Simulator Test Toolkit 8.0

User's 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 4000 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 8. 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: 80stt_us_06-2011_v8.0.001.00

Table of Contents

Preface

Welcome to the *Simulator Test Toolkit 8.0 User's Guide.* This document introduces you to the concepts, terminology, and procedures relevant to this Genesys product.

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

In brief, you will find the following information in this manual:

- Features and functions of Simulator Toolkit
- How to install and configure Test Simulator Server
- How to use Test Simulator Interface and Contact Center Activity Simulator

Intended Audience

This document, primarily intended for integrators and administrators, and assumes that you have a basic understanding of:

- Computer-telephony integration (CTI) concepts, processes, terminology, and applications.
- Network design and operation.
- Your own network configurations.

You should also be familiar with the Genesys Framework 8.0 Configuration Layer.

Chapter Summaries

The *Simulator Test Toolkit 8.0 User's Guide* provides information on installing and using Simulator Toolkit. To help you locate information, the guide begins with a Table of Contents and ends with an Index. The guide also contains the following chapters and appendix:

- Chapter 1, provides an overview of the functions and features of Test Simulator Toolkit.
- Chapter 2, describes how to configure the Test Simulator Toolkit.
- Chapter 3, describes procedures for installing and starting Test Simulator.
- Chapter 4, describes the user interface for Test Simulator.
- Chapter 5, describes the use of Contact Center Activity Simulator.
- The Appendix, "PBX Commands" on page 71, lists commands that you can use in the Test Simulator Server console in place of using Test Simulator Interface.

Making Comments on This Document

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

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document. Please limit your comments to the information in this document only and to the way in which the information is presented. Speak to Genesys Technical Support if you have suggestions about the product itself.

When you send us comments, you grant Genesys a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

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	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 *Simulator Test Toolkit 8.0 User's Guide*. In the future, this section will list topics that are new or have changed significantly since the first release of this document.

Chapter

1

Simulator Test Toolkit Overview

This chapter provides an overview of the features and functions of Test Simulator Toolkit. It covers the following topics:

Introduction

The Genesys Test Tool Kit is a set of programs that provides the ability to create a simulated Genesys environment where T-Server, and other Genesys applications can function together with customer routing strategies, voice applications, and custom CTI–enabled or VoIP applications in real time. These programs simulate a call center infrastructure with multiple switches, customer activity, and call center agent activity.

The Genesys Test Tool Kit:

- Creates environments where customer routing strategies, voice applications, and entire deployment solutions can be tested against projected load and real life scenarios.
- Gives the customer the ability to develop, debug and present applications that will work with the Genesys Framework components without deploying a real telecommunication infrastructure.

Components

Test Simulator Toolkit consists of the following components:

- Test Simulator—a software application that simulates common CTI related functionality of a physical switch (PBX). T-Server can connect to Test Simulator and operate just as it would with a real switch.
- Test Simulator Interface—a special Windows-based client, which connects as a client to the Test Simulator and supports call control functionality for the Test Simulator.
- Generic T-Server—a special edition of the Genesys T-Server for use with the Test Simulator.
- Contact Center Activity Simulator (CCAS)—an application that simulates activity in a contact center, including answering and transferring calls, attaching data, and routing calls. CCAS also provides VoIP functionality. It supports scenarios that simulate SIP Media Gateways, Agents, and Customer SIP phones together with scenarios for RTP stream control.

Features

The Test Toolkit includes the test simulator that supports the Generic or the Avaya Communication Manager CTI link (n_simg3). Support of the Nortel Communication Server 1000 (n_simlink), the Nortel Communication Server 2000 CTI link (n_simdms), and other types of CTI links (with limited functionality) can be ordered by request.

The Test Toolkit components support Genesys High Availability (HA) infrastructure. The Test Simulator opens the primary and the backup link to support connections from primary and backup T-Servers. CCAS scenarios support automatic switchover to the active T-Server.

The Test Toolkit provides the following capabilities:

- To create a multi-tenant distributed environment with external routing communication.
- To create VoIP simulation environments based on SIP/RTP protocols. RTP related capability based on Stream Manager capabilities and scenarios that control Stream Manager.
- To configure any number of linked scenarios to mimic real complex call flows.

The Test Toolkit is a scalable solution with a single point of control. The Test Toolkit components can utilize as many computers as needed to meet load requirements. The Test simulator and CCAS support remote console operation that allows the ability to control test execution from a single location.

Chapter

2 Test Simulator 8.0

This chapter contains information on the following topics:

Overview

The Test Simulator is a software application that simulates the common CTI-functionality of a physical switch (PBX). T-Server can connect to the Test Simulator and operate just as it would with a real switch. You can use the Test Simulator to develop and test the telephony functions of your applications to simulate the actual switch you will be using.

However, since there will inevitably be differences between the behavior of the simulator and an actual switch, you may want to test your application in a real switch environment prior to production use.

Warning! Correct switch simulator functioning is guaranteed for scenarios supported by the CCAS application only. Some switch specific requests may not work. Request for new features support can be submitted through Genesys Technical Support.

Test Simulator Supported Telephony Elements

The toolkit allows the simulation of the following telephony elements:

- "ACD Positions/Extensions"

- “Agent IDs”
- “Queues” (also known as ACD queues, hunt groups, splits)
- “Routing Points”(also known as Controlled DNs on Nortel switches, Vector DNs on Lucent switches, and Call Control Tables on Aspect switches)
- “Trunks”
- Link DNs (also known as access numbers) to remote switches

ACD Positions/Extensions

ACD positions or extensions may be designated from 1 to 99999 and have two lines: Line 0 for incoming calls and Line 2 for outgoing calls. Available operations for ACD positions or extensions are:

- Make inbound, outbound, and internal calls.
- Set phone status (Ready, Not Ready, Busy).
- Log in and log out.
- Hold calls and take off hold.
- Release active calls.
- Transfer calls, with or without a consult call phase.
- Make conference calls.
- External routing from an agent phone; that is, calling a number that will connect to a remote Test Simulator on a specific ACD.

Agent IDs

Agent IDs may be designated with any string, although numeric values may be most convenient. Agent IDs may be associated with a phone station/extension but are not required. An Agent ID is required for logging an agent in to a queue.

Queues

Queues may be designated with values from 1 to 99999. Available operations for a queue are:

- Parking calls until answered by an agent or abandoned by the caller.
- May be used to generate incoming calls, at a given rate, using the DNIS.

Routing Points

Routing points may be designated with any string, although numeric values may be most convenient. Routing points:

- Generate requests for routing destination and park calls until the `RequestRouteCall` is generated or is abandon from the caller.
- May be dialed from an external number.
- May be used to generate incoming calls, at a given rate, using the DNIS.

Trunks

Incoming and outgoing trunks may be designated with any string, although numeric values may be most convenient. Operations are not performed directly with trunks. Rather, they work behind the scenes to handle inbound/outbound calls. Operations/features that are enabled by trunk availability are:

- Inbound calls with ANI.
- Inbound calls with DNIS.
- Inbound calls with collected digits.
- Outbound calls.

Simulator Configuration

The Simulator requires a Switch object in the Configuration Layer. If you are not using the Configuration Layer, you can configure a virtual switch using an Equipment file (see "PBX Commands" on page 71).

Creating a Switch

Create the `Switch` object (see the *Framework Configuration Manager Help* for details) with the desired DNs, including queues, routing points, and external route points. You must add to this switch a special `Annex` section called `Simulator`, shown in Figure 1 on page 16. The simulator gathers all the necessary information to start working as a real switch from the `Annex` section. The `Simulator` section contains all the mandatory options required for simulator operation. Additional sections can be created to collect test specific option values, for example, entry-points, queue assignments and so on. If additional sections are used, it must be specified with "–s" argument in the command line.

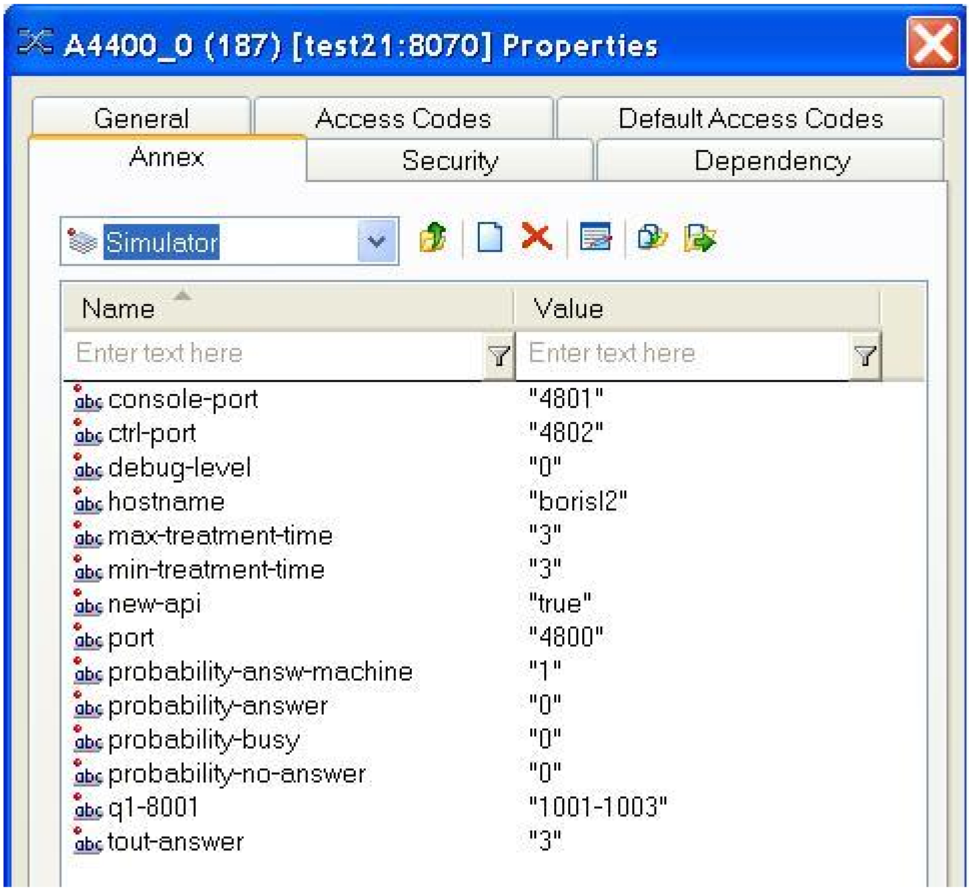


Figure 1: Annex Simulator Section

Configuration Options

The options you can use in the Simulator section include three mandatory options and a number of optional ones.

Mandatory Options

You must add these options and give them values:

hostname

Default value: none
Valid value: any text string

The name of the host where Test Simulator will run. This option is used for external routing simulation. You must specify a value for this option.

port

Default value: none
Valid value: positive integer

Number of the port where Test Simulator will create a server. You must specify a value for this option. T-Server opens the CTI-link to this host and port. You will need to configure T-Server with the corresponding value of host and port.

console-port

Default value: none
Valid value: positive integer

The port for the remote console. The remote console is used to support external routing in multi-site environments. It is also used by the Test Simulator Interface and it can be used to control test simulator remotely. You must specify a value for this option.

Other Options

If you do not add the following options, the attributes they control will either not exist or will have the default values listed.

General Options

ani-file

Default value: none
Valid value: any text string

Name of text file containing list of ANI numbers for inbound call simulation.

cti-protocol-mode

Default value: g3
Valid values: none, g3, generic

When set to `g3`**,** the Test Simulator simulates an Avaya Definity G3 ECS. When set to generic, the Test Simulator supports the Generic T-Server.

data-file

Default value: none
Valid value: any text string

Name of text file containing list of user data for attaching to calls.

start-trunk-num

Default value: 20000
Valid value: positive integer

First number of a range of trunk numbers. See trunk-num-range.

trunk-num-range

Default value: 1000
Valid value: positive integer

Number representing quantity of trunks in the range begun by start-trunk-num. Since each inbound or outbound call allocates one trunk, this option defines how many active inbound and outbound call that can be supported.

queue-tout

Default value: 30
Valid value: positive integer

Maximum time, in seconds, that a call can wait in a queue or routing point.

ringing-tout

Default value: 30
Valid value: positive integer

Maximum time, in seconds, a call can remain on a DN in ringing state, before being abandoned.

Predictive Dialing Options

Note: The Predictive Dialing options are supported with the Generic and Avaya switch simulators only.

probability-answer

Default value: 0.1
Valid value: number between 0 and 1

Probability of answer result for outbound call.

probability-answ-machine

Default value: 0.1
Valid value: number between 0 and 1

Probability of answering machine result for outbound call.

probability-busy

Default value: 0.1
Valid value: number between 0 and 1

Probability of busy result for outbound call.

probability-circuit

Default value: 0.1
Valid value: number between 0 and 1

Probability of circuit result for outbound call.

probability-fax

Default value: 0.1

Valid value: number between 0 and 1

Probability of fax result for outbound call.

probability-no-answer

Default value: 0.1
Valid value: number between 0 and 1

Probability of no-answer result for outbound call.

probability-num-changed

Default value: 0.1
Valid value: number between 0 and 1

Probability of num-changed result for outbound call.

probability-invalid-num

Default value: 0.1
Valid value: number between 0 and 1

Probability of SIT tone detection for outbound call.

probability-unknown

Default value: 0.1
Valid value: number between 0 and 1

Probability of an unknown result for outbound call.

probability-vacant

Default value: 0.1
Valid value: number between 0 and 1

Probability of vacant result for outbound call.

tout-answ-machine

Default value: 5
Valid value: positive integer

Time, in seconds, until call detects an answering machine.

tout-answer

Default value: 5
Valid value: positive integer

Time, in seconds, until call detects an answer.

tout-busy

Default value: 5
Valid value: positive integer

Time, in seconds, until call detects a busy signal.

tout-circuit

Default value: 5
Valid value: positive integer

Time, in seconds, until call detects a circuit.

tout-fax

Default value: 5
Valid value: positive integer

Time, in seconds, until call detects a fax machine.

tout-invalid-num

Default value: 5
Valid value: positive integer

Time, in seconds, until call detects a SIT tone.

tout-num-changed

Default value: 5
Valid value: positive integer

Time, in seconds, until call detects a number-changed result.

tout-no-answer

Default value: 5
Valid value: positive integer

Time, in seconds, until call detects a result of no-answer.

tout-unknown

Default value: 5
Valid value: positive integer

Time, in seconds, until call detects an unknown result.

tout-vacant

Default value: 5
Valid value: positive integer

Time, in seconds, until call detects a result of vacant.?

Configuring DN Properties

qxx-QUEUE

The option `q<ID>-<ACD Queue>` is used to specify the queue assigned for a range of DNs (pre-login queue), auto-answer ,and post-ready properties. This option is not mandatory and can be omitted. However, this option must be specified when using the Nortel Communication Server 1000 and the Nortel

Communication Server 2000 switch simulators in order to assign DNs to ACD Queues.

By default, DNs are not automatically logged in to the queue, and have no assigned auto answer, and post ready properties. You must configure the option as follows:

- Option name = `q<Numeral>-<Queue>`
- Option value = `<DNmin>-<DNmax>, <autoanswer/noautoanswer>, <postready/nopostready>`

  Where:

  - `Numeral` is any unique number.
  - `Queue` is the ACD Queue number assigned to the range of DNs.
  - `DNmin` and `DNmax` specify the DN range.
  - `autoanswer/noautoanswer` instructs the simulator to make automatically answer the call on arrival. The default value is `noautoanswer`.
  - `postready/nopostready` instructs the simulator to set the DN into a ready state after the call is released. The default value is `nopostready`.

Note: These DNs must have already been created in the configuration database.

The following illustrates examples of DN configuration:

- `q1-8111` = `101-105, autoanswer, postready`

  Queue `8111` as assigned to a range of DNs (ACD positions) from `101` to `105` with `autoanswer` and `postready` turned on.

- `q2-8111` = `101-105`

Queue `8111` as assigned to a range of DNs (ACD positions) from `101` to `105` with `noautoanswer` and `nopostready` properties.

Note: The default values for the DNs are `noautoanswer` and `nopostready`. The scenarios supported by the Simulator rely on these default values. Genesys recommends that you do not change the default values.

Configuring Inbound Call Generation

entry-pointXX

The Simulator can be configured to generate inbound calls while in "Run" mode. It is possible to specify the call rate and specific parameters of every entry-point. You must have a DN of type Routing Point or ACD Queue in order to generate inbound calls. The simulator sends an event notification about the new inbound calls on behalf of these DNs. The total call rate is a summary of all call rates configured on the generating DNs.

You must configure the option as follows:

- Option name = `entry-point<Numeral>`
- Option value = `<DN-number>, <DNIS>, <CallRate>`

  Where:
  - `Numeral` is a unique identifying number.
  - `DN-number` is the number of an ACD Queue or Routing Point.
  - `DNIS` is the value of the DNIS attribute for the inbound calls.
  - `CallRate` is a floating number that specifies the inbound call rate in calls per second.

The following illustrates an example of the DN configuration:

- `entry-point1` = `2201, 888, 0.5`

  In "Run" mode this option instructs simulator to generate inbound calls with follow parameters:
  - `2201` is the DN where the inbound call arrives.
  - `888` is the DNIS attribute for the inbound call.
  - `0.5` is the call rate.

  Therefore, the simulator will generate a new inbound call on DN 2201 every two seconds.

Note: The simulator stays in the Stop mode from the beginning. You must enter the "go" instructions to put the simulator in the Run mode.

Configuring the Simulator for External Routing

The switch simulator is designed to support multi-site environments. In other words, a number of simulator instances can work together by exchanging messages that carry information about new and existing calls between the switches. The switch simulator uses the proprietary protocol or message daemon protocol. The message daemon protocol also works with Network T-Server.

Configuring the Simulator for External Routing with the Proprietary Protocol

Using the proprietary protocol is the easiest option for external routing. Add unique `console-port` and `hostname` options in the `Annex` tab for each simulator that is using external routing. Access codes must also be configured on each switch, and the connection between corresponding T-Servers must also be set. If default (route) external routing is used, external routing points must also be configured in the corresponding switch.

Note: The simulator does not support empty Access Codes. Genesys recommends that you specify the Access Codes with a unique number that does not overlap the internal DNs.

Chapter

3 Installing and Running Test Simulator 8.0

This chapter contains information on the following topics:

Installing

On the product CD, locate and double-click `Setup.exe`. Follow the instructions.

Before proceeding, be sure that the Test Simulator Server icon in the Windows Start menu is associated with the proper command line. You can do this in two ways:

- Enter the command line in the Target field of the Properties window of the Test Simulator Server shortcut.
- Put the command line in a batch file. Enter the name of the batch file in the Target field of the Properties window of the Test Simulator Server shortcut.

For command-line parameters, see the next section.

Command-Line Syntax

The command-line syntax for Test Simulator is as follows:

```
<executable> -host <hostname> -port <port> -sw <switchname> -ts
<tservername> -pass <password> [-user<username>],[-clapp<appname>]
```

Where:

- `<executable>` is the name of the executable simulator file.
- `<hostname>` is the name of the host of Configuration Server.

- `<port>` is the port for connecting to Configuration Server.
- `<switchname>` is the name of the Switch object in Configuration Server that Simulator will simulate.
- `<tservername>` is the name of the T-Server object in Configuration Server that connects to the simulated switch.
- `<password>` is a security feature providing data security for different applications within the same Configuration Server.

The parameters listed above are all required. There are also two optional parameters: `-user <username>` and `-clapp <appname>`

Where:

- `<username>` is the user name that you use when connecting to Configuration Server. The default value is `default`.
- `<appname>` is the application name that you use when connecting to Configuration Server. The default value is `default`.

Example:

```
testsim -host test-bdc -port 5070 -sw Aspect-switch -ts TestTSrv
-pass password
```

Starting

To run the Test Simulator, first do the following:

- Be sure you have created and configured a switch in the Configuration Layer, and that the switch includes a Simulator section (see Chapter 2).
- Have the Configuration Layer running.

Then proceed as follows:

1. Start T-Server.
2. Start Test Simulator Server.
3. Start Test Simulator Interface.
4. In the Test Simulator Interface, connect to the Test Simulator using the Connection Tab.

Chapter

4 Test Simulator Interface

The Test Simulator Interface or PbxV (PBX Viewer) is a software component that works with the Test Simulator as a TCP/IP client using the remote console (do not forget to configure the console-port for test simulator), and that supplies a user interface for controlling the Test Simulator. The Test Simulator Interface is an optional component. It is used for convenience and visualization. The Test simulator can work without the Test Simulator Interface, and use the console (and remote console) commands instead.

This chapter describes the use of this interface, including the following topics:

The Main Window and Commands

Test Simulator Interface's main window is shown in Figure 2 on page 28.

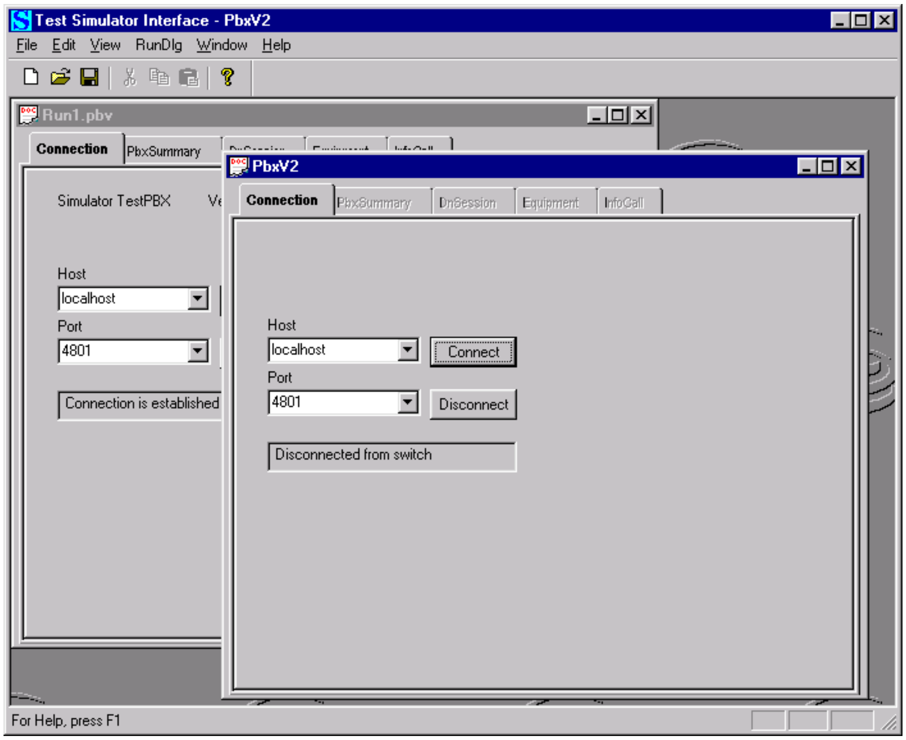


Figure 2: Test Simulator Interface Main Window

Session Window

The main window contains one or more session windows (Figure 2 shows two). Each session window displays a Test Simulator session. Sessions can be saved as *`.pbv` files, which store basic session-defining settings, such as host name, port number, and command line.

Main Menu

Most of the menus accessed from the main menu have the usual basic commands. The following sections describe a few commands that are special to Test Simulator Interface.

File Menu

- `New`—Creates a new Test Simulator session. A new session appears in a new session window with the default title PbxV[n+1], where [n] is the number of sessions previously opened or created in the main window. You can rename the session when you save it. The active session window in Figure 2 is titled PbxV2 since it was opened after Run1.
- `Open`—Opens an existing *`.pbv` file.

- Save—Saves a Test Simulator session as a *.pbv file.
- LoadEquipment—Loads a new equipment file
- SaveEquipment—Saves the current equipment configuration

RunDlg

The RunDlg command brings up a dialog box that allows you to launch Test Simulator or T-Servers (see Figure 3). This is equivalent to launching the applications from the Windows NT® Start Menu. The advantage of the RunDlg command is that it allows you to configure multiple switches and T-Servers, each specifying different equipment, configurations, or runtime options. This allows you to have several different testing environments coexisting.

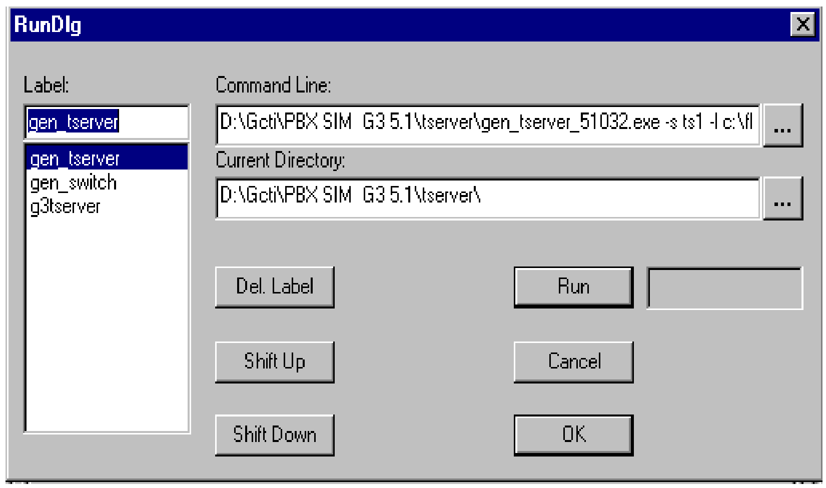


Figure 3: RunDlg Dialog Box

The dialog box allows for the entry of the following information:

- Label—This provides a "tag" that the Test Simulator uses to associate the Command Line and Current Directory information. Choose a label that conveniently describes the Test Simulator or T-Server that is to be run, such as myTServer.
- Command Line—Specifies the executable file and any parameters desired. This command line is identical in syntax to that in the batch file associated with the shortcut provided at install. The Browse button on the right (denoted with an ellipsis) allows you to browse to the correct path for the executable file. After the path has been entered, you must enter the proper parameters.
- Current Directory—Specifies the current directory in which the (Test Simulator or T-Server) executable is located. The Browse button on the right (denoted with an ellipsis) allows you to browse to the correct path for the executable file. By default, this is the directory where the Platform SDK is installed.

The RunDlg box also provides the following commands:

- `Del. Label`—Deletes the current label and dialog information from the Windows NT Registry.
- `Shift Up`—Shifts the selected label up in storage order.
- `Shift Down`—Shifts the selected label down in storage order.
- `Run`—Runs the application specified in the Label text box.
- `Cancel`—Cancels current action and closes dialog box without saving command-line information.
- `OK`—Saves current label and dialog information to the Windows NT Registry.

Window Menu

- `New Window`—Opens a new PBX Window that you can use to connect to an existing or new Test Simulator session.

Session Window Tabs

This section describes the five tabs on the Test Simulator Interface session window.

The Connection Tab

The Connection tab contains controls that enable you to set up connections with the switch simulator (see Figure 4 on page 31).

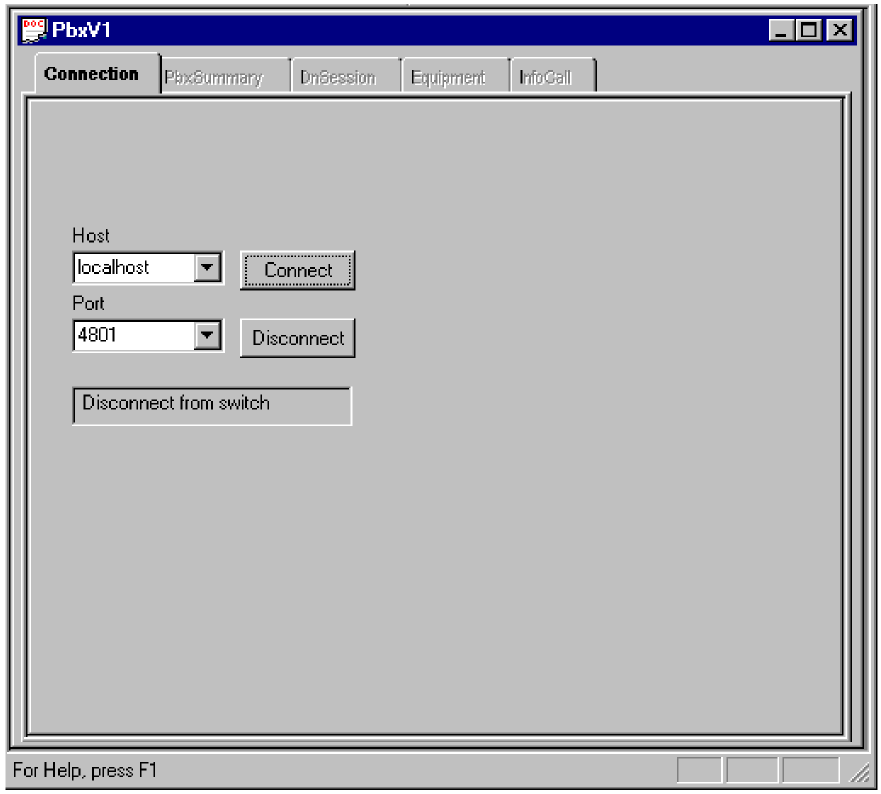


Figure 4: The Connection Tab

Setting Up a TCP/IP Connection to PBX

To set up a TCP/IP connection with the switch:

1. Select a host name from the `Host` drop-down list (or type one in if it is not in the menu). Choose `localhost` for the default value.
2. Select a port number from the `Port` drop-down list (or type one in if it is not in the menu). This port number should match the value of the console-port option. Choose `4801` for the default value.
3. Click `Connect`.

With an active connection to the Test Simulator you can use the four other tabs to monitor and control the PBX.

The PbxSummary Tab

PbxSummary Tab is the second tab in the session window (see Figure 5 on page 32). Clicking the PBxSummary tab allows you to view the entire PBX environment as well as placing the PBX in manual or automatic mode. This tab is primarily for viewing information.

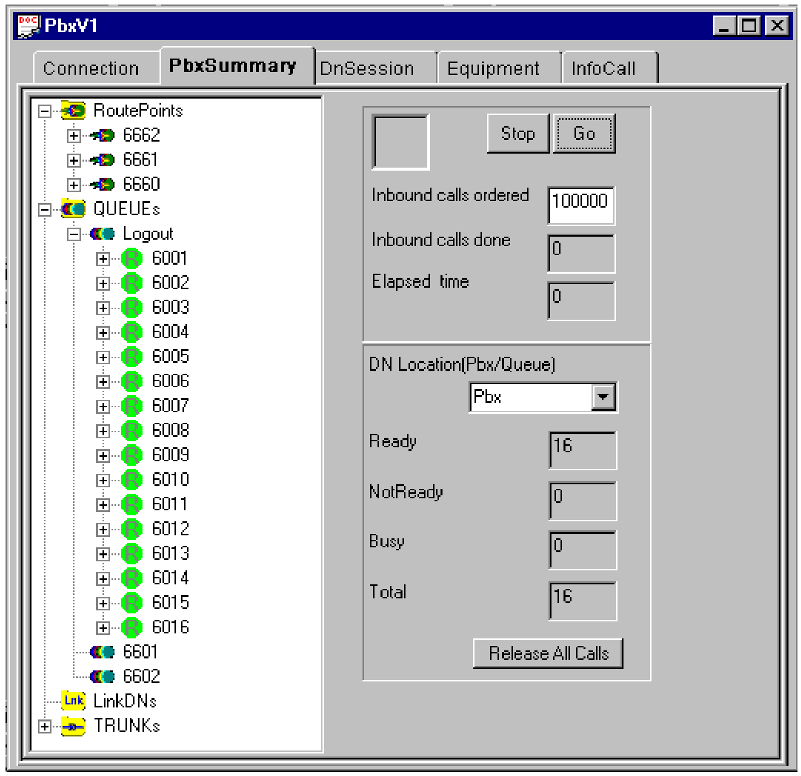


Figure 5: The PBXSummary Tab

The pane on the left side of the PbxSummary tab provides information on routing points, queues, trunks, and ACD positions or extensions. This information also appears in the DnSession tab for convenience. The status of each DN is displayed using color codes and letters as in Table 1.

Table 1: DN Color Codes

Status	Color	Letter
Ready	Green	R
Not ready	Red	N
Busy	Yellow	B

The right side of the PbxSummary tab includes the following (from top to bottom):

- `Stop` and `Go` buttons—clicking `Go` puts the Test Simulator in `Go` mode, which means that it is ready to generate inbound calls.
- Three fields displaying information on simulated inbound calls, as follows:

    - `Inbound calls ordered`—the number of inbound calls that Test Simulator has been set to make
    - `Inbound calls done`·the number of inbound calls made so far
    - `Elapsed time`—the time elapsed since Test Simulator was put in Go mode
- A summary of the overall state of the PBX. From the DN Location drop-down you can choose a queue or PBX for which the following DN totals are displayed:
    - `Ready`
    - `NotReady`
    - `Busy`
    - `Total`
- `Release All Calls` button—releases all calls.

The DnSession Tab

DnSession is the third tab in the session window and provides a secondary means of call control for Test Simulator (see Figure 6 on page 34).

Warning! T-Server may not recognize all the actions that are possible from this tab. Genesys recommends that you use these controls only if you are unable to perform the actions using a T-Server client application, such as a software phone.

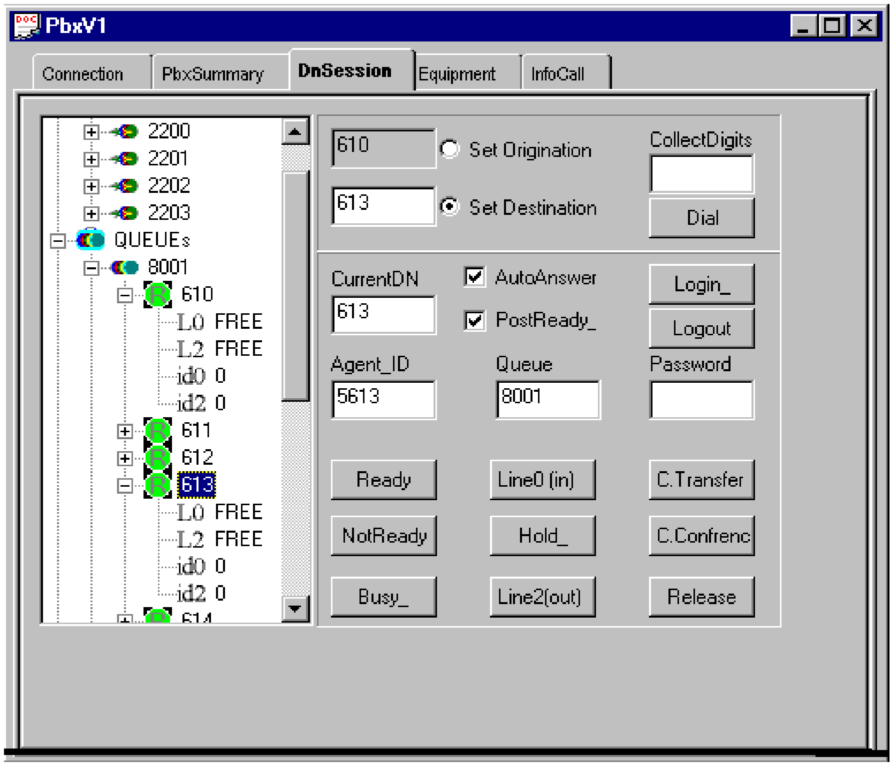


Figure 6: The DnSession Tab

The following sections describe each control available from the DNSession tab.

Making and Receiving Calls

`Set Origination`: To specify the originating DN, click the radio button. Then enter the DN number or double-click the DN icon in the left pane for autofill in the `Set Origination` text box.

`Set Destination`: To specify the destination DN, click the radio button. Then enter the DN number or double-click the DN icon in the left pane for autofill in the `Set Destination` text box.

`Dial Button`: Click this button to make a call from the DN specified in the `Set Origination` text box to the DN specified in the `Set Destination` text box.

To simulate an inbound call:

1. Enter a 6-digit or greater number in the `Set Origination` text box. This number is interpreted as the ANI for the new call.
2. Enter a valid routing point in the `Set Destination` text box.

`CollectDigits:` To simulate customer-entered digits (CED), enter any digits in the `CollectDigits` text box. The CollectedDigits attribute in T-Server is then populated.

Controlling Agent Activities

CurrentDN: This box specifies the DN to which actions apply.

AutoAnswer: If this box is selected, the DN specified in `CurrentDN` answers automatically when called.

PostReady: If this box is selected, the DN specified in `CurrentDN` automatically returns to a `Ready` state after releasing a call.

The Login Button: Click the `Login` button to log an agent in at the DN specified in the `CurrentDN` box. Enter the agent identification in the `Agent_ID` text box and the queue assignment in the `Queue` text box.

The Logout Button: Click the `Logout` button to log an agent out on the DN specified in the `CurrentDN` box.

The Ready Button: Click the `Ready` button to change the status to `Ready` for the DN specified in the `CurrentDN` box.

The NotReady Button: Click the `NotReady` button to change the status to `NotReady` for the DN specified in the `CurrentDN` box.

The Busy Button: Click the `Busy` button to change the status to `Busy` for the DN specified in the `CurrentDN` box.

The Hold Button: Click the `Hold` button to place the call on hold for the DN specified in the `CurrentDN` box.

The Release Button: Clicking the `Release` button releases all active calls on the DN specified in the `CurrentDN` box. This button has the same effect as the `Release All Calls` Button on the PbxSummary tab.

Controlling Lines Within an Extension

The Line0 (in) Button: Click the `Line0` (in) button to have the DN specified in the `CurrentDN` box answer an active call or take a call from hold.

- If a call on line 0 was previously placed on hold, the `Line0` button returns the call to active status.
- If a call arrived previously on line 0, the `Line0` button changes the status of this call from ringing to active.

The Line2(out) Button: Click the `Line2`(out) button to have the DN specified in the `CurrentDN` box answer an active call or take a call from hold.

- If a call on line 2 was previously placed on hold, the `Line2` button returns the call to active status.
- If a call arrived previously on line 2, the `Line2` button changes the status of this call from ringing to active.

Performing Conference and Transfer Calls

The C(omplete) Conferenc(e) Button: If there is an active call at a specific DN, it is possible to simulate a conference call:

1. Click the `Hold_` button. This places the active call on hold.
2. Place a call from one of the active DNs to a third DN.
3. Click `C.Conferenc.` All three DNs are connected.

The C(omplete) Transfer Button: If there is an active call at a specific DN, it is possible to simulate a call transfer, as follows:

1. Click the `Hold_` button. This places the active call on hold.
2. Place a call from one of the active DNs to a third DN.
3. Click `C.Transfer.` The second DN is released and the first and third DNs remain connected.

The Equipment Tab

Equipment tab is the fourth tab of the session window (see Figure 7 on page 37). You can use this tab to create and delete virtual equipment.

Warning! Use the Equipment tab with caution. Actions you take here are not reflected in Configuration Server. Therefore, changes you make in the Equipment tab may create problems for T-Server, which takes its configuration from Configuration Server. Apart from the Probability settings, Genesys recommends that you create, configure, and delete virtual equipment using the Configuration Layer.

You can create the following equipment:

- RP—routing points
- DN—such as ACD positions and queues
- Link DN—access numbers to remote PBXs
- Trunk—main lines for calls coming into and going out of a switch

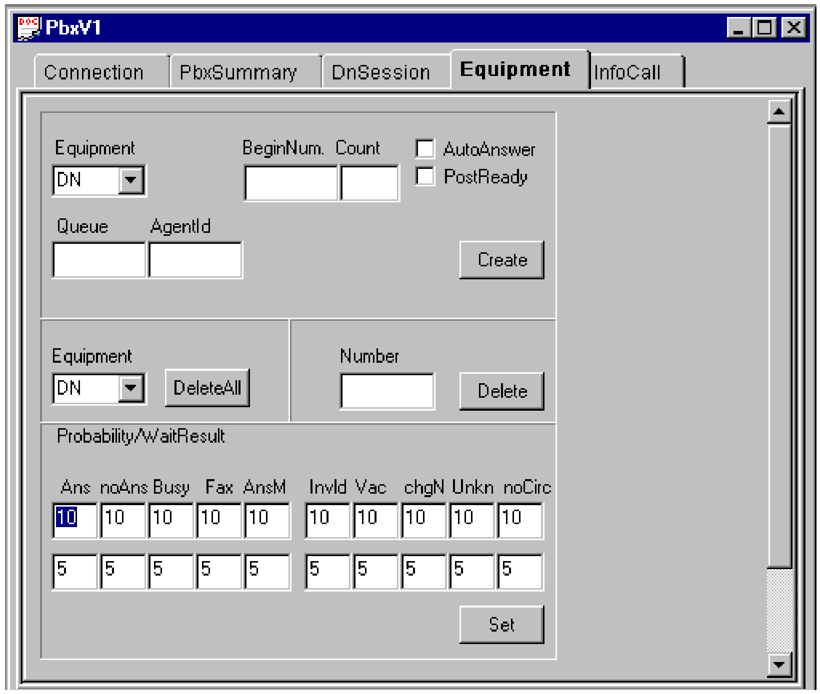


Figure 7: The Equipment Tab

The `Equipment` tab contains three groups of controls; they are concerned with (from top to bottom in Figure 7) creating equipment, deleting equipment, and setting probability.

Creating Equipment

The `Create` group of controls, occupying the top one-third of the `Equipment` tab, contains the buttons you can use to create RPs, ranges of DNs, Link DNs, and Trunks. To create each type:

1. Make a selection from the `Equipment` drop-down list.
2. Enter the required information in the text boxes. The type selected determines which text boxes are available to populate.
3. Click `Create`.

DN

Selecting `DN` allows creation of a range of DNs. Valid values to enter in each text box are `1`–`99999`. You must provide the following parameters before clicking `Create`:

- `BeginNum`—specifies the starting number of a range of DNs. Valid values are `1`–`99999`.
- `Count`—specifies the number of DNs to be created.

- `Queue`—specifies the number of the queue associated with a given DN. Valid values are `8000–8999`.
- `AgentID`—valid values are `0–9999`.

In addition to these text boxes, there are two check boxes associated with the DN equipment type that set how you want your DNs to operate during simulation.

- `PostReady`—select this box if you want the DNs to go to `Ready` status as soon as a call is released.
- `AutoAnswer`—select this box if you want the DNs to have `AutoAnswer` status.

RP

Selecting `RP` allows creation of a range of routing points. Valid values to enter in each text box are `1–99999`. You must provide the following parameters before clicking `Create:`

- `Number`—specifies the routing point designation number. Valid values are `1–99999.`
- `Queue`—contains the number of the queue associated with this RP. Valid values should include a queue configured in the switch.
- `DNIS`—defines the DNIS for inbound call in the `DNIS` text box.
- `Call/sec`—defines the frequency of inbound calls during `Run` mode. If you don't want to generate inbound calls, leave this box empty.

LinkDN

`LinkDNs` establish connections between switches. They are used in simulating external routing/transfers to remote switches (locations). This button may be used to create a `LinkDN` with the following parameters:

- `Number`—the LinkDN number that will be dialed by the local switch to access the remote switch. Valid values are `1–1200`.
- `Host`—the name of the remote host computer that is running a (remote) T-Server.
- `Port`—the management port of the remote switch. This port is specified by the value of the `-p` parameter in the command line of the remote PBX.
- `RemoteCDN`—the remote routing point (on the remote switch) associated with the number specified by the `LinkDN`.

Trunk

Select `Trunk` to create a range of trunks. Test Simulator uses trunks to make outbound calls or receive inbound call traffic.

- `BeginNum`—specifies the beginning number of the range you wish to create. Valid values are `1–99999`.
- `Count`—specifies the number of trunks created.

Deleting Equipment

This group of controls, occupying the middle third of the `Equipment` tab (see Figure 7), is used to delete equipment from the simulator during runtime. You can delete an entire type of DN, such as all routing points, or select an individual number.

1. Select the type of equipment in the `Equipment` drop-down list.
2. Then, do one of the following:
   a. Click `DeleteAll` to delete all equipment of the selected type

   OR

   b. Enter the number of an item of equipment in the `Number` box, then click `Delete`.

Probability/WaitResult

The `Probability/WaitResult` group of controls, occupying the lower third of the `Equipment` tab (see Figure 7), simulates probabilities of call activity for predictive dialing (simple outbound calls are simulated with 100 percent success). This group of controls consists of 10 labeled pairs of text boxes and a Set button. Values entered in the text boxes are used by Genesys Outbound Contact Server/Contact Manager in a Predictive Dialing mode.

The labels are as follows:

- Ans (answer)
- noAns (no answer)
- Busy
- Fax (also includes the probability of a modem answering)
- AnsM (answering machine)
- Invld (invalid number)
- Vac (vacant)
- chgN (number changed)
- Unkn (unknown)
- noCirc (no circuit)

Under each label are two text boxes. The upper box contains a probability value and the lower box contains a timeout value.

- Probability is the probability, as a percentage, that a given outbound call will be of the type designated by the label. Valid ranges are 0 to 100, with total probability adding up to 100. For example, if you enter `20` as the probability value for Busy, Test Simulator gives a result code of `busy` back to Outbound Contact Server for 20 percent of the outbound calls dialed.

- Timeout is the delay time, in seconds, before the simulated hardware detects a given type of call result. For example, if you enter `5` as the timeout value for Ans, the simulation detects a human voice after a five-second delay. The default timeout value is one second.

The Set button loads the specified probability and timeout values for the Test Simulator to use during runtime.

The InfoCall Tab

The `InfoCall` tab provides detailed information on a selected call and on the total number of calls in a selected location. Call information refreshes automatically. See Figure 8 on page 40.

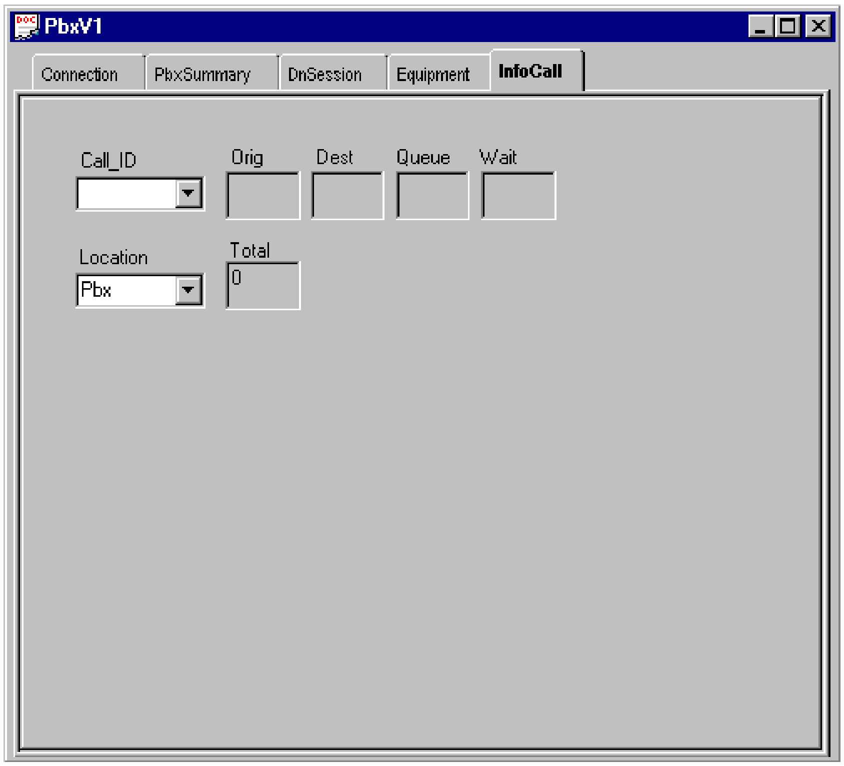


Figure 8: The InfoCall Tab

Call_ID

To receive information about a given call, either enter the Call ID or select from the `Call_ID` drop-down list. The following information appears:

- `Orig`—Origination DN
- `Dest`—Destination DN
- `Queue`—Queue assigned to the call
- `Wait`—Wait time

Location

To receive information about the PBX or a given queue, choose `PBX` or a given queue number from the `Location` drop-down list. The total number of calls appears in the `Total` text box.

Chapter

5 Contact Center Activity Simulator

This chapter describes the Contact Center Activity Simulator (CCAS) user interface, and covers the following topics:

How It Works

CCAS simulates contact center activity, including answering and transferring calls, attaching data, and routing calls. CCAS is a script based application that supports communication with T-Server, SIP Server, Configuration Server, and Stream Manager.

CCAS operates by means of *scenarios.* A scenario consists of three parts:

- The name of the server that it belongs to.
- A predefined scenario name.
- A number of arguments specifying how the scenarios run—for example, the range of DNs, and the length of time spent talking.

For each server type, there exists a set of scenarios that can be assigned to that corresponding server. All scenarios work simultaneously which means that one CCAS project can simulate a complex environment where all components work together with a single point of control.

Note: The list of scenarios are constantly updated. The list of scenarios and parameters may differ from the current document.

Installation

The CCAS installation is included with the Test Toolkit it installation. In order to support VoIP scenarios, CCAS requires a separate installation of Genesys Stream Manager 7.6.

The CCAS Graphical User Interface

The CCAS interface screen has three tabs: `Servers`, `Scenarios`, and `ScriptEngine`. The first two tabs contain fields to hold server and scenario configuration data. To enter data, select a field and double-click. Then you can either select from the resulting drop-down list or enter new data manually. The interface allows you to perform the following activities:

- Open a previously saved project—Select `Open` from the `File` menu.
- Start a simulation—Click the green `Go` button on the toolbar.
- Stop a simulation—Click the red `Stop` button.

Servers Tab

The `Servers` tab (see Figure 9) includes records with server information.

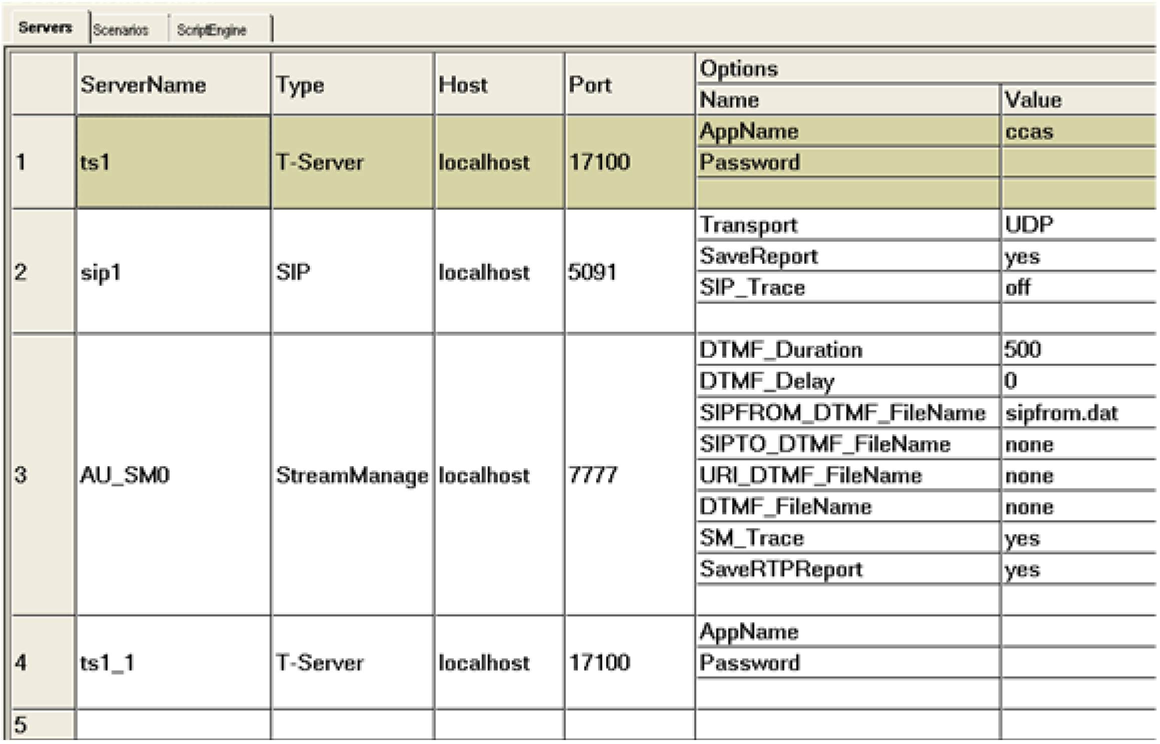


Figure 9: Servers Tab

Scenarios use this information to connect to the specified servers. For example, Genesys SIP Server can be represented in `Servers` tab as two records:

- TLib connection
- SIP connection

Table 2 describes the parameters included in these records.

Table 2: Servers Tab Parameters

Parameter		Description
ServerName		The name of the server that runs the scenario.
Type		The type of server: • T-Server • SIP • StreamManager
Host		The host name of the server.
Port		The TCP/IP port that the server is running on.
Options	Name	The name of the option.
	Value	The value for the option.

Note: The value of the `ServerName` associates the scenario to the server. T-Server and SIP can use an arbitrary value; however, StreamManager must use the same value as assigned to the Stream Manager instance, see StreamManager, page 46.

This document describes scenarios for the following server types:

- T-Server
- SIP
- StreamManager

These scenarios cover simulation of traditional (PBX with CTI) and VOIP contact centers.

T-Server

The `T-Server` type represents a T-Server. Scenarios assigned to this server communicate through the Genesys T-Library protocol, and enable the simulation of a variety of call flows for traditional CTI PBX environments. For the T-Server type server, you need to specify the T-Server host, and the T-Server port. Specify the T-Server application name and password in order for this information to be passed in the `RegisterClient` request from the assigned scenarios to the T-Server.

SIP

Servers with the `SIP` type specify the direction to which the SIP messages will be transmitted. In other words, it represents an outbound SIP proxy. For SIP proxy you must specify the host and port of the SIP Server, and the type of SIP transport (UDP).

Note: CCAS uses the Server tab information to send SIP messages (to find destination IP address). It does not use SIP message headers to find SIP destination. However, the 302 (Move temporary) response processing is an exception. Contact information from the 302 message is used to send SIP messages. This functionality is necessary to support Network SIP Server behavior.

StreamManager

The `StreamManager` type represents a connection with Stream Manager. You must specify the following Stream Manger connection parameters:

- `Host`
- `Port`
- `DTMF_Duration`

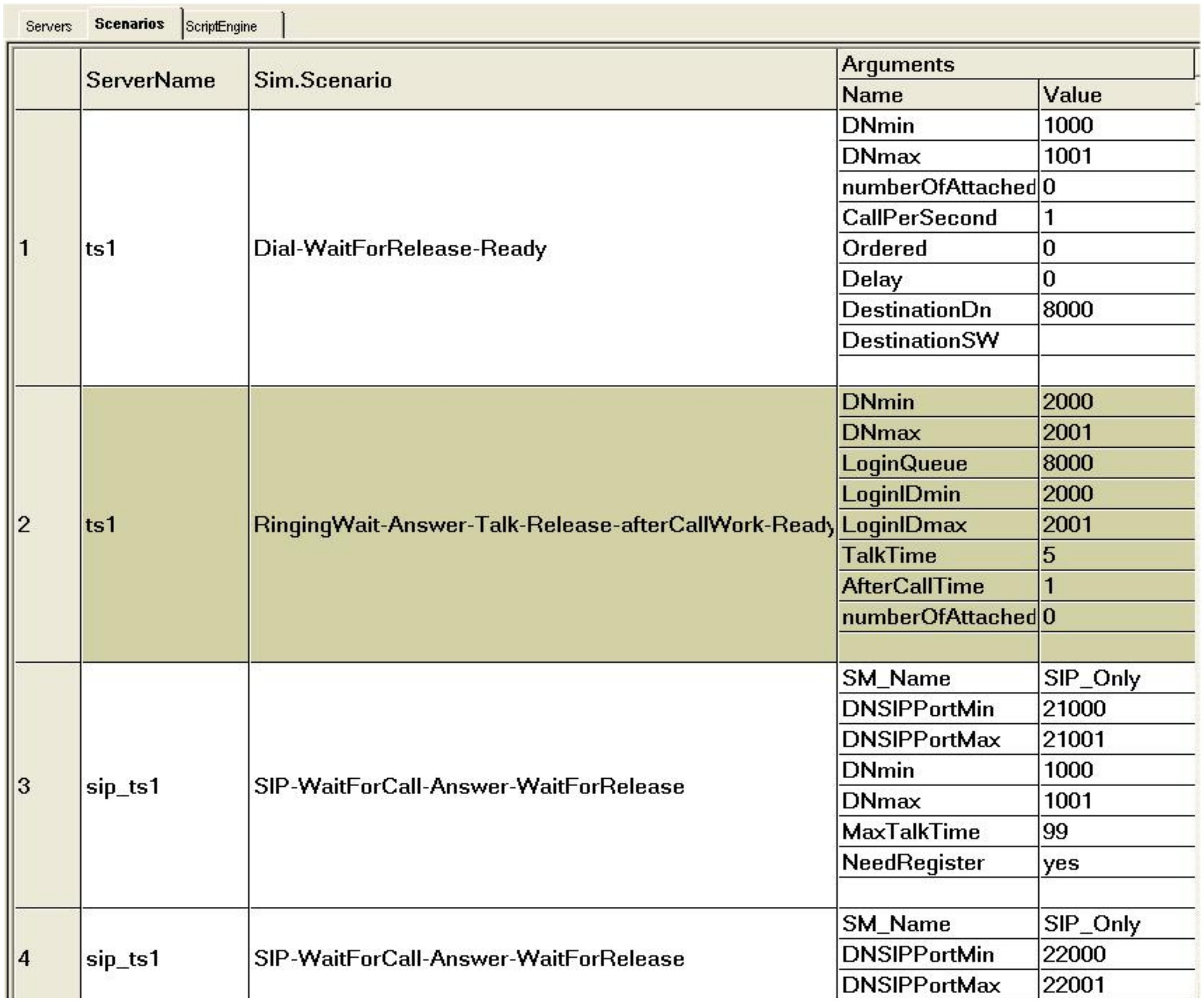
- DTMF_Delay
- SIPFROM_DTMF_FileName
- SIPTO_DTMF_FileName
- URI_DTMF_FileName
- DTMF_FileName

Stream Manager establishes connections to CCAS in order to get stream control instructions. It is very important that Stream Manager runs with the application name and parameters corresponding to those input in the CCAS Servers tab of CCAS.

The data files are the multi-record files where each record represents a set of DTMF sequences or file names to play. This information is used to send DTMF, or to play a file according to the logic of the voice application. In order to use FROM (the ANI information) for the current session as a DTMF key or file name to play, the SIPFROM_DataFile parameter must be specified. If the TO value (the DNIS information) is used, the SIPTO_DataFile parameter must be specified.

Scenarios Tab

The Scenarios tab allows you to assign test scenarios to the servers that you identified in the Servers tab (see Figure 10). CCAS includes a set of predefined scenarios which covers your testing needs.



Servers | Scenarios | ScriptEngine

	ServerName	Sim.Scenario	Arguments Name	Value
1	ts1	Dial-WaitForRelease-Ready	DNmin	1000
			DNmax	1001
			numberOfAttached	0
			CallPerSecond	1
			Ordered	0
			Delay	0
			DestinationDn	8000
			DestinationSW	
2	ts1	RingingWait-Answer-Talk-Release-afterCallWork-Ready	DNmin	2000
			DNmax	2001
			LoginQueue	8000
			LoginIDmin	2000
			LoginIDmax	2001
			TalkTime	5
			AfterCallTime	1
			numberOfAttached	0
3	sip_ts1	SIP-WaitForCall-Answer-WaitForRelease	SM_Name	SIP_Only
			DNSIPPortMin	21000
			DNSIPPortMax	21001
			DNmin	1000
			DNmax	1001
			MaxTalkTime	99
			NeedRegister	yes
4	sip_ts1	SIP-WaitForCall-Answer-WaitForRelease	SM_Name	SIP_Only
			DNSIPPortMin	22000
			DNSIPPortMax	22001

Figure 10: Scenarios Tab

The `Scenarios` tab includes records with scenario descriptions. Table 3 describes the Scenarios tab parameters.

Table 3: Scenarios Tab Parameters

Parameter		Description
Server Name		The name of the server.
Sim.Scenario		The name of the test scenario.
Arguments	Name	The name of the argument.
	Value	The value for the argument.

For more information about scenario arguments, see "Scenario Descriptions" on page 49.

ScriptEngine Tab

The `ScriptEngine` tab provides a console window showing the progress and output of the scenario as it runs, and a small window for entering the commands `Info` and `Print`. These commands provide debug capability. Figure 11 shows the trace of a running test.

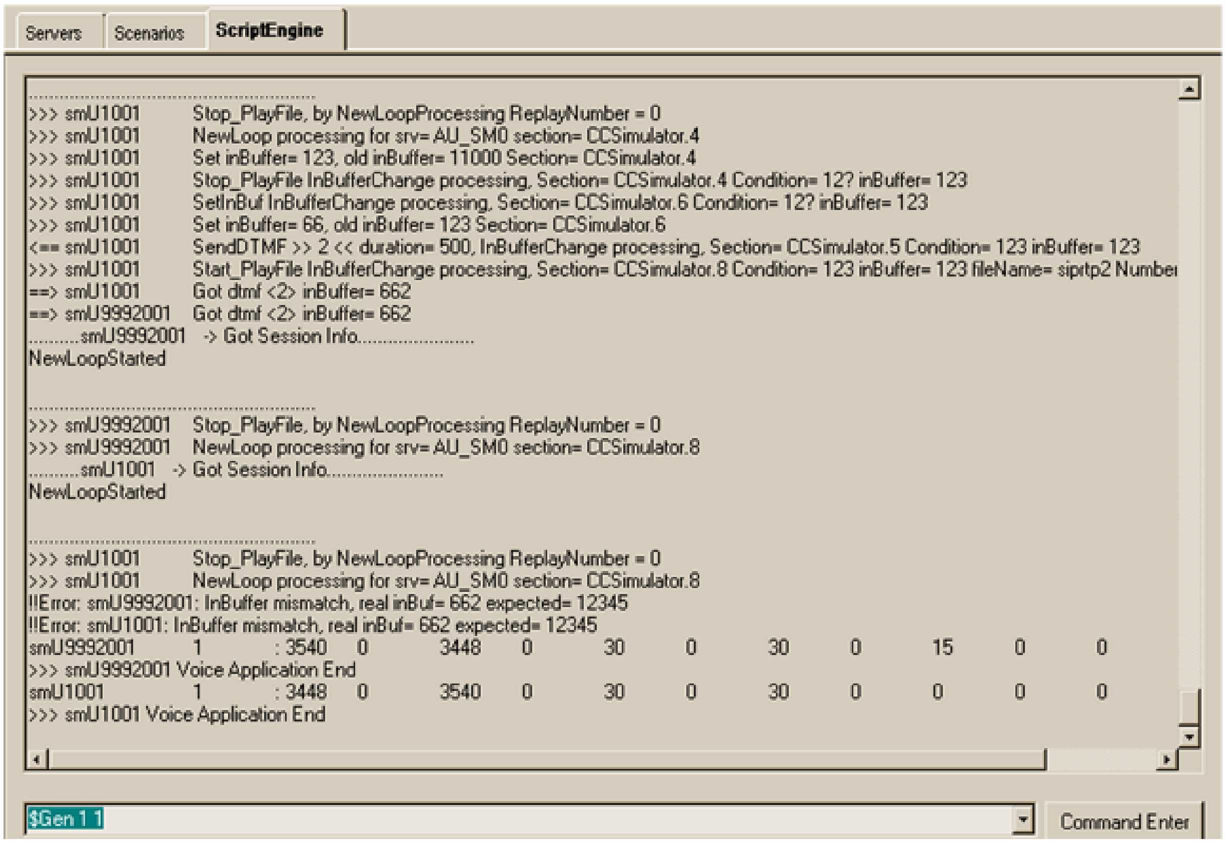


Figure 11: ScriptEngine Tab

Saving Configured Scenarios

Once you have configured a scenario with the argument values that you want, you can save it for later use. Select `Save As` from the `File` menu and enter a name. Your scenario and argument values are then saved as a `*.ccsa` file.

Scenario Descriptions

There are different types of scenarios depending on the server type to which they are assigned. Since the set of built in scenarios and arguments is often modified and updated, the actual scenario and argument list in your distribution may be different from what is described the the following sections.

T-Server Scenarios

All T-Server type scenarios are written with the assumption that all DNs have `noautoanswer` and `nopostready` modes. Scenarios will still run without these DN modes, however, it is recommended to set up the switch simulator with the corresponding DNs mode. This section describes the most common T-Server scenarios.

Dial-WaitForRelease-Ready

This scenario generates calls from a given DN range to a specified destination. DNs in the `DNmin` to `DNmax` range make a call with attached data to a queue or routing point, and then wait for release. The queue or routing point can be in a remote location. Table 4 describes the Dial-WaitForRelease-Ready scenario arguments.

Table 4: Dial-WaitForRelease-Ready

Argument Name	Description
DNmin	The starting number of the DN range.
DNmax	The last number of the DN range.
numberOfAttachedKVPair	The quantity of key-value pairs to attach to call. The default is one.
CallPerSecond	The dialing rate in calls per second (if there are clients ready). The default is one.
Ordered	The total number of calls to dial.
DestinationDN	The destination DN, usually a queue or a routing point.
DestinationSW	The destination switch, using the Switch object name in Configuration Server.

RingingWait-Answer-Talk-Release-afterCallWork-Ready

The RingingWait-Answer-Talk-Release-afterCallWork-Ready scenario deals with calls arriving at DNs of type ACD Position or Extension. Table 5 describes the RingingWait-Answer-Talk-Release-afterCallWork-Ready scenario arguments.Table 5 on page 51.

Note: The `AfterCallTime` argument sets the length of time a DN spends in the After Call Work state.

Table 5: RingingWait-Answer-Talk-Release-afterCallWork-Ready

Argument Name	Description
DNmin	The starting number of the DN range.
DNmax	The last number of the DN range.
LoginQueue	The queue that the DNs in the specified range will log in to.
LoginIDmin	The starting number of the Login ID range.
LoginIDmax	The last number of the Login ID range.
TalkTime	The length of time needed to simulate a conversation.
AfterCallTime	The length of time in the After Call Work state.

RingingWait-Answer-Talk-AttachUserData-TwoStepTransfer-Ready

This scenario answers an incoming call, waits for talking time and initiates a two-step transfer to the specified destination. After the consult call is established, it makes a complete transfer. This scenario uses a set of key-value pairs to serve as the attached data. The keys and values are simple enumerations. The keys are called defaultKey0, defaultKey1, and so on up to defaultKey9. The corresponding values are 0, 1, and so on up to 9. To replace the defaults with custom keys and values, use the "AttachData-Initialization" scenario. Table 6 describes the RingingWait-Answer-Talk-AttachUserData-TwoStepTransfer-Ready scenario arguments.

RingingWait-Answer-Talk-AttachUserData-MuteTransfer-Ready

This scenario answers an incoming call, waits for talking time, and initiates a mute transfer to the specified destination. This scenario uses a set of key-value pairs to serve as attached data. The keys and values are simple enumerations. The keys are called defaultKey0, defaultKey1, and so on up to defaultKey9. The corresponding values are 0, 1, and so on up to 9. To replace the defaults with custom keys and values, use the "AttachData-Initialization"scenario. Table 6 describes the RingingWait-Answer-Talk-AttachUserData-MuteTransfer-Ready scenario arguments.

Table 6: Transfer Scenarios

Argument Name	Description
DNmin	The starting number of the DN range.
DNmax	The last number of the DN range.
LoginQueue	The queue that the DNs in the specified range will log in to.
LoginIDmin	The starting number of the Login ID range.
LoginIDmax	The last number of Login ID range.
TalkTime	The length of time needed to simulate a conversation.
numberOfAttachedKVPair	The quantity of key-value pairs to attach to the call.
DestinationDN	The destination DN, usually a queue or a routing point.
DestinationSW	The destination switch, using the Switch object name in Configuration Server.

Note: The argument `numberOfAttachedKVPair` determines how many key-value pairs are attached to each call. If there are 10 key-value pairs (the default) and you decide to have two pairs attached to each call, the first call will have the first and second key-value pairs attached, the second call will have the third and fourth key-value pairs attached, and so on. After the tenth pair has been attached to a call, the next call receives the first pair and the sequence begins again.

RingingWait-Answer-Conference-Talk-Release-afterCallWork-Ready

Table 7 describes the RingingWait-Answer-Conference-Talk-Release-afterCallWork-Ready scenario arguments.

Table 7: RingingWait-Answer-Conference-Talk-Release-afterCallWork-Ready

Argument Name	Description
DNmin	The starting number of the DN range.
DNmax	The last number of the DN range.
LoginQueue	The queue that the DNs in the specified range will log in to.
LoginIDmin	The starting number of the Login ID range.
LoginIDmax	The last number of the Login ID range.
TalkTime	The length of time needed to simulate a conversation.
DestinationDN	The destination DN, usually a queue or a routing point.
DestinationSW	The destination switch (Switch object name in Configuration Server).
AfterCallTime	The length of time in the After Call Work state.

RoutRequestWait-RouteCall

Table 8 describes the RoutRequestWait-RouteCall scenario arguments.

Table 8: RoutRequestWait-RouteCall

Argument Name	Description
DNmin	The starting number of the DN range.
DNmax	The last number of the DN range.
DestinationDN	The destination DN, usually a queue or a routing point.

Table 8: RoutRequestWait-RouteCall (Continued)

Argument Name	Description
DestinationSW	The destination switch (Switch object name in Configuration Server).
WaitForPartyChange	Instructs the scenario to wait for a PartyChange event for consult call routing. Use `yes` for routing a consult call during mute transfer; otherwise, use default value `no`.
numberOfAttachedKVPair	The quantity of key-value pairs to attach to the call.

AttachData-Initialization

The AttachData-Initialization scenario allows you to customize the keys and values of attached data. Table 9 describes the arguments.

Table 9: AttachData-Initialization

Argument Name	Description
userkey1	You can customize both the name and the value of these key-value pairs
userkey2	
userkey3	

SIP Call Scenarios

There are two servers types required to support VoIP related scenarios—SIP Server and Stream Manager. A server with SIP type is responsible for SIP activity, and can be used with or without real voice streaming. The following sections describe these scenarios.

Configuring Objects for Use with SIP Server Scenarios

Genesys SIP Communication Server (SIP CS) can work in different modes. For more information about SIP Server, see the *Genesys SIP Server Deployment Guide*.

In order to work with the CCAS SIP Server scenarios, some SIP Servers and DN options must be set in specific states. Some options must be synchronised with scenario parameters.

Configuring DN Contact Information

SIP Server retrieves contact information from the Cfg Server DN Annex (Section `TServer`, option contact) tab, or from the DN SIP Register request. The scenario `NeedRegister` parameter specifies how the contact information is obtained.

If the `NeedRegister` parameter is set to `yes`, the scenario will send a Register request with the contact information on behalf of all the DNs from the specified DN range.

If the `NeedRegister` parameter is set to `no`, you must specify the contact information in the configuration database according to the `DNSIPPortMin` and `DNSIPPortMax` scenario parameters.

Configuring DN Options in Configuration Server

CCAS SIP scripts are designed to work in a "single dialog" mode. In order for a “single dialog” to work, set the following options on the DN `Annex` tab for each DN simulated by CCAS:

- `dual-dialog-enabled` = `false`
- `refer-enabled` = `false`

For more information about these SIP Sever options, see the *SIP Server Deployment Guide*.

SIP-MediaGateWay-Call-Talk-Release

The SIP-MediaGateWay-Call-Talk-Release scenario simulates Media Gateway functionality and creates a number of incoming SIP calls with a specified rate. Table 10 describes the SIP-MediaGateWay-Call-Talk-Release scenario arguments.

Table 10: SIP-MediaGateWay-Call-Talk-Release

Argument	Description
SM_Name	The Stream Manager name if stream support is required. The default value is `none`.
Request URI	The URI of the INVITE message that is sent to the SIP destination.
MG Listen SIP Port	The listening SIP communication port of the Media Gateway.
ReferDest	The name of the SIP type on the `Servers` tab. It specifies the SIP destination for transferred calls.
Ext DN Min	The minimum channel number.

Table 10: SIP-MediaGateWay-Call-Talk-Release (Continued)

Argument	Description
Ext DN Max	The maximum channel number.
Talk time	The call duration in seconds.
CallPerSecond	The desired call rate.
Order	The number of calls which are generated during a given session.
NeedRegister	Specifies whether the scenario is to send a Register request.

Note: To estimate the number of active calls, use the following formula:
CNT = Talk time * CallPerSecond

SIP-WaitForCall-Answer-WaitForRelease

The SIP-WaitForCall-Answer-WaitForRelease scenario simulates a number of Agent SIP phones, which answer the call and wait for call release from the calling party. Table 11 describes the SIP-WaitForCall-Answer-WaitForRelease scenario arguments.

This scenario works with the T-Server scenario. The T-Server scenario is required to perform login/ready operations, and for third-party call control functionality.

Table 11: SIP-WaitForCall-Answer-WaitForRelease

Argument Name	Description
SM_Name	The Stream Manager name if stream support is required. The default value is `none`.
AgentSIPPortMin	The first SIP port used for an agent endpoint.
AgentSIPPortMax	The last SIP port used for an agent endpoint.
DN Min	The minimum channel number.
DN Max	The maximum channel number.

Table 11: SIP-WaitForCall-Answer-WaitForRelease (Continued)

Argument Name	Description
MaxTalkTime	The maximum talk time duration, in seconds.
NeedRegister	Specifies whether the scenario is to send a Register request.

Note: `AgentSIPPortMin` and `AgentSIPPortMax` can use the same value if you need to simulate an Agent SIP phones located behind a SIP Proxy.

SIP-WaitForCall-Answer-Release

The SIP-WaitForCall-Answer-Release scenario simulates a number of Agent SIP phones that answer the call and release the call after talking time. Table 11 describes the SIP-WaitForCall-Answer-Release scenario arguments.

RTP Stream Control Call Scenarios

This section describes the set of scenarios which allows simulating a dialog between an external caller/agent and a voice application. The existing RTP stream control scenarios can put a load on production voice applications in the pre-deployment phase and during lab testing.

Any voice application can be represented as a state machine, and can be instrumented by DTMF exchange for synchronization purposes. In this case all test logic is engineered by creating a final state machine table where conditions are represented by accumulated DTMF input from a voice application, and a corresponding action for each buffer state. The current implementation employs the following actions:

- Send DTMF sequence
- Play voice file
- Record voice file with play back in background
- Modify input buffer
- Print text message
- Print RTP report upon call released
- Check input buffer upon call released

After a call is established, CCAS sets an input buffer for the current session to an `empty` value. The condition `empty` will trigger the initial action upon a call being established (no DTMF received yet).

Note: Some stream control scenarios do not use a fixed list of arguments under the argument name column, but a list of conditions which in the argument value column have corresponding actions. It gives the ability to specify in a single scenario list of different conditions and actions. Conditions in notation means that value of the condition is equal to current value of the input buffer, or if the value is notated as `nn?mm`, the third digits in position is not used in the condition.

Configuring Objects for Use with Stream Manager Scenarios

In order to use Stream Manager related scenarios, the corresponding Stream Manager instance (or instances) must be running. Stream Manager can run with or without a Configuration Server connection. This section describes the situations where Stream Manager is running with a Configuration Server connection.

Stream Manager works with CCAS the same way as it works with a DMX application object. (DMX is a Genesys application. CCAS usage doesn't require any knowledge about the DMX application). Create and configure two application objects:

1. Create an application object using the `dmx_client_700` application template with any arbitrary application name.
2. Set the tenant information (use the SIP Server tenant).
3. Set the Server information for this application.
   - Host name must be the host with the CCAS application.
   - Port can be set to any arbitrary value.
   - Set the `sm-port` option to the same value that is specified in CCAS project for the corresponding Stream Manager Server.
4. Create a second application object using the `VOIP_SM_760` application template with the name set to the same value that is specified in CCAS project for the corresponding Stream Manager Server.
5. Set the tenant information (use the SIP Server tenant).
6. Set the Server information for the application.
   - Host name must be the host where Stream Manager is running.
   - Port can be 0 (or any acceptable value).
7. Add a connection to the DMX application.
8. Set the following options in the `contact` section:
   - `sip-port` = 0

- `rtp-address`—Specifies the IP address used by Stream Manager for RTP communication. Set this option to the IP address of the Stream Manager computer, or to a dedicated NIC if the computer has more than one NIC.
- `rtp-port`—Specifies the initial port for RTP/RTCP connections. This option and the `maxports` option together define a range of ports for use by Stream Manager.
- `maxports`—This option and the `rtp-port` option together define a range of ports for use by Stream Manager. This range cannot overlap the range of port used by CCAS SIP scenarios for SIP phone simulation (usually `DNPortMin/DNPortMax`). This range should be more than double the value of the desired active stream in the simulation environment (each stream uses two ports—one for RTP, and one for RTCP).

For more information about these options and other Stream Manager options, see the *Stream Manager Deployment Guide*.

SM-PlayFile

The SM-PlayFile scenario instructs the Stream Manager to play voice files if the specified conditions are true. A number of condition and corresponding voice files can be specified for a single scenario. Table 12 describes the SM-PlayFile arguments.

Table 12: SM-PlayFile

Argument Name (Condition)	Argument Value (File Name to play definition)
Condition 1 (Input Buffer Value 1 or Regular Expression)	The name of the prerecorded voice file1 or the reference to this file.
...	...
ConditionN (Input Buffer ValueN or Regular ExpressionN)	The name of the prerecorded voice fileN or the reference to this file.

The reference to a voice file is expressed as either a `DATA`, `SIPFROM`, or `SIPTO` argument, and a column number in a data file record in the arguments for StreamManager on the `Servers` tab.

- The `DATA` value instructs CCAS to retrieve the actual data file name specified in the Stream Manager Server description under the `DataFile` argument.
- The `SIPFROM` value instructs CCAS to use the `SIPFROM_DataFile` argument.
- The `SIPTO` value refers to the `SIPTO_DataFile` argument.

CCAS chooses the corresponding record in the `SIPFROM_DataFile`, `SIPTO_DataFile`, or from the next record in the `DataFile`.

For `SIPFROM` and `SIPTO` references, CCAS uses the current call “From” or “To” information. For example, `[prompt1]`, `[DATA.1]`, `[SIPFROM.3]`.

The file name is specified as the full name with the `.wav` extension, or part of a file name. Stream Manager will construct a full name by appending the codec name and the `.wav` extension to specified part (see Table 13 for examples). File name can include full or relative path. If the file name has been specified without the full path, Stream Manager looks for the file in its current directory. For more information, see the *Genesys 7.6 Stream Manager Deployment Guide*.

Table 13: Name Construction Examples

Value Specified in the Scenario Argument	Codec	Constructed File Name
music\in_queue	G711	music\in_queue_mulaw.wav
music\in_queue	GSM	music\in_queue_gsm.wav
music\in_queue_gsm.wav	Any	music\in_queue_gsm.wav

Note: Files play once only. To play the file more than once, use the SM-PlayFile-ModifyInBuffer scenario.

SM-PlayFile-ModifyInBuffer

The SM-PlayFile-ModifyInBuffer scenario instructs Stream Manager to play a voice file if the specified `StartPlayCondition` is `true`. Table 14 describes the scenario arguments.

Table 14: SM-PlayFile-ModifyInBuffer

Argument Name	Description
FileName	The name of the prerecorded voice file, or the reference to this file.
StartPlayCondition	The input buffer or regular expression.
StopPlayCondition	The input buffer or regular expression.

Table 14: SM-PlayFile-ModifyInBuffer (Continued)

Argument Name	Description
FileReplayNumbers	The number of times to play the file.
SetInBufferTo	The value stored in the input buffer after the scenario is complete.

SM-RecordInput-PlayFileInBackground

The SM-RecordInput-PlayFileInBackground scenario instructs Stream Manager to record an incoming stream and play voice files in the background if the specified conditions are true. Several conditions and corresponding voice files can be specified for a single scenario. Table 15 describes the scenario arguments.

Table 15: SM-RecordInput-PlayFileInBackground

Argument Name (Condition)	Argument Value (File Name to play definition)
Condition 1 (Input Buffer Value 1 or Regular Expression)	The name of the prerecorded voice file1 or the reference to this file.
...	...
ConditionN (Input Buffer ValueN or Regular ExpressionN)	The name of the prerecorded voice fileN or the reference to this file.

CCAS saves the recorded file to the `<Stream Manager Install Folder>\CCAS\` directory. For each session, CCAS constructs a file using the following rule:

`FileName = smU+<DnNumber>_ <TimeStamp_CodecName>.wav`

For example, `smU9992001_4_18_38_38_pcmu.wav`.

Note: Files play once only. To play the file more then once, use the SM-RecordInput-PlayFileInBackground-ModifyInbuffer scenario.

SM-RecordInput-PlayFileInBackground-ModifyInbuffer

The SM-RecordInput-PlayFileInBackground-ModifyInbuffer scenario instructs Stream Manager to record the incoming stream and play the voice file in the background if the specified condition is true. Table 16 describes the scenario arguments.

Table 16: SM-RecordInput-PlayFileInBackground-ModifyInbuffer

Argument Name	Description
PlayFileName	The name of the prerecorded voice file, or the reference to this file.
StartRecordCondition	The input buffer or regular expression.
StopRecordCondition	The input buffer or regular expression.
FileReplayNumbers	The number of times to play the file.
SetInBufferTo	The value stored in the input buffer after the scenario is complete.

CCAS saves the recorded file to the `<Stream Manager Install Folder>\CCAS\` directory. For each session, CCAS constructs a file using the following rule:

`FileName = smU+<DnNumber>_ <TimeStamp_CodecName>.wav`

For example, `smU9992001_4_18_38_38_pcmu.wav`.

SM-SendDTMF

The SM-SendDTMF scenario instructs Stream Manager to send DTMF sequences if the specified conditions are true. A number of conditions and corresponding DTMF sequences can be specified for a single scenario. Table 17 describes the scenario arguments.

Table 17: SM-SendDTMF

Argument Name (Condition)	Send DTMF Action Definition
Condition 1 (Input Buffer Value 1 or Regular Expression)	The DTMF value or the reference to this file.
...	...
ConditionN (Input Buffer ValueN or Regular ExpressionN)	The DTMF valueN or the reference to this file.

The reference to the DTMF value is a predefined expression of either `DATA`, `SIPFROM`, or `SIPTO`, and a column number in a data file record.

- A `DATA` value instructs CCAS to retrieve the actual data file name specified in the Stream Manager Server description under the DataFile argument.
- A `SIPFROM` value instructs CCAS to use a `SIPFROM_DataFile` argument.
- A `SIPTO` value refers to a `SIPTO_DataFile` argument.

When the `SIPFROM_DataFile`, or the `SIPTO_DataFile` is specified, CCAS chooses the record with the corresponding DNIS/ANI key from the SIP message. This means that the first column in a `SIPFROM` or a `SIPTO` file will always be a key for the other columns. For example, `<SIPTO.1>` instructs CCAS to find the record with the DNIS key in the file specified by the `SIPTO` data file argument in Stream Manager, and take the *first* column. `SIPFROM` performs the same action using the `SIPFROM` as the key.

Specifying a `DATA` argument will not use a key, but rather take the indicated column from the next record in the `DATA` file indicated in Stream Manager.

Note: To keep compatibility with the `SIPFROM/SIPTO` format, the `0` column must be written in a `DATA` file so that `DATA.1` is in the second column.

SM-ModifyInBuffer-as-InBufferChanges

The SM-ModifyInBuffer-as-InBufferChanges scenario changes the input buffer values if the specified conditions are true. A number of conditions and corresponding input buffer values can be specified for a single scenario. This scenario is used to support the logic of a finite state machine. It translates input buffer value into finite state machine states. Table 18 describes the scenario arguments.

Table 18: SM-ModifyInBuffer-as-InBufferChanges

Argument Name	Description
Input Buffer Value 1 or Regular Expression	The new input buffer value.
...	...
Input Buffer ValueN or Regular ExpressionN	The new input buffer value.

SM-Action

The SM-Action scenario is used to execute specific actions if the conditions are true. The following actions are supported:

- Print text messages into console log.
- StopPlayRecord to stop play file or record file procedure.

- StopStream to terminate RTP stream.
- CustomEvent to print the time-stamp for the specified value of the in-buffer to the report file .
- HungUp to release the call if certain conditions are specified.

Table 19 describes the SM-Action scenario arguments.

Table 19: SM-Action

Argument Name	Description
Input Buffer Value 1 or Regular Expression	<Trace message>, StopPlayRecord, or StopStream.
...	...
Input Buffer ValueN or Regular ExpressionN	<Trace message>, StopPlayRecord, or StopStream.

SM-DelayedAction

The SM-DelayedAction scenario provides the ability to change the input buffer value after a specified delay. If the conditions are true, the corresponding action is executed. CCAS can execute any action step by step without DTMF input. The delay action can be canceled if the cancel condition is true. For example, this scenario can send consecutive DTMF sequences with delays in between. It can stop playing the file after a time or can start playing sa file after a specified delay. Table 20 describes the scenario arguments.

Table 20: SM-DelayedAction

Argument Name	Description
StartDelayCondition	The input buffer value or regular expression.
StopDelayCondition	The input buffer value or regular expression.
Delay	The duration, in seconds, of the delay.
SetInBufferTo	The value stored in the input buffer after the scenario is complete.

SM-FinalAction

The SM-FinalAction scenario is used to validate voice application responses and data preparation for reporting. CCAS validates voice application responses by comparing the value of the input buffer with the value from the `CheckBuffer` argument. The result of this comparison can be stored in the report file (column `InBufErr`), and output to the console window.

Each completed call has RTP related data and collected counters (for example, ANI) which is populated to the `Script Engine` tab. It can also be stored in the `ScenarioName_<Number>_ <TimeStamp>` report file which allows the data to be exported to data analyzing tools (MS Excel).

Table 21: SM-FinalAction

Argument Name	Description
PrintRTPReport	Indicates whether to output RTP data. • Yes • No
CheckInBuffer	The expected value of the `inBuffer` used to compare with actual (resulted) inBuffer value. The result of this compare value is stored in corresponding record in the report file.

Scenario Example

This section provides the configuration for an example scenario.

Agents with SIP Phones Receive Call from the Media Gateway

In this example, the CCAS simulates incoming SIP calls from the Media Gateway. A total of 100 calls are generated at a call rate of five calls per second. The SIP calls are directed to the SIP Server that is running on the `test1` host with an open communication port of `5060` and destination Route Point is configured as `9001`. The Universal Routing Server (URS) uses a strategy that routes calls to the ready agent SIP phone. Figures 12, 13, and 14 show how the example is configured.

Servers | Scenarios | ScriptEngine

	ServerName	Type	Host	Port	Options Name	Options Value
1	sip	SIP	test1	5060	Transport	UDP
					SaveReport	no
					SIP_Trace	on
2	CLIENT_SM	StreamManage	localhost	7777	DTMF_Duration	500
					DTMF_Delay	0
					SIPFROM_DataFil	Language.dat
					SIPTO_DataFile	none
					DataFile	none
					SM_Trace	on
					SaveRTPReport	yes
3	ts1	T-Server	test1	6000	AppName	aaa
					Password	ppp
4						

Figure 12: Server Tab Configuration

Servers | Scenarios | ScriptEngine

	ServerName	Sim.Scenario	Arguments Name	Arguments Value
1	sip	SIP-MediaGateWay-Call-Talk-Release	SM_Name	CLIENT_SM
			RequestURI	sip:44651@frbr
			MG_ListenSIPF	5090
			ReferDest	no
			ExtDNmin	9992000
			ExtDNmax	9992100
			TalkTime	30
			CallPerSecond	5
			Order	100
2	sip	SIP-WaitForCall-Answer-WaitForRelease	SM_Name	SIP_Only
			DNSIPPortMin	10000
			DNSIPPortMax	10100
			DNmin	1000
			DNmax	1100
			MaxTalkTime	120
3	ts1	Shift_Login_Logout	DNmin	1000
			DNmax	1100
			LoginQueue	9001
			LoginIDmin	1000
			LoginIDmax	1100
			ShiftStartTime	08:00:00
			ShiftDuration	12:00:00
			ShiftCycle	24:00:00

Figure 13: Scenario Tab

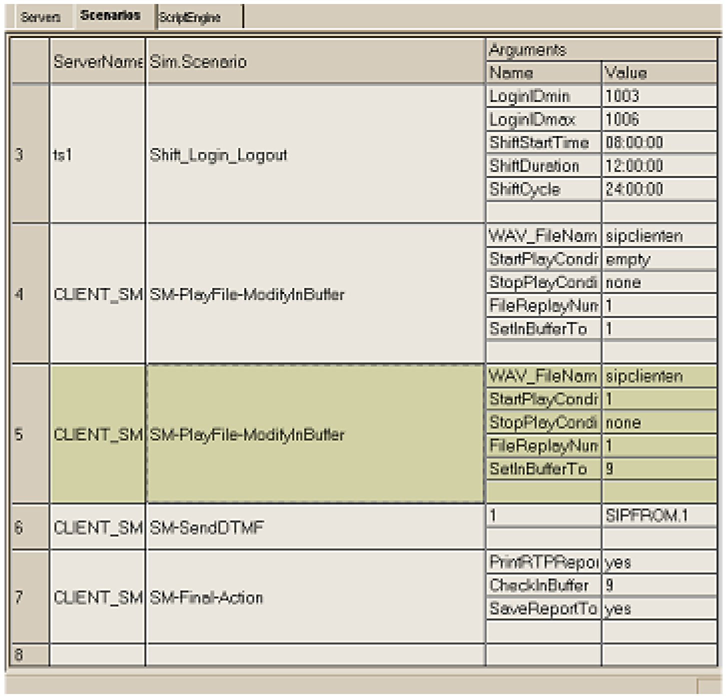


Figure 14: Scenario Tab (Continued)

Note: The `Shift_Login_Logout` scenario requires an initial login/ready procedure, so that URS can route calls to available agents.

Reporting Components

The SM-Final_Action scenario can store session information in a comma delimited file. This file contains a record for each call (session); each record contains the following information:

- The CallID for this session.
- The action time.
- The result of the input buffer verification.
- RTP stream related information:
  - The number of packets sent.
  - The number of packets received.
  - Any errors that occur.
  - Jitter.

Column names are defined in a corresponding header file. The data allows you to calculate the delay between SIP messages, the number of resubmitted SIP messages, the delay between sent and received DTMF tones, and ability to check voice quality for each call.

An example Microsoft Excel template with simple statistical analysis (see Figure 15) accompanies the simulator.

For demonstration purposes an MS Excel template for simple statistical analysis of this file was created and can be found in the reporting folder. To use the spreadsheet, open a header file, copy the contents of it to a data file and save the resulting file. Open the excel spreadsheet, double click clear (and press yes on the resulting dialog), double click browse, select the file you saved, the double click read. This will load the data into the excel spreadsheet.

The initial page of the spread sheet provides an interface to load a results file and calculates essential statistical dependences as shown bellow. All green fields are active buttons and you need to double click on them for action initiation.

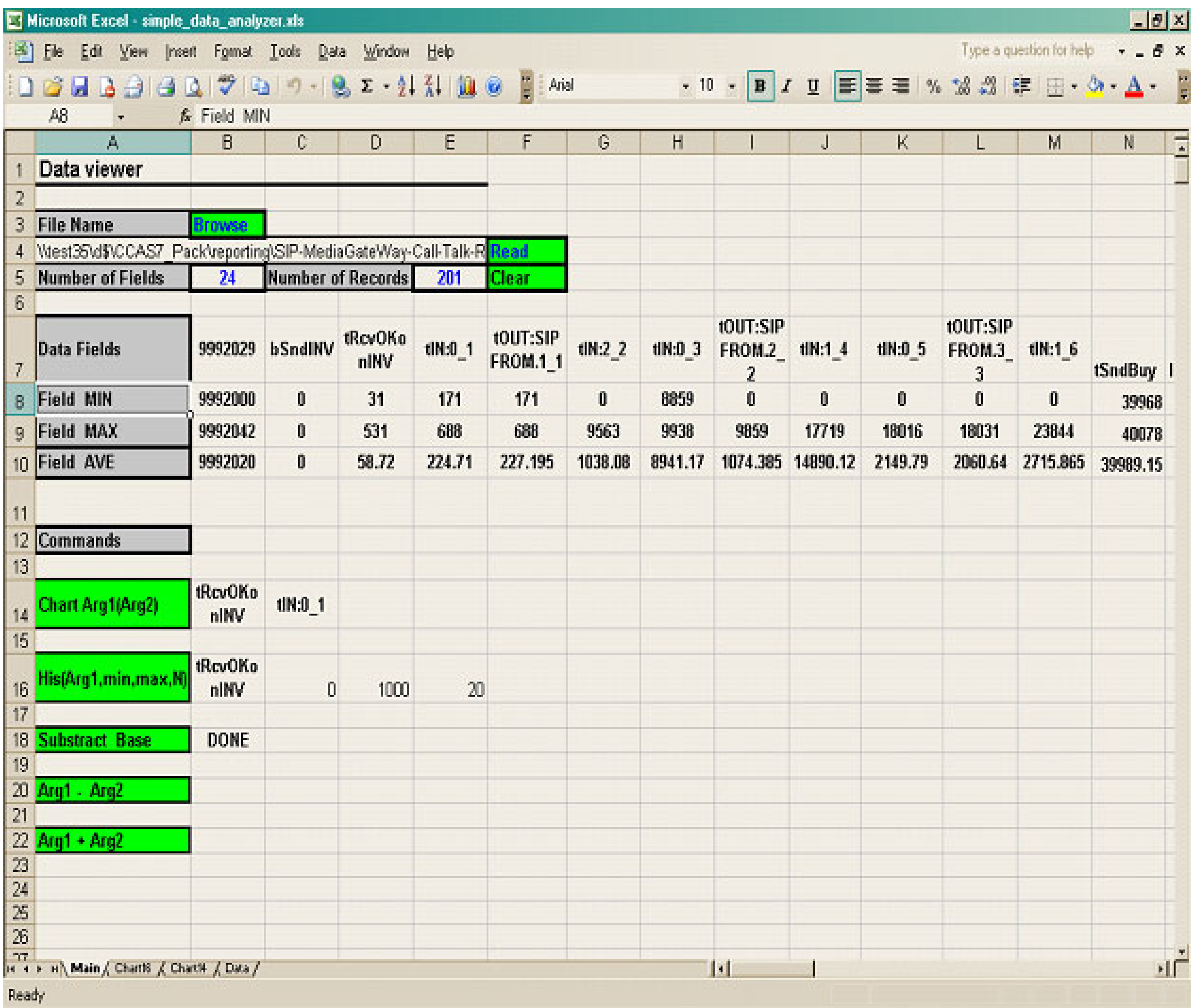


Figure 15: Report Spreadsheet Example

Troubleshooting

You may receive error messages in the case of invalid arguments, performance limitations, or other problems. Such problems may cause the simulator to stop running. If this happens, examine the T-Server and CCAS logs, and check the state of the Switch Simulator. When you have found the problem and fixed it:

- Make sure that all previous calls are released (click `Release All Calls` in Test Simulator Interface, `PbxSummary` tab).
- Restart the simulation from the beginning.

Appendix

PBX Commands

This Appendix describes PBX commands, including the following topics:

Overview

You can use PBX commands to create virtual equipment and to control the Test Simulator by allowing functions for dialing, answering, transferring, agent logins, and so forth. Typically, you control the Test Simulator through the Test Simulator Interface. However, PBX commands may be also be used:

- By means of a script file (similar to a DOS .bat file or a Unix shell script)
- By using the Test Simulator Server Console

This Appendix provides a reference section for all commands available to the Test Simulator, which may be issued from the Test Simulator Console or written into a flat file such as an equipment file.

Equipment Files

Equipment files are text-based files that tell the Test Simulator how it is to be configured; for example, what DNs (ACD positions, Queues, RPs) and Agent IDs are associated with it. These files can include any of the create or console commands described in "Simulator Create Commands" on page 72 and "Simulator Console Commands" on page 74.

Warning! Use equipment files only in the absence of the Configuration Layer.

Simulator Create Commands

The switch equipment file uses Create statements to configure switch attributes. These instructions are executed automatically when the switch simulator starts. The “–c” argument in command line, specifies the equipment file name. Table 22 describes the create commands.

Table 22: Create Commands

Command	Description	Syntax and Examples
create_dn	Creates the agent’s DN, which simulates equipment with type ACD Position or Extension. It is also used to created a LinkDN that is a virtual equipment type used in multi-site exchanges.	`create_dn <DN> <queue> <remote_port> <agent_ID> <force_answer> <after_release>` Where: `<DN>`—possible values are 0-99999. `<queue>`—for ACD position, queue number this DN will log in to; for LinkDN, CDN on remote switch. `<remote_port>`—for agent phone, 0; for LinkDN, port number (same as option -p for remote simulator). `<agent_ID>`—for agent phone, agent ID; for LinkDN, host where remote simulator is operating. <force_answer>—defines how calls will be answered: 0 = call is automatically answered, 1= call is only answered if explicitly commanded. `<after_release>`—defines if agent will automatically be made available after release of a call: 0 = agent is made available, 1 = agent is not made available.

Table 22: Create Commands (Continued)

Command	Description	Syntax and Examples
create_dns	Creates a range of DNs.	create_dns <start> <count> <queue> 0 <agent_ID> <force_answer> <after_release> Where: <start>—first DN number. <count>—number of DNs in range. <queue>—queue number these DNs will log in to. 0—must be included for compatability. <agent_ID>—for agent phone, agent ID; for LinkDN, host where remote simulator is operating. <force_answer>—defines how calls will be answered: 0 = call is automatically answered, 1= call is only answered if explicitly commanded. <after_release>—defines if agent will be made available after release of a call: 0 = agent is made available, 1 = agent is not made available.
create_cdn	Creates a routing point.	create_cdn <CDN> <queue> <call_gen> <DNIS> <fq> Where: <CDN>—defines routing point in the switch simulator; possible values are 0-99999. <queue>—queue associated with this routing point; 0 if Interaction Routing Designer will be used. <call_gen>—sets inbound call generation: 1 = auto call generation, 0 = no generation. <DNIS>—simulates DNIS numbers associated with this routing point. <fq>—sets rate of inbound call generation in calls per second.

Table 22: Create Commands (Continued)

Command	Description	Syntax and Examples
create_cdns	Creates a range of routing points.	`create_cdns <start> <count> <queue> <call_gen> <DNIS> <fq>` Where: `<start>`—first routing point in the sequence. `<count>`—number of routing points to be defined. `<queue>`—queue associated with this range of routing points; 0 if Interaction Routing Designer will be used. `<call_gen>`—sets inbound call generation: 1 = auto call generation, 0 = no generation. `<DNIS>`—simulates DNIS numbers associated with this routing point. `<fq>`—sets rate of inbound call generation in calls per second.
create_trunk	Creates a trunk number. A trunk number is necessary to simulate inbound and outbound calls including multi-site calls (between simulators). Each inbound or outbound call allocates one trunk. You must create enough trunks to handle all active inbound and outbound calls.	`create_trunk <trunk>` Where: `<trunk>`—any number between 1000 and 9999.
create_trunks	Creates a range of trunk numbers.	`create_trunks <begin_trunk_number> <how many trunks>` Where: `<begin_trunk_number>`—first trunk number in range. `<how many trunks>`—number of trunks in range.

Simulator Console Commands

The simulator can execute a number of commands from a local or a remote console. These commands are used to support the simulation of manual actions on DNs (without CTI request from T-Server). For example, the “ready” instruction simulates setting the DN into a ready state.

Console commands are necessary to simulate PSTN connections in multi switch environments. There is set of commands to control simulation process and provide information about the simulator's current state.

It is possible to include console commands as instructions in an equipment file. These instructions are executed automatically when the switch simulator starts. The "–c" argument in command line, specifies the equipment file name.

This section describes how to use the console commands.

Inbound Call Control Commands

Table 23 lists and describes the inbound call control commands.

Table 23: Inbound Call Control Commands

Command	Description	Syntax and Examples
go	Run mode. Starts call generation.	`go[<Cnt>t]` Where: `<Cnt>` = the number of calls to generate (optional). If not used, the simulator will generate inbound calls while in Run mode.
stop	Stop mode. Stops call generation.	`stop`
change_rate	Controls the call generation rate.	`change_rate <DN><Rate>` Where: `<DN>` = the ACD Queue or Route Point. `<Rate>` = the number of calls per second.

place_int_call	Controls call generation from a console or external application to a specific DN.	`place_int_call <DN><Collected Digits><ANI>`
		Where:
		`<DN>` = the ACD Queue or Route Point.
		`<Collected Digits>` = a string of attached data.
		`<ANI>` = the external origination number.
		Example:
		`place_int_call 2200 123456789 55555555555`
		In this example, a call from 55555555555 is placed to Route DN 2200 with 123456789 as attached data.

Forward Control Commands

Forward control commands simulate switch functionality to forward calls to other destinations. It is possible to include forward commands into equipment files. In this case, they will be executed automatically upon switch simulator start. Table 24 lists and describes the forward control commands.

Table 24: Forward Control Commands

Command	Description	Syntax and Examples

set_forward	Forwards the ACD queue or agent DN to another destination with the following conditions `[<COND>]`:
	• 1—Forward if busy. Works on the agent DN only.
	• 2—Forward if no answer. Works on the agent DN only.
	• 3—Forward unconditionally (default if skipped). Works on the ACD queue and the agent DN.
		`set_forward <DN1><DN2>[<COND>]`
		Where:
		`<DN1>` = the agent phone number or ACD queue number.
		`<DN2>` = the forwarding destination.
		`<COND>` = the forward condition.
		Example:
		`set_forward 610 605 2`
		In this example, the command instructs the simulator to forward calls from 610 to 605 if there is no answer on 610.

cancel_forward	Cancels call forwarding.	`cancel_forward<DN>` Where: `<DN>` = the agent's phone number.

Agent Phone Status Control Commands

Agent phone status control commands simulate the agent's manual actions for setting the phone state. Table 25 lists and describes the agent phone status control commands.

Table 25: Agent Phone Status Control Commands

Command	Description	Syntax and Examples

ready	Sets the status of the phone to a ready state.	`ready<DN>`
		Where:
		`<DN>` = the agent's phone number.
		Example:
		`ready 610`
		This example sets the status of phone 610 to the ready state.

idle	Sets the status of the phone into a not ready state.	`idle<DN>`
		Where:
		`<DN>` = the agent's phone number.
		Example:
		`idle 610`
		This example sets the status of phone 610 to the not ready state.

wrapup	Sets the status of the phone into a busy state.	`wrapup<DN>`
		Where:
		`<DN>` = the agent's phone number.
		Example:
		`wrapup 610`
		This example sets the status of phone 610 to the busy state.

Table 25: Agent Phone Status Control Commands (Continued)

Command	Description	Syntax and Examples

login	Logs in an agent to a specified DN with a specified agent ID.	`login<DN><AG_ID><Queue>`
		Where:
		`<DN>` = the agent's phone number.
		`<AG_ID>` = the agent's identification code.
		`<Queue>` = the ACD Queue.
		Example:
		`login 610 16000 8001`
		In this example, the agent with the agent ID 16000 logs into queue 8001 on DN 610.

logout	Logs out an agent from the queue.	`logout<DN>`
		Where:
		`<DN>` = the agent's phone number.
		Example:
		`logout 610`
		In this example, the agent associated with DN 610 logs out.

Agent Phone Call Control Commands

The agent phone call control instruction is partially supported commands. Usually it is not necessary to use these instructions to conduct testing within the Genesys Framework. For different types of switch simulators, there is a different set of supported agent phone control instructions.

Table 26 lists and describes the agent phone call control commands.

Table 26: Agent Phone Call Control Commands

Command	Description	Syntax and Examples
call	Initiates a call to any destination from the agent's phone.	call<DN1><DN2> Where: <DN1> = the agent phone number (origination). <DN2> = the call destination. Example: call 610 1234567890 In this example, the command instructs the simulator to make an outbound call from agent 610 to 1234567890.
line	Answers an active call or retrieves a call from hold on an individual line of an agent's phone. If a call arrives on line0/line2/line1, this instruction changes the status of this call from ringing to active. If the call on line0/line2/line1 is placed on hold, this instruction returns the call to an active status.	line0<DN> line1<DN> line2<DN> Where: <DN> = the agent's phone number.
release	Releases all active calls currently present on the agent's phone.	release<DN> Where: <DN> = the agent's phone number. Example: release 610 In this example, the command instructs the simulator to release all the active calls for agent 610.

Table 26: Agent Phone Call Control Commands (Continued)

Command	Description	Syntax and Examples

conference	Completes a conference call.	`conference<DN>`
		Where:
		`<DN>` = the agent's phone number.
		Example:
		`hold 610`
		`call 610 702`
		`conference 610`
		In this example, the instruction sequence adds the active call on agent phone number 610 to the conference call associated with phone number 702. The hold call instructions and establishing of the consult call must be issued first.

transfer	Completes a transfer.	`transfer<DN>`
		Where:
		`<DN>` = the agent's phone number.
		Example:
		`hold 610`
		`call 610 702`
		`transfer 610`
		In this example, the instruction sequence makes a two-step transfer of an active call on agent phone number 610 to the phone number 702. The hold call instructions and establishing of the consult call must be issued first.

Simulator Status Information

Table 27 lists and describes the simulator status commands.

Table 27: Simulator Status

Command	Description	Syntax and Examples
info_call	Displays call information.	info_call[<call_id>] Where: <call_id> = the caller's number (ANI). If the number is not specified, information for all calls will be displayed. Example: info_call 5040 Output: CALL_ID=5040 ORIG=610 DEST=705 In this example, the output displays that inbound call number 5040 originated at DN 610 and had a destination of DN 705.
info_dn	Displays the current status of the specified DN.	info_dn[<DN>] Where: <DN> = the agent's phone number. If the number is not specified, information for all DNs will be displayed. Example: info_dn 610 Output: DN=610 line0=0 line2=11 ACTIVE QUEUE=8101 AG_ID=1701 BUSY AUTOANSWER=YES POSTREADY=YES In this example, agent phone 610 has the following characteristics: • There is an active call on line2, with CallID 11. • There are no calls on hold. • The agent's status is Logged In. • The DN's status is Busy. • The autoanswer and postready options have been selected.

Table 27: Simulator Status (Continued)

Command	Description	Syntax and Examples

info_trunk	Displays the current status of the specified trunk DN.	`info_trunk[<trunk>]`
		Where:
		`<trunk>` = the trunk number. If the trunk number is not specified, information for all trunk DNs will be displayed.
		Example:
		`trunk_info 1211`

info_queue	Displays the current status of the specified queue number.	`info_queue[<queue>]`
		Where:
		`<queue>` = the ACD queue number. If the queue number is not specified, information for all ACD queues will be displayed.
		Example:
		`info_queue 5000`

info_cdn	Displays the current status of the specified route point.	`info_cdn[<cdn>]`
		Where:
		`<cdn>` = the routing point number. If the routing point number is not specified, information for all route points will be displayed.
		Example:
		`info_cdn 5500`

Supplements

Related Documentation Resources

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

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.
- Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at `http://genesyslab.com/support`.

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

- *Genesys Supported Operating Environment Reference Manual*
- *Genesys Supported Media Interfaces Reference Manual*

Consult these additional resources as necessary:

- *Genesys Hardware Sizing Guide,* which provides information about Genesys hardware sizing guidelines for the Genesys 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 28 describes and illustrates the type conventions that are used in this document.

Table 28: Type Styles

Type Style	Used For	Examples
Italic	• Document titles • Emphasis • Definitions of (or first references to) unfamiliar terms • Mathematical variables Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on page 86).	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 . . .

Table 28: Type Styles (Continued)

Type Style	Used For	Examples
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

G

H

I

L

M

N

O

P

Q

R

S

T

V