwriteDNIS		N
	DDD		N
	IDDD		N
	Direct		N
	Reject		N
	Announcement		N
	PostFeature		N
	DirectAgent		I
	Priority		N
	DirectPriority		N
	AgentID		N
	CallDisconnect		N
Call-Treatment Requests			
TApplyTreatment[i]	Unknown	(EventTreatmentApplied+ EventTreatmentEnd)/ EventTreatmentNotApplied	N
	IVR[j]		N
	Music		Y
	RingBack		Y
	Silence		Y
	Busy		Y
	CollectDigits		N
	PlayAnnouncement		N

Table 13: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TApplyTreatment[k]	PlayAnnouncementAndDigits		N
	VerifyDigits		N
	RecordUserAnnouncement		N
	DeleteUserAnnouncement		N
	CancelCall		Y
	PlayApplication		N
	SetDefaultRoute		N
	TextToSpeech		N
	TextToSpeechAndDigits		N
	FastBusy		Y
	RAN		Y
TGiveMusicTreatment		EventTreatmentApplied	N
TGiveRingBackTreatment		EventTreatmentApplied	N
TGiveSilenceTreatment		EventTreatmentApplied	N
DTMF (Dual-Tone Multifrequency) Requests			
TCollectDigits		EventDigitsCollected	N
TSendDTMF		EventDTMFSent	N
Voice-Mail Requests			
TOpenVoiceFile		EventVoiceFileOpened	N
TCloseVoiceFile		EventVoiceFileClosed	N
TLoginMailBox		EventMailBoxLogin	N
TLogoutMailBox		EventMailBoxLogout	N
TPlayVoice		EventVoiceFileEndPlay	N
Agent and DN Feature Requests			
TAgentLogin		EventAgentLogin	Y

Table 13: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TAgentLogout		EventAgentLogout	Y
TAgentSetReady		EventAgentReady	Y
TAgentSetNotReady		EventAgentNotReady	Y
TMonitorNextCall	OneCall	EventMonitoringNextCall	N
	AllCalls		N
TCancelMonitoring		EventMonitoringCancelled	N
TCallSetForward	None	EventForwardSet	N
	Unconditional		N
	OnBusy		N
	OnNoAnswer		N
	OnBusyAndNoAnswer		N
	SendAllCalls		N
TCallCancelForward		EventForwardCancel	N
TSetMuteOff		EventMuteOff	N
TSetMuteOn		EventMuteOn	N
TListenDisconnect		EventListenDisconnected	N
TListenReconnect		EventListenReconnected	N
TSetDNDOn		EventDNDOn	N
TSetDNDOff		EventDNDOff	N
TSetMessageWaitingOn		EventMessageWaitingOn	N
TSetMessageWaitingOff		EventMessageWaitingOff	N
		EventOffHook	Y
		EventOnHook	Y
		EventDNBackInService	Y
		EventDNOutOfService	Y

Table 13: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
Query Requests			
TQuerySwitch[a]	DateTime	EventSwitchInfo	N
	ClassifierStat		N
TQueryCall[a]	PartiesQuery[l]	EventPartyInfo	Y
	StatusQuery[j]		N
TQueryAddress[a]	AddressStatus	EventAddressInfo	Y
	MessageWaitingStatus		Y
	AssociationStatus		Y
	CallForwardingStatus		N
	AgentStatus[m]		Y
	NumberOfAgentsInQueue[k]		Y
	NumberOfAvailableAgents-InQueue[k]		Y
	NumberOfCallsInQueue[k]		Y
	AddressType[j]		Y
	CallsQuery		N
	SendAllCallsStatus		N
	QueueLoginAudit[n]		Y
	NumberOfIdleTrunks		N
	NumberOfTrunksInUse		N
	DatabaseValue		N
	DNStatus		Y
	QueueStatus		Y

Table 13: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TQueryLocation[a]	AllLocations	EventLocationInfo[o]	I
	LocationData		I
	MonitorLocation		I
	CancelMonitorLocation		I
	MonitorAllLocations		I
	CancelMonitorAllLocations		I
TQueryServer[a]		EventServerInfo	Y
User-Data Requests			
TAttachUserData [Obsolete]		EventAttachedDataChanged	Y
TUpdateUserData		EventAttachedDataChanged	Y
TDeleteUserData		EventAttachedDataChanged	Y
TDeleteAllUserData		EventAttachedDataChanged	Y
ISCC (Inter Server Call Control) Requests			
TGetAccessNumber[c]		EventAnswerAccessNumber	I
TCancelRegGetAccess-Number		EventReqGetAccessNumber -Cancelled	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[p]			
TNetworkConsult		EventNetworkCallStatus	Y

Table 13: Supported T-Library Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TNetworkAlternate		EventNetworkCallStatus	Y
TNetworkTransfer		EventNetworkCallStatus	Y
TNetworkMerge		EventNetworkCallStatus	Y
TNetworkReconnect		EventNetworkCallStatus	Y
TNetworkSingle Step-Transfer		EventNetworkCallStatus	Y
TNetworkPrivateService		EventNetworkPrivateInfo	Y
ISCC Transaction Monitoring Requests			
TTransactionMonitoring		EventACK	Y
		EventTransactionStatus	E

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

b. DN entries submitted in the TRegisterAddress request must be limited to digits. With the current Nortel Communication Server 2000/2100 CTI link configuration, use of nondigit characters can cause unexpected switch behavior.

c. Because 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—a second event response containing the same reference ID as the first event is sent. This second event is either EventRemoteConnectionSuccess or EventRemoteConnectionFailed.

d. Starting with version 7.5, T-Server includes both AuthCode and AccountCode information in the make call message. If the client passes the information in the extensions AccountCode and AuthCode of the TMakeCall request, T-Server includes these in the acctCodeDigits and authCodeDigits fields of the DV-MAKE-CALL request.

e. This subtype is ignored.

f. Supported starting with Nortel Communication Server 2000/2100 SCAI08.

g. Supported starting with Nortel Communication Server 2000/2100 SCAI10. Events are not supported. External parties are supported in SCAI14 or later when the call-progress configuration option is set to true.

h. Request works but RouteType is ignored.

i. See "TApplyTreatment Functionality".

j. Not all Treatment Types are supported and those types not supported have a RingBack default setting.

k. See "TApplyTreatment Functionality".

l. Does not go to the switch. Uses internal information only.

m. For agents only.

n. For queues only.

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

p. 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.

Alternate Call Functionality

Beginning with release 7.2, T-Server for Nortel Communication Server 2000/2100 supports Nortel's "dv-toggle-call" feature (also known as `TAlternateCall`) that was introduced in SCAI version 18. This feature provides an agent with the ability to toggle between the queue and the customer and includes monitoring of call held status, whether it is a manual or a CTI-initiated request.

Switch Requirements

The alternate call functionality and manual hold and unhold functionality, requires these switch resources:

- SCAI service version 18
- `DV-TOGGLE-CALL` message turned `On`
- `DV-CALL-HELD` message turned `On`
- `DV-CALL-UNHELD` message turned `On`

Warning! It is especially important that both `DV-CALL-HELD` and `DV-CALL-UNHELD` are available for extensions and positions. If only one or the other is `On`, T-Server may become unsynchronized during manual toggle/hold/unhold actions.

For more about SCAI, refer to Nortel's functional description document *A00002904.AAxx* or contact your Nortel switch vendor.

TApplyTreatment Functionality

T-Server supports `TApplyTreatment` starting with Nortel Communication Server 2000/2100 SCAI10. When applying a treatment using the `TApplyTreatment` request, T-Server uses either the `MUSIC_DN` or the `ROUTE` parameter for both music and recorded-announcement (RAN) treatments. If both parameters are present in the TApplyTreatment request, T-Server uses the ROUTE parameter first to locate either the music or RAN treatment DN source. T-Server uses the MUSIC_DN parameter when no value is present in the ROUTE parameter.

In the example below, the parameter 512 is used:

```
message RequestApplyTreatment
       AttributeThisDN''1001234567'
       AttributeConnID006c00d54aebc26c
       AttributeTreatmentTypeTreatmentRAN
       AttributeTreatmentParms[29] 00 02 00 00..
              'ROUTE''512'
              'MUSIC_DN''333'
       AttributeReason[32] 00 01 00 00..
              'DefaultReasons''treatment_7'
       AttributeReferenceID3158
```

Unified Party States

T-Server now conforms to the general Unified Party State model by ensuring that all party state transitions are valid. For more information, see “Unified Party States” in the *Genesys Events and Models Reference Manual* and *Voice Platform SDK 8.x .NET (*or *Java) API Reference*.

TDM Call Recording

Starting with release 7.5, T-Server for Nortel Communication Server 2000/2100 supports the TDM Call Recording feature on SCAI 19. This new capability of T-Server requires that the ICM TDM Call Recording functionality be installed and enabled on the switch.

Note: In some outbound scenarios, such as a mute or blind transfer and a conference, trunk information may not be received from the switch until after all corresponding events are distributed. In this case, trunk info is not included in the events.

Inbound Trunk-side

To support the inbound trunk-side of the TDM Call Recording feature, T-Server includes the trunk information extension in the EventRinging, EventQueued, and EventRouteRequest messages. These are provided in the corresponding switch messages: CALL-OFFERED, and CALL-QUEUED. The extensions for these messages are:

- TrunkGatewayType
- TrunkGatewayNumber
- TrunkCarrierNumber

- TrunkChannelNumber.

Outbound Trunk-Side

To support the outbound trunk-side of the TDM Call Recording feature, T-Server includes the trunk information extensions in the EventEstablished and EventNetworkReached messages. This is provided in the corresponding switch message: CALL-PROGRESS farEndAnswered. The extensions for these messages are:

- TrunkGatewayType
- TrunkGatewayNumber
- TrunkCarrierNumber
- TrunkChannelNumber

IP Call Recording

The switch is now able to provide call recording functionality through a new message, DV-RECORD-CALL. To do so, T-Server must expose this recording functionality to clients.

T-Server clients can request that recording be started or stopped on individual DNs. T-Server will ensure that any calls involving such DNs are recorded as the call state allows. This functionality is available on SCAI 20 or above.

Start Recording

Clients initiate call recording by sending a TPrivateService request (with the message: AttributePrivateMsgID 8257566) to T-Server. For more information, see the *Genesys Events and Models Reference Manual* and *Voice Platform SDK 8.x .NET (*or *Java) API Reference*. T-Server replies to the client with an EventACK message if the request is accepted. An EventError message is returned when a DN is not registered by T-Server or the extensions with recording parameters are missing (TERR_INVALID_ATTR). If the request is successfully validated, T-Server stores the host/ports information and will issue the CTI request DV-Record-Call each time a call is active on the DN. If a TDmsRecordingStart message is received when an active call already exists on the DN, T-Server replies with an EventACK message and immediately sends a CTI request for the existing active call.

Stop Recording

Clients stop call recording by sending a TPrivateService request (with the message AttributePrivateMsgID 8257567) to T-Server. For more information, see the *Genesys Events and Models Reference Manual* and *Voice Platform SDK 8.x .NET (*or *Java) API Reference*. T-Server replies with an EventACK message and clears recording parameters stored on the DN. If this request is received while a call is being recorded on the DN, T-Server sends CTI a request to stop recording to the switch.

Automatic recording renewal

T-Server is required to send a start message anytime the configuration of the call changes. This applies to the following scenarios involving any party on the call:

- Hold/Restore
- Add/Drop party
- Transfer party
- Conference party
- Toggle

T-Server uses stored recording parameters received in the TDmsRecordCallStart message to automatically send recording renewal requests as long as the DN used in the TDmsRecordingStart message is connected to the call.

CTI error handling

If the switch returns any error, except invalid call state, in response to the CTI request, T-Server turns off recording for the given DN and distributes an EventHardwareError message to the clients, translating the CTI error code as in Table 14.

Table 14: CTI Error Handling Codes

SCAI error code	Tlib error code	Comment
Not Allowed	TERR_DMS_NOT_ALLOWED	Not subscribed to IPCALLREC in the SCAISSRV table or SOC ICM00085 not ON
Missing parameter	TERR_DMS_MIS_PARAM	Missing required parameter
Invalid parameter	TERR_DMS_INV_PARAM	Unknown/wrong parameter found in message
Invalid parameter content	TERR_DMS_INV_CONTENT	Example: Original address is not valid

T-Server adds extensions with the recording parameters used for the failed CTI request. These are the same extensions as provided by clients in the TDmsRecordingStart message.

Failure Conditions

- HA switchover:

  Call recording information is synchronized between primary and backup T-Servers in hot standby mode. When high-availability (HA) switchover happens due to hardware, software or network failure on the primary T-Server box, backup T-Server becomes primary and continues automatic recording renewal upon changes in recorded call segments.
- Disconnect between T-Server and Call Recorder:

  T-Server does not differentiate between a Call Recorder client and other application connections. When a connection between Call Recorder and T-Server is lost, T-Server does not explicitly stop recording. Recording will continue to be renewed. It is recommended that Call Recorder sends a `TDmsRecordStop` message for all recorded DNs prior to maintenance shutdowns.
- Disconnect between T-Server and CTI link:

  Once connection to CTI link is lost, T-Server has no up-to-date information about calls in progress and cannot issue automatic recording renewals. However, T-Server does continue recording when connection to the CTI link returns. Clients are notified with an `EventLinkDisconnected` message.

IP Call Recording limitations and restrictions

- Recording through replication cannot be supported for calls while on a queue or at an announcement.
- T-Server provides automatic recording renewal only while the DN specified in the recording request stays on the call.
- Only one set of ports per call is supported. When a call is connected to two agents, and both agents need to be recorded using different ports, different parts of the call may be replicated to different ports.
- For calls with more than one recording DN, there is no guarantee which DN will be used to start or renew the call recording with the switch.
- Call recording parameters that are changed (through a client request) while on a call do not take effect until the call is renewed.
- If recording is requested on a call when that call is already being recorded, T-Server does not restart recording.
- If multiple call record requests are made for the same DN, the most recent request takes precedence.
- There is no timeout based retry for a specific `dv-RecordCall` message. The normal `request-timeout` configuration option is used to handle unresponsive `dv-RecordCall` requests.
- Recording is stopped on a call only if:
  - A client requests the recording stopped on a recording DN.
  - There are no other recording DNs on the call.

Supported Nortel Communication Server 2000/2100 SCAI Messages

Table 15 on page 159 details the Nortel Communication Server 2000/2100 SCAI messages supported in the T-Server release 6.1 and this release of T-Server. The table entries use these notations:

N—Not supported

Y—Supported

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

Table 15: Supported Nortel Communication Server 2000/2100 SCAI Messages

Nortel Communication Server 2000/2100 SCAI Messages	T-Server
DV_ADD_PARTY	Y
DV_AGENT_LOGGED_IN_U	Y
DV_AGENT_LOGGED_OUT_U	Y
DV_AGENT_NOT_READY_U	Y
DV_AGENT_READY_U	Y
DV_AGENT_SETACTION_U	Y
DV_AGENT_STATUS_U	Y
DV_ANSWER_CALL	Y
DV_APPL_CONTINUITY_TEST	Y
DV_APPL_LOGOFF	Y
DV_APPL_LOGON	Y
DV_APPL_STAT_QRY	Y
DV_CALL_ANSWERED_U	Y
DV_CALL_CALLINGNAME_U	N
DV_CALL_CONFERENCED_U	Y
DV_CALL_CONSULT_ORIGINATED_U	Y
DV_CALL_OFFERED_U	Y

Table 15: Supported Nortel Communication Server 2000/2100 SCAI Messages (Continued)

Nortel Communication Server 2000/2100 SCAI Messages	T-Server
DV_CALL_PROGRESS_U	Y
DV_CALL_QUEUED_U	Y
DV_CALL_RECEIVED_C	Y
DV_CALL_REDIRECT	Y
DV_CALL_RELEASED_U	Y
DV_CALL_TRANSFERRED_U	Y
DV_CALL_UNHELD_U	Y
DV_CDN_STATUS_U	N
DV_CONFERENCE_PARTY	Y
DV_CONTROLLER_RELEASED_U	N[a]
DV_DN_ASSOCIATE	Y
DV_DN_QUERY	Y
DV_DROP_PARTY	Y
DV_EMK_U	N
DV_GIVE_TREATMENT	Y
DV_HOLD_CALL	Y
DV_LOB_EVENT_U	N
DV_MAKE_CALL	Y
DV_MESSAGE_WAITING_U	Y
DV_MWT_ACTIVATE	N
DV_NONCONTROLLER_RELEASED_U	Y
DV_REASSIGN_AGENT	N
DV_RELEASE_CALL	Y
DV_RESOURCE_QUERY	Y

Table 15: Supported Nortel Communication Server 2000/2100 SCAI Messages (Continued)

Nortel Communication Server 2000/2100 SCAI Messages	T-Server
DV_ROUTE_CALL	Y
DV_SET_CDN_STATE	Y
DV_SET_FEATURE	Y
DV_SET_OFFHOOK_U	Y
DV_TRANSFER_PARTY	Y
DV_TREATMENT_COMPLETE_U	Y
DV_TOGGLE_CALL	Y
DV_UNHOLD_CALL	Y
DV_CALL_HELD_U	Y
DV-RECORD-CALL	Y

a. This message is technically not used as the related NONCONTROLLER_RELEASED message informs T-Server of the same information.

Support for DV_DN_QUERY Messages

T-Server supports the DV_DN_QUERY messages, but its usage is limited by the configuration options described in Table 16:

Table 16: T-Server Support of DV_DN_QUERY Messages

Options	Value	DV_DN_QUERY Usage
call-progress	false	T-Server uses the DV_DN_QUERY message to determine external call connections.
call-progress**	true	T-Server uses the call-progress message, not the DV_DN_QUERY message, to determine external call connections.

Table 16: T-Server Support of DV_DN_QUERY Messages

Options	Value	DV_DN_QUERY Usage
sync-addresses	+extensions	T-Server uses the DV_DN_QUERY message to get the extensions line state after connecting to the link.
use-query-dn**	true	T-Server uses the DV_DN_QUERY message to determine external call connections.
use-query-dn	false	T-Server does not use DV_DN_QUERY message to determine external call connections.

** If the values for the call-progress and use-query-dn configuration option are set to true, the call-progress configuration option takes precedence.

Feature Configuration

The following configuration options support the DV_DN_Query messages feature:

- call-progress
- syncaddresses
- use-query-dn

Support for Make Call Request Handling

When a desktop application sends a TMakeCall request to T-Server, the request is passed on to the Nortel Communication Server 2000/2100 switch as a request to make a call from DN A to DN B. The Nortel Communication Server 2000/2100 rings only the origination DN at DN A and no event is produced at that time. When the agent answers the call (manually or using a desktop answer command), the switch dials the DN at DN B. T-Server then distributes EventDialing for the DN at DN A.

If T-Server monitors the destination DN (DV_DN_ASSOCIATE), it generates EventRinging for the DN at DN B and two instances of EventEstablished when the call is answered by the same DN. If the DN at DN B is an outside party, T-Server periodically sends a message concerning the DN at DN A (DV_DN_QUERY) to the switch to find out when the call is established. After it receives a positive result from the switch, it generates EventEstablished for the DN at DN A.

Note: For additional information about call handling, see "Multi-Site Support" on page 61.

Support for TDeleteFromConference Requests

If a client sends a TDeleteFromConference request on behalf of an internal DN, T-Server responds by sending a DV_RELEASE_CALL message to the switch for that DN, if the complete set of digits is specified in the OtherDN attribute. If the complete set of digits is not specified, T-Server sends a DV_DROP_PARTY message for the internal DN.

For all OtherDNs, T-Server sends a DV_DROP_PARTY message (which does not use the OtherDN attribute) in response to the TDeleteFromConference message. T-Server allows clients to send a TDeleteFromConference request for any call involving at least two parties.

Note: Only the initiator of the conference can delete a consulting party from the call (using DV_DROP_PARTY).

Support for T-Server Dial Plans

Feature Configuration

Three configuration options relate directly or indirectly to the use of dial plans in this T-Server:

- use-dial-plan
- dial-plan-prefix
- set-call-type-with-dialing

If the values for the use-dial-plan and set-call-type-with-dialing configuration options are set to true, the digits in the dial-plan-prefix configuration option are used to determine if the dialed digits are internal or not for the purpose of setting the CallType attribute. If the value for the set-call-type-with-dialing configuration option is set to false then dial plans are not used. If the value for the `use-dial-plan` configuration option is set to `false` then the `dial-plan-prefix` configuration option is not used.

Note: The `CallType` determined by the dial plan is used to determine the behavior in call topology loops (where calls leave and return to the same T-Server). See “new-call-for-unknown-dest” on page 241 for more information on call topology loops.

Table 17 shows examples of various dial plans, indicating which digits T-Server compares and if a match is determined.

Table 17: Dial Plan Examples

dial-plan-prefix	Dialed Digits	Internal Address	Compare Digits	Match
9	1234567	1001234567	1234567	true
9	91234567	1001234567	1234567	true
9	4567	1001234567	4567	true
91	92224567	1002224567	92224567	false
91	911001234567	1001234567	1001234567	true
91	1234567	1001234567	1234567	true
9, 91	91234567	1001234567	1234567	true
9, 91	911001234567	1001234567	1001234567	true
9, 91	1234567	1001234567	1234567	true
9, 91	11001234567	1001234567	11001234567	false
9, 91,1	11001234567	1001234567	1001234567	true

T-Server compares dialed numbers with internal addresses (an address that is configured for a specific switch under the Switches folder in the Configuration Layer) for making processing decisions. If a match between the dialed digits and internal address is determined, then the new call being dialed is Internal, otherwise it is Outbound.

Supported Hot-Standby Configurations

Nortel Communication Server 2000/2100 currently supports the following Hot-Standby configurations:

- `Hot standby` redundancy type for multiple X.25 CTI links with two HA Proxies and two T-Servers.

- `Hot standby` redundancy type for a single CTI link with one HA Proxy and two T-Servers.
- `Hot Standby` redundancy type for Dual CTI Links

Hot-Standby Redundancy Type for Multiple X.25 CTI Links with Two HA Proxies and Two T-Servers

You can deploy HA Proxy for the Nortel Communication Server 2000/2100 in pairs with two CTI links connected to two HA Proxies and two T-Servers configured in the `hot standby` redundancy type (see Figure 14 on page 165). This configuration requires multiple X.25 CTI links.

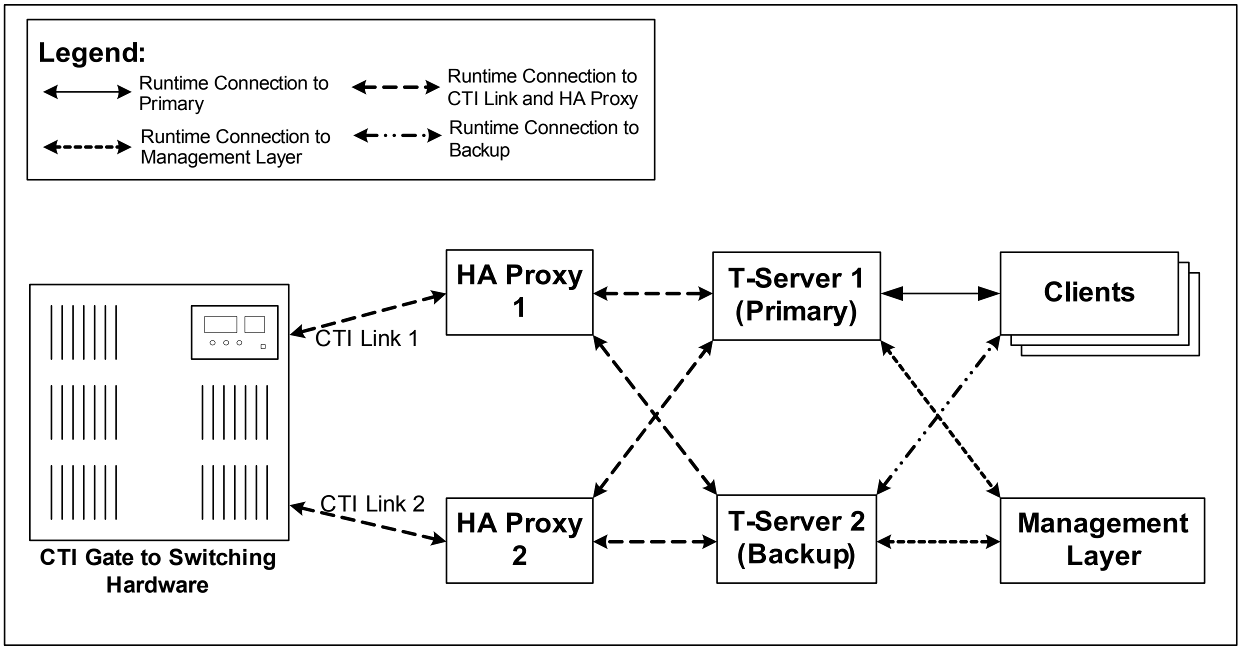
Legend:
Runtime Connection to Primary
Runtime Connection to CTI Link and HA Proxy
Runtime Connection to Management Layer
Runtime Connection to Backup
CTI Link 1
CTI Link 2
CTI Gate to Switching Hardware
HA Proxy 1
HA Proxy 2
T-Server 1 (Primary)
T-Server 2 (Backup)
Clients
Management Layer

Figure 14: Hot-Standby Redundancy Type for Redundant CTI Links with Two HA Proxies and Two T-Servers

Hot-Standby Redundancy Type for a Single CTI Link with One HA Proxy and Two T-Servers

You can deploy HA Proxy for the Nortel Communication Server 2000/2100 in pairs with a single CTI link connected to one HA Proxy and two T-Servers configured in the `hot standby` redundancy type (see Figure 15 on page 166). In this case, the protocol can be either `X25` or `TCP/IP` depending on the type of link on the switch.

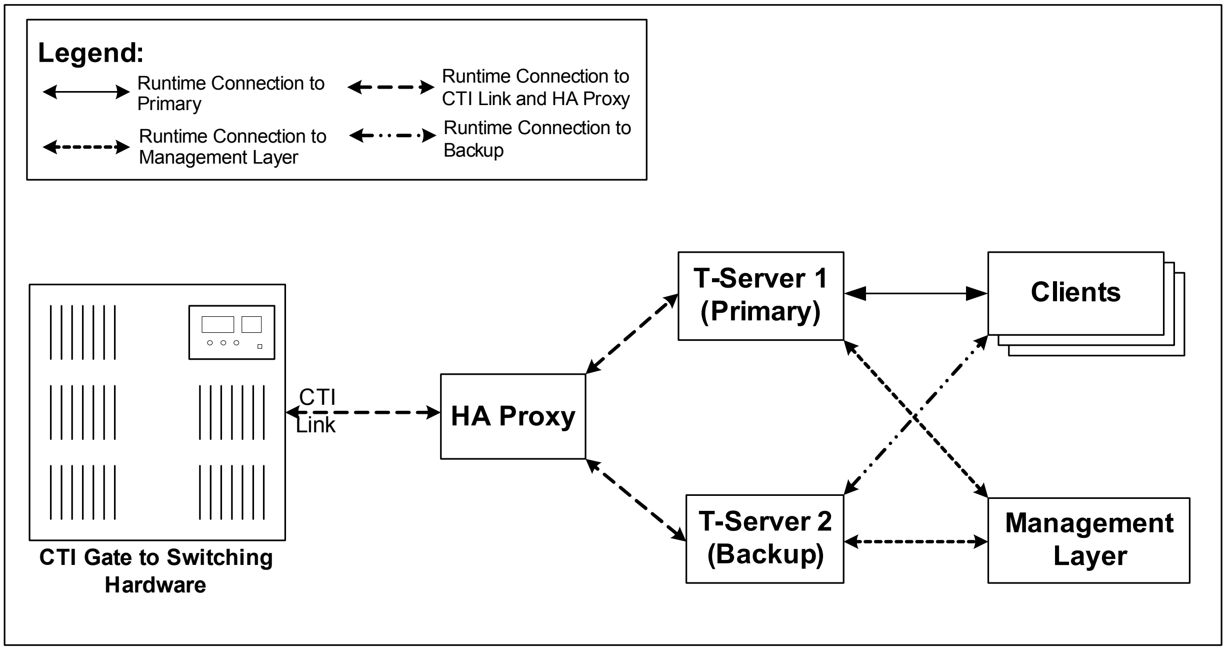
Legend:
Runtime Connection to Primary
Runtime Connection to CTI Link and HA Proxy
Runtime Connection to Management Layer
Runtime Connection to Backup
CTI Link
HA Proxy
T-Server 1 (Primary)
T-Server 2 (Backup)
Clients
Management Layer
CTI Gate to Switching Hardware

Figure 15: Hot-Standby Redundancy Type for Single CTI Link with One HA Proxy and Two T-Servers

The Management Layer is responsible for detection and switchover of a failed T-Server. When it detects a failure, it sends a command to the T-Server in backup mode to switch it to primary mode. That T-Server then registers all telephony resources and acquires all CDNs. Switchover of T-Servers does not affect the HA Proxies. The new T-Server in primary mode also sends a message indicating that it is now in that mode.

A similar process occurs when an HA Proxy fails. In that case, T-Server coordinates the switch to the backup HA Proxy. The primary T-Server sends a command to the `hot standby` HA Proxy to switch it to primary mode. Although the Management Layer can start up and shut down an HA Proxy, it does not control the HA Proxy mode; that control is left to the T-Server in primary mode.

Hot-Standby Redundancy Type for Dual CTI Links

Starting with release 7.5, T-Server for Nortel Communication Server 2000/2100 supports a second mirrored link for TCP/IP connections on SCAI 19 or above. This allows for T-Server Hot Standby implementation without the use of HA Proxy (see Figure 16 on page 167).

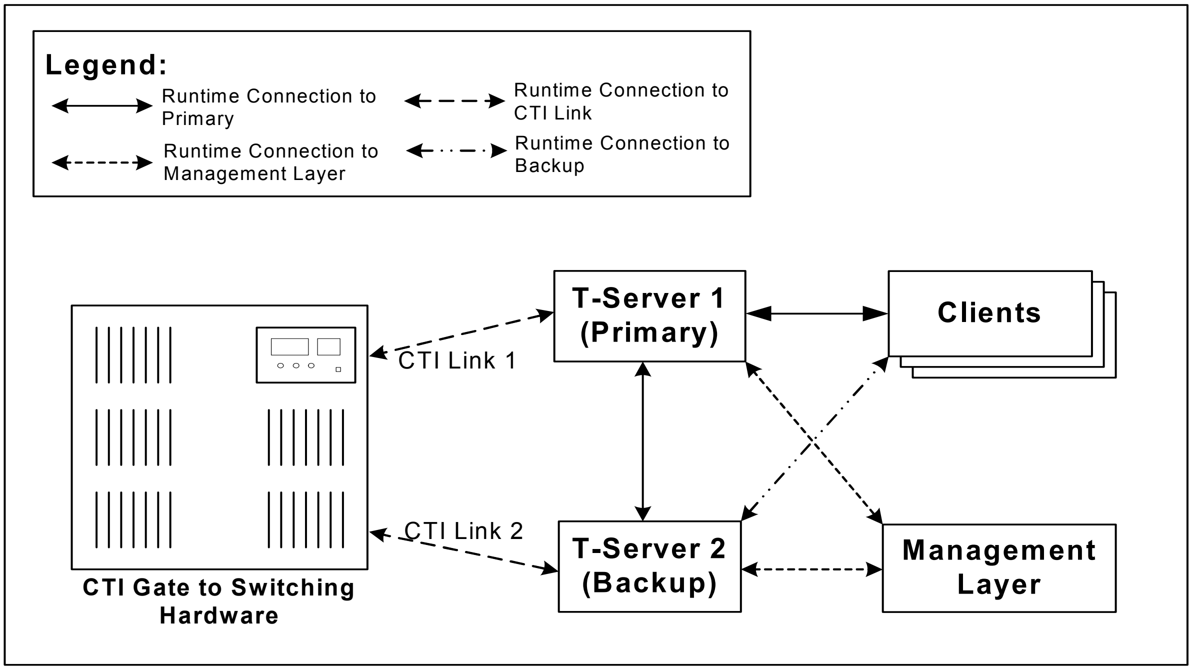
Legend:
Runtime Connection to Primary
Runtime Connection to CTI Link
Runtime Connection to Management Layer
Runtime Connection to Backup
T-Server 1 (Primary)
Clients
CTI Link 1
CTI Link 2
T-Server 2 (Backup)
Management Layer
CTI Gate to Switching Hardware

Figure 16: Hot-Standby Redundancy Type for Dual CTI

The primary and backup T-Servers needed for the Hot Standby redundancy type will each connect to one–and only one–of the two TCP/IP links to the switch from which they will receive identical messages. The response is required from only one of the links; a response from more than one link will result in an error from the switch. For this reason, only the primary T-Server responds to switch messages, with one exception. During continuity testing, the T-Server which receives the message also responds to it.

The Dual CTI Links capability of T-Server requires that the ICM Dual CTI functionality be installed and enabled on the switch. To upgrade from a Hot Standby configuration with HA Proxy, the backup and primary T-Servers must both be stopped before starting the new Dual CTI Links versions.

T-Server is backward compatible with non-dual mode environments, including those using HA Proxy, but cannot be used with a non-dual T-Server in

Hot-Standby mode. When used in a non-dual environment, the value for the dual-links configuration option must be set to false.

Note: With Dual CTI Links activated (the value for the dual-links configuration option is set to true), the backup T-Server does not send the continuity tests at startup if the value for the continuity-test-interval configuration option for the primary T-Server is set to a value of 0 (zero).

Supported Agent Work Mode

Table 18: Supported Agent Work Mode

Agent Work Mode	T-Server[a]
AgentWalkAway	Y
AgentAfterCallWork	Y
AgentReturnBack	Y
AgentWorkModeUnknown	Y

[a] The level of T-Server support for each agent work mode depends on the capabilities of the switch.

Use of the Extensions Attribute

Table 19 indicates how T-Server for the Nortel Communication Server 2000/2100 switch supports the use of the Extensions attribute.

Table 19: Use of the Extensions Attribute

Request/Event	Attribute Extensions		
	Key	Value Type	Value Description
EventAddressInfo with AddressInfoDNStatus[a] QueueStatus	status	integer	0 (idle) or the DN state of a party in the call
	AgentStatus	integer	<0—Unknown 0—LoggedOut 1—LoggedIn 2—Ready 3—Not Ready 4—AfterCallWork
	queue-n	string	An ACD Queue where the agent is logged in (where N is the number of the ACD Queue that can be 1, 2, and so on.)
	conn-n	string	Text representation of the ConnID for a call (if applicable), where N is the number of the call that can be 1, 2, and so on.
	ct-%d	integer	CallType (taken from TCallType) of the call reported by conn-%d (%d is an index).
	mt-%d	integer	CallType taken from TMediaType of the media type information.
	mwl	integer	Used to indicate if message waiting is on or off.
TAgentNotReady	ReasonCode	string	Used to send the ReasonCode to the switch.

Table 19: Use of the Extensions Attribute (Continued)

Request/Event	Attribute Extensions		
	Key	Value Type	Value Description
TInitiateConference or TInitiateTransfer	calledAbtNo	string	Used to pass the `CalledAbtNo` number to the switch. If specified, the number appears on the display of the consulted party's phoneset.
EventAgentNotReady	ReasonCode	string	Used to send the ReasonCode from the switch.
EventMessageWaitingOff or EventMessageWaitingOn	MessageWaiting Type	integer	Specifies the type of Message Waiting: `0`—Message Waiting `1`—Executive Message Waiting
EventQueued	@combined	string	This attribute is included if the ACD Queue DN is also a Routing Point under the control of T-Server
EventRegistered[b] with AddressInfoDNStatus QueueStatus	ct-%d	integer	`CallType` (taken from `TCallType`) of the call reported by `conn-%d` (`%d` is an index)
	mt-%d	integer	`CallType` taken from `TMediaType` of the media type information
	mwl	integer	Used to indicate if message waiting is on or off.
EventTreatmentEnd	ROUTE	integer	Specifies a channel for the treatment.
Call Related Event	FirstFwdNumber	string	Reported by switch—identifies the number from which the first forward was made, in a multiple call forwarding scenario
Call Related Event	LastFwdNumber	string	Reported by switch—identifies the number from which the last forward call was made, in a multiple call forwarding scenario

Table 19: Use of the Extensions Attribute (Continued)

Request/Event	Attribute Extensions		
	Key	Value Type	Value Description
Call Related Event	CalledPartyAddress	string	Included in the event when a call processing message from the switch contains the cpadigs parameter set
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.

a. If the agent's logon failed with ErrorCode:749, and the agent is already logged in, the AgentID is not present in subsequent EventAddressInfo because it was not received from the switch. The switch message with the error does not provide the AgentId.

b. EventRegistered for a position does not contain the queue-1 in the Extensions attribute until a call has been placed to this position. Afterwards, any registration of the DN contains the associated queue-1.

Even though an agent is logged in and is in a Ready state before T-Server startup, EventRegistered reports the agents status as 0 (Unknown) until it receives new information about the agent's activity (Login, Ready).

Error Messages

Table 20 presents the complete set of error messages T-Server distributes in the EventError.

Table 20: Error Messages: T-Server for the Nortel Communication Server 2000/2100 Switch

Code	Symbolic Name	Description
40	TERR_NOMORE_LICENSE	No more licenses are available.
41	TERR_NOT_REGISTERED	Client has not registered for the DN.
42	TERR_RESOURCE_SEIZED	Resource is already seized.
43	TERR_IN_SAME_STATE	Object is already in requested state.
50	TERR_UNKNOWN_ERROR	Unrecognized error

Table 20: Error Messages: T-Server for the Nortel Communication Server 2000/2100 Switch (Continued)

Code	Symbolic Name	Description
51	TERR_UNSUP_OPER	Unsupported operation
52	TERR_INTERNAL	Internal error
53	TERR_INVALID_ATTR	Invalid attribute
54	TERR_NO_SWITCH	The switch is not connected
55	TERR_PROTO_VERS	Incorrect protocol version.
56	TERR_INV_CONNID	Invalid ConnectionID.
57	TERR_TIMEOUT	Timeout expired.
58	TERR_OUT_OF_SERVICE	The link is out of service
59	TERR_NOT_CONFIGURED	DN is not configured in the Configuration Database.
100	TERR_UNKNOWN	Unknown cause
174	TERR_UNSUCC_ANSWER	Unsuccessful answer request
175	TERR_UNSUCC_RELEASE	Unsuccessful release request
496	TERR_INV_CALL_STATE	Call in invalid state
545	TERR_INV_ELEM_VAL	Invalid value within a message element
700	TERR_INV_LOGIN_REQ	Invalid login request
701	TERR_INV_LOGOUT_REQ	Invalid logout request
702	TERR_INV_READY_REQ	Invalid ready request
703	TERR_INV_NOT_RDY_REQ	Invalid not ready request
704	TERR_INV_MAKE_CALL	Invalid make call request
705	TERR_INV_ROUTE_REQ	Invalid route call request
706	TERR_INV_MUTE_TRSFR	Invalid mute transfer request
707	TERR_INV_INIT_CONF	Invalid initiate conference request
708	TERR_INV_INIT_TRSFR	Invalid initiate transfer request
709	TERR_INV_CMPL_CONF	Invalid complete conference request

Table 20: Error Messages: T-Server for the Nortel Communication Server 2000/2100 Switch (Continued)

Code	Symbolic Name	Description
710	TERR_INV_CMPL_TRSFR	Invalid complete transfer request
711	TERR_INV_RETR_REQ	Invalid retrieve original request
712	TERR_INV_CNTL_DN	Invalid control DN
713	TERR_CANT_CONVERT	Cannot convert DN to Position ID
714	TERR_INV_CALL_ID	Invalid call ID
715	TERR_DMS_NOT_ALLOWED	Operation not allowed
716	TERR_DMS_NOT_IDLE	Not idle
717	TERR_DMS_NOT_LOGGED_IN	Agent not logged in
718	TERR_DMS_ORIG_TIME_OUT	Origination timed out
719	TERR_DMS_MAKECALL_RCRS	MakeCall resources unavailable
720	TERR_DMS_MISS_ORIG_ADDR	Missing origination address
721	TERR_DMS_MISS_DEST_ADDR	Missing destination address
722	TERR_DMS_MISS_CALL_TYPE	Missing MakeCall type
723	TERR_DMS_INV_ORIG_ADDR	Invalid origination address
724	TERR_DMS_INV_DEST_ADDR	Invalid destination address
725	TERR_DMS_INV_CALL_TYPE	Invalid MakeCall type
726	TERR_DMS_INV_AUTHCODE	Invalid AuthCode
727	TERR_DMS_INV_ACCTCODE	Invalid AcctCode
728	TERR_DMS_AUTH_OP_NSUBSCR	AuthCode has been sent when the optional parameter has not been subscribed to
729	TERR_DMS_ACCT_OP_NSUBSCR	AcctCode has been sent when the optional parameter has not been subscribed to
730	TERR_DMS_OPER_ABORTED	MakeCall aborted
731	TERR_DMS_MISMATCH_STATE	MakeCall mismatch state
732	TERR_DMS_UNEXP_ACCT	Unexpected AcctCode
733	TERR_DMS_ILL_OPERATION	Illegal operation

Table 20: Error Messages: T-Server for the Nortel Communication Server 2000/2100 Switch (Continued)

Code	Symbolic Name	Description
734	TERR_DMS_INV_ASSOC_DN	Invalid Associated DN
735	TERR_DMS_ASSOC_OTH_SESS	Associated DN already associated with another session
736	TERR_DMS_MAX_NO_OF_LINES	Maximum number of DNs allowed to be DN-associated to that host application has been reached
737	TERR_DMS_NO_RESOURCES	No software resources available to store information for the non-ACD call
738	TERR_DMS_MISSING_ASSOC_DN	Missing the Associated DN parameter
740	TERR_DMS_ALREADY_IN_SET	Associated DN already in set
741	TERR_DMS_NOT_IN_SET	Associated DN not in set
742	TERR_DMS_INV_DN	Invalid DN
743	TERR_DMS_UNKNOWN_DN	Unknown DN
744	TERR_DMS_INV_FILTER	Query DN request has invalid parameter
745	TERR_DMS_MIS_PARAM	Missing parameter
746	TERR_DMS_INV_PARAM	Invalid parameter
747	TERR_DMS_INV_CONTENT	Invalid parameter content
748	TERR_DMS_INV_LINE_CONF	Invalid line configuration
749	TERR_DMS_ALRDY_LOGIN	Agent already logged in
750	TERR_DMS_LOGID_IN_USE	Login ID in use elsewhere
751	TERR_DMS_POS_ALRD_LOGIN	Position already logged in
752	TERR_DMS_INV_SET_STATE	Invalid set state
753	TERR_DMS_INV_PASSWD	Password mismatch
754	TERR_DMS_RSRC_UNAVAIL	Resource unavailable
755	TERR_DMS_AGNT_NOT_LOGIN	Agent not logged in
756	TERR_DMS_AGNT_LOG_PEND	Agent logout pending
757	TERR_DMS_INV_POS_STATE	Invalid agent position state

Table 20: Error Messages: T-Server for the Nortel Communication Server 2000/2100 Switch (Continued)

Code	Symbolic Name	Description
758	TERR_DMS_AGNT_READY	Agent presently ready
759	TERR_DMS_AGNT_NOT_RDY	Agent presently not ready
760	TERR_DMS_SUPERV_OVERRIDE	Supervisor override
761	TERR_DMS_INV_RECON_CALL	The call cannot be reconnected because it is not in an appropriate state for reconnection (no party on hold)
762	TERR_DMS_INV_DEL_FROM_CONF	Invalid TDeleteFromConference request
763	TERR_DMS_AGENT_POSITION_BUSY	Agent position is busy
Network Attended Transfer/Conference Error Messages		
1901	TERR_NATC_UNEXP_CONSULT	Unexpected request TNetworkConsult.
1902	TERR_NATC_UNEXP_ALTERNATE	Unexpected request TNetworkAlternate.
1903	TERR_NATC_UNEXP_RECONNECT	Unexpected request TNetworkReconnect.
1904	TERR_NATC_UNEXP_TRANSFER	Unexpected request TNetworkTransfer.
1905	TERR_NATC_UNEXP_MERGE	Unexpected request for TNetworkMerge.
1906	TERR_NATC_UNEXP_SST	Unexpected request TNetworkSingleStepTransfer.
1907	TERR_NATC_UNEXP_NPS	Unexpected request TNetworkPrivateService.
1908	TERR_NATC_UNEXP_MSG	Unexpected message.

Genesys

Chapter

8 Common Configuration Options

Unless otherwise noted, the common configuration options that this chapter describes are common to all Genesys server applications and applicable to any Framework server component. This chapter includes the following sections:

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 184.

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 184). 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 184). 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 184.

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 188.

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.

Genesys

Chapter

9 T-Server Common Configuration Options

This chapter describes the configuration options that are generally common to all T-Server types, with some exceptions noted. It contains the following sections:

T-Server also supports common log options described in Chapter 8, "Common Configuration Options," on page 177.

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 224.
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 100

Specifies whether DNs associated with the Switch, Switching Office, or Tenant objects will be considered in the T-Server monitoring scope, enabling T-Server to report calls to or from those DNs as internal.

With a value of tenant, all DNs associated with the switches that are within the Tenant will be in the T-Server monitoring scope. With a value of office, all DNs associated with the switches that are within the Switching Office will be in the T-Server monitoring scope. With a value of switch, all DNs associated with the Switch will be in the T-Server monitoring scope.

With a value of undefined (the default), pre-8.x T-Server behavior applies and the switch partitioning is not turned on.

Note: Setting the option to a value of office or tenant, which requires T-Server to monitor a large set of configuration data, may negatively affect T-Server performance.

log-trace-flags

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

Valid Values (in any combination):

`+/-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 201.)

propagated-call-type

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Switch Partitioning" on page 100

Determines what T-Server reports as the value of the `CallType` attribute in events related to calls that have been synchronized with another site via ISCC, as follows:

- When set to `false`, T-Server reports in events related to calls that have been synchronized with another site via ISCC the same value for the `CallType` attribute as it did in pre-8.0 releases and adds the new `PropagatedCallType` attribute with the value of the `CallType` attribute at the origination site. This provides backward compatibility with existing T-Server clients.
- When set to `true`, T-Server reports in events related to calls that have been synchronized with another site via ISCC the same value for the `CallType` attribute as at the origination site, and adds the new `LocalCallType` attribute with the same value as `CallType` in pre-8.0 releases.

server-id

Default Value: An integer equal to the value `ApplicationDBID` as reported by Configuration Server
Valid Values: Any integer from `0–16383`
Changes Take Effect: Immediately

Specifies the `Server ID` that T-Server uses to generate `Connection IDs` and other unique identifiers. In a multi-site environment, you must assign each T-Server a unique `Server ID`, in order to avoid confusion in reporting applications and T-Server behavior.

Configuration of this option is necessary for Framework environments in which there are two or more instances of the Configuration Database.

Note: If you do not specify a value for this option, T-Server populates it with the `ApplicationDBID` as reported by Configuration Server. Each data object in the Configuration Database is assigned a separate DBID that maintains a unique Server ID for each T-Server configured in the database.

Warning! Genesys does not recommend using multiple instances of the Configuration Database.

user-data-limit

Default Value: `16000`
Valid Values: `0–65535`
Changes Take Effect: Immediately

Specifies the maximum size (in bytes) of user data in a packed format.

Note: When T-Server works in mixed 8.x/7.x/6.x environment, the value of this option must not exceed the default value of `16000` bytes; otherwise, 6.x T-Server clients might fail.

license Section

The License section contains the configuration options that are used to configure T-Server licenses. They set the upper limit of the seat-related DN licenses (`tserver_sdn`) that T-Server tries to check out from a license file. See "License Checkout" on page 206.

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 21 shows how to determine the number of seat-related DN licenses that T-Server attempts to check out. See the examples on page 207.

Table 21: 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 21: 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 224.
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 224.
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 211
- Transfer Connect Service Options, page 215
- ISCC/COF Options, page 216
- Event Propagation Options, page 218
- Number Translation Option, page 219
- GVP Integration Option, page 220

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 224.
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 79 for information about supported transaction types by a specific T-Server. The "Multi-Site Support" chapter also provides detailed descriptions of all transaction types.

Notes: For compatibility with the previous T-Server releases, you can use the `direct` value for this option. This is an alias for `direct-callid`.

An alias, `route-notoken`, has been added to the `route` value.

default-dn

Default Value: No default value
Valid Values: Any DN
Changes Take Effect: For the next request for the remote service

Specifies the DN to which a call is routed when a Destination DN (`AttributeOtherDN`) is not specified in the client's request for routing. If neither this option nor the client's request contains the destination DN, the client receives `EventError`.

Note: This option is used only for requests with route types `route`, `route-uui`, `direct-callid`, `direct-network-callid`, `direct-uui`, `direct-notoken`, `direct-digits`, and `direct-ani`.

direct-digits-key

Default Value: `CDT_Track_Num`
Valid Values: Any valid key name of a key-value pair from the `UserData` attribute
Changes Take Effect: For the next request for the remote service

Specifies the name of a key from the `UserData` attribute that contains a string of digits that are used as matching criteria for remote service requests with the `direct-digits` routing type.

Note: For compatibility with the previous T-Server releases, this configuration option has an alias value of `cdt-udata-key`.

dn-for-unexpected-calls

Default Value: No default value
Valid Values: Any DN
Changes Take Effect: Immediately

Specifies a default DN for unexpected calls arriving on an External Routing Point.

network-request-timeout

Default Value: `20 sec`
Valid Values: See “Timeout Value Format” on page 224.
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 224.
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 224.
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 224.
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 224.
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 224.
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 224.
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 224.
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 224.
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 224.
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 224.
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 224.
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 224.
Changes Take Effect: Immediately

Specifies the time interval for periodic checks for stuck calls. These checks affect both notification and cleanup functionality, and are made by checking the T-Server's own call information with call information available in the switch. For performance reasons, T-Server does not verify whether the `notify-idle-tout` or `cleanup-idle-tout` option has expired before performing this check.

Note: Setting this option to a value of less than a few seconds can affect T-Server performance.

Examples

This section presents examples of option settings in the `call-cleanup` section.

Example 1

```
cleanup-idle-tout = 0
notify-idle-tout = 0
periodic-check-tout = 10
```

With these settings, T-Server will not perform any checks for stuck calls.

Example 2

```
cleanup-idle-tout = 0
notify-idle-tout = 5 min
periodic-check-tout = 10 min
```

With these settings, T-Server performs checks every 10 minutes and sends notifications about all calls that have been idle for at least 5 minutes.

Example 3

```
cleanup-idle-tout = 20 min
notify-idle-tout = 5 min
periodic-check-tout = 10 min
```

With these settings, T-Server performs checks every 10 minutes, sends notifications about all calls that have been idle for at least 5 minutes, and attempts to clean up all calls that have been idle for more than 20 minutes.

Translation Rules Section

The section name is specified by the `inbound-translator-<n>` option. It contains options that define translation rules for inbound numbers.

You can choose any name for this section, provided that it matches the value of the section. Every option in this section corresponds to a rule and must conform to the format described below. You can configure as many rules as necessary to accommodate your business needs.

rule-<*n*>

Default Value: No default value
Valid Value: Any valid string in the following format:
`in-pattern=<input pattern value>;out-pattern=<output pattern value>`
Changes Take Effect: Immediately

Defines a rule to be applied to an inbound number. The two parts of the option value describe the input and output patterns in the rule. When configuring the pattern values, follow the syntax defined in “Using ABNF for Rules” on page 88. See “Configuring Number Translation” on page 95 for examples of these rules as well as detailed instructions for creating rules for your installation. For example, a value for this configuration option might look like this:

```
rule-01 = in-pattern=0111#CABBB*ccD;out-pattern=ABD
```

security Section

The `security` section contains the configuration options that are used to configure secure data exchange between T-Servers and other Genesys components. Refer to the *Genesys 8.x Security Deployment Guide* for complete information on the security configuration.

Timeout Value Format

This section of the document describes the values to use for those T-Server common options that set various timeouts. The current format allows you to use fractional values and various time units for timeout settings.

For timeout-related options, you can specify any value that represents a time interval, provided that it is specified in one of the following formats:

`[[[`*hours*`:]`*minutes*`:]`*seconds*`][`*milliseconds*`]`

or

`[`*hours* `hr][`*minutes* `min][`*seconds* `sec][`*milliseconds* `msec]`

Where a time unit name in italic (such as *hours*) is to be replaced by an integer value for this time unit.

Integer values with no measuring units are still supported, for compatibility with previous releases of T-Server. When you do not specify any measuring units, the units of the default value apply. For example, if the default value equals `60 sec`, specifying the value of `30` sets the option to 30 seconds.

Example 1

The following settings result in a value of 1 second, 250 milliseconds:

```
sync-reconnect-tout = 1.25

sync-reconnect-tout = 1 sec 250 msec
```

Example 2

The following settings result in a value of 1 minute, 30 seconds:

```
timeout = 1:30

timeout = 1 min 30 sec
```

Changes from Release 8.0 to 8.1

Table 22 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 22: 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 200.

Chapter

10 T-Server-Specific Configuration Options

This chapter describes the configuration options that are unique to the T-Server for Nortel Communication Server 2000/2100. It includes the following sections:

The options common to all T-Servers are described in Chapter 8, “Common Configuration Options,” on page 177 and Chapter 9, “T-Server Common Configuration Options,” on page 199.

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

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

The following table lists the options you must configure for basic T-Server operation. All other options in this chapter are configured to enable T-Server to support various features.

To establish a link connection, simply configure the link options (TCP/IP or X.25) that are applicable to the connection protocol used in your environment.

Table 23: Mandatory Options

Option Name	Default Value	Details
TServer Section		
application-id	No default value	Identifies the specific customer host application that initiates the logon request. See description on page 233.
business-group-id	No default value	Specifies the customer of the host application and must match the entry in the Nortel Communication Server 2000/2100 table SCAIGRP. See description on page 233.
link-*n*-name	No default value	Specifies the section name containing the configuration options assigned to that link, where *n* is a consecutive number for a CTI link. If an HA Proxy is used, it is not mandatory to configure this option. See description on page 237.
network-node-id	No default value	Enables T-Server to add the option value (which must be equal to the switch's network node ID) to the `CallID` parameter. See description on page 241.

Table 23: Mandatory Options (Continued)

Option Name	Default Value	Details
password	No default value	Lists the switch parameters for application logon operations. This option must match the password field in the entry in the Nortel Communication Server 2000/2100 table S table SCAIGRP. See description on page 243.
service-id	No default value	Identifies the application context to be configured for the session. See description on page 244.
service-version	No default value	Specifies the application-level signaling version the host application utilizes. See description on page 244.
tcp-linkset-name	No default value	Specifies the string sent to the switch to establish a TCP connection for the Nortel Communication Server 2000/2100 SCAI10 or later when using a TCP link to the Nortel Communication Server 2000/2100 switch. See description on page 247.
CTI-Link Section		
comport	`a` or `0`	Specifies the serial port number that T-Server uses for its X.25 connection. This value is the SVC address of the X.25 physical port connected to the switch. Mandatory for X.25 links. See description on page 250.

Table 23: Mandatory Options (Continued)

Option Name	Default Value	Details
dteclass	No default value	Specific to Digital UNIX® machines. This option specifies the DTE class to be used in the link. Mandatory for X.25 links. See description on page 251.
hostname	No default value	Specifies the host of the link according to the switch configuration. Mandatory for a TCP link. See description on page 253.
mode	No default value	Specifies the X.25 mode: either Switched Virtual Circuit (SVC) or Permanent Virtual Circuit (PVC). Mandatory for X.25 links. See description on page 251.
port	No default value	Specifies the TCP/IP port of the link according to the switch configuration. Mandatory for a TCP link. See description on page 253.
protocol	No default value	Specifies the connection protocol T-Server uses in communicating with the switch. Mandatory for X.25 and TCP links. See description on page 251.
restart-attempts	No maximum limits	For Windows users only. Specifies how many repeatedly unsuccessful attempts T-Server makes to connect to the link before considering the connection lost. Mandatory for X.25 links. See description on page 251.

Table 23: Mandatory Options (Continued)

Option Name	Default Value	Details
restart-delay	2	Specifies the delay (in seconds) between attempts to set connections. Mandatory for X.25 links. See description on page 251.
template	No default value	Specifies the name of the Digital Equipment Corporation (DEC) template for X.25. Mandatory for X.25 links. See description on page 252.
userdata	No default value	Specifies the data for the X.25 call request. Mandatory for X.25 links. See description on page 252.
x25address	No default value	Identifies the location of the X.25 address on the local (host) computer. Mandatory for X.25 links. See description on page 252.
x25device	No default value	Indicates the name of the X.25 device being used to access host X.25 services. Only relevant for HP-UX and AIX OS platforms; installation dependent. Mandatory for X.25 links. See description on page 252.
x25localaddr	No default value	Identifies the location of the X.25 address on the host where T-Server is installed. Mandatory for X.25 links. See description on page 252.

TServer Section

This section describes configuration options that are used to support unique T-Server features for Nortel Communication Server 2000/2100.

You must call this section `TServer`.

agent-no-answer-action

Default Value: none
Valid Values: none, notready, walkaway, logout
Changes Take Effect: Immediately

Instructs T-Server to set the agent state to the agent-no-answer-action value when an agent receives a call, and that call is not answered. The agent state is set to this state following the agent-no-answer-timeout period.

agent-no-answer-overflow

Default Value: none
Valid Values:

none	The call will remain ringing on the agent phone.
recall	The call will be redirected back to the Routing Point or the ACD Queue that delivered the call to the agent. **Note:** Recall does not work with Nortel 2000/21000.
Any destination digits	A valid destination DN must be provided (an ACD Queue or a Routing Point on the local switch is recommended).

Changes Take Effect: Immediately

Indicates where to send a call when an agent receives a call, and that call is not answered.

agent-no-answer-timeout

Default Value: 0
Valid Values: 0 to 600
Changes Take Effect: Immediately

Specifies the number of seconds that an agent's phone will ring before the agent state is changed to the agent-no-answer-action value. This configuration option is disabled when its value is set to 0 (zero).

appl-logon-already-ok

Default Value: true
Valid Values: true, false
Changes Take Effect: Immediately

Determines T-Server's response to the appl_logon return error code value of 0x0*. If the value of this configuration option is set to true, T-Server treats the error as a successful logon. If the value of this configuration option is set to false, T-Server treats the return value as a failure and then attempts to retry the logon specified by the logon-interval configuration option. If both configuration options, appl-logon-already-ok and logon-hard-reset, return an error code value of 0x0*, then the logon-hard-reset configuration option takes precedence.

Note: Set the value of this configuration option to `true`, if T-Server is running in `hot standby` mode with HA Proxy.

application-id

Default Value: Mandatory field. No default value.
Valid Value: Depends on switch configuration
Changes Take Effect: After link is restarted

Identifies the specific customer host application that initiates the logon request. A value for this configuration option must be specified.

business-group-id

Default Value: Mandatory field. No default value.
Valid Value: Depends on switch configuration
Changes Take Effect: After link is restarted

Specifies the customer of the host application and must match the entry in the Nortel Communication Server 2000/2100 table SCAIGRP. A value for this configuration option must be specified.

call-held-enabled

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

If the value of this configuration option is set to `true`, the `DV_CALL_HELD` event from the switch is used to trigger an `EventHeld` event from T-Server for the held party. If the value of this configuration option is set to `false`, the return result from the switch for `THoldCall` is used to trigger an `EventHeld` event.

Note: The Service Version must be greater than or equal to SCAI15 for this option to have effect.

call-progress

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: “Support for DV_DN_QUERY Messages” on page 161

If the value of this configuration option is set to `true`, T-Server uses the `CALL_PROGRESS` switch message to process calls more accurately:

- By allowing the distribution of an `EventNetworkReached` message for external DNs in outbound calls without polling the switch with `DV_DN_QUERY`.

- By allowing the distribution of a EventDestinationBusy message when a call is made to a busy DN.

Notes:

- For T-Servers in high availability (hot standby) configuration, Genesys recommends that you use Link Version SCAI14 or later with call-progress messages enabled.
- Call progress functionality requires Service Version SCAI14 or later. If the values of both call-progress and use-query-dn configuration options are set to true, call progress functionality takes precedence in SCAI14 or later.

continuity-test-fail-number

Default Value: 3
Valid Values: 0-10
Changes Take Effect: Immediately

Specifies the number of times T-Server sends a continuity test without receiving a successful response from the switch, after which it restarts the link if the value of the dual-links configuration option is set to true, otherwise it logs a message and restarts the test cycle. No continuity test is sent if the value is less than 1.

continuity-test-interval

Default Value: 120
Valid Values: 0-60000 (values 0 to 9 disable this option)
Changes Take Effect: Immediately

Specifies the interval (in seconds) at which T-Server sends a continuity test to the switch. No continuity test is sent if the interval is less than 10 seconds.

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.

dial-plan-prefix

Default Value: `null`
Valid Values: Any comma-delimited list of dialing prefixes (`1, 9, 19`)
Changes Take Effect: Immediately
Related Feature: "Support for T-Server Dial Plans" on page 163

Specifies a comma-delimited list of dialing plan prefixes that are used to compare dialed digits with internal addresses (DN configured for this T-Server). If a prefix is commonly used before a phone number, the same prefix should be specified as the value for this configuration option. See "Support for T-Server Dial Plans" on page 163 for further information.

In general, the `dial-plan-prefix` configuration option is used for comparing dialed digits with internal numbers in the following scenarios:

- When determining if the call is going outside of T-Server environment so that it can be recognize if it returns (a call topology loop).
- When determining the `CallType` attribute at dial time (when `EventDialing` is distributed).

Note: This configuration option was added for implementations that use simple dialing plans and call scenarios involving call topology loops. Call topology loops occur when a call is routed, transferred, consulted, or conferenced from an address within the T-Server-controlled environment to an external address and then later routed, transferred, consulted, or conferenced back into the environment.

dms-upgrade-time

Default Value: `0` (turns off the option)
Valid Values: `0-3600`
Changes Take Effect: Immediately

Allows upgrading of the switch while T-Server is operating. If each link in its turn has disconnected and reconnected during the period specified in this option (in seconds), T-Server assumes that the switch is being upgraded. As a result, T-Server reregisters all DNs with the switch.

Note: Applicable for multi-link connections between T-Server and the switch

dual-links

Default Value: `false`
Valid Values: `true-false`
Changes Take Effect: After T-Server is restarted

If the value for this configuration option is set to `true`, T-Server runs in Dual CTI Links mode, meaning that each T-Server (primary and backup) directly connects to the switch with its own TCP/IP connection (without the use of HA

Proxy). The switch mirrors all messages across both links, and provides nearly full T-Server and TCP connection redundancy. On the switch side, this mode must be supported and configured correctly. On the T-Server side, only one `link-n-name` configuration option is required for each respective T-Server (primary and backup).

Warning! If more than one `link-n-name` configuration option is set, there is no guarantee which one is used by T-Server.

Note: The `na010-network-node-id` configuration option must be set correctly for both primary and backup T-Servers when the `dual-links` option is set to `true` in order for both T-Servers to successfully logon to the link.

error-on-agent-state

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Distributes an `EventError` message. If the value of this configuration option is set to `true,` and when an agent is in the state of `Login, Logout`, `Ready` or `NotReady` in both the switch and in T-Server, T-Server will distribute an `EventError` message.

ha-heartbeat-failures

Default Value: `0` (no heartbeats)
Valid Values: `0-60000`
Changes Take Effect: Immediately

Shuts down the link after a specified number of heartbeat failures. If the value for this configuration option is set to `0` (zero), an infinite number of heartbeat failures is allowed, so the link does not shut down. T-Server and HA Proxy heartbeats are performed independently of each other. Both T-Server and HA Proxy must support high-availability (HA) heartbeats in order for both to function correctly. If only one of the two supports heartbeats, turning heartbeats on may result in unexpected behavior, such as links shutting down and restarting unexpectedly.

ha-heartbeat-period

Default Value: `0` (no heartbeats)
Valid Values: `0-60000`
Changes Take Effect: Immediately

Sends a heartbeat message to HA Proxy at intervals defined by this configuration option. If the value of this configuration option is set to `0` (zero), the heartbeat message is turned off.

ha-heartbeat-timeout

Default Value: `0` (no heartbeats)
Valid Values: `0-60000`
Changes Take Effect: Immediately

Waits for a heartbeat response from HA Proxy for this amount of time before giving up and starting the timer for the next heartbeat. If the value of this configuration option is set to `0` (zero), the heartbeat message is turned off.

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 be a value of `0` (`zero`).

link-restart-interval

Default Value: 10000
Valid Values: 0 to 600000
Changes Take Effect: Immediately

Specifies the time period (in milliseconds) that T-Server waits before stopping and restarting a link after receiving a `LINK_DOWN_PACKET` message. Although T-Server will usually receive a `LINK_UP_PACKET` message automatically after receiving a `LINK_DOWN_PACKET` message, this configuration option ensures that the link is restarted if the `LINK_UP_PACKET` message is not received. If the value of this configuration option is set to `0` (zero), the option is turned off.

link-stop-delay

Default Value: `0`
Valid Values: `0-3600000`
Changes Take Effect: Immediately

Specifies the time period (in milliseconds) that T-Server waits after receiving an `appl_logoff` return result from the switch before stopping the X.25 link. This can be used to help ensure that T-Server has time to send an X.25 `CLEAR CONFIRM` message in response to an X.25 `CLEAR REQUEST` message from the switch. If the value of this configuration option is set to `0` (zero), there is no delay. This configuration option is applicable to X.25 connections only, and is ignored for TCP connections.

Note: If used in `warm standby` mode, the `warm-standby-link-delay` configuration option must be set to a time period at least as great as the number of X.25 links multiplied by the `link-stop-delay` configuration option. The `warm-standby-link-delay` configuration option is specified in seconds.

logon-hard-reset

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When T-Server receives an `APPL-LOGON` return error with the error message `link-already-in-use`, it sends an `APPL-LOGON` message with a hard reset equal to `1` to clear the existing session data and to establish a new session in the switch.

Note: The configuration option `logon-hard-reset` requires SCAI version 17 or higher.

The `logon-hard-reset` configuration option turns on the hard reset functionality. If the value of this configuration option is set to `true,` T-Server uses the hard reset functionality under the above circumstances. If the value of this configuration option is set to `false`, the return error `link-already-in-use` is processed normally (T-Server assumes the link is indeed already logged in and proceeds to associate the DNs). The default value for this configuration option is `false`.

Note: Hard reset functionality must be turned off (the value of the `logon-hard-reset` configuration option is set to `false),` if it is not available on the switch. If this rule is not followed, T-Server may end up in an endless loop of `APPL-LOGON` return error returning the error message `link-already-in-use`, particularly when the link is slow in responding.

logon-interval

Default Value: `2000`
Valid Values: `1500-60000`
Changes Take Effect: Immediately

Sets the timeout interval (in milliseconds) before the next attempt to logon to the switch.

map-cpa-to-dnis

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies if T-Server places the Called Party Address information from the switch into the `DNIS` attribute in addition to placing the same information, by default, into the `Extensions` attribute.

Note: If the values of both `orig-inbound-to-dnis` and `map-cpa-to-dnis` configuration options are set to `true`, `map-cpa-to-dnis` takes precedence.

max-register-retries

Default Value: `0`
Valid Values: Any positive integer
Changes Take Effect: Immediately

Defines the number of times that T-Server will retry registration of a DN on the switch before giving up.

mute-transfer-delay

Default Value: `0`
Valid Values: `0-60000`
Changes Take Effect: Immediately

Specifies the interval (in milliseconds) that T-Server waits before sending commands to the switch when the `TMuteTransfer` message is received from a client. When this configuration option is enabled, T-Server delays completing the transfer for the interval specified after receiving the `RETURN_RESULT` message from the switch. If the `DV-CALL-PROGRESS farEndRinging` message is received from the switch before this delay expires, T-Server will complete the transfer without further delay.

mute-xfer-retries

Default Value: `5`
Valid Values: `0-100`
Changes Take Effect: Immediately

This configuration option (along with `mute-xfer-retry-delay`) is used to overcome problems in completing transfers to external parties due to timing. Since the switch does not notify when a transfer can be completed, the complete transfer fails occasionally. This configuration option causes T-Server to retry the complete transfer up to the given number of times. If the value of this configuration option is set to `0 (zero),` T-Server does not retry the transfer.

mute-xfer-retry-delay

Default Value: `500`
Valid Values: `0-10000`
Changes Take Effect: Immediately

The delay (in milliseconds) between transfer completion retry attempts. See `mute-xfer-retries` for more details.

na010-network-node-id

Default Value: `None` (turns off the option)
Valid Value: `0` or any positive integer
Changes Take Effect: After link is restarted

Enables T-Server to add the configuration option value (which must be equal to the switch's network node ID) to the `Call ID` parameter. The unique call identification, consisting of `NETWORK_NODE_ID` and `LOCAL_CALL_ID` allows T-Server to handle calls distributed from other switches within the same network group. T-Server sends `NETWORK_NODE_ID` and `LOCAL_CALL_ID` to its clients as `NetworkNodeID` and `CallID` event attributes respectively. This configuration option must match the entry in the Nortel Communication Server 2000/2100 table C7NETWRK.

The configuration option `na010-network-node-id` is included in all but the first logon to a session.

Notes:
- This configuration option is applicable for only NA010 (SCAI version 12) or later and must be enabled on the switch.
- This configuration option is mandatory for logon if the value of the configuration option `dual-cti` is set to `true`.

Warning! The person configuring T-Server must ensure that the `na010-network-node-id` configuration option is specified *only* if necessary for the environment. If the ID is specified when it is not required, logons for each subsequent link will fail. If the ID is not specified and it is required, subsequent logons will fail.

ncr-enabled

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

If the value of this configuration option is set to true, the NONCONTROLLER-RELEASED event from the switch is used in TReconnectCall to trigger the EventReleased event for the released party. If the value of this configuration option is set to false, the return result from the switch for TRequestReconnectCall will trigger the EventReleased event for the released party.

Note: The Service Version must be greater than or equal to SCAI14 for this option to have effect.

network-node-id

Default Value: Mandatory field. No default value.
Valid Value: Depends on switch configuration
Changes Take Effect: After link is restarted

Specifies the switch that the host uses for communication. This configuration option must match the switch entry in the Nortel Communication Server 2000/2100 table SCAIGRP. A value for this configuration option must be specified.

new-call-for-unknown-dest

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

If the value of this configuration option is set to true, and T-Server detects a call topology loop where the call has left the T-Server environment and then returned back to a new party on a new call, a new connectionID and separate user-data is created. If the value of this configuration option is set to false, the same call will instead be used with the new destination party.

Note: If the value of the set-call-type-with-dialing configuration option is set to false when the value of the new-call-for-unknown-dest configuration option is true, T-Server is not always able to detect that the call has left T-Server and therefore does not always create a new call when the call returns.

To prevent the creation of extra calls for internal numbers, set the value of this configuration option to `false` for environments using complex dialing plans that cannot be accommodated with the `dial-plan-prefix` configuration option.

Note: This configuration option is for implementations that use simple dialing plans and call scenarios involving call topology loops. Call topology loops occur when a call is routed, transferred, consulted, or conferenced from an address within the T-Server—controlled environment to an external address and then later routed, transferred, consulted, or conferenced back into the environment.

noncontroller-released-digits

Valid Values: `0-20`, all
Default Value: `4`
Changes Take Effect: immediately

Specifies the right most number of digits from the noncontroller-released message that T-Server uses to determine if an external party has released. In some environments, especially when calls go from one switch to another, the digits provided in the call-progress messages get changed by the time the noncontroller-released message is sent when the party releases. However, since there are usually some right-most digits that remain the same, this configuration option can be used to limit the comparisons to these common digits.

Note: T-Server never requires more digits than are passed in the `call-progress` message. So, if only 7 digits are included in the `call-progress` message and 10 are received in the noncontroller-released message, T-Server will match the two anyway, even if the value of this configuration option is set to greater than `7`.

orig-inbound-to-dnis

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether T-Server populates the `OrigInboundDN` from the switch in the `DNIS` attribute of all events.

Note: If the values of both configuration options `orig-inbound-to-dnis` and `map-cpa-to-dnis` are set to `true`, the configuration option `map-cpa-to-dnis` takes precedence.

password

Default Value: Mandatory field. No default value.
Valid Value: Depends on switch configuration
Changes Take Effect: After link is restarted

Lists the switch parameters for application logon operations. The value of this configuration option must match the `password` field in the Nortel Communication Server 2000/2100 table SCAIGRP. A value for this configuration option must be specified.

relate-primary-supp-qrp

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

Specifies how the Supplementary Queue DNs and Routing Points are treated. If the value of this option is set to `true`, the Supplementary Queue DNs and Routing Points are taken in and out of service based on their corresponding primary DN of the ACD Queues and Routing Points. Each Supplementary Queue DN and Routing Point must specify the primary DN of the ACD Queue or Routing Point to which it belongs by using the `primary-qrp` option on the `Annex` tab of the DN `Properties` dialog box in the `TServer` section. If the value of this option is set to false, all Supplementary Queue DNs and Routing Points are in service regardless of whether their primary DNs of the ACD Queues and Routing Points are in service.

request-timeout

Default Value: `10000`
Valid Values: `1000` to `60000`
Changes Take Effect: Immediately
Specifies the time (in milliseconds) that T-Server waits for a response from the switch after sending the request. After this time, T-Server clears the request and distributes an `EventError TERR_TIMEOUT` message to the requesting client.

send-answer-after-make

Default Value: `0`
Valid Values: `true`, `0-10000`
Changes Take Effect: Immediately

Specifies the number of milliseconds to wait after sending a `dv-make-call` request before automatically sending the corresponding `dv-answer-call` request. T-Server waits for a `make-call return result` response, a `make-call return error` response, a `make-call return reject` response, or a `client answer request` response. If no valid response is received within the `send-answer-after-make` time period, T-Server automatically sends a `dvanswer-call` request. Using this configuration option removes the requirement for clients to answer the call after making it. Setting the value of this configuration option to `zero` turns off this option.

send-result-on-error

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether or not T-Server sends a `ReturnResult` message to the switch on the received switch message if a call redirect was unsuccessful.

Warning! Do not change the default value unless Genesys Technical Support instructs you to do so.

service-id

Default Value: Mandatory field. No default value
Valid Value: Depends on switch configuration
Changes Take Effect: After link is restarted

Identifies the application context to be configured for the session. A value for this configuration option must be specified.

Note: The value specified for this configuration option must match the index information on the Nortel Communication Server 2000/2100 SCAIPROF table.

service-version

Default Value: Mandatory field. No default value
Valid Value: Depends on switch configuration and version
Changes Take Effect: After link is restarted

Specifies the application-level signaling version (SCAI version) the host application utilizes. A value for this option must be specified.

T-Server interprets the parameter representation differently depending on the format being used. If the parameter begins with `0`, T-Server interprets it as an octal representation. If the parameter begins with `0X`, T-Server interprets it as a hexadecimal representation. If the parameter begins with a number other than `0` or `0X`, T-Server interprets it as a decimal representation.

Warning! All T-Servers connected to the same customer group in the Nortel Communication Server 2000/2100 must specify the same service-version in the Configuration Layer. Otherwise, T-Servers may not be able to route calls.

set-call-type-with-dialing

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Support for T-Server Dial Plans" on page 163

Determines the call type at the time that the call is dialed.When the value of this option is set to `true`, T-Server determines the call type at the time that the call is dialed in order to set the `CallType` attribute in `EventDialing`. If the call type is not determined at the time the call is dialed, the `CallType` attribute is set to `unknown`.

soft-login-support

Default Values: `false`
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

Turns on or off the Emulated Agent States (also known as the Soft Agent) feature. If the value of the option is set to `true`, T-Server processes all agent-related 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.
- Only DNs of type Extension are supported for this option.
- If the password is configured in the Configuration Layer, it must be supplied in `TAgentLogin` (otherwise the request fails).
- Workmode is not used in `TAgentSetReady`, but T-Server supports all `NotReady` substates shown on the Agent-State diagram.

Note: The T-Server Common Part implements the Emulated Agent States feature. When it 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 PBX 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-wrap-up-time

Default Value: `0`
Valid Values: See “Timeout Value Format” on page 224.
Changes Take Effect: After agent logs out and then logs in again

Sets the default wrap-up time for the Emulated Agent States (also known as Soft Agent) feature (you can also specify a wrap-up time for each `AgentID` in the `AgentLogin Properties` dialog box); however, it does not affect the state of the agents logged in on the PBX. If the wrap-up time is set to a non-zero value, T-Server emulates the `AfterCallWork` state after each call that was released when the agent was in the `Ready` state.

Note: You must set the value of the `soft-login-support` configuration option to `true` to use `soft-wrap-up-time`.

syncaddresses

Default Value: -positions, -extensions (that is, no synchronization)
Valid Value: +/-positions +/-extensions
Changes Take Effect: Immediately
Related Feature: “Support for DV_DN_QUERY Messages” on page 161

Enables T-Server to synchronize positions and/or extensions with the switch at the link connection time. `EventLinkConnected` is only sent when the synchronization is complete. For the value of `positions`, T-Server synchronizes:

- the agent state
- the call state
- the queue to which the agent belongs

For the value of `extensions`, T-Server synchronizes:

- the call state

In addition, T-Server removes all parties that no longer on calls, and all calls which no longer have any parties.

SCAI 11 is required for the `sync-addresses` option.

The use-query-dn configuration option has no impact on the sync-addresses configuration option.

Notes:
- T-Server does not reconstruct calls that were created during link down. The information provided by the switch is currently not sufficient to allow this. Also, not all information is available from the CTI link, therefore, T-Server may not be able to synchronize all the CTI data.
- Since the query results from the switch do not indicate which agent is logged in to a position, T-Server may not synchronize agent IDs correctly. For example, one agent is logged in, the link goes down, the agent logs out and another logs in, the link comes back up, T-Server queries the switch to find if the agent is still logged in—it does not know that it is a different agent now.

To enable the sync-addresses configuration option, see the Nortel Communication Server 2000/2100 functions, Table 24 and Table 25 for retrieval of switch information.

Table 24: ACD Table SCAISSRV Capabilities and Functions

Service Capabilities	Category	Function	Message
Resource Status	RESOURCE	APPSTQRY	dv-Appl-Stat-Qry

Table 25 presents information about the Meridian Digital Centrex (MDC) and Residential Line (RES) of SCAISSRV Capabilities and Functions.

Table 25: MDC and RES Table SCAISSRV Capabilities and Functions

Service Capabilities	Category	Function	Message
DN Query	DNQUERY	DNQUERY	dv-DN-Query

tcp-linkset-name

Default Value: SCAI Version 10 or later when using a TCP link to the switch
Valid Values: Depends on switch configuration
Changes Take Effect: After link is restarted

Specifies the string sent to the switch to establish a TCP connection for the Nortel Communication Server 2000/2100 SCAI10 or later when using a TCP link to the switch. A value for this configuration option must be specified.

This configuration option is configured for protocol version SCAI10+ (NA008+) with a TCP link only. The value specified for this configuration option must match the name of the TCP link set SCAICOMS table on the switch.

use-dial-plan

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Support for T-Server Dial Plans" on page 163

Uses the `dial-plan-prefix` configuration option when comparing dialed digits with internal addresses. If the value for this configuration option is set to `true`, T-Server uses the `dial-plan-prefix` configuration option when comparing dialed digits with internal addresses (DNs configured for this T-Server). That is, T-Server ignores the dial plan prefix when performing the comparison. See "Support for T-Server Dial Plans" on page 163 for further information.

Dial plans are used when comparing DNs in the following two cases:

- When setting the call type at dialing time. See "set-call-type-with-dialing" on page 245.
- When determining if a call is going outside of the T-Server environment.

use-query-dn

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Support for DV_DN_QUERY Messages" on page 161

Specifies whether T-Server uses a call status change on the DN from the switch for outbound calls.

Notes:
- The configuration option `use-query-dn` is not applicable to ACD Positions because the switch does not support `DV-DN-QUERY` requests for ACD Positions.
- If both the `call-progress` and `use-query-dn` configuration options are set to `true`, call progress functionality takes precedence in SCAI versions 14 and later. See "call-progress" on page 233.

use-supp-in-queued

Default Value: `false`
Valid Value: `true, false`
Changes Take Effect: Immediately

If the value of this configuration option is set to `true`, T-Server distributes `EventQueued` and `EventDiverted` events on the Supplementary Queue DN if it is the one called. This distribution allows statistics to be collected on calls to

Supplementary Queue DN, as well as the Primary Queue DNs. This configuration option does not affect the events distributed for Routing Points.

Note: The Supplementary Queue DN must be defined in the Configuration Layer in order for this configuration option to take affect. If the Supplementary Queue DN is not defined in the Configuration Layer and the value of `use-supp-in-queued` is set to `true`, T-Server may not properly queue and divert calls made to the Supplementary Queue DN. In some cases, a call may get stuck on the Primary Queue DN. If using this configuration option, ensure that all Supplementary Queue DNs are defined in the Configuration Layer.

warm-standby-link-delay

Default Value: `2`
Valid Values: `0-600`
Changes Take Effect: Immediately

Specifies the delay (in seconds) before links are started when transferring from the backup to the primary T-Server during a switch over.

Flow Control Options

Flow control options support a new flow control mechanism to limit the number of messages sent to the switch per given time period. This avoids situations where more messages are sent to the switch at a given time than the switch is able to handle. The link bandwidth of the switch is estimated by the number of messages sent and received per second.

flow-control-period

Default Value: `250`
Valid Values: `100-1000`
Changes Take Effect: Immediately

The period, in milliseconds (ms), used for sending flow-controlled messages to the switch. At the period defined by this configuration option (every `flow-control-period` milliseconds) T-Server sends buffered messages to the link. The number of messages sent at each cycle is determined by the `flow-control-rate` configuration option and the number of messages received from the link.

flow-control-warning

Default Value: `500`
Valid Values: `10-10000`
Changes Take Effect: Immediately

T-Server logs one LMS alarm when the number of buffered messages goes above this amount, and logs another LMS alarm when it returns below again.

link-alarm-high

Default Value: `0`
Valid Values: Any unsigned integer
Changes Take Effect: Immediately

Sets the percentage of the `flow-control-rate` configuration option that will trigger the `link-alarm-high` LMS message. This alarm is raised to warn that the link bandwidth is approaching an unsafe level. A value of `0` (zero) turns the alarm off.

Note: The value for this configuration option is measured as a percentage, for example `90`. This configuration option is set when flow control is required.

CTI-Link Section

The section name is specified by the `link-n-name` configuration option when you use X.25 links or a TCP link to the switch with T-Server (without HA Proxy) and is only valid for Nortel Communication Server 2000/2100 versions SCAI10+ (NA008+). One section per link is required. See “link-n-name” on page 237.

- If you are using X.25 links to the switch, configure options as they are described under “X.25 Protocol Options”.
- If you are using a TCP link to the switch, configure options as they are described under “TCP Protocol Options” on page 253.

Warning! Do not update the link configuration while T-Server is running.

X.25 Protocol Options

Note: On the Windows platform, T-Server supports the use of the Eicon X.25 card to establish an X.25 link to the switch. Other X.25 cards may not function with T-Server unless the card is compatible with the Eicon API. Check with your hardware vendor for details on compatibility.

comport

Default Value: `a` or `0`
Valid Values: `a, A, b, B,` or any integer from `0-9`
Changes Take Effect: After T-Server is restarted

Specifies the serial port number that T-Server uses for the X.25 connection. This is the SVC address of the X.25 physical port connected to the switch.

dteclass

Default Value: Mandatory field. No default value.
Valid Value: Any valid DTE class
Changes Take Effect: After T-Server is restarted

Specific to Digital Unix machines.

mode

Default Value: Mandatory field. No default value
Valid Values: `svc, pvc`
Changes Take Effect: After T-Server is restarted

Specifies the SVC (Switched Virtual Circuit) or PVC (Permanent Virtual Connection) X.25 mode.

Note: Switched Virtual Circuit (SVC) X.25 mode is currently the only mode that the T-Server for Nortel Communication Server 2000/2100 supports.

protocol

Default Value: Mandatory field. No default value.
Valid Value: `X25`
Changes Take Effect: Immediately

Specifies the connection protocol T-Server uses in communicating with the switch.

pvc-channel

Default Value: `0` (zero)
Valid Values: Any positive integer
Changes Take Effect: After T-Server is restarted

Specifies the Permanent Virtual Circuit (PVC) channel for the X.25 connection.

restart-attempts

Default Value: No maximum limits
Valid Value: Any positive integer
Changes Take Effect: After T-Server is restarted

For Windows users only. Specifies how many repeatedly unsuccessful attempts T-Server makes to connect to the link before considering the connection lost.

restart-delay

Default Value: `2`
Valid Value: Any positive integer
Changes Take Effect: After T-Server is restarted

Specifies the delay (in seconds) between attempts to set connections.

template

Default Value: Mandatory field. No default value.
Valid Value: Any valid template name
Changes Take Effect: After T-Server is restarted

Specifies the name of the Digital Equipment Corporation (DEC) template for X.25.

userdata

Default Value: None
Valid Value: Any sequence of integers, comma-separated, of less than `256`
Changes Take Effect: After T-Server is restarted

Specifies the data for the X.25 call request.

x25address

Default Value: Mandatory field. `000000990100`
Valid Value: Any valid X.25 address
Changes Take Effect: After T-Server is restarted

Identifies the location of the X.25 address on the switch. A value for this option must be specified.

Note: The X.25 address of the T-Server host can be located in the Nortel Communication Server 2000/2100 SCAICOMS table.

x25device

Default Value: None
Valid Value: Any name of X.25 device in the system
Changes Take Effect: After T-Server is restarted

Indicates the name of the X.25 device being used to access host X.25 services. Only relevant for HP UX and AIX OS platforms; installation dependent.

x25localaddr

Default Value: Mandatory field. No default value.
Valid Value: Any valid X.25 address
Changes Take Effect: After T-Server is restarted

Identifies the location of the X.25 address on the host where T-Server is installed. A value for this option must be specified.

Note: The X.25 address of the Nortel Communication Server 2000/2100 MPC can be located in the Nortel Communication Server 2000/2100 MPCLINK table.

TCP Protocol Options

hostname

Default Value: Mandatory field. No default value.
Valid Value: Any valid host name
Changes Take Effect: Immediately

Specifies the host of the link, which is either the TCP/IP host name or IP address of the Nortel Communication Server 2000/2100 CTI connection.

port

Default Value: Mandatory field. No default value.
Valid Value: Any valid port address
Changes Take Effect: Immediately

Specifies the TCP/IP port of the link according to the switch configuration. The Nortel Communication Server 2000/2100 switch uses port `2500`.

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.

DN-Level Options

These options are set in the `TServer` section on the `Annex` tab of a `DN` configuration object in Configuration Manager. Each option applies only to the DN for which it is configured.

primary-qrp

Default Value: blank string
Valid Values: Any valid ACD Queue or Routing Point number
Changes Take Effect: Immediately

Specifies the primary ACD Queues and Routing Points to which this Supplementary Queue DN and Routing Point DN belong.

Note: Use this option only for Supplementary Queue DNs and Routing Points DNs. See `relate-primary-supp-qrp` for more details.

Changes from 8.0 to 8.1

For reference, Table 26 lists configuration options that have been changed between the 8.0 and the 8.1 releases of T-Server. If a configuration option has been replaced with another one that enables the same functionality, the new option name and location in this chapter are noted**.**

Table 26: Changes from 8.0 to 8.1

Option Name		Type of Change	Details
TServer Section			
relate-primary-supp-qrp		New	See page 243 for details.
DN-Level -> TServer Section			
primary-qrp		New	See page 253 for details.

Chapter

11 HA Proxy Configuration Options

This chapter describes the configuration options that are unique to the HA Proxy for Nortel Communication Server 2000/2100. It includes these sections:

- Mandatory Options, page 255
- HA Proxy Section, page 258
- CTI-Link Section, page 260
- Changes from 8.0 to 8.1, page 263

The options common to all T-Servers are described in Chapter 8, "Common Configuration Options," on page 177 and Chapter 9, "T-Server Common Configuration Options," on page 199.

Mandatory Options

Table 27 on page 256 lists the options that you must configure for basic HA Proxy operation. All other options in this chapter are configured to enable HA Proxy to support various features.

To establish a link connection, simply configure the link options (TCP/IP or X.25) that are applicable to the connection protocol used in your environment.

Table 27: Mandatory Options

Option Name	Default Value	Details
HA Proxy Section		
link-*n*-name	No default value	Specifies the section name containing the link configuration options for that link, where `n` is the consecutive number of a CTI link. See description on page 259.
CTI-Link Section		
comport	`a` or `0`	Specifies the serial port number that HA Proxy uses for its X.25 connection. This value is the PVC number or the SVC address of the X.25 physical port connected to the switch. Mandatory for X.25 links. See description on page 261.
dteclass	No default value	Specific to Digital Unix machines. This option specifies the DTE class to be used in the link. Mandatory for X.25 links. See description on page 261.
hostname	No default value	Specifies the host of the link according to the switch configuration. Mandatory for a TCP link. See description on page 260.
mode	No default value	Specifies the X.25 mode: either `SVC` (Switched Virtual Circuit) or `PVC` (Permanent Virtual Circuit). Mandatory for X.25 links. See description on page 261.

Table 27: Mandatory Options (Continued)

Option Name	Default Value	Details
port	No default value	Specifies the TCP/IP port of the link according to the switch configuration. Mandatory for a TCP link. See description on page 260.
protocol	No default value	Specifies the connection protocol HA Proxy uses in communicating with the switch. Mandatory for a TCP link and X.25 links. See description on page 260.
restart-attempts	No maximum limits	For Windows users only. Specifies how many repeatedly unsuccessful attempts HA Proxy makes to connect to the link before considering the connection lost. Mandatory for X.25 links. See description on page 262.
restart-delay	2	Specifies the delay (in seconds) between attempts to set connections. Mandatory for X.25 links. See description on page 262.
template	No default value	Specifies the name of the DEC template for X.25. Mandatory for X.25 links. See description on page 262.
userdata	No default value	Specifies the data for the X.25 call request. Mandatory for X.25 links. See description on page 262.
x25address	No default value	Identifies the location of the X.25 address on the local (host) computer. Mandatory for X.25 links. See description on page 262.

Table 27: Mandatory Options (Continued)

Option Name	Default Value	Details
x25device	No default value	Indicates the name of the X.25 device being used to access host X.25 services. Only relevant for HP-UX and AIX OS platforms; installation dependent. Mandatory for X.25 links. See description on page 263.
x25localaddr	No default value	Identifies the location of the X.25 address on the host where HA Proxy is installed. Mandatory for X.25 links. See description on page 263.

HA Proxy Section

This section describes configuration options that support features for a High Availability configuration. These options are located in the `proxy` section on the `Options tab` for the HA Proxy Application object in the Configuration Layer.

Note: Nortel Communication Server 2000/2100 does not support HA Proxies in a Warm Standby configuration. See Figure 5, "Warm Standby Redundancy Architecture," on page 46.

You must call this section `proxy`.

buffering-timeout

Default Value: `10`
Valid Values: `0-6000`
Changes Take Effect: Immediately

Specifies the maximum interval (in seconds) that switch events are buffered after one T-Server connection is lost. If the connection is not restored within the specified interval, the buffered events are sent to the remaining T-Server.

ha-heartbeat-failures

Default Value: `0` (never fail if heartbeats is turned on)
Valid Values: `0-60000`
Changes Take Effect: Immediately

Specifies the number of heartbeat failures after which HA proxy shuts down the link. A value of `0` enables an infinite number of heartbeat failures, so the link does not shut down.

HA Proxy and T-Server heartbeats are performed independently of each other.

Both HA Proxy and T-Server must support HA heartbeats for both to function correctly. If only one of the two supports heartbeats, turning on heartbeats may result in unexpected behavior, such as links shutting down and restarting unexpectedly.

ha-heartbeat-period

Default Value: `0` (no heartbeats)
Valid Values: `0-60000`
Changes Take Effect: Immediately

Specifies the interval at which HA Proxy sends a heartbeat message to T-Server. Setting this configuration option to a value of `0` disables this functionality.

ha-heartbeat-timeout

Default Value: `0` (no heartbeats)
Valid Values: `0-60000`
Changes Take Effect: Immediately

Specifies the interval during which HA Proxy waits for a heartbeat message from T-Server. Setting this configuration option to a value of `0` disables this functionality.

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. You must specify a value for this configuration option.

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

link-start-timeout

Default Value: `5`
Valid Values: `1-6000`
Changes Take Effect: Immediately

Specifies the interval (in seconds) that HA Proxy waits before disconnecting T-Servers once the connection to the link is unavailable.

logon-timeout

Default Value: `5`
Valid Values: `0-6000`
Changes Take Effect: Immediately

Specifies the interval (in seconds) that HA Proxy waits before attempting to log on to a disconnected link.

CTI-Link Section

The section name is specified by the `link-n-name` configuration option when X.25 links or a TCP link to the switch are used with HA Proxy and is only valid for Nortel Communication Server 2000/2100 versions SCAI10+ (NA008+). One section per link is required. For more information, see `link-n-name` on page 259.

- If you are using a TCP link to the switch, configure options as they are described under "TCP Protocol Options" below.
- If you are using X.25 links to the switch, configure options as they are described under "X.25 Protocol Options" on page 261.

Warning! Do not update the link configuration while T-Server is running.

TCP Protocol Options

hostname

Default Value: Mandatory field. No default value.
Valid Value: Any valid host name
Changes Take Effect: Immediately

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

port

Default Value: Mandatory field. No default value.
Valid Value: Any valid port address
Changes Take Effect: Immediately

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

protocol

Default Value: Mandatory field. No default value.
Valid Value: `tcp`
Changes Take Effect: Immediately

Specifies the connection protocol HA Proxy uses in communicating with the switch. You must specify a value for this configuration option.

X.25 Protocol Options

Note: On the Windows platform, T-Server supports the use of the Eicon X.25 card to establish an X.25 link to the switch. Other X.25 cards may not function with T-Server unless the card is compatible with the Eicon API. Check with your hardware vendor for details on compatibility.

comport

Default Value: `a` or `0`
Valid Values: `a, A, b, B,` or any integer from `0-9`
Changes Take Effect: After T-Server is restarted

Specifies the serial port number that HA Proxy uses for the X.25 connection. This is the PVC circuit number or the SVC address of the X.25 physical port connected to the switch.

dteclass

Default Value: Mandatory field. No default value.
Valid Value: Any valid Data Terminal Equipment (DTE) class
Changes Take Effect: After T-Server is restarted

Specific to UNICX machines. You must specify a value for this configuration option.

mode

Default Value: Mandatory field. No default value.
Valid Values: `svc, pvc`
Changes Take Effect: After T-Server is restarted

Specifies the SVC (Switched Virtual Circuit) or PVC (Permanent Virtual Connection) X.25 mode. You must specify a value for this configuration option.

Note: Currently, this T-Server only supports the SVC X.25 mode.

protocol

Default Value: Mandatory field. No default value.
Valid Value: `x25`
Changes Take Effect: Immediately

Specifies the connection protocol HA Proxy uses in communicating with the switch. You must specify a value for this configuration option.

restart-attempts

Default Value: No maximum limits
Valid Value: Any positive integer
Changes Take Effect: After T-Server is restarted

For Windows users only. Specifies how many repeatedly unsuccessful attempts HA Proxy makes to connect to the link before considering the connection lost.

restart-delay

Default Value: 2
Valid Value: 0 or any positive integer
Changes Take Effect: After T-Server is restarted

Specifies the delay (in seconds) between attempts to set connections.

template

Default Value: Mandatory field. No default value.
Valid Value: Any valid template name
Changes Take Effect: After T-Server is restarted

Specifies the name of the Digital Equipment Corporation (DEC) template for X.25.

userdata

Default Value: None
Valid Value: Any sequence of integers, comma-separated, of less than 256.
Changes Take Effect: After T-Server is restarted.

Specifies the data for the X.25 call request.

x25address

Default Value: Mandatory field. No default value.
Valid Value: Any valid X.25 address
Changes Take Effect: After T-Server is restarted

Identifies the location of the X.25 address on the switch.You must specify a value for this configuration option.

Note: The X.25 address of the HA Proxy host can be located in the Nortel Communication Server 2000/2100 SCAICOMS table.

x25device

Default Value: None
Valid Value: Any name of X.25 device in the system
Changes Take Effect: After T-Server is restarted

Indicates the name of the X.25 device being used to access host X.25 services. Only relevant for HP UX and AIX O/S platforms; installation dependent.

x25localaddr

Default Value: Mandatory field. No default value.
Valid Value: Any valid X.25 address
Changes Take Effect: After T-Server is restarted

Identifies the location of the X.25 address on the host where HA Proxy is installed. You must specify a value for this configuration option.

Note: The X.25 address of the Nortel Communication Server 2000/2100 MPC can be located in the Nortel Communication Server 2000/2100 MPCLINK table.

Changes from 8.0 to 8.1

There were no changes in the configuration options for HA Proxy for Nortel Communication Server 2000/2100 between release 8.0 and release 8.1.

Appendix

Using LinkPlexer with T-Server

This chapter describes the LinkPlexer functionality that T-Server for Nortel Communication Server 2000/2100 supports and provides guidelines for using LinkPlexer with T-Server. This chapter includes these sections:

LinkPlexer Configurations

LinkPlexer, a Nortel product that functions much like Genesys HA Proxy[2], allows switch messages to be distributed to multiple applications, including T-Server. LinkPlexer allows multiple applications to send switch messages to the same link. This permits both Genesys and non-Genesys applications to monitor and control switch resources on the same link.

Note: See the Nortel Networks document *LinkPlexer Installation and Configuration Guide* for more information.

To avoid conflicts in the control of switch resources when using T-Server with LinkPlexer, carefully review the configuration scenarios shown in Figure 17, Figure 18, and Figure 19. You cannot replace HA Proxy with LinkPlexer.

Figure 17 on page 266 shows the possible Nortel Communication Server 2000/2100 switch, LinkPlexer and T-Server configuration.

[2] LinkPlexer also attempts to behave as a switch

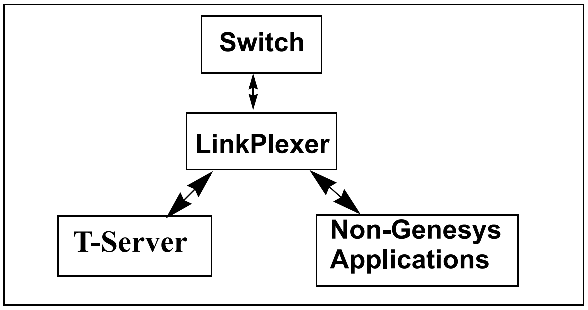

Figure 17: LinkPlexer and T-Server Configuration

Figure 18 shows the possible Nortel Communication Server 2000/2100 switch, LinkPlexer and HA Proxy configuration.

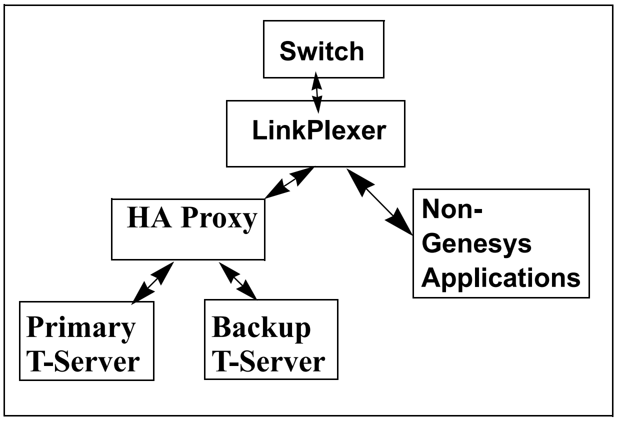

Figure 18: LinkPlexer and HA Proxy Configuration

Figure 19 shows an unsupported configuration that tries to use the Nortel Communication Server 2000/2100 switch and LinkPlexer, but no HA Proxy.

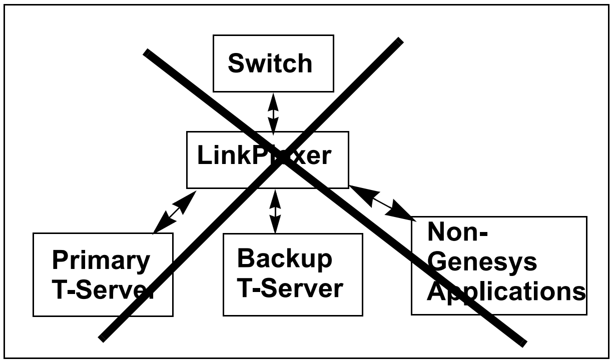

Figure 19: Unsupported Configuration

LinkPlexer Guidelines

LinkPlexer operation may vary from one version of LinkPlexer to another so be sure to read and understand Nortel's LinkPlexer documentation before using it with T-Server. Below are some guidelines to follow to help ensure seamless operability with T-Server.

Session Management

LinkPlexer manages session logons and DN association differently from one version to another. Be sure to fully understand how your version of LinkPlexer handles these things before using it with T-Server.

Resource Controlling

Understanding the control of switch resources between the different applications connected to LinkPlexer is essential. Ensure that all such applications control switch resources in a mutually exclusive way, otherwise problems may result in call processing. For example, ACD Queue resources, such as Supplementary Queue DNs and ACD Positions, should be controlled by the same application.

Logon Options

Depending on your version and configuration of LinkPlexer, session logons may be performed without the use of T-Server's logon parameters. However, since T-Server may use some of these values internally, ensure that they agree with whatever values are actually used in establishing the session. This requires you to fully understand how your version of LinkPlexer performs session logons, and, in particular it's use of LinkPlexer configuration file(s), if any.

The `TServer` options used for session logons are:

- `na010-network-node-id`
- `network-node-id`
- `service-version`
- `business-group-id`
- `application-id`
- `password`
- `tcp-linkset-name`
- `dms-upgrade-time`

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 and HA Proxy for Nortel Communication Server 2000/2100

- The Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at `http://genesyslab.com/support`.

Management Framework

Consult these additional resources as necessary:

- The *Framework 8.1 Deployment Guide*, which will help you configure, install, start, and stop Framework components.
- The *Framework 8.1 Configuration Manager Help*, which describes how to use Configuration Manager in either an enterprise or multi-tenant environment.
- The *Framework 8.1 Genesys Administrator Help,* which describes how to use Genesys Administrator in either an enterprise or multi-tenant environment.
- The *Framework 8.0 Configuration Options Reference Manual*, which will provide you with descriptions of configuration options for other Framework components.

Platform SDK

- The *Genesys Events and Models Reference Manual*, which contains an extensive collection of events and call models describing core interaction processing in Genesys environments.

- The *Voice Platform SDK 8.x .NET (or Java) API Reference*, which contains technical details of T-Library functions.

Genesys

- *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms used in this document.
- *Genesys Migration Guide*, which ships on the Genesys Documentation Library DVD, and which provides documented migration strategies for Genesys product releases. Contact Genesys Technical Support for more information.

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

- *Genesys Supported Operating Environment Reference Manual*
- *Genesys Supported Media Interfaces Reference Manual*

Consult these additional resources as necessary:

- *Genesys Hardware Sizing Guide,* which provides information about Genesys hardware sizing guidelines for the Genesys 7.x and 8.x releases.
- *Genesys Interoperability Guide,* which provides information on the compatibility of Genesys products with various Configuration Layer Environments; Interoperability of Reporting Templates and Solutions; and Gplus Adapters Interoperability.
- *Genesys Licensing Guide,* which introduces you to the concepts, terminology, and procedures relevant to the Genesys licensing system.
- *Genesys Database Sizing Estimator 8.0 Worksheets,* which provides a range of expected database sizes for various Genesys products.

For additional system-wide planning tools and information, see the release-specific listings of System Level Documents on the Genesys Technical Support website, accessible from the `system level documents by release` tab in the Knowledge Base `Browse Documents` Section.

Genesys product documentation is available on the:

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

Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

81fr_ref_06-2011_v8.1.000.01

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 28 describes and illustrates the type conventions that are used in this document.

Table 28: 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 272).	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