Framework 8.1

Network T-Server for AT&T

Deployment Guide

About Genesys

Alcatel-Lucent's Genesys solutions feature leading software that manages customer interactions over phone, Web, and mobile devices. The Genesys software suite handles customer conversations across multiple channels and resources—self-service, assisted-service, and proactive outreach—fulfilling customer requests and optimizing customer care goals while efficiently using resources. Genesys software directs more than 100 million customer interactions every day for 4,000 companies and government agencies in 80 countries. These companies and agencies leverage their entire organization, from the contact center to the back office, while dynamically engaging their customers. Go to www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Genesys, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the regional numbers provided on page 11. 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_att_01-2012_v8.1.001.00

Table of Contents

List of Procedures

Preface

Welcome to the *Framework 8.1 Network T-Server for AT&T Deployment Guide.* This document introduces you to the concepts, terminology, and procedures relevant to T-Servers® in general and provides detailed reference information about Network T-Server for AT&T. The reference information includes, but is not limited to, configuration options, limitations, and switch-specific functionality. You must configure the configuration objects and options described in this document in the Framework Configuration Layer.

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

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

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

This preface contains the following sections:

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on page 231.

About Network T-Server for AT&T

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:

- NIO AT&T.
- The current name is Network T-Server for AT&T.

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

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
Malaysia	1-800-814-472 (toll-free) +61-7-3368-6868 (International)	support@genesyslab.com.au
India	000-800-100-7136 (toll-free) +61-7-3368-6868 (International)	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp
Before contacting technical support, refer to the *Genesys Technical Support Guide* for complete contact information and procedures.		

Document Change History

This is the first release of the *Framework 8.1 Network T-Server for AT&T Deployment Guide*. In the future, this section will list topics that are new or that have changed significantly since the first release of this document.

Part

1 T-Server Deployment

Part One of this *Network 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 15, 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 29, presents configuration and installation procedures for all T-Servers.
- Chapter 3, "Multi-Site Support," on page 43, details the variations available for T-Server implementations across geographical locations.
- Chapter 4, "Starting and Stopping T-Server Components," on page 101, 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 154 for details.

- T-Server now supports the Unresponsive Process Detection feature. The following configuration options enable this feature:
  - "heartbeat-period" on page 149
  - "hangup-restart" on page 150

  For more information, refer to the *Framework 8.1 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 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 179.
- 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 13, 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.

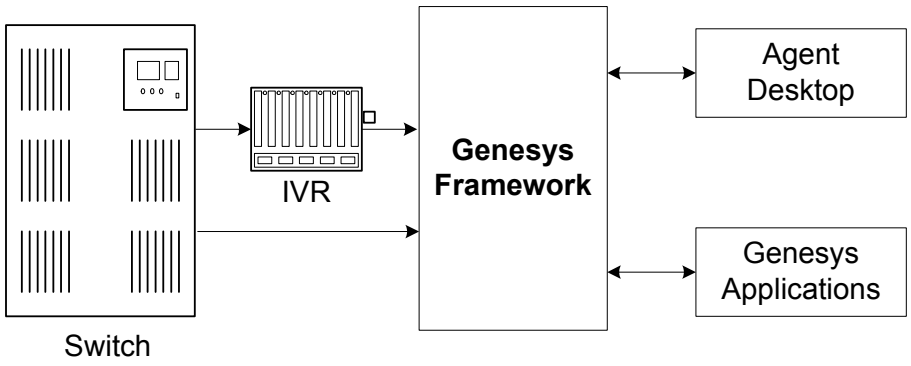

Figure 1: Framework in a Genesys Solution

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

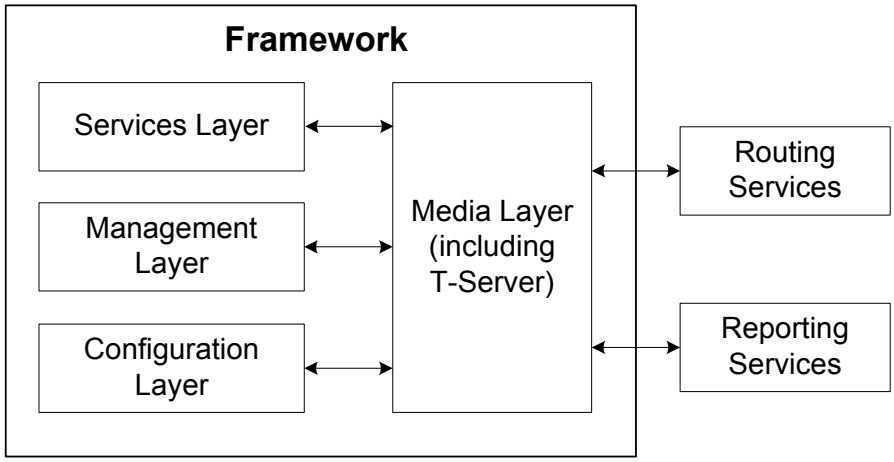

Figure 2: The Media Layer in the Framework Architecture

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.

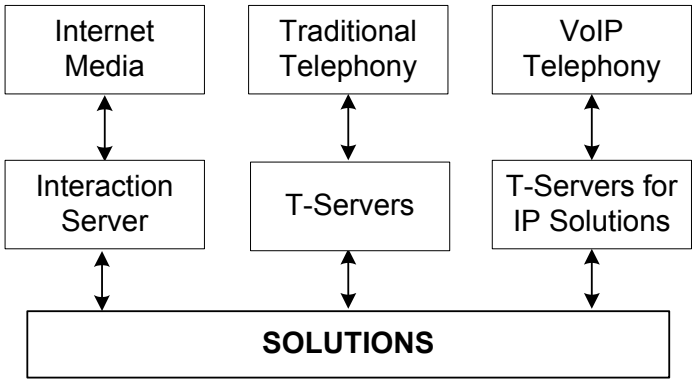

Figure 3: Media Layer Architecture

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 17 (and other figures) depicts T-Server that works with telephony systems as a single product, this is a simplification. Because

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

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

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

T-Server Functional Steps During a Sample Call

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

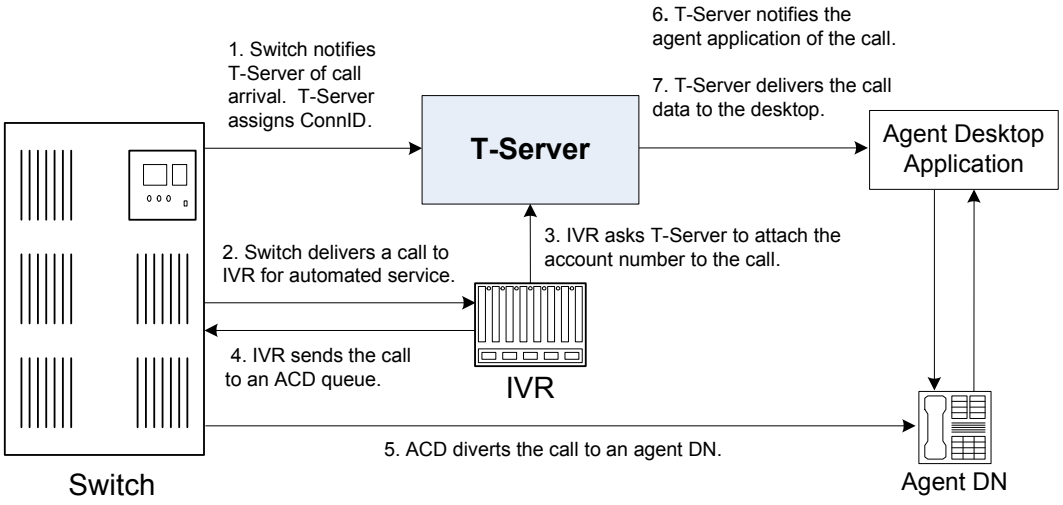

Figure 4: Functional T-Server Steps

Step 1

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

Step 2

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

Step 3

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

Step 4

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

Step 5

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

Step 6

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

Step 7

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

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

Advanced Disconnect Detection Protocol

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

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

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

ADDP applies only to connections between Genesys software components.

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

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

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

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

Redundant T-Servers

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

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

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

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

Support for Hot Standby Redundancy in Various T-Servers

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

Table 1 only summarizes `hot standby` redundancy support in various T-Servers. For detailed, up-to-date information on the subject, see the *Genesys Supported Media Interfaces Reference Manual* located on the Technical Support website at http://genesyslab.com/support/dl/retrieve/default.asp?item=A9CB309AF4DEB8127C5640A3C32445A7&view=item.

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

T-Server Type	Hot Standby Supported	HA Proxy Required	Number of HA Proxy Components
Aastra MXONE CSTA I	Yes	No	—
Alcatel A4200/OXO	Yes	No	—
Alcatel A4400/OXE	Yes	No	—
Aspect ACD	Yes	No	—
Avaya Communication Manager	Yes	No[a]	—
Avaya INDeX	Yes	No	—
Avaya TSAPI	Yes	No	—
Cisco UCCE	Yes	No	—
Cisco Unified Communications Manager	Yes	No	—
DataVoice Dharma	Yes	No	—
Digitro AXS/20	Yes	No	—

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

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

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

T-Server Type	Hot Standby Supported	HA Proxy Required	Number of HA Proxy Components
Concert	No	—	—
CRSP	No	—	—
DTAG	No	—	—
GenSpec	No	—	—
ISCP	No	—	—
IVR Server, using network configuration	Yes	—	—
KPN	No	—	—
MCI	No	—	—
NGSN	No	—	—
Network SIP Server	No	—	—
Sprint	No	—	—
SR3511	No	—	—
Stentor	No	—	—

a. With release 7.1, T-Server for Avaya Communication Manager no longer uses HA Proxy for its support of `hot standby`. Earlier releases of this T-Server require two HA Proxies to support `hot standby`.

b. For T-Server for Nortel Communication Server 2000/2100 in high-availability (`hot standby`) configuration, Genesys recommends that you use link version SCAI14 or above with call-progress and noncontroller-released messages enabled. See the switch-specific information in Part 2 of this *Deployment Guide* for additional information on HA configurations.

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

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

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

Multi-Site Support

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

For instructions on installing and configuring a multi-site environment, including information on the Inter Server Call Control (ISCC) features, please see Chapter 3, "Multi-Site Support," on page 43.

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 45), 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 52 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 162 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 162).

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

Chapter

2 T-Server General Deployment

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

This chapter contains these sections:

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 8, "T-Server Common Configuration Options," on page 153. *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 4, "Starting and Stopping T-Server Components," on page 101.

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 3, “Multi-Site Support,” on page 43, 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 86, 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 3, "Multi-Site Support," on page 43.

End of procedure

Next Steps

- See "Installation of T-Server" on page 38.

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 37 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 41.
- To test your configuration and installation, go to Chapter 4, "Starting and Stopping T-Server Components," on page 101, and try it out.
- To install T-Servers for a multi-site environment, proceed to Chapter 3, "Multi-Site Support," on page 43.

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 37 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 41.
- To test your configuration and installation, go to Chapter 4, "Starting and Stopping T-Server Components," on page 101, and try it out.

- To install T-Servers for a multi-site environment, proceed to Chapter 3, "Multi-Site Support," on page 43.

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 39
- Procedure: Installing T-Server on Windows, on page 40

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 4, "Starting and Stopping T-Server Components," on page 101, and try it out. If you want to install T-Servers for a multi-site environment, proceed to Chapter 3, "Multi-Site Support," on page 43.

Chapter

3 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 8, "T-Server Common Configuration Options," on page 153.

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 61 and Table 4 on page 66.

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 52 and "Transfer Connect Service Feature" on page 64.
  - 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 65).
  - Number Translation feature (see page 69).
  - Network Attended Transfer/Conference (NAT/C) feature (see page 77).

  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 80)
  - Party Events propagation (see page 81)

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

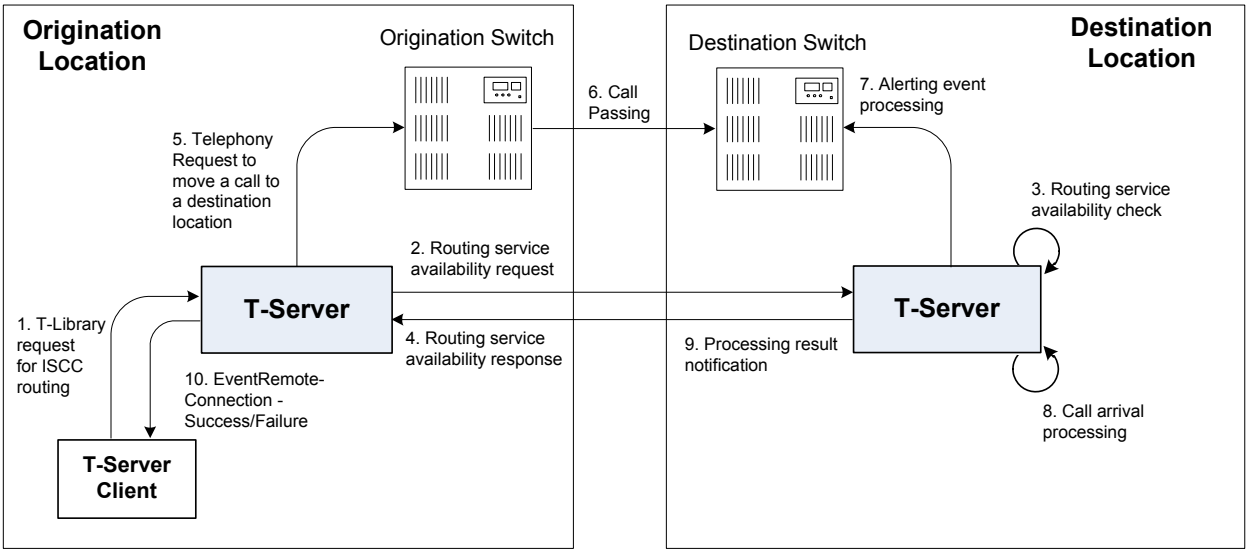

Figure 5: Steps in the ISCC Process

ISCC Call Flows

The following section identifies the steps (shown in Figure 5) 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 88.

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 8, "T-Server Common Configuration Options," on page 153 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 53).

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 53. Use Table 3 on page 61 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 8, "T-Server Common Configuration Options," on page 153 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 53
- `direct-notoken`, page 55
- `dnis-pool`, page 56
- `pullback`, page 57
- `reroute`, page 58
- `route` (aliased as `route-notoken`), the default transaction type, page 59

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

- `direct-callid` (aliased as `direct`), page 54
- `direct-digits` (reserved for Genesys Engineering)
- `direct-network-callid`, page 54
- `direct-uui`, page 55
- `route-uui`, page 60

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 97 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 6.

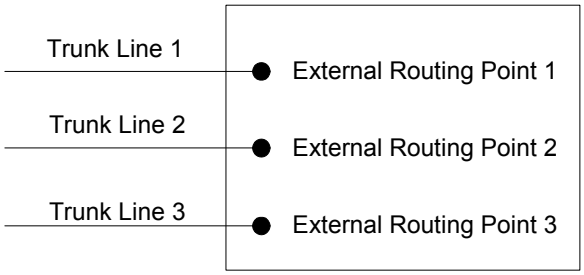

Figure 6: Point-to-Point Trunk Configuration

Note: Dedicated DNs of the `External Routing Point` type must be configured in a switch. See “Configuring Multi-Site Support” on page 86.

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 7.

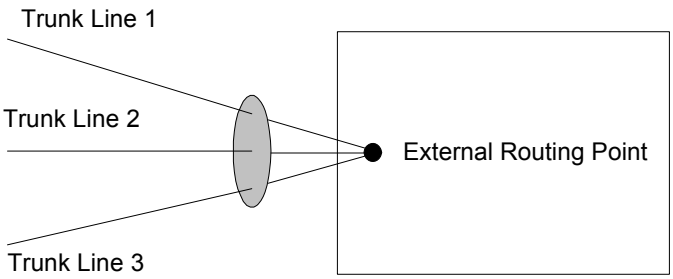

Figure 7: Multiple-to-Point Trunk Configuration

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 59) and the UUI matching feature of the `direct-uui` transaction type (page 55). 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	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 `TRequestRouteCall` 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 69.

ISCC/COF Call Flow

Figure 8 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.

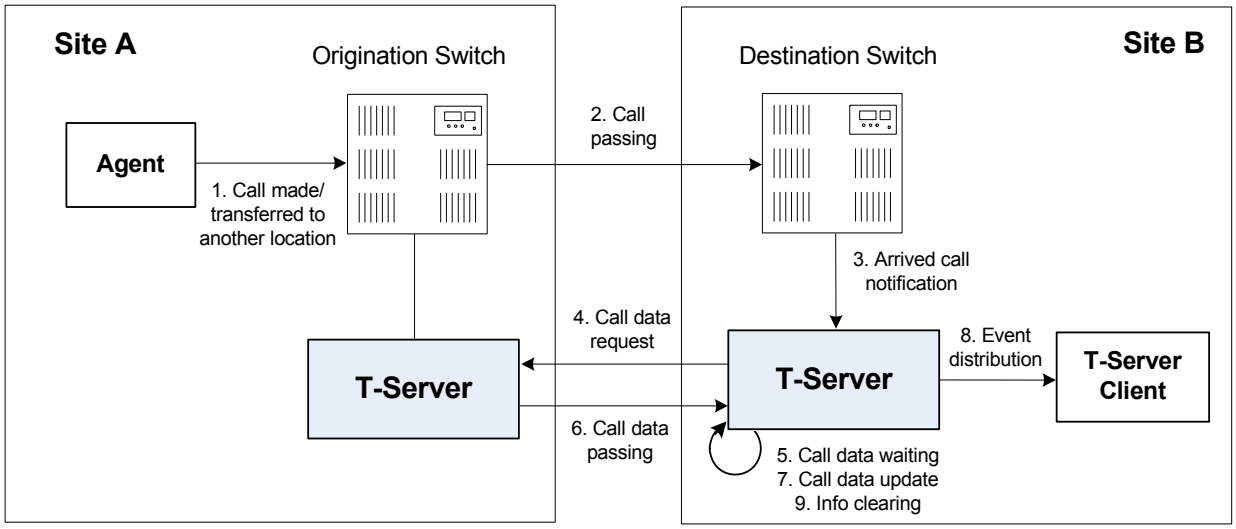

Figure 8: Steps in the ISCC/COF Process

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 70 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 75 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 77.

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 75) 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 8, "T-Server Common Configuration Options," on page 153.
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 9 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).

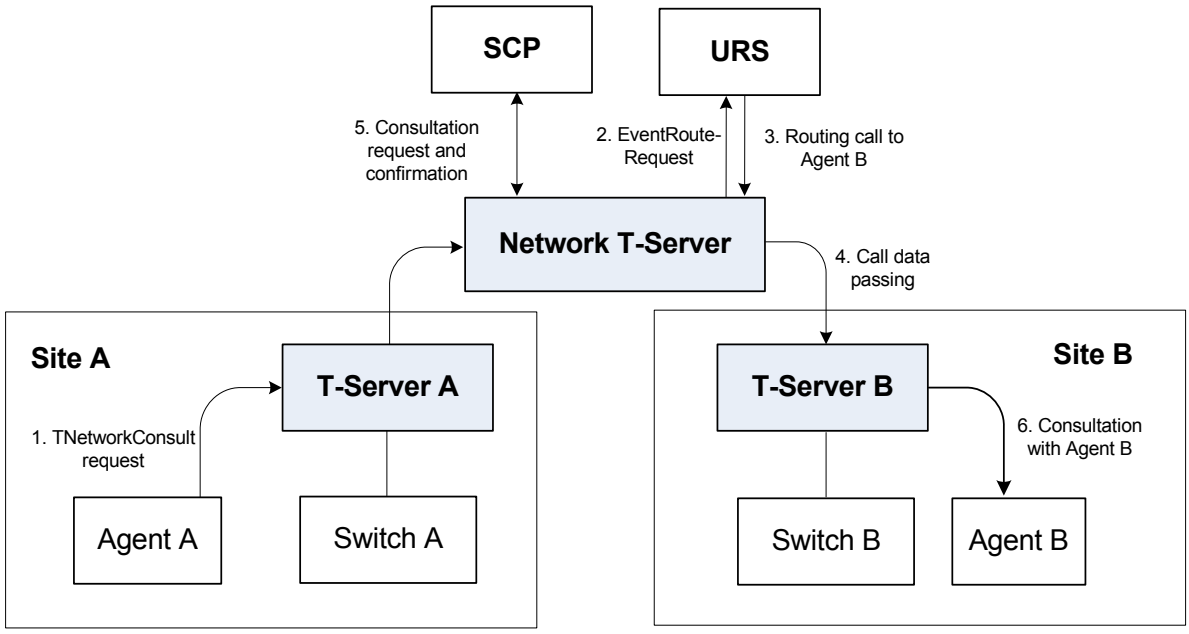

Figure 9: Steps in the NAT/C Process in URS-Controlled Mode

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 45 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 10.)

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.

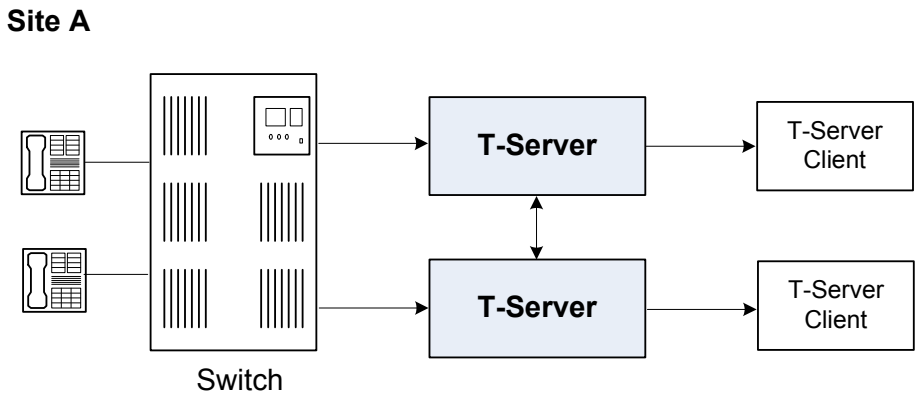

Figure 10: Switch Partitioning Architecture

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 158.

To control race conditions that may occur in the switch-partitioned environment, use the `epp-tout` configuration option (see page 173).

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 8, "T-Server Common Configuration Options," on page 153.
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 84

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 84

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 89 for instructions. If no `Access Codes` are configured, see page 90 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 31, as well as previous sections of this chapter on multi-site deployment. In particular, Table 3 on page 61 shows which transaction types are supported by a specific T-Server, while Table 4 on page 66 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 163 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 94 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 163, 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 93.

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 84.
`direct-network-callid=<>`	For configuration information, see Part Two of this document. (Use Table 4 on page 66 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 66 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 88.

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 94.
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 90.

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 90.

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 4, "Starting and Stopping T-Server Components," on page 101 to test your configuration and installation.

Chapter

4 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 101.
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 105 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 101.

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 109.

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 101.

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 106 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 106 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 101.

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 106 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 106 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 8.1 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 101 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 *Network 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 5, "AT&T–Specific Configuration," on page 115, describes compatibility and configuration information specific to ATT, including recommendations for the switch configuration.
- Chapter 6, "Supported Functionality in Network T-Server for AT&T," on page 117, describes which features this T-Server supports, including T-Library functionality, use of the `Extensions` attribute, and error messages.
- Chapter 7, "Common Configuration Options," on page 131, describes log configuration options that are common to all Genesys server applications.
- Chapter 8, "T-Server Common Configuration Options," on page 153, describes configuration options common to all T-Server types including options for multi-site configuration.
- Chapter 9, "Configuration Options in Network T-Server for AT&T," on page 181, describes configuration options specific to this T-Server, including the link-related options—those which address the interface between T-Server and the switch.

New in Network T-Server for AT&T

The following new feature is now available in the initial 8.1 release of Network T-Server for AT&T:

- Network T-Server now supports version 9.1 of Natural Access software.
- Network T-Server now supports vSphere 4 Hypervisor.

Notes:
- For a list of new features common to all T-Servers, see "New for All T-Servers in 8.1" on page 13.
- Configuration option changes that apply to Network T-Server for AT&T are described in "Changes from 8.0 to 8.1" on page 193.

Chapter

5 AT&T–Specific Configuration

This chapter presents reference information specific for configuring Network T-Server for AT&T. It contains the following section:

Note: For general instructions on T-Server deployment procedures, refer to Chapter 2, "T-Server General Deployment," on page 29. Use the *Framework 8.1 Deployment Guide* to prepare accurate configuration information. You may also consult *Configuration Manager Help*, which contains detailed information on configuring objects.

Manual Configuration of Telephony Objects

This section describes the telephony objects that you need to configure for Network T-Server for AT&T using Configuration Manager.

Switching Office

The `Switching Office Type` for the Network T-Server for AT&T is: `AT&T 800 ICP Gateway`.

Service Numbers

The toll-free numbers used as network-level routing points must be configured under the network switch as DNs of type `Service Number`.

Note: For option information common to all T-Servers, refer to Chapter 8, "T-Server Common Configuration Options," on page 153. For specific option information, see Chapter 9, "Configuration Options in Network T-Server for AT&T," on page 181.

Chapter

6 Supported Functionality in Network T-Server for AT&T

This chapter describes the telephony functionality supported by the Network T-Server for AT&T, and includes these sections

T-Library Functionality

The tables in this chapter present T-Library functionality supported in the AT&T switch. The table entries use these notations:

- **N**—Not supported.
- **Y**—Supported.
- **E**—Event only is supported.
- **I**—Internal reserved for Genesys Engineering.

In Table 10, 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 the *Voice Platform SDK 8.x .NET (or Java) API Reference.*

Table 10 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 10 are reserved for internal use and are listed here merely for completeness of information.

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

Table 10: Supported Functionality

Feature Request	Request Subtype	Corresponding Event(s)	Supported
General Requests			
TOpenServer		EventServerConnected	Y
TOpenServerEx		EventServerConnected	Y
TCloseServer		EventServerDisconnected	Y
TSetInputMask		EventACK	Y
TDispatch		Not Applicable	Y
TScanServer		Not Applicable	Y
TScanServerEx		Not Applicable	Y
Registration Requests			
TRegisterAddress[a]		EventRegistered	Y
TUnregisterAddress[a]		EventUnregistered	Y
Call-Handling Requests			
TMakeCall[b]	Regular	EventDialing	N
	DirectAgent		N
	SupervisorAssist		N
	Priority		N
	DirectPriority		N
TAnswerCall		EventEstablished	N
TReleaseCall		EventReleased	N
TClearCall		EventReleased	N
THoldCall		EventHeld	N
TRetrieveCall[c]		EventRetrieved	N
TRedirectCall		EventReleased	N
TMakePredictiveCall		EventDialing* EventQueued	N

Table 10: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
Transfer/Conference Requests			
TInitiateTransfer[b]		EventHeld EventDialing*	N
TCompleteTransfer		First arriving EventReleased* EventPartyChanged	N
TInitiateConference[b]		EventHeld EventDialing*	N
TCompleteConference		EventReleased* EventRetrieved EventPartyChanged EventPartyAdded	N
TDeleteFromConference		EventPartyDeleted* EventReleased	N
TReconnectCall[d,d]		EventReleased EventRetrieved*	N
TAlternateCall		EventHeld* EventRetrieved	N
TMergeCalls	ForTransfer	EventReleased* EventPartyChanged	N
	ForConference	EventReleased* EventRetrieved EventPartyChanged EventPartyAdded	N
TMuteTransfer[b]		EventHeld, EventDialing* EventReleased EventPartyChanged	N
TSingleStepTransfer[b]		EventReleased* EventPartyChanged	N
TSingleStepConference		EventPartyAdded* EventRinging* EventEstablished	N

Table 10: Supported Functionality (Continued)

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

Table 10: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TApplyTreatment (cont.)	PlayAnnouncement	(EventTreatmentApplied + EventTreatmentEnd)/ EventTreatmentNotApplied	N
	PlayAnnouncementAnd-Digits		N
	VerifyDigits		N
	RecordUserAnnouncement		N
	DeleteUserAnnouncement		N
	CancelCall		N
	PlayApplication		N
	SetDefaultRoute		N
	TextToSpeech		N
	TextToSpeechAndDigits		N
	FastBusy		N
	RAN		N
TGiveMusicTreatment		EventTreatmentApplied	N
TGiveRingBackTreatment		EventTreatmentApplied	N
TGiveSilenceTreatment		EventTreatmentApplied	N
DTMF 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

Table 10: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
Agent & DN Feature Requests			
TAgentLogin		EventAgentLogin	N
TAgentLogout		EventAgentLogout	N
TAgentSetIdleReason		EventAgentIdleReasonSet	N
TAgentSetReady		EventAgentReady	N
TAgentSetNotReady		EventAgentNotReady	N
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

Table 10: Supported Functionality (Continued)

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

Table 10: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TQueryLocation[a]	AllLocations	EventLocationInfo[e]	I
	LocationData		I
	MonitorLocation		I
	CancelMonitorLocation		I
	MonitorAllLocations		I
	CancelMonitorAllLocations		I
	LocationMonitorCanceled		I
	AllLocationsMonitor-Canceled		I
TQueryServer[a]		EventServerInfo	Y
User-Data Requests			
TAttachUserData (Obsolete)		EventAttachedDataChanged	Y
TUpdateUserData		EventAttachedDataChanged	Y
TDeleteUserData		EventAttachedDataChanged	Y
TDeleteAllUserData		EventAttachedDataChanged	Y
ISCC (Inter Server Call Control) Requests			
TGetAccessNumber[b]		EventAnswerAccessNumber	I
TCancelReqGetAccess-Number		EventReqGetAccessNumber-Canceled	I
ISCC Transaction Monitoring Requests			
TTransactionMonitoring		EventACK	Y
		EventTransactionStatus	E
Special Requests			
TReserveAgent		EventAgentReserved	Y
TSendEvent		EventACK	I
TSendEventEx		EventACK	I

Table 10: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TSetCallAttributes		EventCallInfoChanged	I
TSendUserEvent		EventACK	Y
TPrivateService		EventPrivateInfo	N
Network Attended Transfer Requests			
TNetworkAlternate		EventNetworkCallStatus	N
TNetworkConsult		EventNetworkCallStatus	N
TNetworkMerge		EventNetworkCallStatus	N
TNetworkReconnect		EventNetworkCallStatus	N
TNetworkSingleStep-Transfer		EventNetworkCallStatus	N
TNetworkTransfer		EventNetworkCallStatus	N
TNetworkPrivateService		EventNetworkPrivateInfo	N

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

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

c. Supported only for T-Library clients using version 5.0 or later.

d. `TReconnectCall` will not function properly if `Auto Hold Allowed` is enabled on the switch.

e. Two subtypes are supported by `EventLocationInfo LocationInfoLocationMonitorCanceled` and `LocationInfoAllLocationsMonitorCanceled`.

Load Balancing

Network T-Server for AT&T uses Load Balancing, a type of load-sharing redundancy, instead of the `warm standby` or `hot standby` redundancy type. With this functionality, two or more Network T-Servers are attached to an SCP. The SCP distributes calls across the group of load-balanced Network T-Servers and determines the backup solution in case of failure.

The Solution Control Interface (SCI) sees each one of the attached Network T-Servers as an active, primary server.

Note: The primary/backup methodology found in other applications does not apply to load balancing, and the SCI switchover functionality does not apply to load-balanced Network T-Servers.

The Network T-Server applications can be started and stopped manually. Once they are running, you can manually shut down any Network T-Server, and the others will continue to run. If a software failure occurs on any given Network T-Server, any calls already in process on that Network T-Server are lost, but all the other Network T-Servers continue to operate.

To implement Network T-Server Load Balancing, create multiple, separate Network T-Server `Application` objects that share the same `Switch` object. Enter `Not Specified` for the redundancy type, and do not select a backup server.

T-Server Error Messages

The following tables present the complete set of error messages T-Server distributes with `EventError`. While some guidelines are given on how to handle T-Server Common Part (TSCP) error messages, refer to the switch documentation for resolution of switch-related errors.

Table 11: T-Server Common Part (TSCP) Error Messages

T-Library Error Code	Symbolic Name	Description	Recommendations
40	TERR_NOMORE_LICENSE	No more licenses are available	Ensure that the license file contains enough licenses, or increase the number of licenses in T-Server configuration.
41	TERR_NOT_REGISTERED	Client had not registered for the DN	T-Server reports unauthorized access. Make sure that the T-client successfully registers the DN before sending other requests involving the DN.
42	TERR_RESOURCE_SEIZED	Resource is already seized	The resource (DN) cannot be registered, because another application has registered for it in private mode. Select another DN for registration, or have that application unregister the DN first.

Table 11: T-Server Common Part (TSCP) Error Messages (Continued)

T-Library Error Code	Symbolic Name	Description	Recommendations
50	TERR_UNKNOWN_ERROR	Unrecognized error	T-Server could not identify the reason for the error. Check the error message text for possible explanation of the error.
51	TERR_UNSUP_OPER	Unsupported operation	The combination of this T-Server release with this switch version does not support the requested operation.
52	TERR_INTERNAL	Internal error	Contact Genesys Technical Support.
53	TERR_INVALID_ATTR	Invalid attribute value	Check the content of the client request for correctness.
54	TERR_NO_SWITCH	No connection to the switch	Ensure that the connection to the switch exists.
55	TERR_PROTO_VERS	Incorrect protocol version	T-Server cannot recognize the client version. Make sure that the client uses the T-Library protocol; check the T-Server and client version compatibility.
56	TERR_INV_CONNID	Invalid Connection ID	At the time when T-Server received the request, the Connection ID was invalid. Ensure that the specified Connection ID is associated with a live call.
57	TERR_TIMEOUT	Timeout expired	The request processing was cancelled due to a timeout. Resubmit the request or check that the request is valid in association with the subject of the request.

Table 11: T-Server Common Part (TSCP) Error Messages (Continued)

T-Library Error Code	Symbolic Name	Description	Recommendations
58	TERR_OUT_OF_SERVICE	Out of service	The referenced resource (for example, DN) is out of service.
59	TERR_NOT_CONFIGURED	DN is not configured in the Configuration Database	A client attempts to register for a DN that must be configured in the Configuration Database. Verify that the specified DN number is valid and, if it is, add the DN to the switch configuration in the Configuration Database.

Voice-Processing Failure Messages

Table 12: Voice-Processing Failure Messages

T-Library Error Code	Symbolic Name	Description	Switch Error Code
231	TERR_DN_BUSY	DN Is Busy	3009
232	TERR_DN_NO_ANSWER	No Answer at DN	300A
234	TERR_CONN_ATMPT_FAIL	Call Connection Attempt Has Failed	300C

Known Limitations

Several known limitations exist in the current Network T-Server for AT&T environment:

1. Network T-Server uses load-sharing redundancy instead of the `warm standby` or `hot standby` redundancy type. With load-sharing redundancy, two or more Network T-Servers are attached to an SCP. The SCP distributes the workload between the attached Network T-Servers. Since `hot standby` redundancy is not supported, the configuration options described in the "Backup-Synchronization Section" in Chapter 8 on page 153, do not apply to this Network T-Server. See "Load Balancing" on page 125 for more information.

2. Network T-Server will not attempt to connect to a backup Configuration Server in a switchover scenario where the backup was configured for the primary Configuration Server after the AT&T Network T-Server was started. In this scenario, a manual restart of Network T-Server is required in order to establish the connection to the backup Configuration Server.
3. During installation, configure Network T-Server as case sensitive to match similar configuration in other Genesys products.
4. Network T-Server does not pass attached user data to a remote T-Server when these two conditions occur simultaneously:
   - The `joint` method of handling user data for consultation calls is used.
   - A call is overflowed or transferred to a remote switch using the External Routing feature.

   Consultation calls retain all data when overflowed to a remote switch using the External Call-Handling feature.
5. If Local Control Agent starts as a Windows NT service or as a UNIX background process, do not use console output for the Application log.
6. To enable multi-site functionality, restart all T-Servers after adding the ExtRouter feature in the T-Server license file. T-Server verifies any modification to the license file at startup.
7. T-Server may incorrectly generate events in the following scenario:
   - DN1 on Site 1 uses the External Routing feature to place a call to DN2 on remote Site 2.
   - DN2 answers the call sent from DN1.
   - DN2 uses the External Routing feature to perform a call transfer to DN3 at Site 1.
   - DN3 answers the call sent from DN2.
   - With the call answered, DN2 completes the transfer.
   - DN1 releases the call.

   In this scenario, the T-Server at Site 1 may incorrectly generate events for the call and report the incorrect status of DN1 to T-Server clients.

Chapter

7 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 138.

Note: For definitions of the Standard, Interaction, Trace, and Debug log levels, refer to the *Framework 8.0 Management Layer User's Guide*, *Framework 8.1 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 138). 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 8.1 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 138). 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 138.

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 142.

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

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

all

Default Value: No default value
Valid Values (log output types):

`stdout`	Log events are sent to the Standard output (`stdout`).
`stderr`	Log events are sent to the Standard error output (`stderr`).
`network`	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. Setting the `all` log level option to the `network` output enables an application to send log events of the `Standard`, `Interaction`, and `Trace` levels to Message Server. `Debug`-level log events are neither sent to Message Server nor stored in the Log Database.
`memory`	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends all log events. The log output types must be separated by a comma when more than one output is configured. For example:

```
all = stdout, logfile
```

Note: To ease the troubleshooting process, consider using unique names for log files that different applications generate.

alarm

Default Value: No default value
Valid Values (log output types):

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

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the Alarm level. The log output types must be separated by a comma when more than one output is configured. For example:

```
standard = stderr, network
```

standard

Default Value: No default value
Valid Values (log output types):

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

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Standard` level. The log output types must be separated by a comma when more than one output is configured. For example:

```
standard = stderr, network
```

interaction

Default Value: No default value
Valid Values (log output types):

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

Changes Take Effect: Immediately

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

```
interaction = stderr, network
```

trace

Default Value: No default value
Valid Values (log output types):

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

Changes Take Effect: Immediately

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

```
trace = stderr, network
```

debug

Default Value: No default value
Valid Values (log output types):

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

Changes Take Effect: Immediately

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

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

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

Log File Extensions

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

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

  Note: Provide `*.snapshot.log` files to Genesys Technical Support when reporting a problem.

- `*.memory.log`—Assigned to log files that contain the memory output snapshot when you configure output to memory and redirect the most recent memory output to a file. For example, if you set `standard = memory` and `memory = confserv` for Configuration Server, it prints the latest memory output to a file called `confserv.<time_stamp>.memory.log`.

Examples

This section presents examples of a `log` section that you might configure for an application when that application is operating in production mode and in two lab modes, debugging and troubleshooting.

Production Mode Log Section

```
[log]
verbose = standard
standard = network, logfile
```

With this configuration, an application only generates the log events of the `Standard` level and sends them to Message Server, and to a file named `logfile`, which the application creates in its working directory. Genesys recommends that you use this or a similar configuration in a production environment.

Warning! Directing log output to the console (by using the `stdout` or `stderr` settings) can affect application performance. Avoid using these log output settings in a production environment.

Lab Mode Log Section

```
[log]
verbose = all
all = stdout, /usr/local/genesys/logfile
trace = network
```

With this configuration, an application generates log events of the `Standard`, `Interaction`, `Trace`, and `Debug` levels, and sends them to the standard output and to a file named `logfile`, which the application creates in the `/usr/local/genesys/` directory. In addition, the application sends log events of the `Standard`, `Interaction`, and `Trace` levels to Message Server. Use this configuration to test new interaction scenarios in a lab environment.

Failure-Troubleshooting Log Section

```
[log]
verbose = all
standard = network
all = memory
memory = logfile
memory-storage-size = 32 MB
```

With this configuration, an application generates log events of the `Standard` level and sends them to Message Server. It also generates log events of the `Standard`, `Interaction`, `Trace`, and `Debug` levels, and sends them to the memory output. The most current log is stored to a file named `logfile`, which the application creates in its working directory. Increased memory storage allows an application to save more of the log information generated before a failure.

Note: If you are running an application on UNIX, and you do not specify any files in which to store the memory output snapshot, a core file that the application produces before terminating contains the most current application log. Provide the application's core file to Genesys Technical Support when reporting a problem.

Debug Log Options

The options in this section enable you to generate `Debug` logs containing information about specific operations of an application.

x-conn-debug-open

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about "open connection" operations of the application.

Warning! Use this option only when requested by Genesys Technical Support.

x-conn-debug-select

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about "socket select" operations of the application.

Warning! Use this option only when requested by Genesys Technical Support.

x-conn-debug-timers

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about the timer creation and deletion operations of the application.

Warning! Use this option only when requested by Genesys Technical Support.

x-conn-debug-write

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about "write" operations of the application.

Warning! Use this option only when requested by Genesys Technical Support.

x-conn-debug-security

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about security-related operations, such as Transport Layer Security and security certificates.

Warning! Use this option only when requested by Genesys Technical Support.

x-conn-debug-api

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about connection library function calls.

Warning! Use this option only when requested by Genesys Technical Support.

x-conn-debug-dns

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about DNS operations.

Warning! Use this option only when requested by Genesys Technical Support.

x-conn-debug-all

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about open connection, socket select, timer creation and deletion, write, security-related, and DNS operations, and connection library function calls. This option is the same as enabling or disabling all of the previous `x-conn-debug-<op type>` options.

Warning! Use this option only when requested by Genesys Technical Support.

log-extended Section

This section must be called `log-extended`.

level-reassign-<eventID>

Default Value: Default value of log event `<eventID>`
Valid Values:

`alarm`	The log level of log event `<eventID>` is set to `Alarm`.
`standard`	The log level of log event `<eventID>` is set to `Standard`.
`interaction`	The log level of log event `<eventID>` is set to `Interaction`.
`trace`	The log level of log event `<eventID>` is set to `Trace`.
`debug`	The log level of log event `<eventID>` is set to `Debug`.
`none`	Log event `<eventID>` is not recorded in a log.

Changes Take Effect: Immediately

Specifies a log level for log event `<eventID>` that is different than its default level, or disables log event `<eventID>` completely. If no value is specified, the

log event retains its default level. This option is useful when you want to customize the log level for selected log events.

These options can be deactivated with the option `level-reassign-disable`.

Warning! Use caution when making these changes in a production environment.

Depending on the log configuration, changing the log level to a higher priority may cause the log event to be logged more often or to a greater number of outputs. This could affect system performance.

Likewise, changing the log level to a lower priority may cause the log event to be not logged at all, or to be not logged to specific outputs, thereby losing important information. The same applies to any alarms associated with that log event.

In addition to the preceding warning, take note of the following:

- Logs can be customized only by release 7.6 or later applications.
- When the log level of a log event is changed to any level except `none`, it is subject to the other settings in the `[log]` section at its new level. If set to `none`, it is not logged and is therefore not subject to any log configuration.
- Using this feature to change the log level of a log changes only its priority; it does not change how that log is treated by the system. For example, increasing the priority of a log to `Alarm` level does not mean that an alarm will be associated with it.
- Each application in a High Availability (HA) pair can define its own unique set of log customizations, but the two sets are not synchronized with each other. This can result in different log behavior depending on which application is currently in primary mode.
- This feature is not the same as a similar feature in Universal Routing Server (URS) release 7.2 or later. In this Framework feature, the priority of log events are customized. In the URS feature, the priority of debug messages only are customized. Refer to the *Universal Routing Reference Manual* for more information about the URS feature.
- You cannot customize any log event that is not in the unified log record format. Log events of the `Alarm`, `Standard`, `Interaction`, and `Trace` levels feature the same unified log record format.

Example

This is an example of using customized log level settings, subject to the following log configuration:

```
[log]
verbose=interaction
all=stderr
```

```
interaction=log_file
standard=network
```

Before the log levels of the log are changed:

- Log event 1020, with default level `standard`, is output to `stderr` and `log_file`, and sent to Message Server.
- Log event 2020, with default level `standard`, is output to `stderr` and `log_file`, and sent to Message Server.
- Log event 3020, with default level `trace`, is output to `stderr`.
- Log event 4020, with default level `debug`, is output to `stderr`.

Extended log configuration section:

```
[log-extended]
level-reassign-1020=none
level-reassign-2020=interaction
level-reassign-3020=interaction
level-reassign-4020=standard
```

After the log levels are changed:

- Log event 1020 is disabled and not logged.
- Log event 2020 is output to `stderr` and `log_file`.
- Log event 3020 is output to `stderr` and `log_file`.
- Log event 4020 is output to `stderr` and `log_file`, and sent to Message Server.

level-reassign-disable

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

When this option is set to `true`, the original (default) log level of all log events in the `[log-extended]` section are restored. This option is useful when you want to use the default levels, but not delete the customization statements.

log-filter Section

The `log-filter` section contains configuration options used to define the default treatment of filtering data in log output. This section contains one configuration option, `default-filter-type`. Refer to the chapter "Hide Selected Data in Logs" in the *Genesys 8.x Security Deployment Guide* for complete information about this option.

log-filter-data Section

The `log-filter-data` section contains configuration options used to define the treatment of filtering data in log output on a key-by-key basis. This section contains one configuration option in the form of `<key name>`. Refer to the chapter "Hide Selected Data in Logs" in the *Genesys 8.0 Security Deployment Guide* for complete information about this option.

security Section

The `security` section contains configuration options used to specify security elements for your system. In addition to other options that may be required by your application, this section contains the configuration option `disable-rbac`, which is used to enable or disable Role-Based Access Control for an application. Refer to the chapter "Role-Based Access Control" in the *Genesys 8.x Security Deployment Guide* for complete information about this option.

sml Section

This section must be called `sml`.

Options in this section are defined in the `Annex` of the `Application` object, as follows:

- in Genesys Administrator—`Application` object > `Options` tab > `Advanced View (Annex)`
- in Configuration Manager— `Application` object > `Properties` dialog box > `Annex` tab

Warning! Use the first three options in this section (`heartbeat-period`, `heartbeat-period-thread-class-<n>`, and `hangup-restart`) with great care, and only with those applications of which support for this functionality has been announced. Failure to use these options properly could result in unexpected behavior, from ignoring the options to an unexpected restart of the application.

heartbeat-period

Default Value: None

Valid Values:

`0`	This method of detecting an unresponsive application is not used by this application.
`3-604800`	Length of timeout, in seconds; equivalent to 3 seconds–7 days.

Changes Take Effect: Immediately

Specifies the maximum amount of time, in seconds, in which heartbeat messages are expected from an application. If Local Control Agent (LCA) does not receive a heartbeat message from the application within this period, it assumes the application is not responding and carries out corrective action.

This option can also be used to specify the maximum heartbeat interval for threads registered with class zero (0). This thread class is reserved for use by the Management Layer only.

If this option is not configured or is set to zero (`0`), heartbeat detection is not used by this application.

heartbeat-period-thread-class-<n>

Default Value: None
Valid Values:

`0`	Value specified by `heartbeat-period` in application is used.
`3-604800`	Length of timeout, in seconds; equivalent to 3 seconds–7 days.

Changes Take Effect: Immediately

Specifies the maximum amount of time, in seconds, in which heartbeat messages are expected from a thread of class <n> registered by an application. If a heartbeat message from the thread is not received within this period, the thread is assumed to be not responding, and therefore, the application is unable to provide service.

If this option is not configured or is set to zero (`0`), but the application has registered one or more threads of class <n>, the value specified by the value of `heartbeat-period` for the application will also be applied to these threads.

Refer to application-specific documentation to determine what thread classes, if any, are used.

hangup-restart

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

If set to `true` (the default), specifies that LCA is to restart the unresponsive application immediately, without any further interaction from Solution Control Server.

If set to `false`, specifies that LCA is only to generate a notification that the application has stopped responding.

suspending-wait-timeout

Default Value: `10`
Valid Values: `5-600`
Changes Take Effect: Immediately

Specifies a timeout (in seconds) after the Stop Graceful command is issued to an application during which the status of the application should change to

`Suspending` if the application supports graceful shutdown. If the status of the application does not change to `Suspending` before the timeout expires, it is assumed that the application does not support graceful shutdown, and it is stopped ungracefully.

Use this option if you are unsure whether the Application supports graceful shutdown.

Note: Genesys recommends that you do not set this option for any Management Layer component (Configuration Server, Message Server, Solution Control Server, or SNMP Master Agent) or any DB Server. These components by definition do not support graceful shutdown, so this option is not required.

common Section

This section must be called `common`.

enable-async-dns

Default Value: `off`
Valid Values:

`off`	Disables asynchronous processing of DNS requests.
`on`	Enables asynchronous processing of DNS requests.

Changes Take Effect: Immediately

Enables the asynchronous processing of DNS requests such as, for example, host-name resolution.

Warnings!
- Use this option only when requested by Genesys Technical Support.
- Use this option only with T-Servers.

rebind-delay

Default Value: `10`
Valid Values: `0`–`600`
Changes Take Effect: After restart

Specifies the delay, in seconds, between socket-bind operations that are being executed by the server. Use this option if the server has not been able to successfully occupy a configured port.

Warning! Use this option only when requested by Genesys Technical Support.

Changes from 8.0 to 8.1

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

Chapter

8 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 7, "Common Configuration Options," on page 131.

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 178.
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 82

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 155.)

propagated-call-type

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Switch Partitioning" on page 82

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 160.

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 36.

License Checkout

Table 13 shows how to determine the number of seat-related DN licenses that T-Server attempts to check out. See the examples on page 161.

Table 13: 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 13: 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 26 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 178.
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 178.
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 166
- Transfer Connect Service Options, page 170
- ISCC/COF Options, page 170
- Event Propagation Options, page 173
- Number Translation Option, page 174
- GVP Integration Option, page 174

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 178.

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 61 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 178.
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 178.
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 178.
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 178.
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 178.
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 178.
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 178.
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 178.
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 178.
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 events received from the service provider and containing a `Location` attribute before distributing them 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 the way ISCC handles events from, and requests to, an external service provider registered for a DN using the `AddressType` attribute set to `VSP`.

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 178.
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 178.
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.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 by terminating the call and 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 178.
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 178.
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.Specifies the time interval for periodic checks for stuck calls. These checks affect both notification and cleanup functionality. For performance reasons, T-Server does not verify whether the `notify-idle-tout` or `cleanup-idle-tout` option has expired before performing this checking.

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

Examples

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

Example 1

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

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

Example 2

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

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

Example 3

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

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

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 70. See "Configuring Number Translation" on page 77 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 14 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 14: 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 154.

Chapter

9 Configuration Options in Network T-Server for AT&T

This chapter describes configuration options unique to the Network T-Server for AT&T, and includes these sections:

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

Mandatory Options

This section lists the options that must be configured for basic T-Server operation. All other options in this chapter are configured to enable T-Server to support various features. See Table 15 for a list of mandatory options.

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 15: Mandatory Options

Option Name	Default Value	Details
crp Section		
def-label	No default value	Specifies the default label to route calls that were not processed by Interaction Router during the interval specified in the `Call Timeout` option in the `Timers` section. See description on page 186.
AsnSap Section		
type-table-file	`CRP.tt`	Specifies the relative or absolute path to the `ASN.1` type table file. See description on page 183.
codec-data-location-specified	false	Determines whether the data to be encoded/decoded is taken from the data buffer in the message, or from the property map in the message. See description on page 184.
encode-module-name	No default value	Determines whether the data to be encoded/decoded is taken from the data buffer in the message, or from the property map in the message. See description on page 184.
encode-type-name	No default value	Specifies encode instructions to the ASN SAP. See description on page 184.
decode-module-name	No default value	Specifies decode instructions to the ASN SAP. See description on page 184.
decode-type-name	No default value	Specifies decode instructions to the ASN SAP. See description on page 184.

Table 15: Mandatory Options (Continued)

Option Name	Default Value	Details
data-location	No default value	Specifies the location in the message property map to take/place the decoded/encoded ASN information. See description on page 185.
property-map-location	No default value	Specifies the location in the message property map to take/place the decoded/encoded ASN information. See description on page 185.
pgf Section		
ptc-file	tserver.ptc	Specifies the file name of the application file for this T-Server. See description on page 186.
OpenCall Section		
ss7Name	No default value	Stack related. Specifies the HP SS7 class name as declared in the HP SS7 `global.conf` file. See description on page 192.

AsnSap Section

This section must be labeled `AsnSap`.

type-table-file

Default Value: `CRP.tt`
Valid Value: Any valid relative or absolute path to the `CRP.tt` file
Changes Take Effect: After T-Server is restarted

Specifies the relative or absolute path to the `ASN.1 type` table file.

codec-data-location-specified

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

Determines whether the data to be encoded/decoded is taken from the data buffer in the message, or from the property map in the message. If set to `true`, `property-map-location` and `data-location` become required options.

Warning! In order for this T-Server to operate properly, the value must be set to `true`.

encode-module-name

Default Value: No default value
Valid Value: `CustomerRoutingPoint`
Changes Take Effect: After T-Server is restarted

Specifies encode instructions to the ASN SAP.

Warning! In order for this T-Server to operate properly, the value must be set to `CustomerRoutingPoint`.

encode-type-name

Default Value: No default value
Valid Value: `ICPRouteResponse`
Changes Take Effect: After T-Server is restarted

Specifies encode instructions to the ASN SAP.

Warning! In order for this T-Server to operate properly, the value must be set to `ICPRouteResponse`.

decode-module-name

Default Value: No default value
Valid Value: `CustomerRoutingPoint`
Changes Take Effect: After T-Server is restarted

Specifies decode instructions to the ASN SAP.

Warning! In order for this T-Server to operate properly, the value must be set to `CustomerRoutingPoint`.

decode-type-name

Default Value: No default value
Valid Value: `ICPRouteRequest`
Changes Take Effect: After T-Server is restarted

Specifies decode instructions to the ASN SAP.

Warning! In order for this T-Server to operate properly, the value must be set to `ICPRouteRequest`.

data-location

Default Value: No default value
Valid Value: `TCAP.ComponentList.Parameters`
Changes Take Effect: After T-Server is restarted

Specifies the location in the message property map to take/place the decoded/encoded ASN information. When encoding, place the encoded data in this location in the property map. When decoding, take the data to be decoded from this location in the property map. This property is not used unless `codec-data-location-specified` is set to `true`, and then it becomes required.

Warning! In order for this T-Server to operate properly, the value must be set to `TCAP.ComponentList.Parameters`.

property-map-location

Default Value: No default value
Valid Value: `ASN`
Changes Take Effect: After T-Server is restarted

Specifies the location in the message property map to take/place the decoded/encoded ASN information. When encoding, take the properties to be encoded from this location in the property map. When decoding, place the decoded properties in this location in the property map. This option is not used unless `codec-data-location-specified` is set to `true`, and then it becomes required.

Warning! In order for this T-Server to operate properly, the value must be set to `ASN`.

pgf Section

This section must be labeled `pgf`.

ptc-file

Default Value: `tserver.ptc`
Valid Value: Any valid relative or absolute path to the `CRP.smx` file
Changes Take Effect: After T-Server is restarted

Specifies the file name of the application file for this T-Server.

Warning! This must be set to the path name of the `CRP.smx` file.

crp Section

This section must be labeled `crp`.

def-route-type

Default Value: `Destination`
Valid Values: `Destination, Announcement, PostFeature`
Changes Take Effect: Immediately

Specifies the default route type for calls not processed by Interaction Router during the interval specified in the `Call Timeout` option in the `Timers` section.

def-label

Default Value: No default value
Valid Value: Any valid AT&T routing label
Changes Take Effect: Immediately

Specifies the default label to route calls that were not processed by Interaction Router during the interval specified in the `Call Timeout` option in the `Timers` section.

use-purge-by-size

Default Value: `no`
Valid Values: `yes, no`
Changes Take Effect: Immediately

Specifies whether an unprocessed call is routed to the default destination when the number of unprocessed calls exceeds the value specified in the `max-queue-size` option.

max-queue-size

Default Value: `1000`
Valid Value: Any positive integer greater than or equal to `1000`
Changes Take Effect: Immediately

Specifies the number of calls that can be queued. After this limit is reached, unprocessed calls are routed to the default destination if the `use-purge-by-size` option is set to `yes`.

min-dialed-len-national

Default Value: `10`
Valid Value: Any integer from `0`–`255`
Changes Take Effect: Immediately

Specifies the minimum number of digits for a nationally dialed number. If the number of digits dialed is less than specified, T-Server rejects the routing request.

max-dialed-len-national

Default Value: `11`
Valid Value: Any integer from `0`–`255`
Changes Take Effect: Immediately

Specifies the maximum number of digits for a nationally dialed number. If the number of digits dialed is more than specified, T-Server rejects the routing request.

min-dialed-len-international

Default Value: `10`
Valid Value: Any integer from `0`–`255`
Changes Take Effect: Immediately

Specifies the minimum number of digits for an internationally dialed number. If the number of digits dialed is less than specified, T-Server rejects the routing request.

max-dialed-len-international

Default Value: `11`
Valid Value: Any integer from `0`–`255`
Changes Take Effect: Immediately

Specifies the maximum number of digits for an internationally dialed number. If the number of digits dialed is more than specified, T-Server rejects the routing request.

min-ani-len-national

Default Value: 10
Valid Value: Any integer from 0–255
Changes Take Effect: Immediately

Specifies the minimum number of digits for an ANI in the national format. If the number of digits dialed is less than specified, T-Server rejects the routing request.

max-ani-len-national

Default Value: 10
Valid Value: Any integer from 0–255
Changes Take Effect: Immediately

Specifies the maximum number of digits for an ANI in the national format. If the number of digits dialed is more than specified, T-Server rejects the routing request.

min-ani-len-international

Default Value: 3
Valid Value: Any integer from 0–255
Changes Take Effect: Immediately

Specifies the minimum number of digits for an ANI in the international format. If the number of digits dialed is less than specified, T-Server rejects the routing request.

max-ani-len-international

Default Value: 15
Valid Value: Any integer from 0–255
Changes Take Effect: Immediately

Specifies the maximum number of digits for an ANI in the international format. If the number of digits dialed is more than specified, T-Server rejects the routing request.

min-ced-len

Default Value: 0
Valid Value: Any integer from 0–255
Changes Take Effect: Immediately

Specifies the minimum number of digits for a CED. If the number of digits dialed is less than specified, T-Server rejects the routing request.

max-ced-len

Default Value: `30`
Valid Value: Any integer from `0`–`255`
Changes Take Effect: Immediately

Specifies the maximum number of digits for a CED. If the number of digits dialed is more than specified, T-Server rejects the routing request.

Timers Section

This section must be labeled `Timers`.

Valid Units of Time

The default unit of time for all timers is milliseconds (`ms`). If you do not supply a unit of time, then `ms` is assumed. Valid unit of time are:

`ms` = milliseconds

`s` = seconds

`min` = minutes

`h` = hours

`d` = days

Call Timeout

Default Value: `750 ms`
Valid Value: Any positive integer and valid unit of time.
Changes Take Effect: Immediately

Specifies the amount of time that a call can exist before being default routed.

DecSs7 Section

This section must be labeled `DecSs7`.

tcap-allow-multi-comp

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

Specifies whether the SAP will allow more than one component to be placed in a package.

Warning! In order for this T-Server to operate properly, the value must be set to `false`.

NMS Section

This section must be labeled `NMS`.

serviceName

Default Value: `TCAP`
Valid Value: <string>
Changes Take Effect: After T-Server is restarted

The name of the Natural Access service under which the application will run.

serviceManager

Default Value: `TCAPMGR`
Valid Value: <string>
Changes Take Effect: After T-Server is restarted

The name of the Natural Access service manager that will be used.

paramFlags

Default Value: `shared`
Valid Values: `shared, local`
Changes Take Effect: After T-Server is restarted

Specifies whether global default parameters are stored in shared memory or in the process local memory.

traceFlags

Default Value: `enable`
Valid Values: `enable, disable, on error`
Changes Take Effect: After T-Server is restarted

Applies only if `paramFlags` is not `local`. If not disabled, trace information is sent to the ctdaemon.

boardNum

Default Value: `1`
Valid Value: <integer>
Changes Take Effect: After T-Server is restarted

The number of the TCAP board.

appid

Default Value: `32`
Valid Value: Any positive integer from `32–63`
Changes Take Effect: After T-Server is restarted

The application (entity) ID.

sapid

Default Value: `0`
Valid Value: <integer>
Changes Take Effect: After T-Server is restarted

The Natural Access service access point ID defined for the application.

ssn

Default Value: `254`
Valid Value: Any integer from `0`–`254`
Changes Take Effect: After T-Server is restarted

The subsystem number for the application.

ansi_param_container

Default Value: `sequence`
Valid Values: `sequence, set`
Changes Take Effect: Immediately

Specifies the container type that parameters will be placed in. This is required ANSI.

switchover

Default Value: `10`
Valid Value: <integer>
Changes Take Effect: Immediately

Specifies the time interval (in milliseconds) used to check for TCAP messages received by the stack.

tcap-allow-multi-comp

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

Specifies whether the SAP will allow more than one component to be placed in a package.

Warning! In order for this T-Server to operate properly, the value must be set to `false`.

OpenCall Section

This section must be labeled `OpenCall`.

ssn

Default Value: `0`
Valid Value: Any integer from `0`–`254`
Changes Take Effect: On connection open

Stack related. Specifies the SS7 subsystem number that T-Server services. You must specify a value for this option.

ss7Name

Default Value: Mandatory field. No default value.
Valid Value: <string> (no longer than 20 alphanumeric characters)
Changes Take Effect: On connection open

Stack related. Specifies the HP SS7 class name as declared in the HP SS7 `global.conf` file (see the *HP OpenCall SS7 Platform Management Guide* for details). You must specify a value for this option.

sccpServiceType

Default Value: `Regular`
Valid Values: `White, Ansi, Regular`
Changes Take Effect: On connection open

Stack related. Determines the type of SCCP layer being used. Should be set to `Regular`.

appId

Default Value: `1`
Valid Value: Any integer from `0`–`255`
Changes Take Effect: On connection open

Application related. Identifies the user application. Load sharing is possible by setting up several connections with the same `appId` (see the *HP OpenCall SS7 Platform Management Guide* for details). You must specify a value for this option.

instId

Default Value: `1`
Valid Value: Any positive integer from `0`–`255`
Changes Take Effect: On connection open

Specifies the number to differentiate between T-Servers having the same ID as specified in the `appId` option (see the *HP OpenCall SS7 Platform Management Guide* for details). You must specify a value for this option.

openTimeout

Default Value: `10000`
Valid Value: Any positive integer
Changes Take Effect: On connection open

Specifies (in milliseconds) the timeout after which an `open` attempt fails.

reconnectTimeout

Default Value: `5000`
Valid Value: Any positive integer
Changes Take Effect: On connection open

Specifies (in milliseconds) the timeout after which an attempt is made to re-open a failed connection.

tcap-allow-multi-comp

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

Specifies whether the SAP will allow more than one component to be placed in a package.

Warning! In order for this T-Server to operate properly, the value must be set to `false`.

Changes from 8.0 to 8.1

No configuration options have been changed between the 8.0 and 8.1 releases of Network T-Server for AT&T.

Appendix

Installing and Configuring NMS SS7, TX3220 and TX4000

This document contains the following sections:

Note: This appendix provides instructions for the Windows 2000 Advanced Server SP3 and Solaris 8 (64-bit) platforms. In most cases, the instructions are the same for both platforms. When an instruction differs by platform, the instruction begins "On Windows" or "On Solaris."

Installing and Configuring TX3220

Procedure: Installing and configuring TX3220

Purpose: This section provides detailed information for installing and configuring SS7 and the TX3220 board.

Start of procedure

1. Download NMS Software. Contact NMS Communications (www.nmscommunications.com) to obtain the latest versions of the SS7 and Natural Access software supported by the Network T-Server. Information on supported versions can be found in the Network T-Server for AT&T Release Note.
2. Installing NMS Software.
   a. Genesys recommends that you install the SS7 and Natural Access software on the same machine as your Network T-Server.
   b. Review the SS7 and Natural Access readme files to verify that you have downloaded the correct software for your operating system and that your Network T-Server machine meets the NMS hardware and memory requirements.
   c. To install the software, you must log on with Administrator privileges on Windows or as Superuser on Solaris. You do not need Administrator or Superuser privileges to use the software; these privileges are needed only to install SS7 and Natural Access.
   d. Install the SS7 software, and reboot when prompted.
      - On Windows, the installation software adds product directories to the `%PATH%` variable, so SS7 utilities can be invoked from any directory.
      - On Solaris, you need to set the following environmental variables for the user account which is used to run Genesys products:

      The `LD_LIBRARY_PATH` variable for this account should include the directory `/opt/nms/lib:/opt/nmstx/lib`

      The `PATH` variable should include the directory `/opt/nms/bin:/opt/nmstx/bin`

      Note: The way you set up these variables depends on the shell you use. See the `readme_ss7.txt` file (which downloads with the SS7 installation package) for examples and other installation information.

   e. Install the Natural Access software, and reboot when prompted.

      Note: Genesys has tested the Network T-Server for AT&T on Windows 2000 Advanced Server SP3 and on Solaris 8 (64-bit).

3. Install the NMS TX3220 board in a spare PCI slot on the machine where you just installed SS7 and Natural Access.

Note: Do not use a shared IRQ PCI slot to install the TX3220. Refer to your machine's documentation to see where this slot and any unused slots are located.

4. When you power up after installing the TX3220:
   - On Windows, the operating system should detect a new hardware device (called PCI Bridge). The operating system should then install a driver from the previously installed SS7 package automatically. As part of this installation, the operating system updates the system `%PATH%` variable so that the SS7 utilities become available from any working directory.
   - On Solaris, no additional installation steps are required. However, you can use the `dmesg` command to check your system startup log to see if the new board was detected.

Warning! Do not connect any cables to the TX3220 board at this time. You must download the TX3220 firmware and configure the board before connecting the cables.

5. Upload NMS Firmware to the TX3220
   a. Edit the `ss7load` file to enable SCCP and TCAP.
   b. From a command prompt (in any directory), type `ss7load` and press `Enter`. This command downloads pre-configured firmware into the default board (Board 1). On Solaris, the output of `ss7load` should be similar to the example shown in Figure 11:

```
$ ./ss7load board(1) statistics cleared

TX32XX FLASH Interface Utility V3.0
Copyright 1997-2001, NMS Communications

CP number 1 booted.
Loading: diag       Version 1.0 07/15/02
Loading: inf        Version 4.0 07/15/02
Loading: mvip       Version 2.0 07/15/02
Loading: t1e1mgr    Version 2.0 07/15/02
Loading: arp        Version 2.0 07/15/02
Loading: mtp        Version 7.1.13 03/06/03
Loading: sccp       Version 6.1.2 03/21/03
Loading: tcap       Version 4.1 07/15/02
```

Figure 11: Output from ss7load Command

c. Open a new command window (on Windows) or terminal (on Solaris), and type `txalarm` and press `Enter`. This starts the `txalarm` utility, which prints out the stack diagnostic (trace) information.

d. Issue the `ss7load` command again, and then check the output of the `txalarm` utility.This utility reports any errors and the versions of the firmware that has been loaded. The set of packages and the versions shown in your `txalarm` window may differ from the sample output in Figure 11, but you should not see any errors in the `txalarm` window. The first time you run `ss7load` after updating the firmware, you may get a message from `ss7load` that the card flash memory was updated.

Note: You should check the output of `txalarm` for errors each time you run `ss7load`.

6. Verify Firmware and SS7 Installation

a. Verify that SS7 and the TX3220 firmware have been installed correctly and are running on your system:

— On Windows:

i. Locate the `NMS` applet in the `Services` folder. Select `Start->Administrative Tools->Component Services`. In the `Component Services` window, select the `Services (Local)` folder.

ii. Double-click `NMS` in the `Services (Local)` folder to run the `NMS` applet. Select `TX3220` when prompted.

iii. After the `NMS` applet runs, check the statistics it generates to verify that the TX3220 is running without errors. Figure 12 shows an example of the statistics.

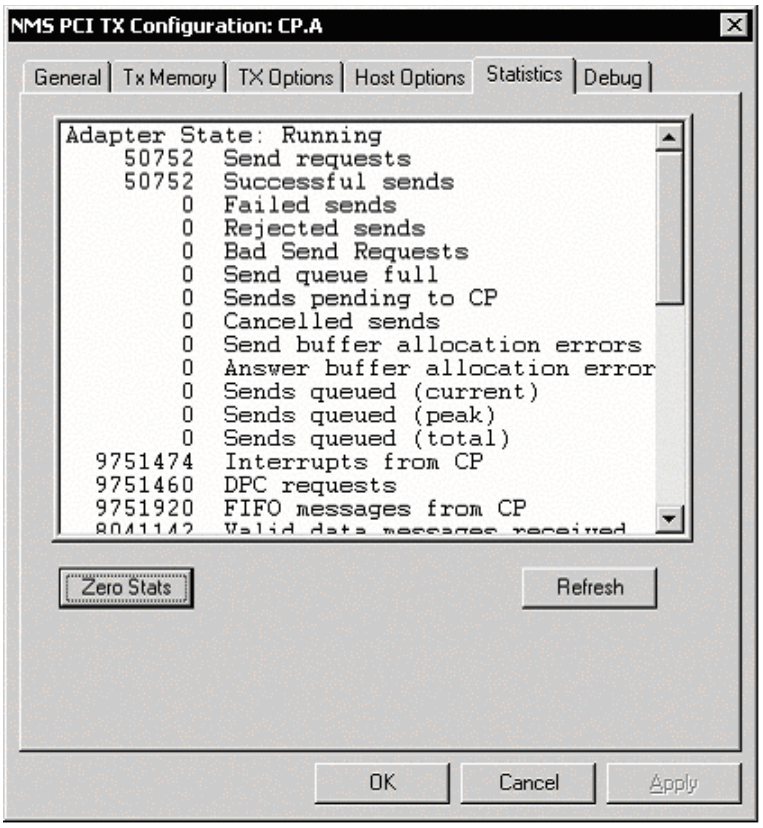

Figure 12: TX3220 Statistics—Windows Platform

— On Solaris:

i. At a shell prompt, run the shell command `cputil -b 1 -s`

ii. Check the output of this command (on the shell prompt screen) to verify that the TX3220 adapter is running without errors. Figure 13 shows an example of the statistics.

```
TX3220 board(1) statistics:

SENDING DATA (from host to TX):
Unacked Send Msgs_____________[0]
SEND REQUESTs from apps______[1122]
SEND REQUESTs posted to TX___[1122]
SEND REPLYs from TX__________[1122]
OK ACKs to app_______________[1122]
ERROR ACKs (flow controlled)_[0]
ERROR ACKs (bad state)_______[0]
ERROR ACKs (write failed)____[0]
Postponed (TX msg limit)_____[0]
No TX trans index____________[0]
Non-data requests from apps__[3980]

RECEIVING DATA (from TX to host):
SEND REQUESTs from TX________[73778]
SEND REQUESTs (unbound chan)_[51223]
SEND REQUESTs (have chan)____[22555]
SEND REQUESTs (payload)______[33]
Data indicators to app_______[22555]
Data payloads to app_________[33]
SEND REPLYs to TX____________[22555]
SEND REPLYs drop (bad state)_[0]
SEND REPLYs (app closed)_____[0]
SEND REPLYs (No RX wait)_____[0]

ERROR CONDITIONS:
SEND ERROR ACKs to app_______[2]
Out of State errors__________[0]
Out of Memory errors_________[0]
Anomaly errors_______________[0]
Empty free message FIFO______[0]

PCI INTERFACE:
Total Interrupts_____________[75984]
InPostListCount______________[0]
OutPostListStatus____________[0]
OutPostListMask______________[0]
OutPostListCount_____________[0]
```

Figure 13: TX3220 Statistics—Solaris Platform

b. On Windows or Solaris, from a command prompt, run the command `txalarm`. This utility prints out the stack diagnostic (trace) information. The utility also reports firmware upload status immediately after the `ss7load` command is run.

Note: You should check the output of `txalarm` for errors each time you run the `NMS` applet, the `cputil` command, or the `ss7load` command.

7. Upgrade TX3220 Firmware. NMS Communications Technical Support periodically posts newer versions of the TX3220 firmware to its website to resolve specific problems. You may need to update the TX3220 firmware.
   a. Locate the firmware files on your system (the path depends on your operating system). These are binary files with extension `*.lot`.
   b. Back up all the existing `*.lot` files.
   c. Check the NMS Communications website `http://www.nmscommunications.com` for updated firmware.

Note: At the time of this release, you must install the patches listed in Figure 14 for the SS7 stack to work on the AT&T network.

```
$ ./ss7load board(1) statistics cleared

TX32XX FLASH Interface Utility V3.0
Copyright 1997-2001, NMS Communications

CP number 1 booted.
Loading: diag       Version 1.0 07/15/02
Loading: inf        Version 4.0 07/15/02
Loading: mvip       Version 2.0 07/15/02
Loading: t1e1mgr    Version 2.0 07/15/02
Loading: arp        Version 2.0 07/15/02
Loading: mtp        Version 7.1.13 03/06/03
Loading: sccp       Version 6.1.2 03/21/03
Loading: tcap       Version 4.1 07/15/02
```

Figure 14: TX3220 Firmware Patches

   d. Copy the updated firmware files from the website to the same location on your system as the existing files.

Note: You may need to contact NMS Technical Support to get these versions of the firmware files (`.lots`). The firmware is identical for Windows and Solaris.

   e. After you update the firmware files, issue the `ss7load` command from a command prompt, and then issue the `txalarm` command.
   f. Review the output of `txalarm`, to verify the updated version information and check for error messages.

8. Modify the `ss7load` Batch File
   a. Locate the `ss7load` file on your system:
      - On Windows, the `ss7load.bat` file is located in the `\nms\tx\bin` directory.

- On Solaris, the `ss7load` shell script is located in the `/opt/nmstx/bin/` directory.

b. Within the Network T-Server installation directory, create a subdirectory called `ss7config`. (Although the `ss7config` subdirectory name is used throughout the instructions in this appendix, you could use a different subdirectory name on your system.)

c. Copy the `ss7load` file into the `ss7config` subdirectory, and open the file in a text editor.

d. Edit the `ss7load` file to enable the SCCP and TCAP subsystems. Do not load the ISUP and TUP subsystems. Comment out the lines that refer to `ISUPcp%BRD%.cfg` and `TUPcp%BRD%.cfg`. On Windows, the standalone configuration (`:stdalncfg`) settings should be similar to the settings shown in Figure 15.

```
:stdalncfg
rem
rem  Load standalone configuration files - note that each board in a
standalone
rem  configuration gets a configuration file unique to that board
rem
@echo [%TIME%] Loading : %TXUTIL%\sccpcfg -b %BRD% -f
%TXCONFIG%\SCCPcp%BRD%.cfg
%TXUTIL%\sccpcfg -b %BRD% -f %TXCONFIG%\SCCPcp%BRD%.cfg
```

Figure 15: Standalone Configuration (:stdalncfg) Settings

e. The `ss7load` file refers to several firmware files (`.lot`) and SS7 configuration files (`.cfg`) located in the NMS installation directory. Copy these files to the `ss7config` subdirectory.

f. Modify the `ss7load` script variable `TXCONFIG` to point to the copies of the SS7 configuration files in the `ss7config` subdirectory.

g. While it is not required, you might want to configure your system so the `ss7load` script starts each time your operating system starts.

- On Windows, you can either create a shortcut to run the `ss7load` upload script and add the shortcut to the `Run->Program->Startup` folder, or add the `ss7load` command as the first line of the batch file that starts the Network T-Server. (See Step 14 on page 208 for information about creating a batch file to start the Network T-Server.)
- On Solaris, add `ss7load` to the beginning of the shell script that runs the Network T-Server. (See Step 14 on page 208 for information about creating a shell script to start the Network T-Server.)

9. Configure the TX3220, SS7, MTP2, and MTP3 files. The TX 3220 board and SS7 stack parameters are scattered across several text-based configuration files. These configuration files are compiled to binary form and then uploaded to the TX3220 board using the `ss7load` command.

   You need to update the following configuration files:

   - MTP2cp1.cfg
   - MTP3cp1.cfg
   - SCCPcp1.cfg
   - TCAPcp1.cfg

   a. Locate these configuration files on your system. In a standard installation, the files are located in the following directories:
   - **On Windows:** `\nms\tx\config\standalone\ansi`
   - **On Solaris:** `/opt/nmstx/etc/standalone/ansi/`

   b. Copy the configuration files into the `ss7config` directory, and open the files in a text editor.

   c. Before updating the SS7 configuration file, collect the following information:
   - Point codes assigned to the Genesys Network Routing Solution (NRS) platform
   - Point codes assigned to AT&T signaling transfer points (STPs)
   - Link configuration—how the NRS platform is connected to the STPs
   - Routing information—the rules that define how messages should be sent across the links

   d. Open the SS7 configuration file `sccpcp1par.cfg` in a text editor, and `Save As` the file with a name that follows your company's naming convention. Then enter the data you collected in the previous step.

   e. Before updating an MTP2 or MTP3 configuration file, collect the following information:
   - Whether the board should be configured as DTE or DCE
   - Physical interface type (V.35 Serial or T1/E1)
   - Baud rate
   - SLCs (SS7 service link codes) assigned to the links

   f. Open the MTP2 or MTP3 configuration file (`mtp2cp1par.cfg` or `mtp3cp1par.cfg`) in a text editor, and `Save As` the file with a name that follows your company's naming convention. Then enter the data you collected in the previous step.

   Note: See the sample configuration files in on "Sample Configuration Files" on page 208 for more configuration information.

g. After you make changes to the configuration files, run the `ss7load` command to update the configuration. After `ss7load` runs, check the output of `txalarm` for errors. The most common errors are misspelled options and incompatible settings. For a list of available SS7 options, refer to the NMS Communications document *SS7 Installation and Configuration Manual*.

Note: Make sure the `txalarm` utility is running before you issue the `ss7load` command.

h. If you use the DCE hardware configuration, you will need to open the V.35 pod connected to the NMS board and manually change the jumper configuration. V.35 pod configuration is described in the NMS Communications document *TX3220 Installation Manual*.

10. Modify the Network T-Server for AT&T Configuration.
    a. Open the Genesys Configuration Manager.
    b. From the `Configuration Manager` main menu, select `View > Options`.
    c. In the `Environment` directory, select `Applications`, and then double-click your Network T-Server application to open its Properties dialog.
    d. From the `Properties` dialog, select the `Options` tab.
    e. On the `Options` tab, double-click the `NMS` object.
    f. Change (or create) the following options:
       - `ssn=254` (This is the subsystem number. You must use `254` for this value.)
       - `sapid=0` (This value corresponds to the `USER_SAP` parameter as defined in both the `sccpcp1.cfg` and `tcapcp1.cfg SS7` configuration files. The value may be different on your system.)

Note: The `ssn` and `sapid` should match the values defined in the NMS configuration files. If you use the template files provided with Network T-Server, these values are already configured. See Step 9 on page 203 for information about configuring these files. Also see the configuration samples in "Sample Configuration Files" on page 208.

    g. Create a `Switching Office` object for the AT&T switch.
    h. Define a `switch` object for the switch that is connected to the Network T-Server for AT&T. The properties for this switch should correspond to the properties you just defined for the AT&T `Switching Office` object.
    i. Define Service Numbers (DNs) for this switch. The DN values are assigned by the provider (AT&T).

Note: See "Chapter 5, "AT&T–Specific Configuration," on page 115 and the *Framework 8.1 Deployment Guide* for information on how to complete these configuration steps (and other necessary configuration) for the Network T-Server for AT&T in the Configuration Layer.

11. Start the Network T-Server for AT&T.
    a. Use a batch file (Windows) or a script (Solaris) to start the Genesys Network T-Server for AT&T. Do not run it as a service on Windows. You may want to add the `ss7load` command to the beginning of the batch file or script to ensure that you are using the current configuration for the SS7 stack. You also may need to specify a Genesys license file for the Network T-Server.
    b. In Configuration Manager, open the Network T-Server's `Properties` dialog box and select the `Options` tab. Turn on the maximum verbosity level and choose `log output to the console` and `log output to a file`.
    c. Check the output of the `txalarm` command immediately after you start the Network T-Server. The `txalarm` output should report that the subsystem number (SSN) corresponding to the Network T-Server is in service. Because the TX3220 cables are still disconnected at this point, you also should see link alignment attempts (SS7 SIO/SIOS messages) in the `txalarm` output about every 10 seconds. If the MTP2 and MTP3 layers are aligned, you should get a `txalarm` message similar to Figure 16.

```
<06/20/2003 14:30:36> sccp      1  18673 DEBUG: SCCP - Local
Broadcast Msg 4
<06/20/2003 14:30:36> sccp      1  18434 SCCP User Sap 0 [SSN 9]
In Service
```

Figure 16: txalarm Message

12. Connect TX3220 Cables
    a. Configure and provision the T1 lines or V.35 cables before you connect them to the TX3220 board.
    b. Connect the T1 lines or V.35 cables to your TX3220 board. A T1 line is connected directly to the on-board RJ-45 connector. Use a Quad V.35 pod to connect V.35 cables.
13. This step contains troubleshooting tips that you can use before connecting the TX3220 cables but after the entire system has been connected and configured.
    a. When the TX3220 cables are disconnected, you should be able to see link alignment attempts (SS7 SIO/SIOS messages) in the `txalarm` output every several seconds, as shown in Figure 17.

```
<06/20/2003 14:40:38> mtp       1  18180 MTP3 Link 1 Down
<06/20/2003 14:40:38> mtp       1      1 Flushing Buffers
(OutPktCnt=0) (Lnk 1)
<06/20/2003 14:40:38> mtp       1      1 Starting Alignment
(Lnk 1)
<06/20/2003 14:40:38> mtp       1      1 Sent SIO (Lnk 1)
<06/20/2003 14:40:38> mtp       1      1 ALIGN TIMER 2 EXPIRED
iacSt=8 (Lnk 0)
<06/20/2003 14:40:38> mtp       1      1 LinkFailure (Lnk 0):
Alignment Not Possible
<06/20/2003 14:40:38> mtp       1      1 Sent SIOS (Lnk 0)
```

Figure 17: Link Alignment Attempts in txalarm Output

b. Once you have properly configured the SS7 stack and TX3220 hardware, soon after you connect the cables to the TX3220, the `txalarm` output should show that the links are aligned.

c. After the physical links have been connected and configured, with the Network T-Server running, issue the `ss7load` command at a command prompt.

d. Check the `txalarm` output for the following items. You should see log messages similar to those shown in Figure 18.

- The MTP3 links are aligned—you should see SS7 SIO, SLTM, and other messages.
- The Network T-Server SSN is in service.
- Requests from the AT&T network for configured Service Numbers are processed, and can be seen in the Network T-Server log.

```
<06/20/2003 14:36:54> mtp       1      1 Starting Alignment (Lnk 0)
<06/20/2003 14:36:54> mtp       1      1 Sent SIO (Lnk 0)
<06/20/2003 14:36:54> mtp       1      1 Flushing Buffers
(OutPktCnt=0) (Lnk 1)
<06/20/2003 14:36:54> mtp       1      1 Starting Alignment (Lnk 1)
<06/20/2003 14:36:54> mtp       1      1 Sent SIO (Lnk 1)
<06/20/2003 14:36:54> mtp       1      1 Rcvd SIO (Lnk 1)
<06/20/2003 14:36:54> mtp       1      1 Sent SIN (Lnk 1)
<06/20/2003 14:36:54> mtp       1      1 Rcvd SIOS (Lnk 0)
<06/20/2003 14:36:54> mtp       1      1 Rcvd SIO (Lnk 1)
<06/20/2003 14:36:54> mtp       1      1 Rcvd SIN (Lnk 1)
<06/20/2003 14:36:55> mtp       1      1 Rcvd SIO (Lnk 0)
<06/20/2003 14:36:55> mtp       1      1 Sent SIN (Lnk 0)
<06/20/2003 14:36:55> mtp       1      1 Rcvd SIO (Lnk 0)
<06/20/2003 14:36:55> mtp       1      1 Rcvd SIN (Lnk 0)
<06/20/2003 14:36:56> mtp       1      1 Timer 4 Expired (LINK 1
ALIGNED at layer 2)
<06/20/2003 14:36:56> mtp       1     1 Setting link 1 ACTIVE from
SLTA
```

Figure 18: Link Alignment Confirmations in txalarm Output

e. Correct any problems you find in the log.

By editing the SS7 configuration files, you can turn on additional trace information for MTP2 and MTP3. Use the SS7 management utilities described in the NMS Communications document *SS7 Installation and Configuration Manual* for troubleshooting.

If the links are not aligned or the SS7 messages are not being processed, use these troubleshooting checklists to pinpoint the problem:

- MTP2 Layer
  - — Is the DTE/DCE type set correctly?
  - — Are the baud rate and clock sources (Receive, Transmit, and External) set correctly?
  - — Are the DTR/DSR and CTS/RTS handshakes working?
- MTP3 Layer
  - — Is the node type set correctly? (It should be set to `SP`.)
  - — Is the SLC set as defined by AT&T?
  - — Are the point codes set as defined by AT&T?
- SCCP/TCAP

— Is the stack type properly configured? (The `LINK_TYPE` should be set to `ANSI`, and the `SWITCH_TYPE` should be set to `ITU92`.)

— Are the routing rules set for both UP and DOWN? (See the sample configurations in "Sample Configuration Files" on page 208.)

f. After correcting any problems you find in the `txalarm` output, run the `ss7load` command again, and check the `txalarm` output again.

14. Configure the SS7 Stack and Network T-Server for Production

a. Disable the trace mode in the SS7 configuration files.

b. Turn off debug logging for the Network T-Server in Configuration Manager.

c. Although it is not required, you might want to configure your system so the Network T-Server starts each time your operating system starts.

- On Windows, you can create a shortcut to run the Network T-Server, and move the shortcut to the `Start->Programs->Startup` folder so the Network T-Server will be started automatically at system startup.

  Instead of a shortcut for the Network T-Server application, you might want to create a batch file that starts the Network T-Server at system startup. If you choose this option, Genesys recommends that you make the `ss7load` command the first line in this batch file, before the Network T-Server startup command. You should move the batch file to the `Start->Programs->Startup` folder.

  As a third alternative, you can set up a Solution Control Server entry for the Network Routing Solution.
- On Solaris, create a shell script to run the Network T-Server. Genesys recommends that you make the `ss7load` command the first line in this shell script, before the Network T-Server startup command.

End of procedure

Sample Configuration Files

This section contains sample configuration files for a configuration with two V.35 links and with the TX3220 board configured as DTE. You can use these files as a reference; however, the configuration in these files is generic. You will need to replace several of the parameter settings with real values.

Parameters you need to modify are enclosed in angle brackets, like this: `<GENESYS_PC>`.

Sample MTP3 File (MTP3cp1.cfg)

You can use this MTP3 sample configuration file (`mtp3cp1.cfg`) as a template.

Note: Replace the sample values with values that apply to your environment.

```
#   $Id: mtp3cp1par.cfg,v 1.1 2003/05/03 02:49:53 kozin Exp $
#
#  Template MTP3 Configuration file for AT&T/Genesys NRS/NMS
#  SS7 TX3220 board
#
#    Written by Genesys/ATG
#
#    Replace the following parameters with real values:
#
#  <GENESYS_PC>::= Point Code assigned to NRS platform,
#  e.g. 231.10.4
#     <STP1_PC>::= Point Code assigned to the first ATT STP, e.g.
254.152.0
#     <STP2_PC>::= Point Code assigned to the second ATT STP, e.g.
254.153.0
#
#  These files only show the use of some basic configurable
#  parameters. Please refer to the MTP3 Developer's Guide
#  and the SS7 Installation and Configuration Guide for
#  information about all possible parameters.
#
#-------------------------------------------------
# Overall MTP3 Parameters
#-------------------------------------------------
NODE_TYPE          SP        # choose STP [routing] or SP
[non-routing]
PC_FORMAT          DFLT      # Point code format:  DFLT (8.8.8) /
INTL (3.8.3) / JNTT (7.4.5)
POINT_CODE         <GENESYS_PC>
RESTART_REQUIRED   TRUE
VALIDATE_SSF       TRUE
MAX_LINKS          4
MAX_USERS          1          # sccp only
MAX_ROUTES         64
MAX_ROUTE_ENTRIES  32
MAX_LINK_SETS      2
MAX_ROUTE_MASKS    2
ROUTE_MASK         0xFFFFFFFF
ROUTE_MASK         0xFFFF0000
MTP3_TRACE_DATA    TRUE     # for debugging only, set to FALSE in
production
END
#
```

```
#-----------------------------------------------
# Link Parameters
#-----------------------------------------------
# LINK 0 (V.35) configuration
#
LINK    0
PORT              S1        #  T<n> for T1/E1/MVIP, S<n> for serial
(V.35)
LINK_SET           1
LINK_TYPE          ANSI      #  ANSI / ITU / JNTT
ADJACENT_DPC       <STP1_PC>
LINK_SLC           0
LSSU_LEN           2
TIMER_T34   90#remove if in production
MAX_SLTM_RETRY     0
SSF                NATIONAL   # NATIONAL / INTERNATIONAL
INT_TYPE           DTE
BAUD               56000
END
#
# LINK 1 (V.35)
#
LINK    1
PORT              S2        #  T<n> for T1/E1/MVIP, S<n> for serial
(V.35)
LINK_SET           2
LINK_TYPE          ANSI      #  ANSI / ITU / JNTT
ADJACENT_DPC       <STP2_PC>
LINK_SLC           0
TIMER_T34   90#remove if in  production
MAX_SLTM_RETRY     0
LSSU_LEN           2
SSF                NATIONAL   # NATIONAL / INTERNATIONAL
INT_TYPE           DTE
BAUD               56000
END

#-----------------------------------------------
# User Parameters (NSAP definition)
#-----------------------------------------------
NSAP               0          # isup must be NSAP 0 if its present
LINK_TYPE          ANSI       #  ANSI / ITU / JNTT
END
#
#
#-----------------------------------------------
# Routing Parameters
#-----------------------------------------------
#
# Route UP from network to applications on this node
#
```

```
ROUTE        0
DPC          <GENESYS_PC>      # this node
SPTYPE   SP
LINK_TYPE   ANSI
DIRECTION    UP
END
#
# Route to STP 1
#
ROUTE              1
DPC          <STP1_PC>
SPTYPE   STP
LINK_TYPE   ANSI
DIRECTION    DOWN
END
#
# Route to STP 2
#
ROUTE         2
DPC         <STP2_PC>
SPTYPE   STP
LINK_TYPE   ANSI
DIRECTION     DOWN
END
#
# Route to AT&T Network
#
ROUTE      3
DPC          254.0.0      # AT&T Network 254.x.x
SPTYPE   SP
LINK_TYPE   ANSI
DIRECTION    DOWN
ADJACENT_ROUTE     FALSE
END
#
#
#-------------------------------------------------
# Linkset Parameters
#-------------------------------------------------
LINK_SET_DESCRIPTOR       1
ADJACENT_DPC              <STP1_PC>
MAX_ACTIVE_LINKS          1
ROUTE_NUMBER              1,0
ROUTE_NUMBER              2,1
ROUTE_NUMBER              3
END
#
LINK_SET_DESCRIPTOR       2
ADJACENT_DPC              <STP2_PC>
MAX_ACTIVE_LINKS          1
ROUTE_NUMBER              1,1
```

```
ROUTE_NUMBER              2,0
ROUTE_NUMBER              3
END
#
```

Sample SS7 File (SCCPcp1.cfg)

You can use this SS7 sample configuration file (`SCCPcp1.cfg`) as a template.

Note: Replace the sample values with values that apply to your environment.

```
#
#    $Id: sccpcp1par.cfg,v 1.1 2003/05/03 02:49:53 kozin Exp $
#
#  Template of SSCP config file for AT&T/ Genesys NRS / NMS SS7
TX3220
#
#    Written by Genesys/ATG
#
#  Parameters to modify:
#
#  <GENESYS_PC>::= Point Code of Genesys NRS platform, e.g. 231.10.4
#
#     <STP1_PC>::= Point Code of the first  ATT STP, e.g. 254.152.0
#     <STP2_PC>::= Point Code of the second ATT STP, e.g. 254.153.0
#

# General Configuration Section
MAX_USERS               4               # Max SCCP user applications
MAX_NSAPS               1               # Number of MTP3 interface
(max 1 per switch type)
MAX_SCLI               20              # Max simultaneous sequenced
connectionless
                                        # data xfers (Class 1 only)
MAX_ADDRS               0               # Max Address translation
entries
MAX_ROUTES              10              # Max far point codes SCCP
knows
DEF_ROUTING             TRUE            # Set Default Routing(FALSE
= OFF, TRUE = ON)
SAVE_CONNS              FALSE           # Drop connections on lost
link (FALSE) or don't drop (TRUE).
SCCP_ALARM_LEVEL        3               # Set alarm level reporting
                                       # (0=off, 1=default, 2=debug,
3=detail)
TRACE_DATA              TRUE           # Set data tracing (FALSE =
OFF, TRUE = ON)
MAX_ADJDPC              2               # Max far point codes
directly adjacent to us
```

```
MAX_MSGDRN              5              # Max msgs to send in a batch
when MTP comes up
                                         # (prevents flooding when
link(s) come up)
MAX_XUDT                1              # Number of control blocks to
allocate for
                                         # reassembling segmented
eXtended UnitDaTa
                                         # (ITU-92 only) ????
MAX_XUDTXREF            2               # Max number of local
references used to
                                         # segment eXtended UnitDaTa
????
MAX_CONN                512            # Max number of simultaneous
connections
CONN_THRESH             3               # Minimum number of SCCP
buffers that must be
                                       # available for new connection
to be accepted
QUEUE_THRESH            8              # Max number of buffers that
can be queued for
                                       # connection waiting for conn
window to open
SOG_THRESH              3               # Minimum number of SCCP
buffers that must be
                                         # available for SOR request
from replicated
                                         # (backup) subsystem to be
accepted

# Note all timer values are in seconds (0 disables timer)
XREFFRZ_TIMER           2               # wait before reusing local
reference
#ASMB_TIMER             0               # wait for all segments of
segmented XUDT (ITU-92 only)
FREEZE_TIMER            2               # wait before reusing
connection local reference
CONN_TIMER              180             # wait for response to
connection request
TXINACT_TIMER           600             # wait with no outgoing
packets on a connection
                                         # before issuing Inactivity
test (IT) message
RXINACT_TIMER           900             # wait with no incoming
packets on a connection
                                       # before releasing connection
(should be > TXINACT_TIMER)
REL_TIMER               10              # wait for response to
release request
#REPREL_TIMER           0               # wait for response to 2nd
release request (ITU only)
```

```
#INTERVAL_TIMER         0               # wait before reporting
abnormal release (ITU only)
GUARD_TIMER             2               # wait after MTP3 traffic
restart before
                                        # application traffic
RESET_TIMER            30              # wait for response to Reset
Request
#SCLI_TIMER             0               # max time sequenced
connectionless
                                       # transmission can take (class
1)
SST_TIMER              30               # time between subsystem
status tests
SRT_TIMER              30               # time between subsystem
routing tests
NSAP_TIMER             1                # time between bursts of
messages to MTP3 when
                                        # draining built up queue
(prevents congestion
                                        # when link comes back up)
IGNORE_TIMER           30               # delay after receiving SOG
before actually
                                        # going out of service
COORD_TIMER            30              # wait for grant to go out of
service (SOG)
                                        # after issuing SOR request
END

#
# User SAP configuration for 1st application
#
USER_SAP               0                # Sap number start at 0
SWITCH_TYPE            ANSI92           # one of ITU92, ITU88,
ANSI92, ANSI88
CONC_PC                <STP1_PC># Concerned point code
CONC_PC                <STP2_PC># Concerned point code
ADDR_MASK              FFF0000000      # match on only 1st 3 digits
of GT
MAX_HOPS               10               # maximum network hops
END                                     # User application 0

#
# User SAP configuration for 2nd application
#
USER_SAP               1                # Application 1
SWITCH_TYPE            ANSI92           # one of ITU92, ITU88,
ANSI92, ANSI88
CONC_PC                <STP1_PC># Concerned point code
CONC_PC                <STP2_PC># Concerned point code
ADDR_MASK              FFF0000000      # match only 1st 3 digits of
GT
```

```
MAX_HOPS                10              # maximum network hops
END                                     # User application 1

#
# Network (MTP3) Saps - one per switch type
#
NSAP                    0              # SCCP must be NSAP 1 if isup
present too
SWITCH_TYPE             ANSI            # one of ITU, ANSI
DPC                     <GENESYS_PC>    # REQUIRED - this nodes point
code
DPC_LEN                 24              # normally would'nt specify
this - let it
                                       # default based on switch type
MSG_LEN                 256             # MTU length on this network
TXQ_THRESH              20             # max packets queued to this
MTP3
ADDR_MASK               FFF0000000      # match 3 digits for global
title translation
                                        # of incoming packets
MAX_HOPS                10              # maximum network hops
END                                     # of ANSI MTP3 NSAP

#
# Routes: one route per node known to the SCCP layer
#
ROUTE                   <STP1_PC>       # destination point code
SWITCH_TYPE             ANSI            # one of ITU, ANSI
ADJACENT                TRUE           # this dest directly adjacent
#TRANSLATOR             FALSE          # not a translator node
END                                     # of route

ROUTE                   <STP2_PC>       # destination point code
SWITCH_TYPE             ANSI            # one of ITU, ANSI
ADJACENT                TRUE           # this dest directly adjacent
#TRANSLATOR             FALSE          # not a translator node
END                                     # of route
```

Sample TCAP File (TCAPcp1.cfg)

You can use this TCAP sample configuration file (`TCAPcp1.cfg`) as a template.

Note: Replace the sample values with values that apply to your environment.

```
#   $Id: tcapcp1par.cfg,v 1.1 2003/05/03 02:49:53 kozin Exp $
#
#    Template of TCAP configuration file for
#    AT&T / Genesys NRS / NMS SS7 / TX3220 board
#
```

```
#    Written by Genesys/ATG
#
#    Parameters to modify: None
#

# General Configuration Section
TCAP_ALARM_LEVEL        1               # standard alarms
MAX_TCAP_USERS          4               # Max TCAP user applications
MAX_TCAP_DIALOGS        200             # Max TCAP simultaneous
dialogs
MAX_TCAP_INVOKES        200             # Max TCAP simultaneous
invokes
MIN_TID_LEN4# Required by ATT
END

#
# User SAP configuration for 1st application
#
USER_SAP                0               # Sap number start at 0
SWITCH_TYPE             ITU92          # one of ITU92, ITU88,
ANSI92, ANSI88
INACTIVITY_TIMER        0
END                                     # User application 0

#
# User SAP configuration for 2nd application
#
USER_SAP                1               # Application 1
SWITCH_TYPE             ITU92          # one of ITU92, ITU88,
ANSI92, ANSI88
END                                     # User application 1
```

Installing and Configuring TX4000

Procedure:
Installing and configuring TX4000

Purpose: The following procedure outlines the steps required to configure the SS7 stack, install Natural Access, and configure the TX4000 PCI board. The procedure for installing the TX4000 board is similar to the TX3220 board. Differences are documented in the following procedure. Refer to the appropriate step in the TX3220 procedure where indicated.

Prerequisites

Download NMS Software (TX4000). Contact NMS Communications (www.nmscommunications.com) to obtain the latest versions of the SS7 and Natural Access software supported by the Network T-Server. Information on supported versions can be found in the Network T-Server for AT&T Release Note.

Start of procedure

1. Download the latest software from `http://www.nmscommunications.com`.
2. Install the TX4000 board in a spare PCI slot in your computer.

> **Note:** Do not use a shared IRQ PCI slot to install the TX4000. Refer to your machine's documentation to see where this slot and any unused slots are located.

> **Warning!** Do not connect any cables to the TX4000 board at this time. You must download the TX4000 firmware and configure the board before connecting the cables.

3. Install the Natural Access software, accept the default settings. Refer to the Natural Access *Installation Guide* for further details.
4. Review the installed software.
   - On Windows, select `Start->Programs->NMS->Natural Access->Version Utility`. The window appears similar to Figure 19.
   - On Solaris, At a shell prompt, run the shell command `cputil -b 1 -s`.

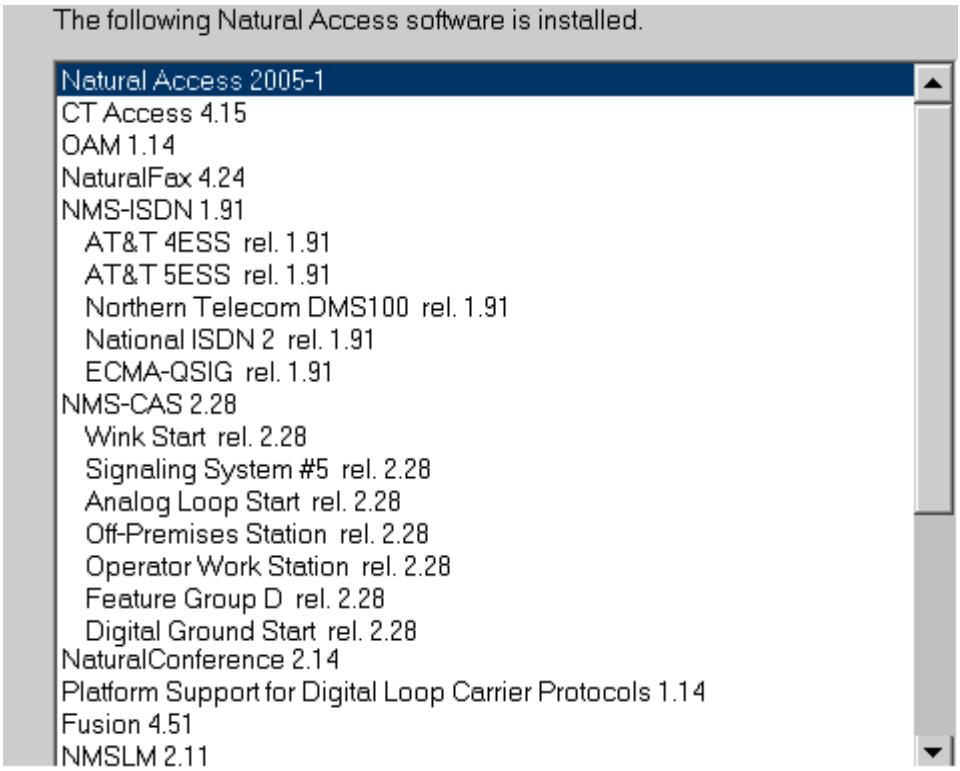

Figure 19: Natural Access Installed Software

5. Select the hardware type, choose either the `AG2000` or `AG4000` board.

> **Note:** The TX4000 board may not appear in the list of available hardware.

6. Specify the country where the software is deployed:
   - On Windows, select `United States` from the drop down list.
   - On Solaris, select `United States` from a menu.
7. Install the SS7 Software, accept the default settings, and reboot when prompted.

> **Note:** Refer to Step 2 on page 196, and the NMS *C*ommunications document *TX4000 Installation Manual* for further details. You will need to specify the specific Operating System type and version for your particular installation.

8. On Windows or Solaris open a command prompt. Assign a board number to the TX4000 board by using the command line utility `txcpcfg`. The following example shows how to complete this process on a Windows system:

Note: Samples are provided for reference only, Bus/slot numbers may be different for your particular installation.

a. Check the installed hardware, use the `txcpcfg` command.

```
C:\Documents and Settings\Administrator>txcpcfg

TX Configuration Utility V1.1
Copyright 1997-2004, NMS Communications

Bus    Slot    CP Number    CP Model
---    ----    ---------    --------
  0      13    UNDEFINED    TX 4000

Assign a new board number, by typing the bus, slot, and cp number.

C:\Documents and Settings\Administrator>txcpcfg 0 13 1

TX Configuration Utility V1.1
Copyright 1997-2004, NMS Communications

txcpcfg: configuring bus: 0, slot: 13, cpid: 1

Verify that the new setting has been accepted, use the txcpcfg
command.

C:\Documents and Settings\Administrator>txcpcfg

TX Configuration Utility V1.1
Copyright 1997-2004, NMS Communications

Bus    Slot    CP Number    CP Model
---    ----    ---------    --------
  0      13     1           TX 4000
```

9. Upgrade the TX4000 firmware. The latest version of the TX4000 firmware comes with the NMS SS7 Stack. To upgrade the firmware use the `ss7load.bat` batch file for a Windows system, or `ss7load.sh` for a Solaris system. The procedure is similar for both the TX3220 and TX4000 boards. Refer to the Step 7 on page 201 for further information.

Warning! Prior to running the batch file it is recommended that you modify the `ss7load` script file. Modify the script so the path to the configuration files points to a local directory and not the installation directory itself.

10. Modify the `ss7load` Batch File for TX4000.

    The files which must be modified and placed in the `ss7load` directory are described in Table 16.

Table 16: SS7load Files Requiring Modification

File Name	Description
TXcfg.txt	Defines low-level link configuration (T1 settings)
MTP3cp1.cfg	Defines MTP2 and MTP3 settings
SCCPcp1.cfg[a]	Defines SCCP settings
TCAPcp1.cfg[a]	Defines TCAP settings

a. This file is identical for both the TX3220 and TX4000 boards.

    Refer to Step 9 on page 203, and the NMS Communications *TX4000 Installation Manual* for detailed instructions on how to complete this task.

11. Configure TX4000 TXCFG and MTP3 files. Complete the tasks outlined in Step 9 on page 203 to modify the `SCCPcp1.cfg`, and `TCAPcp1.cfg` files.

12. Before updating the MTP3 configuration file, collect the following information:
    - Whether the board should be configured as DTE or DCE
    - Physical interface type
    - Baud rate
    - SLCs (SS7 service link codes) assigned to the links

13. Open the MTP3 configuration file (`mtp3cp1.cfg`) in a text editor, and `Save As` the file with a name that follows your company's naming convention. Then enter the data you collected in the previous step.

14. Open the TXCFG configuration file (`txcfg1.txt`) in a text editor, `Save As` the file with a name that follows your company's naming convention.

Note: Sample TX4000 MTP3 and TXCFG configuration files are located on page 221.

15. After you make changes to the configuration files, run the `ss7load` command to update the configuration. After `ss7load` runs, check the output of `txalarm` for errors. The most common errors are misspelled options and incompatible settings. For a list of available SS7 options, refer to the NMS Communications document *SS7 Installation and Configuration Manual*.

16. Complete the procedure to modify the configuration for Network T-Server for AT&T, see Step 10 on page 204.
17. Complete the procedure to start the Network T-Server for AT&T described in Step 11 on page 205.
18. Connect the TX4000 cables. Configure and provision the T1 lines or V.35 cables before you connect them to the TX4000 board.
19. Connect the V.35 cables to your TX4000 board. A T1 line is connected directly to the on-board RJ-45 connector. Use a Quad V.35 pod to connect V.35 cables.
20. Troubleshooting the TX4000 board. The troubleshooting procedures are the same for the TX3220, and TX4000 boards. The only difference is that the initialization messages for the TX4000 board displayed in the `txalarm` console are related to the T1 interface for the TX4000, and not the V.35 interface. Refer to Step 13 on page 205 for more details on troubleshooting procedures.
21. Complete the TX4000 installation and start the Network T-Server for AT&T by following the procedure outlined in Step 14 on page 208.

End of procedure

Sample TX 4000 Configuration Files

This section contains sample configuration files for the T1 configuration of the TX4000 board. You can use these files as a reference; however, the configuration in these files is generic. You will need to replace several of the parameter settings with real values.

Parameters you need to modify are enclosed in angle brackets, like this: `<portnum>`.

Note: The TCAPcp1.cfg, and SCCPcp1.cfg files are the same for the TX-3220 and TX-4000 boards. Refer to the section labelled “Sample Configuration Files” on page 208 for examples of these files.

Sample TXCFG File (TXcfg1.txt)

```
# file : txcfg1.txt - an example of the T1 config for TX4000
# CONFIGURE CLOCK

# CLOCK: clock derived from network 1 drives H.100|H.110 A clocks
clock net=1 a

# CONFIGURE TRUNKS
```

```
#     Trunk     Framing   Encoding   Buildout   Loop Master
t1cfg 1         esf       b8zs       0          false

# CONFIGURE PORTS
# syntax : port <portnum> <bus> <outstream> <slot> <speed>

#    PortNum    L|H|E|T|J   Trunk   Channel     Speed
port 1          t1          1       0           56

#  Command Descriptions:
#
#
#  Clocking configuration commands:
#
#  clock    | Set the H.100/H.110 clock registers
[ConfigBoardClock]
#  netref   | Set the H.100/H.110 NETREF signal
[ConfigNetrefClock]
#
#  T1/E1 trunk configuration commands:

#  e1cfg    | Configure trunk as E1 mode
[ConfigLocalStream]
#  t1cfg    | Configure trunk as T1 mode
[ConfigLocalStream]
#  j1cfg    | Configure trunk as J1 mode
[ConfigLocalStream]
#  e1opt    | Set E1 configuration option
#  t1opt    | Set T1 configuration option
#  j1opt    | Set J1 configuration option
#
#  Connection commands:

#  connect  | Make half-duplex connection
[MakeConnection]
#  port     | Make full-duplex connection
[ConfigLocalTimeslot]
#
#  Clocking Control:
#
#  CLOCK Command syntax:

#  clock <source>[=<network>] <outsigs> [<netref speed>
[<FALLBACK>]]
#    where
#      <source>           = A | B | NR1 | NR2 | NET=<network> | OSC
#      <network>          = 1..n (where n is maximum network
available)
#      <outsigs>          = A | B | - (None)
#      <netref speed>     = 8K | 15M | 20M | - (None)
#
```

```
#    <FALLBACK> = <fallback source>[=<fallback network>]
#    where
#      <fallback source>  = A | B | NR1 | NR2 | NET=<fallback
network> | OSC
#      <fallback network> = 1..n (where n is maximum network
available)
#
#    examples:
#      clock a -            source from H.100 A clocks; no output
clocks
#      clock a b            source from H.100 A clocks; drive H.100
B clocks
#      clock nr1 - 8k       source from NETREF_1 signal (at 8 KHz);
no outputs
#     clock net=1 a        source from network 1 clock; drive H.100
A clocks
#      clock osc b          source from internal oscillator; drive
H.100 B clks
#      clock a - - b        source from H.100 A; fallback to H.100 B
clocks
#
#  NETREF Command syntax:
#  netref <network> <outsigs> [<netref speed>]
#    where
#      <network>      = 1..n (where n is maximum network available)
#      <outsigs>      = NR1 | NR2 | NR12 | - (None)
#      <netref speed> = 8K | 15M | 20M    [DEFAULT 8KHz]
#
#  T1/E1 Trunk Configuration:
#  T1CFG|J1CFG Command syntax:

#  t1cfg <trunk_num> <framing> <encoding> <build_out> <master>
#    where
#      <trunk_num> = number of trunk to configure (1..n)
#      <framing>    = D4 | F4 | ESF | F72
#      <encoding>   = NOZCS | B7ZS | B8ZS
#      <build_out>  = 0(0 dB) | 1(-7.5 dB) | 2(-15 dB) | 3(-22.5 dB)
#      <master>     = TRUE | FALSE
#
#  T1CFG|J1CFG Parameter Choices:
#  Framing:
#      D4  = D4 (193S) Framing - 12-frame multiframe format (F12,
D3/4)
#      F4  = 4-frame multiframe format (F4)
#     ESF = Extended Superframe Format - 24-frame multiframe format
(ESF)
#      F72 = 72-frame multiframe format (F72, remote switch mode)
#
#  Encoding:
#      NOZCS = AMI encoding with NO Zero Code Suppression
#      B7ZS  = Bit 7 stuffing with Zero code Suppression
```

```
#      B8ZS  = Bipolar Eight Zero Substitution
#
#  Buildout:
#      0 =   0   dB
#      1 = - 7.5 dB
#      2 = -15   dB
#      3 = -22.5 dB
#
#  Loop Master:
#      TRUE  = Local side of connection acts as timing source for
this circuit
#      FALSE = Remote side of connection expected to act as timing
source
#
#  T1OPT|J1OPT Command Syntax and Options:

#  t1opt|j1opt [!]<flag_name> | <value_name>=<value>
#    where:
#      !            = do NOT set flag (i.e. clear flag)
#      <flag_name>  = name of flag to set (or clear if '!'
specified)
#      <value_name> = name of value to change
#      <value>      = new value for named parameter
#
#  T1|J1 OPTION FLAGS:
#    Name   Default Description

#    EXZE           extended code violation or excessive zero
detection
#    SRAF           F12:FS-bit of frame 12; ESF:bit2=0
#    CRC    SET     CRC6 check/generation enabled [ESF format only]
#    AIS3           AIS detection only in asynchronous state
#    SSC2           LFA declared if more than 320 CRC6 errors per
second
#    RRAM           allow detect of remote(yellow) alarm during bit
error rates
#    LOS1           additional condition for LOS alarm cleared:
GR-499-CORE
#    SJR   [J1:SET] alarm handling done according to ITU-T JG.
704+706
#    AXRA   SET     remote alarm bit set automatically if receiver
async state
#    CRCI           automatic CRC[4] bit inversion
#    XCRCI          transmit CRC[4|6] bit inversion
#    DCOXC          center function of transmit circuitry is enabled
#    DAIS           automatic AIS insertion disabled
#    DAXLT          disable auto high impedence transmit pins on
short detect
#    DXJA           disable internal transmit jitter attenuation
#    DCF            disable center function of receive circuitry
```

```
#    EQON           -36 dB receiver; long haul mode (CLR = -10 dB;
short haul)
#    RLM            receiver mode for receive line monitoring
#    CLOS           received data is cleared as soon as LOS detected
#   SCF           corner frequency of DCO-R reduced by factor of 10
to 0.6 Hz
#    MCSP           multiple candidates synchronization procedure
#    SSP            select synchronization/resynchronization
procedure
#
#  T1|J1 OPTION VALUES:
#    Name   Default Range   Description

#    XP0    0xFF[^] 0x00-1F transmit pulse shape mask (for 1st
level)
#    XP1    0xFF[^] 0x00-1F transmit pulse shape mask (for 2nd
level)
#    XP2    0xFF[^] 0x00-1F transmit pulse shape mask (for 3rd
level)
#    XP3    0xFF[^] 0x00-1F transmit pulse shape mask (for 4th
level)
#    RIL    0x00    0x00-07 receive input threshold
#    SLT    0x02    0x00-03 voltage thresh when receive slicer
generates mark
#    PCD    0x00    0x00-FF LOS alarm if no trans in 16x(pcd+1)
consec pulses
#    PCR    0x00    0x00-FF LOS alarm clr'ed if pcr+1 pulses in
detect interval
#    XY     0x00    0x00-1F spare bits for national use
#    SSC    0x00    0x00-03 select sync conditions
#
#    [^] = special indicator: transmit pulse shape determined by
line buildout
#
# Boot-Time Connection Definitions:
#
#  PORT Command:
#      The port command defines a pair of half-duplex connections
(resulting
#      in a single full-duplex connection) between the
communications processor
#     on the TX board and a remote channel. Port connections can be
made
#      over an H100 stream [connecting to another board in the same
chassis]
#      or over a local stream [connecting to a remote chassis via
E1/T1/J1].
#      The port command includes the specification of the outbound
stream.
#      For local streams the inbound stream number is the same as
the outbound.
```

```
#      For H.100/H.110 the inbound stream number is determined as
follows:
#        if <outstream> even; then <instream> = <outstream> + 1
#        if <outstream> odd; then <instream> = <outstream> - 1

#  port <portnum> <bus> <outstream> <slot> [<speed>]
#    where
#      <portnum>   = communications port number to connect with
#      <bus>       = H100 (H.100/H.110) | LOCAL (local bus) |
#                    E1 (local bus to trunk in E1 mode) |
#                    T1 (local bus to trunk in T1 mode) |
#                    J1 (local bus to trunk in J1 mode)
#      <outstream> = outbound stream #: 0-31(Hbus), 0-n(Local)
#                    [if <bus>=E1|T1|J1 then <outstream>=1-based
trunk number]
#      <slot>      = inbound and outbound timeslot: 0-127(Hbus),
0-31(Local)
#                    [if <bus>=E1|T1|J1 then <slot>=1-31(E1),
0-23(T1/J1)]
#      <speed>     = 56 | 48 (specify to change from default 64 Kb
speed)
#
#  CONNECT Command:
#      The connect command defines a single half-duplex connection,
connecting
#      a TDM input to a TDM output.  Knowledge of the TX board's
local stream
#      mapping scheme is required to properly use the connect
command
#      (see 'Local Stream Usage' table below).
#
#  CONNECT Command syntax:
#  connect <inbus> <instream> <inslot> <outbus> <outstream>
<outslot>
#    where
#      <inbus>     = H (H.100/H.110) | L (local bus)
#      <instream>  = inbound stream # : 0-31(Hbus), 0..n(Local)
#      <inslot>    = inbound timeslot : 0-127(Hbus), 0-31(Local)
#      <outbus>    = H (H.100/H.110) | L (local bus)
#      <outstream> = outbound stream #: 0-31(Hbus), 0..n(Local)
#      <outslot>   = outbound timeslot: 0-127(Hbus), 0-31(Local)
#
#  TX 4000[C] Local Stream Usage:
#  ------------------------------
#  The TX 4000 provides the following local streams:
#
#   0 and  1 - Trunk 1
#   4 and  5 - Trunk 2
#   8 and  9 - Trunk 3
#  12 and 13 - Trunk 4
#
```

```
#  16 and 17 - Trunk 5 [CompactPCI only]
#  20 and 21 - Trunk 6 [CompactPCI only]
#  24 and 25 - Trunk 7 [CompactPCI only]
#  28 and 29 - Trunk 8 [CompactPCI only]
#
#  72 and 73 - primary connection to communications processor
(MPC8260)
#  74 and 75 - secondary connection to communications processor
(MPC8260)
#
#  80        - Trunk 1 as single stream (rather than as stream pair)
#  81        - Trunk 2 as single stream (rather than as stream pair)
#  82        - Trunk 3 as single stream (rather than as stream pair)
#  83        - Trunk 4 as single stream (rather than as stream pair)
#
#  84        - Trunk 5 as single stream [CompactPCI only]
#  85        - Trunk 6 as single stream [CompactPCI only]
#  86        - Trunk 7 as single stream [CompactPCI only]
#  87        - Trunk 8 as single stream [CompactPCI only]
#####################################################################
```

Sample MTP3 File (mtp3cp1.cfg)

```
# file: mtp3cp1.cfg - example of the MTP config for TX4000 board
# Overall MTP3 Parameters
NODE_TYPE          SP        # choose STP [routing] or SP
[non-routing]
PC_FORMAT          DFLT       # Point code format:  DFLT (8.8.8) /
INTL (3.8.3) / JNTT (7.4.5)
POINT_CODE         231.10.4
RESTART_REQUIRED   TRUE
VALIDATE_SSF       TRUE
MAX_LINKS          4
MAX_USERS          1          # sccp only
MAX_ROUTES         64
MAX_ROUTE_ENTRIES  32
MAX_LINK_SETS      2
MAX_ROUTE_MASKS    2
ROUTE_MASK         0xFFFFFFFF
ROUTE_MASK         0xFFFF0000
MTP3_TRACE_DATA   TRUE
END

#
# LINK CONFIGURATION (for T1 ports)
#
LINK             0
PORT              T1        #  T<n> for T1/E1/MVIP, S<n> for serial
(V.35)
LINK_SET           1
```

```
LINK_TYPE          ANSI       #  ANSI / ITU / JNTT
ADJACENT_DPC       254.152.0  #  AT&T STP-1 DPC
LINK_SLC           0
LSSU_LEN           2
SSF                NATIONAL   # NATIONAL / INTERNATIONAL
# the following settings are to pass AT&T certification
TIMER_T34    90         # remove this for production deployment
MAX_SLTM_RETRY 0
LINK_TRACE_DATA   TRUE
END
#
# LINK CONFIGURATION (for T1 ports)
#
LINK             1
PORT              T1        #  T<n> for T1/E1/MVIP, S<n> for serial
(V.35)
LINK_SET           2
LINK_TYPE          ANSI       #  ANSI / ITU / JNTT
ADJACENT_DPC       254.153.0  #  AT&T STP-1 DPC
LINK_SLC           0
LSSU_LEN           2
SSF                NATIONAL   # NATIONAL / INTERNATIONAL
LINK_TRACE_DATA   TRUE
END
#-----------------------------------------------
# User Parameters (NSAP definition)
#-----------------------------------------------
NSAP               0          # isup must be NSAP 0 if its present
LINK_TYPE          ANSI       #  ANSI / ITU / JNTT
END
#-----------------------------------------------
# Routing Parameters
#-----------------------------------------------
# Route UP from network to applications on this node
ROUTE              0
DPC                231.10.4      # this node
SPTYPE   SP
LINK_TYPE   ANSI
DIRECTION          UP         # default is DOWN
END
#
# Route to STP 1
ROUTE              1
DPC                254.153.0      # STP 1 point code
SPTYPE   STP
LINK_TYPE   ANSI
DIRECTION          DOWN         # default is DOWN
END
#
# Route to STP 2
ROUTE              2
```

```
DPC                 254.154.0       # STP 2 point code
SPTYPE   STP
LINK_TYPE   ANSI
DIRECTION           DOWN          # default is DOWN
END
#
# Route to AT&T Network
ROUTE               3
DPC                 254.0.0       # AT&T Network 254.x.x
SPTYPE   SP
LINK_TYPE   ANSI
DIRECTION           DOWN          # default is DOWN
ADJACENT_ROUTE      FALSE
END
#-----------------------------------------------
# Linkset Parameters
#-----------------------------------------------
LINK_SET_DESCRIPTOR       1
ADJACENT_DPC              254.153.0       # STP-1
MAX_ACTIVE_LINKS          1
ROUTE_NUMBER              1,0
ROUTE_NUMBER              2,1
ROUTE_NUMBER              3
END
#
LINK_SET_DESCRIPTOR       2
ADJACENT_DPC              254.154.0       # STP-2
MAX_ACTIVE_LINKS          1
ROUTE_NUMBER              1,1
ROUTE_NUMBER              2,0
ROUTE_NUMBER              3
END
```

Related Documentation Resources

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

Network T-Server for AT&T

- 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:

80fr_ref_06-2008_v8.0.001.00

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

Screen Captures Used in This Document

Screen captures from the product graphical user interface (GUI), as used in this document, may sometimes contain minor spelling, capitalization, or grammatical errors. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Type Styles

Table 17 describes and illustrates the type conventions that are used in this document.

Table 17: 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 234).	Please consult the *Genesys Migration Guide* for more information. Do *not* use this value for this option. A *customary and usual* practice is one that is widely accepted and used within a particular industry or profession. The formula, $x + 1 = 7$ where *x* stands for . . .
Monospace font (Looks like `teletype` or `typewriter text`)	All programming identifiers and GUI elements. This convention includes: • The *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages. • The values of options. • Logical arguments and command syntax. • Code samples. Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.	Select the `Show variables on screen` check box. In the `Operand` text box, enter your formula. Click `OK` to exit the `Properties` dialog box. T-Server distributes the error messages in `EventError` events. If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls. Enter `exit` on the command line.
Square brackets ([ ])	A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.	`smcp_server -host [/flags]`
Angle brackets (< >)	A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise. **Note:** In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.	`smcp_server -host <confighost>`

Index

Symbols

A

B

C

D

E

F

H

I

K

L

M

N

O

P

R

S

T

U

V

W

X