GENESYS®

Call Concentrator 7

Deployment Guide

About Genesys

Genesys Telecommunications Laboratories, Inc., a subsidiary of Alcatel, is 100% focused on software for call centers. Genesys recognizes that better interactions drive better business and build company reputations. Customer service solutions from Genesys deliver on this promise for Global 2000 enterprises, government organizations, and telecommunications service providers across 80 countries, directing more than 100 million customer interactions every day. Sophisticated routing and reporting across voice, e-mail, and Web channels ensure that customers are quickly connected to the best available resource—the first time. Genesys offers solutions for customer service, help desks, order desks, collections, outbound telesales and service, and workforce management. Visit 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 CD, 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.

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 following regional numbers:

Region	Telephone	E-mail
North and Latin America	+888-369-5555 or +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44 (0) 118 974 7002	support@genesyslab.co.uk
Asia Pacific	+61 7 3368 6868	support@genesyslab.com.au
Japan	+81-3-5649-6821	support@genesyslab.co.jp

Prior to contacting technical support, please refer to the *Genesys Technical Support Guide* for complete contact information and procedures.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the ***Genesys 7 Licensing Guide*.**

Released by

Genesys Telecommunications Laboratories, Inc.
http://www.genesyslab.com

Document Version: 70cc_dep_4-2005_v3.01

Table of Contents

Chapter

1 About This Document

Welcome to the *Call Concentrator 7 Deployment Guide*. This guide introduces you to the concepts, terminology, and procedures relevant to Call Concentrator installation, configuration, starting, and stopping.

Call Concentrator 7 is a Reporting product that collects and processes call-based data on activity in your enterprise. It draws on information from Configuration Server and T-Server® to create detailed representations of each leg of a call and of the call as a whole. Call Concentrator then stores this data in various tables, several of which you can customize to best suit your interaction management environment. These tables provide data that your reporting applications can take up, perform additional processing on, and present.

This guide is valid only for the 7.0 release(s) of this product.

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

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

Intended Audience

This guide, primarily intended for system administrators, 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.
- Editing and running database scripts and managing database tables.

You should also be familiar with Genesys Framework, in particular Management Layer, Load Distribution Server (LDS), T-Server, and DB Server.

Chapter Summaries

In addition to this opening chapter, this guide contains these chapters:

- Chapter 2, "Configuration" on page 11, explains how to create the Call Concentrator Application object using Configuration Manager. It also includes basic application configuration instructions.
- Chapter 3, "Customize Your Configuration" on page 15, contains detailed descriptions of all `Options` tab options in your Call Concentrator application. Use this chapter to enter settings for key Call Concentrator functions and log options.
- Chapter 4, "Installation" on page 35, gives step-by-step instructions for installing Call Concentrator on Windows and Unix platforms..
- Chapter 5, "Deploying SQL Scripts" on page 45, explains how to run the scripts to create the necessary database tables and introduces script customization.
- Chapter 6, "Starting and Stopping" on page 49, describes the various procedures for starting and stopping Call Concentrator on both Windows and Unix platforms.
- Chapter 7, "Uninstalling Call Concentrator" on page 55, explains how to remove Call Concentrator from both Windows and Unix platforms.

Document Conventions

This document uses some stylistic and typographical conventions with which you might want to familiarize yourself.

Version Number

A document version number appears at the bottom of the inside front cover of this guide. Version numbers change as new information is added to this guide. Here is a sample version number:

70fr_ref_09-2003_v1.00

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

Type Styles

Italic

In this document italic is used:

- When a term is being defined.

  Example

  - *A customary and usual practice* is one that is widely accepted and used within a particular industry or profession.
- For emphasis. For example, "Do *not* use this value for this option."
- For variables, for example, *x* +1 = 7 where *x* stands for . . .

Monospace

A monospace font, which is shown in the following examples, is used for:

- All programming identifiers and GUI elements—*except* for instances of these occurring in tables and figures. This convention includes the *names* of directories, files, folders, paths, scripts, dialog boxes, options, fields, text and list boxes, all buttons including radio buttons, check boxes, commands, tabs, CTI events, and error messages; the values of options; logical arguments and command syntax; and code samples.

  Examples

  - Select the `Show variables on screen` check box.
  - Click the `Summation` button.
  - On the `Properties` dialog box, enter the value for the host server in your environment.
  - In the `Operand` text box, enter your formula.
  - Click `OK` to exit the `Properties` dialog box.
  - The following table presents the complete set of error messages T-Server distributes 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.
- For any text the user must manually enter during a configuration or installation procedure:

Example

- Enter `exit` at the command line.

Correction of Errors in Screen Captures

Screen captures taken from the product GUI (graphical user interface) and used in this document may sometimes contain a minor spelling, capitalization, or grammatical error. The text accompanying and explaining the screen captures corrects such errors.

Use of Square Brackets

In any logical arguments, commands, and programming syntax presented in this document, square brackets are used to indicate that a particular parametric value is optional. That is, the value is not required to resolve a command, argument, or programming syntax. The customer/user decides whether to supply a value and what that value is. Here is a sample:

```
smcp_server -host [/flags]
```

Use of Angle Brackets

Angle brackets are used to indicate that a value in a logical argument, command, or programming syntax is required, but that the user must supply the data for the value. Because the value is specific to an individual enterprise—for example, DNs or port numbers—the program cannot predict (that is, program in) what the value is. Here is a sample:

```
smcp_server -host <confighost>
```

Related Resources

Consult these additional resources as necessary:

- *Call Concentrator 7 Getting Started Guide,* which introduces Call Concentrator 7, presents an overview of its architecture, and provides deployment-planning recommendations.
- *Call Concentrator 7 Reference Manual,* which provides detailed explanations of the Call Concentrator database tables and instructions for customizing them, when appropriate.
- *Technical Reference Guide for the Reporting 6.5 Release*, which explains the Genesys call model in detail.
- The *Framework 6.5 Load Distribution Server User's Guide*, which explains how to use LDS with Call Concentrator.
- The documentation provided for your Genesys T-Server(s).

- The *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library CD and which provides a comprehensive list of the Genesys and CTI terminology and acronyms used in this document.
- The *Genesys Migration Guide*, also on the Genesys Documentation Library CD, which contains a documented migration strategy for Genesys product releases 5.x and later. Contact Genesys Technical Support for additional information.
- The Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at http://genesyslab.com/support.

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

- *Genesys 7 Supported Operating Systems and Databases*
- *Genesys 7 Supported Media Interfaces*

Genesys product documentation is available on the:

- Genesys Technical Support website at http://genesyslab.com/support.
- Genesys Documentation Library CD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.

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 book 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.

Chapter

2 Configuration

This chapter describes how to configure your Call Concentrator application. You must configure the application before performing the Call Concentrator installation. The chapter includes these sections:

Before Starting

You use Configuration Manager to create and configure a Call Concentrator Application object. Before starting Configuration Manager, be sure that the following components are running:

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

Import the Application Template

You only need to import the application template one time, no matter how many Call Concentrator applications you are creating. If you have already imported the template into Configuration Manager, skip to “Create the Application” on page 12.

To import the Call Concentrator template:

1. In Configuration Manager, right-click the `Application Template` folder, located under `Environment`.
2. Select `Import` from the shortcut menu that appears.

3. Browse to the location of the Call Concentrator application template, `Call_Concentrator_700.apd`. The template is located on your installation CD, or might also have been copied to your network.
4. Click `OK` to import the template.

Note: Do not change any settings on the template. You must configure settings separately for each Call Concentrator application.

Create the Application

Use the imported application template as a source for your application. To create the application:

1. Right-click the `Applications` folder and select `New` from the shortcut menu that appears.
2. Browse to the application template and select it.
3. Click `OK` to confirm.

Configure the Application

After creating your application, you must change some settings to conform to your environment:

General Tab

1. Open the application. On the `General` tab, enter an application name for this Call Concentrator object.
2. Make sure the `State Enabled` box is selected.

Tenants Tab

3. If your environment consists of multiple tenants, select the `Tenants` tab and list the tenants that will use this Call Concentrator application.

Server Info Tab

4. Select the `Server Info` tab and enter the name and port number of the host machine on which Call Concentrator will reside.

Start Info Tab

5. Select the `Start Info` tab and specify the following information:
   - **Working directory**—the path to the directory where you will install Call Concentrator, for example, `C:\GCTI\Reporting\Call Concentrator70`
   - **Command line**—the path to the Call Concentrator application executable, for example, `C:\GCTI\Reporting\CallConcentrator70\ccon.exe`
   - **Command-line arguments**—the parameters used when starting or restarting Call Concentrator. For example:

```
-host <Configuration Server host name> -port
<Configuration Server port number> -app <Call Concentrator
```

application name> -l <license_server_port@license_server_host>

Startup time—the time interval, in seconds, that Local Control Agent (LCA) is to wait for confirmation of the successful start of the Call Concentrator application, for example, 15. This setting has no impact if the Auto-Restart check box (see below) is cleared.

Note: The initialization time Call Concentrator needs on startup depends mainly on the volume of the configuration data it must read from the Configuration Database. If you intend to use Management Layer to start or automatically restart Call Concentrator, you should experimentally determine the optimal value of the setting.

- **Shutdown time**—the time interval, in seconds, that it takes Call Concentrator to shut down, for example, 30 seconds.
- **Auto-Restart**—selecting this box allows the LCA to restart Call Concentrator automatically in case of an application failure.

Connections Tab

6. Select the Connections tab and click Add to specify all the components to which the Call Concentrator application should connect. The list must include:
   - A *Database Access Point* (DAP) that points to the database where Call Concentrator will write data.

Note: The DAP Application object should point to the DB Server configured for your database.

- One or more *T-Servers* from which Call Concentrator will collect data.
- *Message Server* for Genesys logging. For information on common and Call Concentrator-specific log messages, see the *Framework 7 Combined Log Events Help*.
- *Load Distribution Server (LDS)* if you plan to use LDS.

Note: For detailed information on using Call Concentrator and LDS together, see the *Framework 6.5 Load Distribution Server User's Guide*.

7. To enable Advanced Disconnect Detection Protocol (ADDP) between Call Concentrator and the other servers, specify addp as the Connection Protocol when configuring the connection between applications. ADDP helps detect connection failures on both the client and server side.

Note: When using ADDP, you must set values for the Local Timeout, Remote Timeout, and Trace Mode properties.

Options Tab

8. Select the `Options` tab and specify values for the Call Concentrator configuration options. These options are divided into two sections:
   - The `callconcentrator` section specifies settings controlling core functionality.
   - The `Log` section specifies the options controlling Call Concentrator's logging feature.
   - The optional, user-created `license` section. You can use this section to specify the location of your license file. The settings you specify using this option are superceded by any licensing information entered on the command line.

Setting well-planned option values is the most complex part of the Call Concentrator deployment process. If you want, you can leave the `Options` tab settings at their default values until you complete the installation, and then return to the application to customize your settings.

Note: All configuration options are explained in detail in Chapter 3, "Customize Your Configuration," on page 15.

9. When you have finished entering configuration settings, click `OK`.

Making Dynamic Configuration Changes

You can change settings or configure additional options for your Call Concentrator application object at any time. See Chapter 3, "Customize Your Configuration," on page 15 for more information.

You might enter changes during runtime or during the period between two subsequent runs. If you make a change in runtime, Call Concentrator may adopt the change, depending on whether the option settings can be dynamically updated. If an option does not recognize dynamic changes, restart Call Concentrator to make the new settings take effect.

Note: Dynamic configuration change information is described for each option in Chapter 3, "Customize Your Configuration," on page 15.

For detailed information about configuring application objects using Configuration Manager, refer to *Framework 7 Configuration Manager Help*.

Chapter

3 Customize Your Configuration

This chapter tells you how you can customize your configuration settings to best suit your environment. It includes these sections:

Configuration Options

The Call Concentrator configuration options are listed below in alphabetical order. Configuration options are specified on the `Options` tab of the Call Concentrator application. The description of each option consists of:

- **Default Value**—the value that Call Concentrator uses if you do not specify an alternate option value.
- **Valid Values**—the set of permissible values for the option. In this part of the description, a Boolean specification may indicate any of the following values: `true, on, yes, false, off, no`. The values are case insensitive. The first three values are equivalent to `on` and the last three are equivalent to `off`.
- **Dependencies**—the options that may impact or modify the current option's functionality.
- **Dynamic Change**—whether Call Concentrator will respond dynamically to changes made to the configuration option while the Call Concentrator application is running. `Yes` signifies that Call Concentrator will dynamically adjust to the new value of the option. `No` signifies that Call Concentrator must be restarted for the change to take effect.

- **Functional description**—what the option does depending on the value you set and any additional information that you might need to understand how to decide on a value for the option that is suitable for your environment.

call_concentrator Option Descriptions

Select the settings most appropriate for your enterprise environment. The options are listed in alphabetical order.

AddSwitchCallID

Default Value: `off`
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

If this option is set to `on,` Call Concentrator assumes that the SCDR table has an additional `SSwitchCallID` field adjacent to the `SCallType` field. Call Concentrator fills this field with the single call identifier assigned by the physical switch (retrieved from T-Events).

Note: To enable this option, run the appropriate SQL script to create the SCDR table. Otherwise, Call Concentrator reports an error and quits. For more information, see Chapter 5, “Deploying SQL Scripts” on page 45.

AgentRecordUserTypes

Default Value: No default value
Valid Values: See “Customizing the AREC Table” in the “Associated Records” chapter of the *Call Concentrator 7 Reference Manual*.
Dependencies: `AgentUserFields`
Dynamic Change: No

May be used in conjunction with the `AgentUserFields` configuration option to specify that custom agent states should be stored in the AREC table.

AgentStatuses

Default Value: No default value
Valid Values: Comma-separated list of integers
Dependencies: `CustomAgentStatusesOnly`, `HeldAgentStatusesOnly`
Dynamic Change: No

Lists the codes of statuses that should be recorded into the AREC table. If the option is set, only statuses with the codes listed are recorded and the values of the `CustomAgentStatusesOnly` and `HeldAgentStatusesOnly` configuration options are ignored.

AgentUserFields

Default Value: No default value
Valid Values: See “Custom the AREC Table” in the “Associated Records” chapter of the *Call Concentrator 7 Reference Manual*.
Dependencies: AgentRecordUserTypes
Dynamic Change: No

May be used in conjunction with the AgentRecordUserTypes configuration option to specify that custom agent states should be stored in the AREC table.

arec

Default Value: on
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

Enables or disables Call Concentrator output to the AREC table.

BindThreshold

Default Value:1
Valid Values: Positive integers
Dependencies: UseBinding, BindTimeout
Dynamic Change: Yes

The BindThreshold and BindTimeout configuration options govern the use of the DB Server Bulk-Operation feature.

Note: The Data-Binding and Bulk-Operation features are supported only for an Oracle database. Otherwise, the UseBinding, BindThreshold, and BindTimeout configuration options have no effect.

Bulk-operation sets the number of bound records to be inserted into the database immediately. If you set a bind threshold, Call Concentrator does not write data into the database immediately after the data is available. Instead, Call Concentrator holds the bound data records in memory until the limit that has been set with the BindThreshold configuration option has been reached, or the timeout that has been set with the BindTimeout configuration option expires.

Whichever of these two conditions is met first triggers Call Concentrator to write the collected bound data records into the database with a single bulk request to the DB Server. If the request was caused by the BindTimeout expiration, then the number of bound records written may be less than the BindThreshold value.

Note: The BindThreshold configuration option only takes effect if the UseBinding configuration option is set to on.

BindTimeout

Default Value: 300
Valid Values: Positive integers
Dependencies: UseBinding, BindThreshold
Dynamic Change: Yes

In conjunction with the BindThreshold configuration option, controls the usage of the DB Server Bulk-Operation feature. This option sets a timeout, in seconds, to force Call Concentrator to issue a bulk request to the DB Server even if the BindThreshold value has not yet been reached. For more information, see the description of the BindThreshold configuration option.

CC_ID

Default Value: 0
Valid Values: Integers between 0 and the value set in the CCON_COUNT configuration option
Dependencies: CCON_COUNT, ESequenceInitFactor
Dynamic Change: No

The value of this option is used, along with the values of the CCON_COUNT and ESequenceInitFactor options, to calculate the ESequence field of the EVREF and EVDATA tables. The value of the CC_ID configuration option must be different from that in any other Call Concentrator application that is stored in the Configuration Database.

CCON_COUNT

Default Value: 8
Valid Values: Positive integers greater than the number of Call Concentrator objects specified in the Configuration Database
Dependencies: CC_ID, ESequenceInitFactor
Dynamic Change: No

The value of this option is used, along with the values of the CC_ID and ESequenceInitFactor configuration options, to calculate the ESequence field of the EVREF and EVDATA tables.

CountQueuing

Default Value: on
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

Enables/disables a check that Call Concentrator performs after receiving EventDiverted, EventAbandoned, or EventRouteUsed T-Events related to a call. The check is used when a call has been sent to multiple queues to determine if the call, which has generated a call completed event in one queue, is still present in another queue or routing point. If so, depending on the exact scenario, Call Concentrator may keep the call active rather than closing it.

CustomAgentStatusesOnly

Default Value: off
Valid Value: Boolean
Dependencies: AgentRecordUserTypes, AgentUserFields
Dynamic Change: Yes

If this option is set to on, Call Concentrator places in the AREC table only those records that correspond to the custom agent statuses (if any) defined in the AgentRecordUserTypes and AgentUserFields configuration options.

DataFilter

Default Value: off
Valid Values: Boolean
Dependencies: evref, EventData
Dynamic Change: No

Designed to be used in conjunction with the evref and EventData configuration options. If both the evref and DataFilter configuration options are set to on, Call Concentrator records into the EVREF and EVDATA tables only the attached data corresponding to the keys listed in the EventData configuration option. If the evref configuration option is set to on and the DataFilter configuration option is set to off, Call Concentrator records all attached data.

DbDumpCommitFrequency

Default Value: 1
Valid Values: Positive integers
Dependencies: DbDumpCommitTime, SendBeginTransaction
Dynamic Change: Yes

The number of records Call Concentrator writes into the database before it sends a commit request to DB Server. This option only takes effect if SendBeginTransaction is set to on.

DbDumpCommitTime

Default Value: 1
Valid Values: Positive integers
Dependencies: DbDumpCommitFrequency, SendBeginTransaction
Dynamic Change: Yes

Time, in seconds, that Call Concentrator has to write records to the database before sending a commit request to DB Server. This option only takes effect if SendBeginTransaction is set to on.

Note: Both the DbDumpCommitFrequency and DbDumpCommitTime configuration options set limits on the number of records Call Concentrator writes to the database before sending a commit request. The option that applies first is the one used. For example, if the

DbDumpCommitFrequency configuration option is set to 5 and Call Concentrator writes 5 records to the database before DbDumpCommitTime has expired, then Call Concentrator sends a commit request to DB Server.

DbDumpMaxInsertsInProgress

Default Value: 10
Valid Values: Positive integers
Dependencies: None
Dynamic Change: Yes

The number of records Call Concentrator sends to DB Server without receiving a confirmation. Setting a large value for this option improves performance but increases memory consumption.

DbMaxErrorsBeforeShutdown

Default Value: 0
Valid Values: Positive integers
Dependencies: None
Dynamic Change: Yes

Specifies the maximum number of allowable database errors encountered. Call Concentrator terminates when the limit is reached. If set to 0, Call Concentrator continues to function regardless of the number of database errors encountered.

DebugLevel

Default Value: Statistics
Valid Values: All or a comma-separated list including one or more of these debug levels: Init, Server, Debug, Lib, Tel, SQL, Mngmnt, Statistics
Dependencies: DebugStatus
Dynamic Change: Yes

Each debug level specifies a class of messages that a user wants to see in Call Concentrator's execution log. Table 1 describes the message class for each debug level.

Table 1: Debug Level Definitions

Debug Level	Log Message Contents
All	Equivalent to the list consisting of all log levels except Debug
Debug	Additional miscellaneous messages that can be used for troubleshooting

Table 1: Debug Level Definitions (Continued)

Debug Level	Log Message Contents
Init	Messages about the initial configuration of the running instance of Call Concentrator as well as about its dynamic changes during execution
Lib	Messages providing detailed information about the call recognition process (most numerous)
Mngmnt	Messages related to Simple Network Management Protocol (SNMP) support
Server	Messages about common connectivity events like connection to or disconnection from a T-Server or DB Server
Statistics	Messages that record current summary information about the internal state of the Call Concentrator instance (number of calls, pending database requests, and so on)
SQL	Messages relating to the interaction between Call Concentrator and DB Server
Tel	Messages recording T-Events

DebugStatus

Default Values: `on`
Valid Value: Boolean
Dependencies: `DebugLevel`
Dynamic Change: Yes

An obsolete option that turns the logging feature on or off. Genesys recommends using the `Verbose` log option instead. For more information, refer to “Log Options” on page 33.

DeleteTime

Default Value: `10`
Valid Values: Positive integers
Dependencies: None
Dynamic Change: Yes

Occasionally, Call Concentrator cannot determine if a call is completed. The `DeleteTime` configuration option sets a time interval, in seconds, during which Call Concentrator holds this call in memory before recording it into the database.

If the time interval expires and the call is still in a questionable state, Call Concentrator writes it to the database and deletes the call information from memory. If the call then continues with the same connection ID, Call

Concentrator creates a new record in the GCDR and SCDR tables with a duplicate connection ID.

- If you find records for two calls with the same `ConnID` in the GCDR and SCDR table, try increasing the `DeleteTime` value. This may eliminate records with duplicate `ConnID`s.

  For example, if a call is sent to an IVR not monitored by Genesys, Call Concentrator cannot track what happens to it there. If you set the `DeleteTime` configuration option to double the maximum IVR delay, Call Concentrator may delay closing the call long enough to see the call again after the IVR has handled it. Call Concentrator then recognizes it as a single call and writes one record to the database.

Genesys recommends that you do not use the `DeleteTime` configuration option in simple call models where it is immediately clear that a call has ended, for instance, when `EventAbandoned` is received for a two-party call.

Note: The `DeleteTime` configuration option also affects postcall attached data. For attached data to be stored with a call after the call has ended, the user must send attached data from the desktop application before the time specified by the `DeleteTime` configuration option has expired.

DeleteTimeFNA

Default Value: `300`
Valid Values: Positive integers
Dependencies: None
Dynamic Change: Yes

Similar to the `DeleteTime` configuration option except that it controls Call Concentrator's behavior with respect to those released calls that have been forwarded after the destination DN does not answer. When the call is released with the call state indicating forwarding, the call is kept in memory for at least the number of seconds specified by `DeleteTimeFNA`. This is so that if it reappears somewhere else (the destination of the forwarding), further events with the same connection ID should be considered a continuation of the original call.

DNRegistrationDelay

Default Value: `10`
Valid Values: Positive integers
Dependencies: `DNRegistrationFactor`
Dynamic Change: Yes

Sets the duration of the break, in seconds, that Call Concentrator will take between two sequential attempts to register a group of DNs as specified by the `DNRegistrationFactor` configuration option. Also sets the time interval between receiving a notification from the Configuration Server that a new DN has been added to the Configuration Database and the moment when Call Concentrator is to register the newly-added DN with T-Server.

Note: This option goes into effect only if the DNRegistrationFactor configuration option is set to a positive value.

DNRegistrationFactor

Default Value: 0
Valid Values: Positive integers
Dependencies: DNRegistrationDelay
Dynamic Change: Yes

Improves the scalability of Call Concentrator and its ability to handle situations where Call Concentrator needs to register a large number of DNs with T-Server (as can happen in a large contact center). This option explicitly sets the number of DNs that Call Concentrator must register immediately. After that is done, Call Concentrator takes a break for the time specified by the DNRegistrationDelay configuration option. If the option is set to 0, Call Concentrator will register a DN at the time it reads the DN information from the Configuration Database.

EnvironmentDBID

Default Value: 1
Valid Values: Valid DBID values
Dependencies: EnvironmentName
Dynamic Change: No

Defines the DBID of the so-called default tenant. If Call Concentrator cannot recognize the tenant of a call, it assigns the call to the default tenant, specified in the EnvironmentName option.

EnvironmentName

Default Value: Environment
Valid Values: Valid names allowed by Configuration Layer
Dependencies: EnvironmentDBID
Dynamic Change: No

Defines the name of the so-called default tenant. If Call Concentrator cannot recognize the tenant of a call from the T-Server events, it assigns the call to the default tenant. By default, the name of the default tenant is Environment.

ESequenceInitFactor

Default Value: 32
Valid Values: Positive integers
Dependencies:CC_ID, CCON_COUNT
Dynamic Change: No

Defines a constant to be used, along with the value set in the CCON_COUNT configuration option, to calculate a unique value for the ESequence field in the EVREF and EVDATA tables. If the call rate is *N* single calls per second, the following condition should hold:

```
ESequenceFactor > N
```

to ensure the uniqueness of the ESequence values during

```
2 * 32/CCON_COUNT * ESequenceFactor  seconds
```

If the options are set to the default values CCON_COUNT= *8*, ESequenceInitFactor=*32*, the values of ESequence will be unique for 194 days. The default value ESequenceInitFactor=*32* corresponds, approximately, to the rate of 10 incoming calls per second (assuming an average of 3 single calls per second). If the call rate is larger, increase the value of ESequenceInitFactor and either decrease the value of CCON_COUNT or accept a shorter uniqueness period. Manipulating the CCON_COUNT and ESequenceInitFactor configuration options ensures that the generated ESequence values in an adequate time range are unique depending on the projected call rate.

EventData

Default Value: No default value
Valid Value: Boolean; See the “Specifying Fields in the User Data Tables” section in Chapter 4 of the *Call Concentrator 7 Reference Manual*.
Dependencies: evref, DataFilter
Dynamic Change: No

Designed to specify the custom fields in the EVREFEX table. If both the evref and DataFilter configuration options are set to on, Call Concentrator also considers the EventData configuration option as a filter for the attached data and will write to the EVREF and EVDATA tables only those keys that are listed in the EventData configuration option.

You can extend the EventData configuration option if the number of fields defined exceeds 255 characters. To do this, add additional fields (EventData_1, ...EventData_n) as needed. When you use this format, each EventData line entry (except the last) should end with a comma. For example:

```
EventData char,Customer,char,Phone,
EventData_1 char,Address,char,Comment,
EventData_2 char,CallCenter
```

evref

Default Value: `on`
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

Enables or disables Call Concentrator output to the EVREF table. If it is set to `off`, Call Concentrator output into the EVREF table is disabled. If it is set to `on`, Call Concentrator writes the user data to the EVREF and EVDATA tables. Additionally, Call Concentrator may write the data in the EVREFEX table, depending on the value of the `evrefex` option. For more information, please refer to the "User Data Tables" chapter in the *Call Concentrator 7 Reference Manual*.

Note: If the `DataFilter` configuration option is set to `on`, Call Concentrator will record to the EVREF and EVDATA tables only the attached data that corresponds to the keys listed with `EventData`. If `DataFilter` is set to `off`, Call Concentrator records all attached data.

evrefex

Default Value: `off`
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

Enables or disables Call Concentrator output to the EVREFEX table. If it is set to `on`, Call Concentrator writes user data to the EVREFEX table. If it is set to `off`, output to the EVREFEX table is disabled. In addition, Call Concentrator may write the user data to the EVREF and EVDATA tables, depending on the value of the `evref` configuration option. For more information, see Chapter 4 of the *Call Concentrator 7 Reference Manual*.

gcdr

Default Value: `on`
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

Enables or disables Call Concentrator output to the GCDR table.

GDATAEX

Default Value: `off`
Valid Value: Boolean
Dependencies: `GlobalData`
Dynamic Change: No

Enables or disables Call Concentrator output to the GDATAEX table.

GlobalData

Default Value: No default value
Valid Values: See the "Specifying Custom Fields" section in Chapter 4 of the *Call Concentrator 7 Reference Manual*.
Dependencies: GDATAEX
Dynamic Change: No

Enables you to specify custom fields in the GDATAEX table. You can extend GlobalData if the number of fields defined exceeds 255 characters. To do this, add additional fields (GlobalData_1, ...GlobalData_n) as needed. When you use this format, each GlobalData line entry (except the last) should end with a comma. For example:

```
GlobalData char,Customer,char,Phone,
GlobalData_1 char,Address,char,Comment,
GlobalData_2 char,CallCenter
```

HeldAgentStatusesOnly

Default Value: off
Valid Value: Boolean
Dependencies: DataFilter, CustomAgentStatusesOnly, AgentRecordUserTypes, AgentUserFields
Dynamic Change: No

If set to on, Call Concentrator will write to the AREC table only those records that correspond to the ONHOLD agent status. This option goes into effect only if DataFilter is set to on. If CustomAgentStatusesOnly is also set to on, Call Concentrator also records custom agent statuses into the AREC table.

ignore-observing-dns

Default Value: off
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

If set to on, Call Concentrator does not include observing DNs in the list of DNs involved in a call leg. This eliminates the possibility that observing DNs will not be correctly released at the end of the call segment and that, as a result, the call will become stuck.

This option does not appear in the application template. You must create it manually.

IgnoreRingingTime

Default Value: `on`
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

Note: This option does not appear in the application template. To use it, you must create and configure this option using Configuration Manager. See *Framework 7 Configuration Manager Help* for instructions on creating application options.

Used when a callsegment is abandoned from the ringing state the second that it starts ringing and the following conditions apply:

- The call status (or `SResult`) is `386`.
- The duration of the call (`SDuration`) equals the duration of the `RingTime` and the duration spent in all queues before being abandoned (`WtTime`).

If the option is set to `on,` Call Concentrator then changes `RingTime` to `0`. Otherwise, Call Concentrator records the unchanged `RingTime`.

IgnoreRoutingPoints

Default Value: `off`
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

When set to `on,` causes Call Concentrator to ignore all events relating to routing points.

Note: This option is designed to handle special supplementary routing points on DMS switches. Otherwise, it should be set to `off` (default).

ignore_trunks

Default Value: `on`
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

Configuring this option enables you to specify whether Call Concentrator should register Trunk DNs. If the option is set to `on`, then Call Concentrator ignores DNs of the Trunk type. If the option is set to `off`, Call Concentrator tries to register Trunk DNs as if they were configured as Extensions.

This option does not appear in the application template. You must create it manually.

IgnoreVRP

Default Value: off
Valid Value: Boolean
Dependencies: UseCfgDNType, IgnoreRoutingPoints
Dynamic Change: No

Along with UseCfgDNType, affects how Call Concentrator treats DNs defined in the Configuration Database as routing points, external routing points, virtual routing points, and routing queues. Call Concentrator treats these DNs according to how you set UseCfgDNType and IgnoreVRP.

If this option is set to on, Call Concentrator treats the above DNs in the same manner as it does virtual queues; namely, no T-Server event (T-Event) related to those DNs has any effect on the number and content of GCDR and SCDR records in the recorded call history. The T-Events may, however, generate records in the AREC table.

If this option is set to off (the default), Call Concentrator treats these DNs according to how you set UseCfgDNType and IgnoreRoutingPoints.

LostRouting

Default Value: off
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

Modifies the algorithm Call Concentrator uses to determine what should be done on the RouteUsed event. When the option is set to on, this event is unconditionally treated as indicating the possible termination of a call segment. When the option is set to false (default), the RouteUsed event is treated as indicating the termination of a call segment only if no DNs are attached to the call segment.

management-port

Default Value: <must be supplied>
Valid Values: Valid TCP/IP port numbers
Dependencies: None
Dynamic Change: Yes

Specifies the TCP port for SNMP support.

MediaType

Default Value: off
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

If set to on, Call Concentrator assumes that the GCDR table has an additional MediaType field next to the CallType field. Call Concentrator fills this field with the values of the MediaType attribute retrieved from the T-Server messages.

To enable this option, create the GCDR table using the appropriate SQL script. Otherwise, Call Concentrator reports an error and quits. For more information, see Chapter 5, "Deploying SQL Scripts" on page 45.

NetworkCallFlow

Default Value: off
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

Tells Call Concentrator how to treat T-Server messages that indicate that a call has been routed. It should be set to on when Call Concentrator is used to monitor only network T-Servers and the External Router is not used. If the option is set to on, Call Concentrator will consider the call completed after leaving a routing point. If the option is set to off (default), Call Concentrator does not automatically assume that after routing the call the call was completed and applies the DeleteTime timeout.

Note: If the External Router is used along with the Network routing, set the value to off to prevent duplicate rows from being written to the Call Concentrator tables.

newtables

Default Value: yes
Valid Values: yes, 1, dart, 2
Dependencies: None
Dynamic Change: No

Tells Call Concentrator which table formats to use. If the option is set to yes or 1, Call Concentrator uses the short format of the GCDR and SCDR tables that does not include the EndTime field (and several others) in the SCDR table. If the option is set to dart or 2, Call Concentrator uses the long format of the GCDR and SCDR tables. Use an appropriate SQL script to ensure that a chosen setting for the newtables configuration option works correctly. For more information, see Chapter 5, "Deploying SQL Scripts" on page 45.

RecordTreatment

Default Value: off
Valid Value: Boolean
Dependencies: AgentRecordUserTypes, AgentUserFields
Dynamic Change: No

When set to on, RecordTreatment causes Call Concentrator to store call treatment data to the AREC table. For more information about recording treatment data, see Chapter 5 in *Call Concentrator 7 Reference Manual.*

scdr

Default Value: `on`
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

Enables or disables Call Concentrator output to the SCDR table.

SendBeginTransaction

Default Values: `off`
Valid Value: Boolean
Dependencies: `DbDumpCommitTime, DbDumpCommitFrequency`
Dynamic Change: Yes

If set to `on,` Call Concentrator sends `begin transaction` and `commit transaction` requests to the DB Server in accordance with the settings you configure for the `DbDumpCommitFrequency` and `DbDumpCommitTime` options. Otherwise, Call Concentrator tries to set the database into `Auto-Commit` mode and does not use transaction instructions.

show-attached-data

Default Value: `off`
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

Controls whether attached data appears in Call Concentrator logs. Set this option to `off` (default) to prevent sensitive customer data from appearing in the logs.

SingleCDRLimit

Default Value: `0`
Valid Values: Positive integers or `0`
Dependencies: None
Dynamic Change: Yes

After a call leg is completed, Call Concentrator creates an internal Call Details Record (CDR) to hold information about the leg. It keeps the CDR in memory until it can be written into the database. `SingleCDRLimit` sets an upper limit for the number of these records to be kept simultaneously in memory. This may be useful from the point of memory consumption. The downside is that if the limit is exceeded, Call Concentrator stops creating CDRs for the completed call legs. It just deletes call segment information until the moment a pending

CDR is written to the database. During this period, some information may be lost.

If the SingleCDRLimit configuration option is set to 0 (zero), then no limit on the number of CDRs is placed. The number can grow as memory resources permit.

If the connection to DB Server is lost, Call Concentrator keeps in memory as many CDRs as specified by the SingleCDRLimit setting until the connection to DB Server is reestablished. See StatesLimit.

StatesLimit

Default Values: 0
Valid Values: Positive integers or 0
Dependencies: None
Dynamic Change: Yes

During execution, Call Concentrator creates internal records holding information about the states of agents and DNs, and it keeps these state records in memory until the moment they can be written into the database. StatesLimit sets an upper limit for the number of those state records to be simultaneously kept in memory. This may be useful from the point of memory consumption. The downside is that if the limit is exceeded, Call Concentrator stops creating state records. It just deletes the state information until the moment a pending state record is written to the database. During this period, some information may be lost.

If StatesLimit is set to 0 (zero), no limit on the number of state records is placed. The number can grow as memory resources permit.

If the connection to DB Server is lost, Call Concentrator keeps in memory as many state records as specified by the StatesLimit value, until the connection to DB Server is reestablished. See also SingleCDRLimit.

StatInterval

Default Value: 30
Valid Values: Positive integers
Dependencies: None
Dynamic Change: Yes

Specifies the interval of time, in seconds, between two subsequent runtime statistics output in the execution log.

UseBinding

Default Value: off
Valid Value: Boolean
Dependencies: BindThreshold, BindTimeout
Dynamic Change: No

Enables or disables the use of DB Server's Data-Binding feature. The feature allows DB Server client applications, like Call Concentrator, to issue an SQL statement (for example, an INSERT statement) only once and then repeatedly execute by sending to DB Server only the data needed for the execution (bound data records), not the statement itself.

If the option is enabled, the BindThreshold and BindTimeout configuration options can be used to deploy another DB Server feature, Bulk Operations. If BindThreshold is set to a value greater than 1, Call Concentrator does not send the bound data to DB Server (one record per request basis). Instead, it tries to write as many bound data records as specified by BindThreshold with a single bulk request.

Note: The Data-Binding and Bulk-Operation features are supported only for an Oracle database. Otherwise, UseBinding, BindThreshold, and BindTimeout have no effect.

UseCfgDNType

Default Value: off
Valid Value: Boolean
Dependencies: IgnoreVRP, IgnoreRoutingPoints
Dynamic Change: No

Along with IgnoreVRP, affects how Call Concentrator treats directory numbers (DNs) which are defined in the Configuration Database as routing points, external routing points, virtual routing points, and routing queues. Call Concentrator treats these DNs according to how you set UseCfgDNType and IgnoreVRP.

If this option set to on, Call Concentrator relies solely on the DN type information that the Configuration Database provides and further treats those DNs according to how you set IgnoreRoutingPoints and IgnoreVRP.

If this option set to off (the default), Call Concentrator uses the DN type information that T-Server provides. If T-Server reports one of the DNs as having the RouteQueue type, Call Concentrator treats the DN as a combination of a queue and a routing point (in other words, a virtual directory number) independent of how you set IgnoreRoutingPoints and IgnoreVRP.

use-null-values

Default Value: false
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

If set to true, this option directs Call Concentrator to use NULL instead of "0"/"" (empty string) as the default field value for the EVDATA and EVREFEX tables. The default value, false, causes Call Concentrator to use "0"/"" (empty string) as the default field value.

use_original_connid

Default Value: `on`
Valid Value: Boolean
Dependencies: None
Dynamic Change: No

Use when multi-site routing scenarios include the external routing of a call back to the site from which it has been previously routed. When the option is set to `on,` Call Concentrator uses the original `ConnID` (`ConnID` of the call that has been externally routed to another site) to identify the call when it is routed back to the original site. When this option is set to `off,` Call Concentrator might use a new `ConnID` instead of the original `ConnID` for some calls.

Note: For Call Concentrator to use this option, a version of T-Server that supports the `FirstTransferConnectionID` attribute is required.

license Option Settings

This is an optional, user-created section. You can use this section to specify the location of your license file. Any settings configured here take lower priority than licensing parameters specified on the command line or at startup.

license-file

Default Value: No default value
Valid Values: <string>.
Dynamic Change: No

Specifies the license address in either format:

The host name and port of the license server, as specified in the `SERVER` line of the license file, in the `port@host` format. For example:

```
7260@ctiserver
```

The full path to and the exact name of the license file. For example:

```
/opt/mlink/license/license.dat
```

Note: Genesys recommends that you specify the License Manager's host and port parameter. This value eliminates the need to store a copy of a license file on all computers running licensed applications.

Log Options

Call Concentrator uses the Genesys Framework common logging. For more information about configuring the options in the `Log` section, refer to the "Common Log Options" chapter in the *Framework 7 Configuration Options Reference Information*. For explanations of the Call Concentrator-specific log events, see the Call Concentrator section of the *Framework 7 Combined Log Events Help*.

Configuring a Windows NT Service

If you installed Call Concentrator as a Windows NT Service, you can adjust the Service settings. For example, to adjust the startup settings:

1. Select `Start > Settings > Control Panel`.
2. Double-click the `Services` icon in the `Control Panel` folder. The `Services` window appears.
3. Select the Call Concentrator entry.
4. Click `Startup...`
5. In the `Service` window that appears, select `Automatic` or `Manual Startup Type`.
6. If using Windows NT resources that require user identification (protected databases, for example), select the `This Account` button and specify the user account that the server should use to gain access to those resources.
7. Click `OK` to confirm and then close the `Service` window.

Consult the Windows NT documentation for more information about Windows NT Services.

Chapter

4 Installation

This chapter contains step-by-step instructions for installing Call Concentrator. The chapter includes these sections:

- Overview, page 35
- Windows Installation, page 35
- UNIX Installation, page 43

Overview

This chapter describes how to install Call Concentrator. When you deploy Call Concentrator, you should first configure the Call Concentrator application, then perform the installation.

You can install Call Concentrator on either Windows or Unix platforms. For a list of supported operating systems, see *Genesys 7 Supported Operating Systems and Databases*.

Refer to the "Deployment Planning" chapter in the *Call Concentrator 7 Getting Started Guide* and Chapter 2, "Configuration," on page 11 of this document prior to installing Call Concentrator.

After Performing the Installation

After installation, and before you begin using Call Concentrator, you must create the Call Concentrator tables in the database of your choice by running the appropriate script from the Call Concentrator installation package. The scripts are described in Chapter 5, "Deploying SQL Scripts," on page 45.

Windows Installation

To install Call Concentrator on a Windows platform:

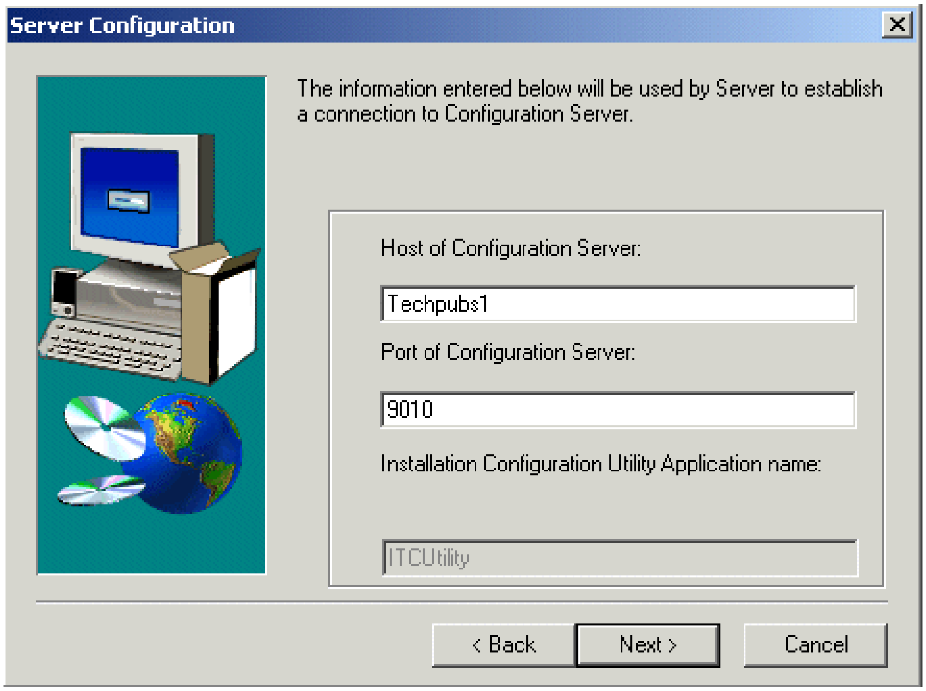
1. Insert the product CD into your CD-ROM drive.
2. Browse to and open the `ccon\windows` directory.

3. Double-click `Setup.exe` to start the installation wizard.

   You will receive prompts as you move through the setup process. Enter the appropriate information and then click `Next` to move to each successive window.

> **Note:** By default, the setup process creates a directory called `\gcti\ccon` in the root directory of your hard drive and installs Call Concentrator there unless you direct otherwise.

4. On the `Welcome` screen, read the text, and then click `Next`.
5. On the `Server Configuration` window, shown in Figure 1, enter the host of your Configuration Server and the Configuration Server port number, and then click `Next` to continue.



Server Configuration

The information entered below will be used by Server to establish a connection to Configuration Server.

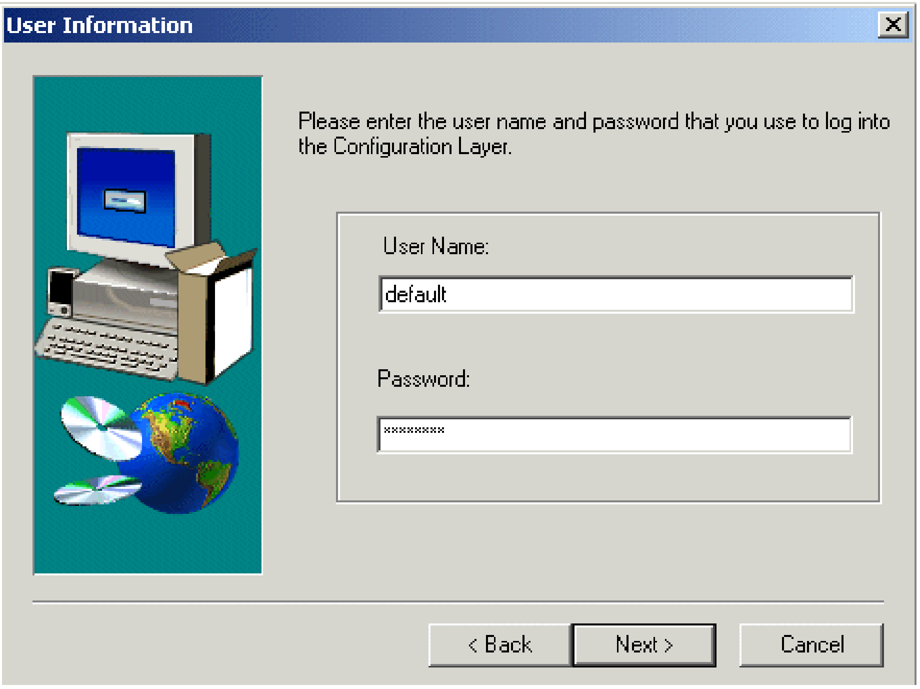
Host of Configuration Server: Techpubs1

Port of Configuration Server: 9010

Installation Configuration Utility Application name: ITCUtility

Figure 1: Server Configuration Window

6. On the `User Information` window, shown in Figure 2 on page 37, enter the user name and password used to log in to the Configuration Layer, and then click `Next`.



Please enter the user name and password that you use to log into the Configuration Layer.

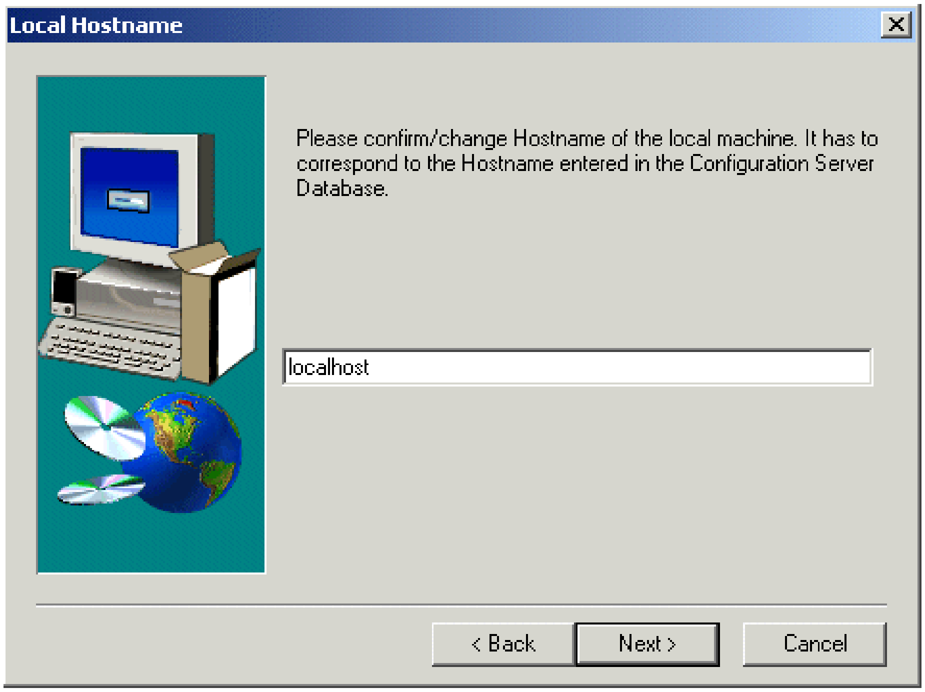
User Name: default

Password: *********

Figure 2: User Information Window

7. Confirm or change the host name of the local machine on the `Local Hostname` window, shown in Figure 3 on page 38, and then click `Next`.

Note: This host must be registered in the Configuration Server database.



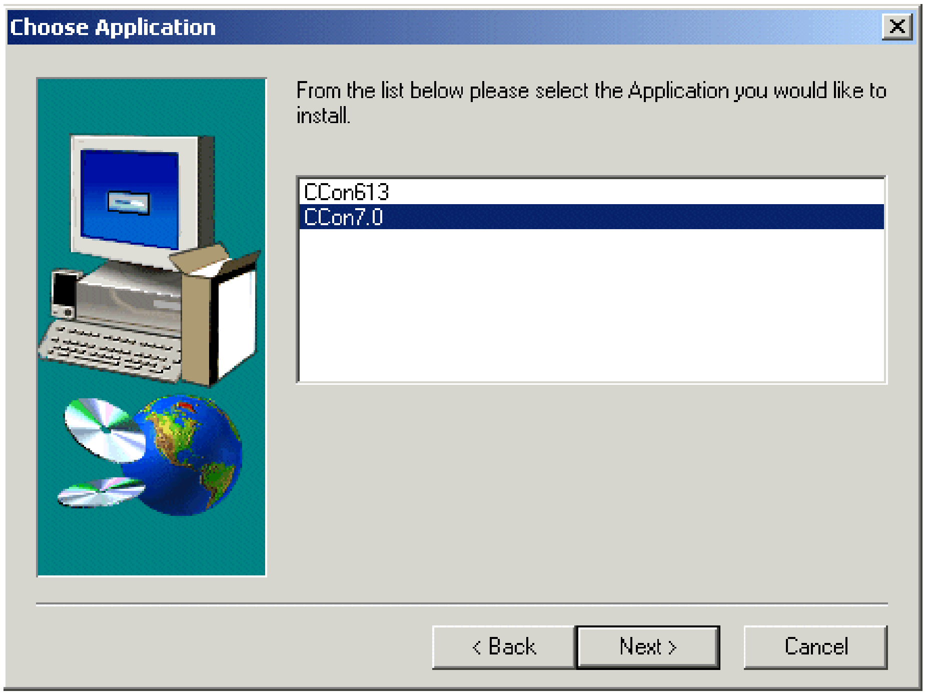
Local Hostname

Please confirm/change Hostname of the local machine. It has to correspond to the Hostname entered in the Configuration Server Database.

localhost

Figure 3: Local Hostname Window

8. On the `Choose Application` window, shown in Figure 4 on page 38, select the Call Concentrator Application object you have configured.



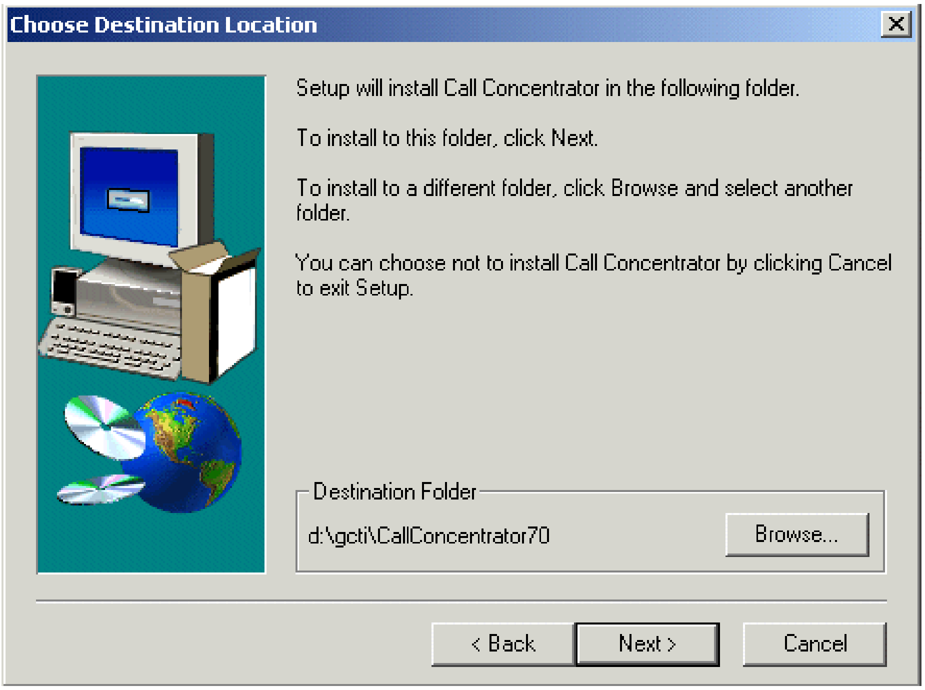
Choose Application

From the list below please select the Application you would like to install.

CCon613
CCon7.0

Figure 4: Choose Application Window

9. On the `Choose Destination Location` window, shown in Figure 5 on page 39, accept the installation default folder or use the `Browse` button to find an alternate destination folder in which to install Call Concentrator. Then click `Next`.



Choose Destination Location

Setup will install Call Concentrator in the following folder.

To install to this folder, click Next.

To install to a different folder, click Browse and select another folder.

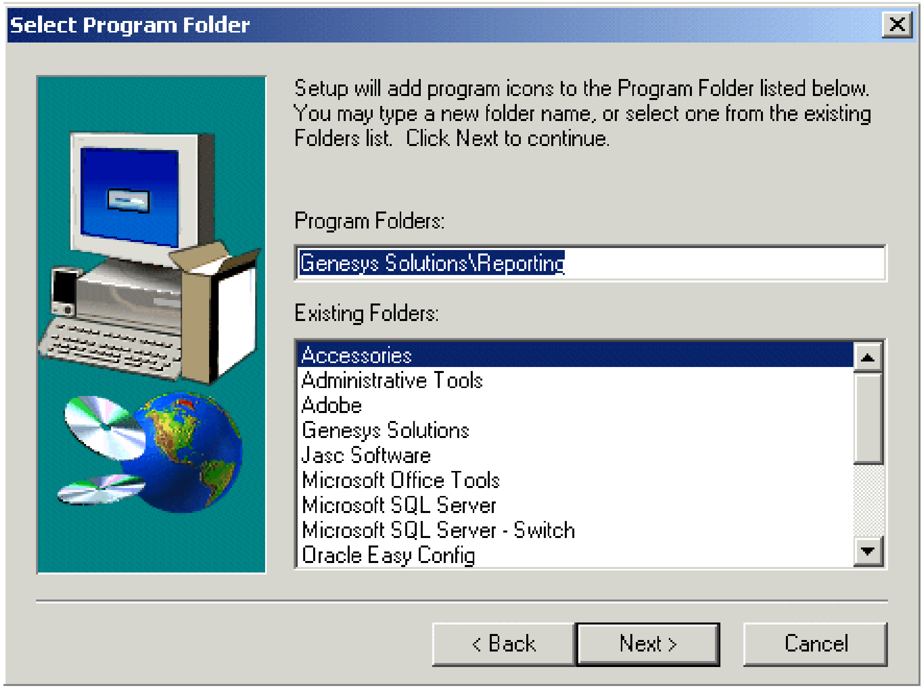
You can choose not to install Call Concentrator by clicking Cancel to exit Setup.

Destination Folder

d:\gcti\CallConcentrator70

Figure 5: Choose Destination Location Window

10. On the `Select Program Folder` window, shown in Figure 6 on page 40, accept the default Program Folder name (`Genesys Solutions\Reporting`) or enter another, and then click `Next`.



Select Program Folder

Setup will add program icons to the Program Folder listed below. You may type a new folder name, or select one from the existing Folders list. Click Next to continue.

Program Folders:

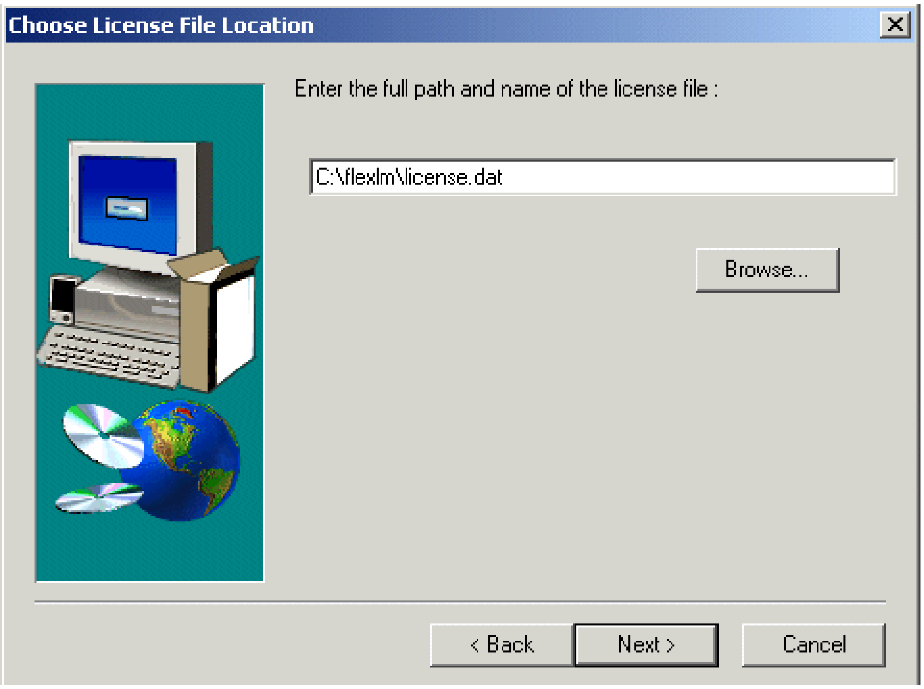
Genesys Solutions\Reporting

Existing Folders:

Accessories
Administrative Tools
Adobe
Genesys Solutions
Jasc Software
Microsoft Office Tools
Microsoft SQL Server
Microsoft SQL Server - Switch
Oracle Easy Config

Figure 6: Select Program Folder Window

11. On the `Choose License File Location` window, shown in Figure 7 on page 41, enter the either the license file location or license server location, specified using the format `<port number>@<host name>`, and then click `Next`.



Choose License File Location

Enter the full path and name of the license file :

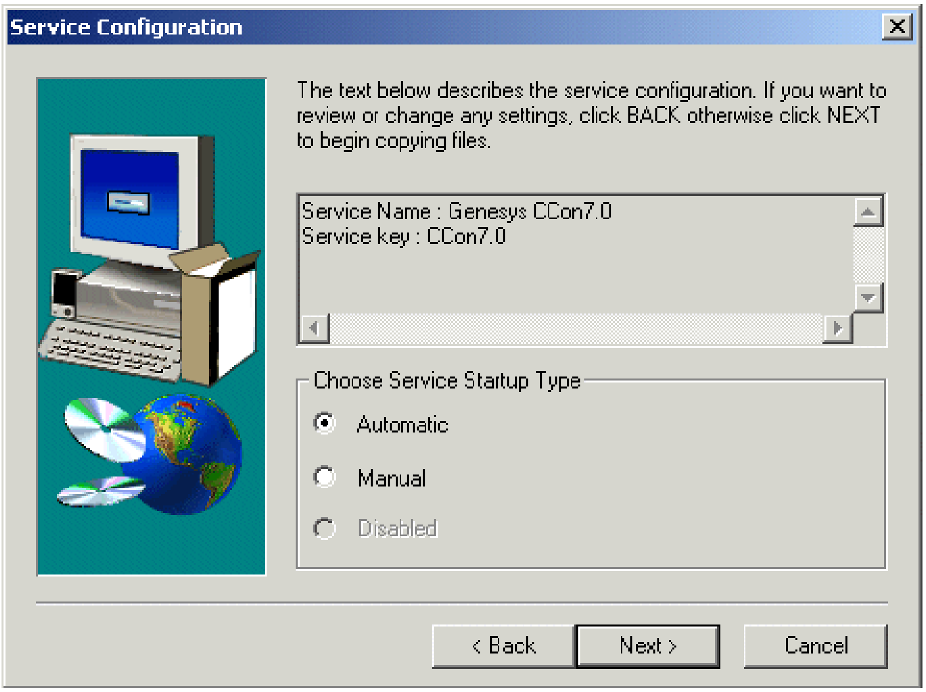
C:\flexlm\license.dat

Figure 7: Choose License File Location Window

12. The Wizard next prompts you to install Call Concentrator as a Windows NT Service. Click `Yes` if you want to start Call Concentrator as a Windows NT Service (recommended).

13. If you clicked `Yes` to install Call Concentrator as a Windows NT Service, the `Service Configuration` window, shown in Figure 8 on page 42, appears. To review or change any settings, click the `Back` button; otherwise click `Next`.

Note: You can make changes to the Windows NT Service settings after installation. For instructions, see “Configuring a Windows NT Service” on page 34.


Service Configuration

The text below describes the service configuration. If you want to review or change any settings, click BACK otherwise click NEXT to begin copying files.

Service Name : Genesys CCon7.0
Service key : CCon7.0

Choose Service Startup Type

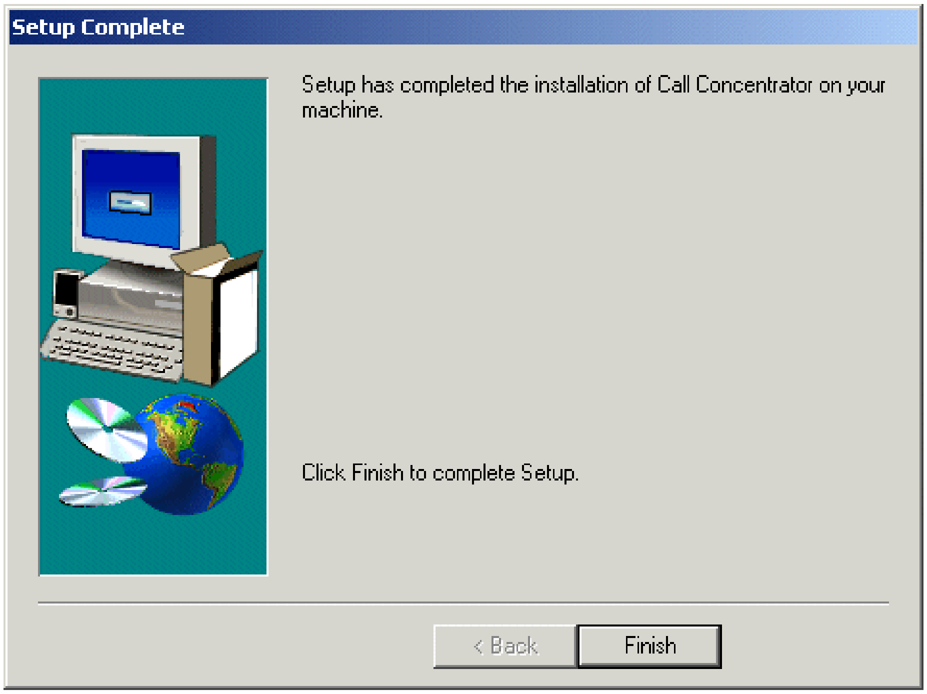
● Automatic

○ Manual

○ Disabled

Figure 8: Service Configuration Window

14. On the `Setup Complete` window, shown in Figure 9 on page 42, click `Finish` to complete the installation. A Call Concentrator icon now appears in your `Start` menu folder.


Setup Complete

Setup has completed the installation of Call Concentrator on your machine.

Click Finish to complete Setup.

Figure 9: Setup Complete Window

Installation is now complete. The information you entered in the various Call Concentrator Configuration windows is stored in the `service.cfg` file which is located in the Call Concentrator installation directory.

UNIX Installation

To install Call Concentrator on a Unix platform:

1. On the Call Concentrator CD, locate the correct installation directory for your platform, for example, `ccon\solaris`.
2. Save all contents of the directory to a local folder.
3. Following extraction, locate and run the `install.sh` shell script.
4. When prompted, specify the host name of the computer where Call Concentrator is to be installed.
5. If prompted to install Local Control Agent (LCA) on this host:
   a. Select `Yes.`
   b. Specify that LCA be started automatically.
   c. Type `y` when prompted, `Add LCA to startup (rc) files (y/n)?` In this case, LCA will start automatically when the computer is rebooted. After a successful LCA installation, continue installing Call Concentrator.
6. Select the Call Concentrator application to install.
7. Specify the destination directory where Call Concentrator is to be installed.

As soon as the installation process finishes, a message appears indicating a successful installation. The installation process creates a directory which has the name specified during the installation. This directory contains the Call Concentrator executable. If you installed LCA during Call Concentrator installation, the process also creates an additional directory containing LCA.

Chapter

5 Deploying SQL Scripts

Before running Call Concentrator, you must first create the tables to which Call Concentrator will write. The Call Concentrator installation package provides a set of sample SQL scripts. You can deploy these scripts as is or you can be modify them to fit the needs of your specific deployment environment. This chapter explains the deployment of these sample SQL scripts. It contains these sections:

Scripts and Tables Overview

The names of the sample scripts provided with Call Concentrator specify whether they create tables in the short (new) or long (dart) formats. The script names look like this:

```
makecdr_DATABASE_new.sql
```

or

```
makecdr_DATABASE_dart.sql
```

where DATABASE is one of the supported RDBMSs listed in *Genesys 7 Supported Operating Systems and Databases*.

Available Table Formats

Call Concentrator uses two basic table formats—new (or short) format and dart (or long) format—as prefixes in the name of the respective SQL script. For example, the makecdr_oracle_dart.sql script creates tables in the long format.

The long table format has some fields that do not appear in the short format. The tables affected are the Call Details tables: SCDR (Single Call Details Records) and GCDR (Global Call Details Records).

- The long GCDR structure has the `awgtime` field; the short format of the table does not.
- The long SCDR structure has four additional fields: `F_obsrv`, `EndTime`, `LocLQ`, and `RmtLQ`.

For detailed information about the structure of the GCDR and SCDR tables, refer to the "Global Call Details Records" and the "Single Call Details Records" sections of *Call Concentrator 7 Reference Manual*.

During runtime, Call Concentrator distinguishes the short and long table formats based on the value of the `newtables` configuration option, which is discussed in "newtables" on page 29.

Note: If the value for `newtables` does not fit the SQL script used (for example, if the `newtables` configuration option specifies the long format while the tables have been created with a `*.new.sql` script,) Call Concentrator reports the error and quits.

Optional Fields

Call Concentrator tables can contain optional fields that are independent of the table formats (long or short) used. These fields depend on the configuration options specified and can be edited using SQL scripts. They include:

- Custom fields in the user data tables (EVDATA, EVREF, and EVREFEX) and in the AREC table. For more information about custom fields, refer to the "Customizing User Data Tables" and the "Customizing the AREC Table" sections in *Call Concentrator 7 Reference Manual*.
- The `SSwitchCallID` field appears only in the SCDR table and only then if the `AddCallSwitchID` configuration option is set to `on`. For more information, see "AddSwitchCallID" on page 16.
- The `MediaType` field appears only in the GCDR table and only then if the MediaType configuration option is set to `on`. For more information, see "MediaType" on page 28.

Editing SQL Scripts

You can edit the sample SQL scripts with any text-editing tool. Typically, you will want to edit a script to:

- Add or delete optional fields.
- Change the data type of a table field.

The procedures for customizing the user data tables (EVDATA, EVREF, and EVREFEX) and the AREC table are described in the “Customizing User Data Tables” and the “Customizing the AREC Table” sections in *Call Concentrator 7 Reference Manual*.

Adding or Deleting Fields

To add or delete an optional field, edit the appropriate script following the instructions given in the comments within the script. You can also use a database-editing tool to restructure an existing table by adding or deleting fields.

Changing the Field Data Type

Sometimes you may need to change the data type of a particular field, possibly to provide more space for `STRING` data. For instance, you may want to increase the length of the `Phone` field in the GCDR table (by default defined in the sample SQL script as 40 characters) to fit a long e-mail address.

Call Concentrator sends changes in a field’s data type to the DB Server via textual `INSERT` statements of the form:

```
INSERT INTO TABLE VALUES(value1, value2, ...);
```

where each value is a literal string representing an integer or a character string. For example:

```
INSERT INTO EVREFEX VALUES(..., 'account', 1156888, ...);
```

In turn, DB Server passes the statement to a SQL Server that converts the values of the literal strings to the data types defined by the table’s structure.

Note: The only rule to remember when changing the data type of a field is that as long as the SQL Server can perform the conversion without error, Call Concentrator accepts it.

For example, if a field is defined as an integer in an Oracle database, the data type of the field can be changed to `character(32)` because the Oracle database server will automatically convert any integer (like `12345`) into a character string. Of course, the specification of the length (in the example, `32`) should be enough to fit any possible value of the field after the conversion.

Migration Scripts

Call Concentrator can store user data and present it in two tables—EVDATA and EVREF—or in a single EVREFEX table. The installation package includes a group of SQL scripts that facilitate the migration from the two-table form of user-data storage to the EVREFEX table. These scripts are named `migrate_to_evrefex_DATABASE.sql` and support the migration on the Oracle, Microsoft SQL Server, and Sybase platforms only.

Warning! Before running these migration scripts, be sure to back up all your existing data.

For a more detailed description about how these scripts work, refer to the "Migration to the EVREFEX Table" section in Chapter 4 of the *Call Concentrator 7 Reference Manual*.

Chapter

6 Starting and Stopping

This chapter describes the various starting and stopping procedures for Call Concentrator 7. It contains these sections:

Starting on Windows Platforms

You can start Call Concentrator 7 from a Windows platform in several ways:

- Using Management Layer (SCI)
- From a console window
- From the `Programs` menu
- From the desktop
- As a Windows NT Service

Starting from Solution Control Interface

The recommended approach for starting Call Concentrator is through Solution Control Interface (SCI), which provides a user-friendly interface for managing Genesys solutions. Refer to *Framework 7 Solution Control Interface Help* for complete details on using SCI.

To start Call Concentrator from SCI:

1. Start SCI and go to the `Applications` view.
2. Select `Call Concentrator` on the `List` pane.

3. Click the `Start` button on the toolbar, or select `Start` either from the `Action` menu or from the `Solutions` shortcut menu; then confirm the action in the resulting dialog box.

   SCI sends the start command to the Solution Control Server, which uses the Local Control Agent (LCA) to activate the Call Concentrator Application object.

Application startup may take some time. SCI reports the successful start of an application if its status changes from `Pending` to `Started` within the configured timeout.

Starting from a Console Window

To starting Call Concentrator from a console window:

1. Open an MS-DOS console window.
2. At the command prompt, change to the directory in which Call Concentrator is installed.
3. Enter the name of the Call Concentrator executable followed by the appropriate command-line parameters using the following syntax:

   ```
   ccon -host [host] -port [port] -app [app] -l [license
   information]
   ```

   where:

   - `host` refers to the name of the computer on which Configuration Server is running.
   - `port` refers to the communication port on which Configuration Server is running.
   - `app` refers to the name of the Call Concentrator Application object as defined in Configuration Manager.

   Note: If the application name contains blanks or hyphens (-), enclose it in double quotation marks.

   - `l` should be followed by either the license file location or license server location, specified using the format `<port number>@<host name>`.

Starting from the Programs Menu

To start Call Concentrator from the `Programs` menu:

1. Select the program folder created during installation.
2. Click the `Call Concentrator` shortcut.

Starting from the Desktop

If you have a `Call Concentrator` icon on your desktop, double-click the icon to start Call Concentrator.

Note: You must edit the location to which this shortcut points if you move your Call Concentrator application after installation.

Starting as Windows NT Service

To start a Call Concentrator installed as a Windows NT Service, perform the following steps:

1. Click the Windows NT Services Manager `Start` button, choose `Settings` > `Control Panel.`
2. Double-click `Services.`
3. Select your Call Concentrator application from the list and click `Startup.`
4. At the `Service` dialog box, choose a start-up type: `Automatic` or `Manual.`

   Genesys recommends selecting `Automatic.` In this manner, whenever Windows NT starts, Call Concentrator automatically starts too.

Starting on Unix Platforms

Call Concentrator 7 can be started on Unix using these steps:

1. Open a terminal window.
2. Change to the directory where Call Concentrator is installed.
3. Enter the name of the Call Concentrator executable followed by the appropriate command-line parameters using the following syntax:

   ```
   ccon -host [host] -port [port] -app [app] -l [license information]
   ```

   where:

   - `host` refers to the name of the computer on which Configuration Server is running.
   - `port` refers to the communication port on which Configuration Server is running.
   - `app` refers to the name of the Call Concentrator Application object as defined in Configuration Manager.

   Note: If the application name contains blanks or hyphens (-), enclose it in double quotation marks.

- `l` should be followed by either the license file location or license server location, specified using the format `<port number>@<host name>`.

Stopping on Windows Platforms

You can stop Call Concentrator from a Windows platform in several ways:

- Use SCI.
- Use a console window.
- Stop the Windows NT Service, if Call Concentrator is running as a Service.

Stopping with SCI

To stop Call Concentrator from SCI:

1. Go to the `Applications` view.
2. Select `Call Concentrator` from the `List` pane.
3. Click `Stop` on the toolbar

   or

   right-click anywhere and select `Stop` from the shortcut menu that appears

   or

   select `Action` > `Stop.`
4. Confirm the action in the resulting dialog box.

Stopping from a Console Window

Stop Call Concentrator within a console window using any of the following methods:

- Press `CTRL+C.`
- Click the `Close Windows` button on the right side of the title bar.
- Click the `DOS Control` icon on the left side of the title bar and select `Close.`

Stopping the Windows NT Service

If Call Concentrator was started as Windows NT Service, either automatically or manually, stop it from the Windows Service Control Manager:

- Click the `Stop` button. Be sure that `Autorestart` is not selected.

Note: If you started Call Concentrator from SCI, do not use Windows Services Manager to stop it.

Stopping on Unix Platforms

Stop Call Concentrator on Unix using either of the following methods:

- At the command line, enter `kill -SIGTERM [processid]` where `processid` is Call Concentrator's Unix process ID.

  or

- Enter `CTRL+C` from the active Call Concentrator window.

Chapter

7 Uninstalling Call Concentrator

This chapter describes how to uninstall Call Concentrator. Genesys recommends that you uninstall your current Call Concentrator installation before migrating to a more recent version. This chapter contains these sections:

Warning! Genesys recommends that you always back up your Call Concentrator database before uninstalling.

Uninstall Procedure for Windows

You must stop Call Concentrator before uninstalling. For the correct procedure for your environment, see Chapter 6, "Starting and Stopping" on page 49. Then perform these steps:

1. Select `Start` > `Programs` > `Call Concentrator` > `UnInstall Call Concentrator`.
2. The unInstallShield Wizard opens and asks you to confirm that you want to uninstall Call Concentrator. Click `Yes` to continue.
3. When unInstallShield has finished uninstalling Call Concentrator, click `OK` to close the window.
4. Verify that the Call Concentrator directories have been deleted.

Uninstall Procedure for UNIX

You must stop Call Concentrator before uninstalling. For the correct procedure for your environment, see Chapter 6, “Starting and Stopping” on page 49.

To uninstall Call Concentrator:

- Delete the directory where Call Concentrator is installed.

Index