Voice Callback 7.1

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	+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.
www.genesyslab.com

Document Version: 71vcb_dep_04-2005_v7.1.001.00

Table of Contents

Chapter

1 About This Document

Welcome to the *Voice Callback 7.1 Deployment Guide*. This document introduces you to the deployment process, which entails the configuration and installation of the Universal Callback Server and other components that enable the Voice Callback option for Universal Routing. VCB 7.1 includes support for Web API Server for callback orders and the Universal Contact Server for contact history.

This chapter has these sections:

Note: For the 7.1 release, the *Voice Callback Getting Started Guide* and the *Voice Callback Deployment Guide* have been combined. For the *Voice Callback Getting Started Guide* and the *Voice Callback Deployment Guide* for previous releases, please visit the Genesys Technical Support website `http://genesyslab.com/support`, or request the Documentation Library CD, which you can order by e-mail from Genesys Order Management at `orderman@genesyslab.com.`

In brief, you will find in this guide:

- Explanation of VCB and description of its features and functions.
- Architectural overview with a discussion of the applications and components that enable VCB functionality.
- Information for planning the implementation of VCB at a contact center; references to more-detailed instructions for the configuration and installation process.
- Introduction to the deployment process.
- Manual configuration of VCB.

- Design of routing strategies and IVR (Interactive Voice Response) scripts.
- Installation of Dialogic boards.
- Starting and stopping procedures.

Note: You can find information about VCB templates in the *Reporting Technical Reference Guide for the Genesys 7.x Release*.

Intended Audience

This guide is intended primarily for system engineers and other members of an implementation team who will be configuring and installing the components of VCB and for system administrators who will maintain VCB.

To successfully implement VCB, you must have a basic understanding of and familiarity with:

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

You should also be familiar with the software that supports and enables VCB:

- Genesys Framework.
- Universal Routing (Enterprise Routing or Network Routing).
- Genesys-compliant Interactive Voice Response (IVR) interface.
- Agent desktop: either Genesys Agent Desktop, Genesys Contact Navigator or a desktop application created with a VCB-compatible SDK.
- Genesys Reporting.

Chapter Summaries

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

- Chapter 2, "Voice Callback Overview" on page 17, describes Voice Callback's features and functions, and lists software requirements and supported platforms.
- Chapter 3, "Architecture" on page 31, discusses Voice Callback components and how they interact to achieve VCB functionality.
- Chapter 4, "Deployment Procedures," on page 61, explains how to import Application templates, create and configure Application objects for the Universal Callback Server, CPD Server, Web API Server, and Universal Contact Server, and install VCB on Windows or UNIX.

- Chapter 5, “Other Deployment Considerations,” on page 81, describes Automatic Call Distribution (ACD), routing extensions, operation without an agent desktop, deployment for different dialing modes, and configuration of table access points for a black list.
- Chapter 6, “Configuration of Other Objects,” on page 99, provides instructions for configuring other objects that support and enable VCB functionality.
- Chapter 7, “Designing IVR Scripts and Routing Strategies,” on page 125, offers guidelines for writing IVR scripts and designing routing strategies.
- Chapter 8, “Sample Genesys Voice Portal Script,” on page 139, describes the Genesys Studio project file `vcb_gvp_sample.vws` and a number of `VOX` files, which are voice prompts used in the sample script.
- Chapter 9, “Installing Dialogic Boards,” on page 153, discusses how to install and use Dialogic Boards.
- Chapter 10, “Starting and Stopping Procedures,” on page 171, demonstrates how to stop and start the VCB.

Note: For information about the types of statistical reporting available in VCB and the use of filters, please consult the Genesys Reporting documentation.

- The Appendix on page 181 includes a Sample Deployment-Planning Worksheet.

Document Conventions

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

Document 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:

70vcb_ref_09-2004_v1.00

You will need this 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. This convention includes the *names* of directories, files, folders, paths, scripts, dialog boxes, options, modes, 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:

- *Voice Callback Reference Manual* provides the following information about the Universal Callback option of Universal Routing: configuration objects and options, communication protocols, enumerations for callback record statuses, treatments, data tables used as persistent information storage, and licensing requirements.
- *Framework Deployment Guide* describes the Framework architecture and the configuration, installation, starting, and stopping procedures for Genesys Framework.
- *Universal Routing Deployment Guide* describes how to configure, install, start, and stop Enterprise Routing and Network Routing, which together comprise Universal Routing.
- *Voice Treatment Server User's Guide* describes the Voice Treatment Server (VT Server), which is the server component of Voice Treatment Option. The guide is intended for those who install, configure, maintain, and administer the server side.
- *Voice Treatment Manager User's Guide* describes Voice Treatment Manager which serves as the user interface for Voice Treatment Option (VTO) and is primarily for those who create VTO scripts.
- *Reporting Technical Reference Guide for the Genesys 7.x Release* provides information about statistics, reporting templates, and reporting layouts that CC Analyzer and CCPulse+ use for reporting on Genesys products, including Multi-Channel Routing 7.
- *Multi-Channel Routing Genesys Web Media Getting Started Guide* provides a high-level overview of features and functions of Genesys Web Media, together with architecture information and deployment-planning materials.

- *Multi-Channel Routing Event Media Deployment Guide* describes deployment procedures for all Multi-Channel Routing components.
- *Multi-Channel Routing Event Media User's Guide* provides overall information and recommendations on the use and operation of Multi-Channel Routing.
- *Multi-Channel Routing Universal Contact Server Manager Help* serves as a guide to the user interface for Universal Contact Server Manager.
- *Multi-Channel Routing Web API Reference* is a Javadoc listing of classes, methods, fields, and constants of the Web API portion of the Web API Server component.
- *Multi-Channel Routing Web API Client Developer's Guide* describes the structure of the Web API, explains the Web Compound Samples, and describes procedures for customizing them.
- *Multi-Channel Routing Manual Deployment of MCR Web Components and UCS* updates part of the *MCR Deployment Guide*.
- *Multi-Channel Routing Knowledge Management: Categories and Standard Responses* supersedes the "Categories and Standard Responses" section of chapter 2 of the *MCR Event Media User's Guide.*
- *Multi-Channel Routing MCR Web API Callback Sample* updates part of the *MCR Web API Client Developer's Guide.*
- *Genesys Agent Desktop .NET Toolkit Reference Manual* describes the development of custom desktop client-applications in a Genesys Framework environment.
- *Genesys Agent Desktop .NET Server Deployment Guide* introduces the concepts, terminology, and procedures relevant to this server-side application that provides services to a client-side application for building a custom agent desktop application based on .NET technology. It shows how to configure and install Genesys .NET Server using the Installation Package CD.
- *Genesys 7 Migration Guide* is also on the Genesys Documentation Library CD contains a documented migration strategy for Genesys product releases. Contact Genesys Technical Support for additional information.
- *Genesys 7 Licensing Guide* discusses what types of licenses you need for specific installations.
- *Genesys Hardware Sizing Guide for the Genesys 7.x Release* provides hardware sizing information and performance guidelines for Genesys products.
- System Requirements chart on the Technical Support website at `http://genesyslab.com/support`.
- The *Genesys Master Glossary* document, which is on the Genesys Documentation Library CD, provides a fairly comprehensive list of Genesys and CTI terminology and acronyms.

- 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 Reference Manual*
- *Genesys 7 Supported Media Interfaces Guide*

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.`

Chapter

2

Voice Callback Overview

This chapter explains the Voice Callback option for Universal Routing and describes VCB features and functions. It has the following sections:

Introduction to Voice Callback

Voice Callback (VCB) helps contact centers manage periods of high, inbound call volume. It is an option for Universal Routing (Enterprise Routing and Network Routing, releases 6.5.1 and above). VCB provides an additional channel for customer contact when the call load is heavy at a contact center and allows callers to request a callback from an agent instead of waiting on hold. VCB supports two types of callbacks:

- `ASAP` (as soon as possible) callback
- `Scheduled` callback (for a specific date and time)

With VCB, Genesys provides a universal outbound calling-service, and the caller still has the choice to wait on hold until an agent is available to answer the call.

It is also possible to order callbacks through the Web API Server. The Web interface provides callback request submission and management functionality and allows the user to request callbacks and to manage requested callbacks from the company's website.

The Universal Contact Server is an optional component that provides the ability to store interaction and contact history information in a centralized database and to retrieve data to the agent desktop. See the detailed description of VCB features in the next section. New features for the 7.1 release are described at the end of this section.

VCB Features and Functions

This section describes the features and functionality for Voice Callback. It also refers you to more specific information contained in this guide and other product guides.

VCB Features

Caller Options

Voice Callback offers callers a choice of the following options:

- `ASAP` callback option allows the caller to request a callback as soon as possible. The caller disconnects, but the callback request serves as a virtual placeholder that retains the customer's position in the queue. The caller then receives a callback from the first available agent who is best qualified to serve the customer.
- `Scheduled` callback option allows the caller to request a date and time for the callback. The callback service request is resubmitted to a live calling queue on the date and time requested or sooner if VCB is configured to take estimated waiting time into consideration. In either case, an agent contacts the caller *as near as possible* to the requested time.
- `Waiting on hold` is an option for a customer who prefers to remain on the line and wait for an available agent. The caller who chooses to wait on hold retains his or her position in the queue until the call is routed to an agent.

Estimated Waiting Time

An IVR can be set to inform the customer of the estimated waiting time (EWT) and propose a callback if EWT exceeds the contact center's configured customer-service goals.

Note: The EWT is only an estimate. A sudden change of contact center dynamics, such as an increase in abandoned calls or higher-priority calls, can lead to an inaccurate EWT. So the callback might occur sooner or later than the EWT.

Managing Duplicate Callback Requests

Duplicate callback requests are callbacks ordered by the same caller (same callback number, that is, same `value` of `VCB_CONTACT` attribute) on the same Routing Point while the original callback request is being processed. The Universal Callback Server handles duplicate requests as follows:

- Universal Callback Server replaces a previously submitted request for a `Scheduled` or an `ASAP` callback if the new callback request is the `Scheduled` type. The second callback request overrides the first one if the second request arrives when the first request for the same telephone number is in a queue or waiting for the scheduled time.

- If an `ASAP` callback request arrives when another `ASAP` or `Scheduled` request for the same telephone number is in a queue, Universal Callback Server rejects the second request. See Figure 1.

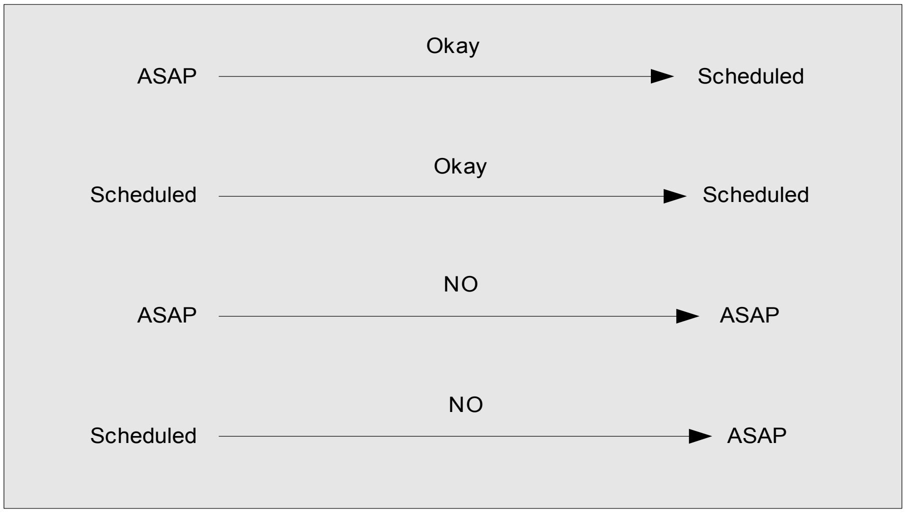


Figure 1: ASAP or Scheduled Callback Request Handling

Contact Center Service Options

A contact center typically offers Voice Callback to prevent extensive delays in its responses to incoming calls and to improve the quality of service. VCB offers a contact center various ways to use the callback request to meet its business objectives:

- Queue-by-queue selection: Some business queues, but not others, may offer customers the callback options. For example, the calling queues for the Sales and Customer Service Departments might offer VCB, but the Manufacturing Department might not. (See also "Routing Targets" on page 40.)
- Call-by-call selection: VCB may be offered for calls that meet a specific criterion. For example, calls from Gold cardholders, but not from Bronze cardholders, may have the VCB option. In other words, the customer's service level determines if the VCB option is available to callers.
- Hour/day/time selection: VCB can be offered at designated times or on specified days.

Multi-Site Support

VCB can deliver a callback request to the agent best qualified to handle a specific type of call across a multi-site environment.

Multi-Tenant Support

VCB functionality can be deployed on the network level, managed by a network provider, and made available for multiple tenants.

Note: The Universal Callback Server controls all Routing Point DNs which have been configured for its monitoring, regardless of tenant configuration and VCB tenant assignment. That is, tenants set in the `Tenants` tab in the Universal Callback Server application configuration object do not determine tenant scope. However, the Universal Callback Server processes only those switches which are configured for T-Servers present in the `Connections` tab of the Universal Callback Server application configuration object.

Call-Result Treatments

VCB provides treatment options for unsuccessful callback attempts. A contact center can customize treatments according to its business rules. For example, you can have:

- Separate treatments for different types of call results.
- Treatments determined by the number of unsuccessful callback attempts for a particular reason.
- Simple treatment actions, such as:
  - `Redial`—Call the same number again (for example, `Retry in 5 minutes`).
  - Retry at specified date/time (for example, `03/23/2004 5:00 PM`).
  - No treatment.

When configuring treatments, you can also specify:

- Number of callback attempts.
- Time period for attempting callbacks.
- Interval between callback attempts.

See the Configuration Manager in Genesys Framework for additional information. Refer to the Configuration Layer in Figure 5 on page 37.

IVR in VCB

An interactive voice response (IVR) interface with Genesys Framework can enable VCB-specific functions. The IVR must be Genesys-compliant; that is, it must be compatible with Genesys IVR drivers. You can write an IVR script to perform functions to:

- Prompt the caller and validate his or her response by means of text-to-speech or recorded voice capability.
- Provide automated number identification (ANI) and announce (play back) the caller’s telephone number for confirmation.
- Accept announced ANI or user-entered digits for the callback request.
- For a scheduled callback request, verify that the user-entered digits for the date and/or time are outside the range of the estimated waiting time (EWT) and standard operation hours (SOH). If either the date or the time that the caller requests for the callback is within the EWT, the IVR responds to the invalid entry with a prompt such as “Since your entry is within the estimated waiting time, you will be called back ASAP.”

- Process responses from VCB Server to provide caller with interactively correct callback-request attributes.
- Retain customer data from the original call.

Note: You must write an IVR script to utilize these capabilities; they are not out-of-the-box functions. See particularly "Designing IVR Scripts" on page 125 and "Implementation of IVR Script Logic" on page 129.

For additional information about IVR, see "Overview" on page 31 and "Interactive Voice Response" on page 41.

Routing in VCB

As an option for Enterprise Routing (ER) and for Network Routing (NR), Voice Callback relies heavily on the Universal Routing Server (the main component of ERS and NRS) and on routing strategies to perform VCB functions. Routing is responsible for:

- Providing EWT in call-attached data if IVR cannot request it directly from Stat Server.
- Routing a callback request to an agent with the appropriate skill set to assist the caller.
- Positioning calls in queues by priority or first-in-first-out (FIFO).

For more information about routing, see "Overview" on page 31 and "Universal Routing: Enterprise Routing and Network Routing" on page 39. See also "Designing Routing Strategies" on page 134.

Agent Desktop in VCB

An agent desktop presents callback requests to an agent. All user data that is present in the callback submission request is delivered to the agent desktop. If the callback is requested via `EventRouteRequest` or `EventTreatmentEnd`, then all user data that was attached to the call before it was passed to the Routing Point controlled by Universal Callback Server is delivered to the agent's desktop. In addition, all user data modifications, applied to the virtual call (which represents the callback request), while this call is being routed on Virtual Routing Point, is processed and delivered to the agent's desktop.

The choices for an agent desktop include:

- Genesys Agent Desktop (GAD)
- Another agent desktop with VCB functions implemented by Genesys APIs.

VCB can deliver a callback request to the agent desktop in the following two modes:

- `Preview` mode: The Voice Callback request record is presented to an agent selected by the routing strategy, the callback number and other customer-related information are autopopulated in the GUI, and the agent manually dials the customer's number. In this way, the agent can view the customer's record before dialing the callback number.

The `Preview` dialing mode depends on support from the agent desktop and requires agent-level routing performed by a Genesys Routing solution.

- `Autodial` mode: The system automatically connects the customer to an agent.

  The `Autodial` mode offers the option of Call Progress Detection on a CPD Server. The system automatically places the call at the appropriate time, listens to the progress of the call—for example, ringing, busy signal, no answer—and delivers the call to an agent when it detects a customer on the line.

In both `Preview` and `Autodial` modes, the callback request delivered to the agent desktop would include, for example:

- Quoted EWT and the time that the quote was given.
- Number of callbacks already attempted.

An agent may reschedule, cancel, or reject a callback request, or may apply a call result (`reason code`) to a callback attempt in the following ways:

- An agent may cancel a call request even without any dialing attempt (in `Preview` mode). The system discontinues dialing for that callback request.
- An agent can reject a callback request. In this case, the callback request is redistributed.
- The agent may reschedule a callback request. This option overrides the default treatment values and allows the agent to modify the callback date and time.
- Out-of-the-box `reason codes` (call results) are available from a drop-down list in Genesys Agent Desktop (GAD). The agent may apply one of these call results to an unsuccessful callback attempt. The standard treatment(s) associated with a call result will apply to subsequent callback attempt(s).

A system administrator can customize treatments for call results based on contact-center business rules. See the `Call Result` field description in the section on "Creating a Treatment Object" on page 108.

Connection Modes

The following features deal with callback routing and connection:

Agent-Reserved Connection

This automated dialing solution requires CPD Server. First, Universal Callback Server "engages" the agent by an internal call that contains all `UserData` for the corresponding callback request and plays an announcement (if configured) to the agent. Then Universal Callback Server transfers the call to the customer's phone number. This process ensures that the agent is available for processing the callback and that he or she has had time to prepare for the callback.

This mode enables ACD Queue callback parking (the ability to use an internal call as a callback placeholder). You can use ACD callback routing (ability to use ACD Queue for callback request distribution to agent) for agent-level routing as well.

Customer Reserved Connection

This automated dialing solution requires CPD Server.It has CPD Server voice announcement capability and ACD routing capability (ability to use the ACD Queue for callback request distribution to an agent).

Customer callback tasks (dialing callback number, playing announcement to the customer, and reaching the right agent) are automated.

The routing strategy can maintain a low number of calls on the ACD Queue by setting a threshold. See the *Universal Routing Deployment Guide* for more information.

CPD Server Announcements

CPD announcements are optional and are based on the user's configurations:

Announcements for Agent Reserved Connection

Enhancements for announcements of callback to the agent:

- Beep Tone announcement

  Configured Beep Tone may be played to the agent after the call has been established on the agent's DN (see "Agent-Reserved Connection" on page 22). The agent is then aware that it is a callback request independent of the screen pop.

- Voice announcement

  Prerecorded voice announcement may be played to the agent after the call has been established on the agent's DN (see "Agent-Reserved Connection" on page 22). The same announcement may be used for all callbacks, or the announcement can be recorded by IVR during callback submission. This announcement can be constructed from several voice files.

Announcements for Customer-Reserved Connection

Enhancement for announcements of callback to the customer:

- Voice announcement

  Pre-recorded voice announcement may be played to the customer after the call has been answered (see "Customer Reserved Connection" on page 23). The same announcement may be used for all callbacks, or the announcement can be recorded by IVR during callback submission. This announcement can be constructed from separate voice files.

Scheduled Callback Accuracy

Scheduled Callback Requests

Scheduled callback requests are processed according to EWT in queue. That data is then used to schedule callback processing.

Support for Switch Call Progress Detection

This capability allows Universal Callback Server to utilize the CPD capabilities of PBX (for PBX so equipped) with no CPD Server required.

Configurable Operating Hours

Enables the contact center administrator to configure operating hours for the center, so that this data is taken into account upon scheduled callback submission and acceptance.

Black-List Blocking

Black-list blocking enables “black list” support for VCB. Phone numbers from that list are not allowed to be used for callbacks.

Time Zone Support for Scheduled Callbacks

Adds time zone support for scheduled callbacks. The customer’s time zone may be determined during the callback-offering phase by an IVR script, for example, submitted to VCB, and then taken into consideration upon scheduled callback processing.

No Agent Desktop Operation

Provides the option of working without an agent desktop for automatic dialing modes. The agent desktop is not required for automatic dialing modes other than for screen-pop enhancements.

VCB IVR Integration

VCB does not contain an internal IVR. Genesys leverages the existing customer IVR. Sample IVR scripts are provided.

Reporting

Reporting capabilities can be applied to Voice Callback. The Stat Server is able to distinguish live calls from callback requests and calculate statistics for callback-related activity on virtual queues.

Out-of-the-box VCB reporting templates support callback submissions scheduled through any supported interface. All information available in VCB reporting templates is applicable to all callbacks, regardless of the origin of the callback request: Web, IVR, Agent Desktop.

New Features and Enhancements

Web Callback

VCB is now able to process callback requests originating from the Web, using a Web interface that enables callback ordering and management via the Web API Server.

The Web interface:

- Allows the user to specify the following callback parameters in the Web interface:
  - Scheduled time for scheduled callback
  - Phone number or other media address
  - Time interval for callback, which should not be dialed outside of this interval, regardless of treatment configuration
  - Call reason required to define appropriate Routing Point for callback distribution
  - Other customer-specified parameters attached as user data to the callback request
- Automatically reflects the user’s time zone, which is taken into account when dialing the callback.
- Provides the use of statistics for indication of queue length. For example:
  - Number of calls

- Callback position
- Estimated waiting time (EWT) for the entire queue
- EWT for the particular callback request

See *Multi-Channel Routing Event Media Deployment Guide* for implementation of the Web API Server component.

Contact History

The communication history between agents and customers, regardless of callback request origination (Web, IVR, Agent Desktop), is maintained by the Universal Contact Server in one centralized database that all agents may view. VCB works as a client of Universal Contact Server, providing it with callback-related details. These contact details are available to agents during callback preview and processing. In other words, callbacks are automatically registered in the contact history database that stores all interactions, including contact information.

The following information is available for the agent's review of contact history for each callback interaction:

- Scheduled time
- Callback status (for example, wait for scheduled time, wait for agent, fulfilled, failed, canceled)
- Call result (if applicable)
- Number of callback attempts
- User data

Universal Callback Server also has the capability to create automatically a new contact in the contact history database if the existing contact is not found for callback interaction.

Note: Functionality depending on the contact history database is optional for callback operation: VCB is able to operate without it.

See *Multi-Channel Routing Event Media Deployment Guide* for implementation of the Universal Contact Server.

Agent Desktop Processing of Web Callbacks

Callback requests can originate from the Web, and Genesys Agent Desktop provides the same preview and autodial functionality (GUI) for Web callbacks as it does for other sources of callback origin (IVR and Agent Desktop). The callback origin is indicated visually in the interface.

Note: VCB 7.1 is used for Genesys 7 Web callback features instead of ICS 6.5 Callback Server. Agent desktops providing preview support for ICS Web callback 6.5 also provide VCB 7.1 preview support.
For agent desktops supporting both VCB 7.1 and ICS 6.5 Web callback, the interface is merged into one: callbacks are supported by the same agent's UI regardless of callback request origin.

Time Interval for Callback Processing

Time interval limitations for callback processing are set up when ordering the callback. The expiration time for the callback can be specified for any ordering interface (Web, IVR, Agent Desktop) and is supplied in the form of start time and end time attributes.Time interval limitations take priority over any configured treatments and also ensure that the callback is not processed outside the stipulated time window. If the specified time expires and the callback is not delivered to the caller, the callback processing is stopped, and the callback is finalized in the database.

The default expiration time (option `max_proc_interval`) is configured either per Universal Callback Server Application or per Routing Point and applies to requests for which the expiration time is not specified in the callback submission request.

See the *Genesys Agent Desktop .NET Server Deployment Guide*, *Genesys Agent Desktop .NET Toolkit Developer's Guide.*

Reporting Enhancements

The way VCB calculates statistics and reports them to Stat Server has been enhanced. Now VCB uses Java extensions for statistics calculations and passes them to Stat Server. See *Reporting Technical Reference Guide for the Genesys 7.x Release.*

Software Requirements

Since Voice Callback is an option of Universal Routing, the existing users of Enterprise Routing (ER) and/or Network Routing (NR), version 6.5.1 or above, will have already configured and installed the prerequisite routing product for Voice Callback.

The following software products are required for a VCB deployment. Install them first:

- Framework
- Stat Server—required for providing statistical data on VCB operations
- Universal Routing: Enterprise Routing or Network Routing
- Interactive Voice Response (IVR)—required for interactions with customer
- An Agent Desktop—required for presenting call-related information to an agent
- Universal Callback Server
- CPD Server—required for Call Progress Detection for VCB
- Web API Server—required for enabling callback requests from Web browser
- Universal Contact Server—required for creation of interactions in Contact Database

Framework

All Genesys products are built on Genesys Framework.

Note: VCB requires the Framework components that are specified in the *Universal Routing Deployment Guide*.

New customers who require deployment procedures for Framework and/or Enterprise Routing (ER) or Network Routing (NR) can find this information in these documents:

- *Framework Deployment Guide*
- *Universal Routing Deployment Guide*

Stat Server

Stat Server tracks real-time states of interaction management resources and collects statistics about contact center performance. Genesys solutions use the statistical data to manage interactions more "intelligently." Use Genesys Reporting to generate real-time and historical contact center reports based on data that Stat Server collects.

For specific recommendations on Stat Server, refer to Stat Server documentation. *Genesys Framework Stat Server User's Guide*.

Universal Routing

Universal Routing offers VCB as an option; therefore, either Enterprise Routing (ER) or Network Routing (NR) is required for VCB. The Universal Routing Server, which enables both ER and NR, requires additional configuration for VCB.

Note: VCB has the capability to work without Universal Routing Server (URS) by using the default routing mechanism.

You can find information about Universal Routing and ER and NR in the following manual:

- *Universal Routing Deployment Guide*

Interactive Voice Response

VCB may use an Interactive Voice Response (IVR) product, such as Genesys Voice Portal (GVP), Voice Treatment Option (VTO), or another Genesys-compliant IVR. VTO is a `behind-the-switch` IVR. If you want to use another type of IVR, such as an `in-front-of-the-switch` IVR, you will also need the IVR Driver and IVR Server.

Note: IVR is not required since the Web or agent desktop can be the source of voice callback requests.

You can find the configuration and installation procedures for these products in these documents:

- *Genesys Voice Portal Deployment Guide*
- *Voice Treatment Option Voice Treatment Manager User's Guide*
- *Voice Treatment Option Voice Treatment Server User's Guide*

Agent Desktop

An agent desktop significantly increases productivity. Genesys offers Genesys Agent Desktop .NET toolkit to customize an existing agent desktop for Voice Callback.

Note: VCB is capable of working without an agent desktop in `Autodial` mode.

You can find information about GAD in these sources:

- *Genesys Agent Desktop .NET Toolkit Reference Manual*
- *Genesys Agent Desktop .NET Server Deployment Guide*

Note: Another desktop toolkit that enables the customer’s existing Agent Desktop to work with the VCB client-server protocol is an acceptable alternative.

Universal Callback Server

The Universal Callback Server is the primary component of VCB. You can find the configuration and installation instructions for the Universal Callback Server in “Installing Voice Callback” on page 71. Detailed information about Universal Callback Server is found in the *Voice Callback Reference Manual*.

CPD Server

If you want Call Progress Detection (CPD) in the `Autodial` mode, then you need the CPD Server. See “Licensing Requirement” on page 29.

Note: The configuration procedures in this document focus primarily on the configuration and installation of the Universal Callback Server, the CPD Server (optional), and other configuration objects required to support VCB.

Web API Server

You can order callbacks using the Web API Server. See *Multi-Channel Routing Event Media Deployment Guide* for information about installation and configuration.

Universal Contact Server

This component keeps a historical record of communication between agents and customers. See *Multi-Channel Routing Event Media Deployment Guide* for further information about installation and configuration.

Other Resources

The following resources are also available to assist you:

- *Configuration Manager Help* contains details about setting up solution objects and applications and performing other important configuration tasks.
- To prepare for Voice Callback deployment, Genesys recommends that you complete a Deployment-Planning Worksheet. See “Deployment Planning Worksheet” on page 181 for a sample.

Supported Platforms for Universal Callback Server

See *Genesys 7 Supported Operating Systems and Databases Reference Manual* for further information on supported platforms and databases for VCB components.

Licensing Requirement

Consult the following:

- Chapter 4, "Deployment Procedures," on page 61 for licensing requirements specific to VCB and detailed information on implementation.
- "VCB Licensing" in *Voice Callback Reference Manual* for configuration dependencies and options.
- *Genesys 7 Licensing Guide* for further information about enhancements to Genesys license control.

Globalization

VCB conforms to Genesys globalization standards. Further information on these standards is available at `http://genesyslab.com/support`.

Chapter

3 Architecture

This chapter discusses VCB and the applications/components that enable VCB functions. This chapter includes the following sections:

Overview

Like all Genesys products, Voice Callback (VCB) is built to function within the Framework software architecture. VCB itself is a packaged option for Enterprise Routing (ER) and Network Routing (NR), both of which are included in the term *Genesys Universal Routing*. The Universal Routing Server (URS) executes the routing strategies for both ER and NR. (See page 39 for further information.) You can find a sample routing strategy on the VCB product CD.

The primary component of VCB is the Universal Callback Server, which performs these operations:

- Processes callback orders
- Keeps a list of ordered callbacks
- Delivers callback request data to agents
- Requests the callback destination from the router

An Interactive Voice Response (IVR) product can be used along with VCB. Genesys offers two IVR products: Genesys Voice Portal (GVP) and Voice Treatment Option (VTO). VCB can work with any Genesys compliant IVR Driver. To be Genesys-compliant, an IVR Driver must be able to communicate the gathered user data to the Genesys Framework so that data is accessible to other applications (URS, Universal Callback Server, an agent desktop). A Genesys-compliant IVR works with a Genesys IVR Driver or interacts with the Genesys Framework through a T-Library interface. See page 41 for further information.

VCB uses Web API Server to provide Web callbacks.See page 45 for further information.

Universal Contact Server is an optional component that provides a history of contacts. See page 45 for further information.

The Call Progress Detection (CPD) Server is an optional dialing server; however, if a contact center uses autodialing with extended call-treatment features, the CPD Server is required. See page 44 for further information.

An agent desktop is also required for VCB. Genesys recommends the Genesys Agent Desktop (GAD). See page 44 for further information.

IVR Behind-the-Switch and IVR In-Front-of-the-Switch Modes

Figure 2 below and Figure 3 on page 34 provide high-level views of the applications/components that comprise Voice Callback. Figure 2 on page 33 shows IVR behind-the-switch mode. Figure 3 on page 34, shows IVR in-front-of-the-switch mode.

Note: Arrows between applications/components indicate interactions between them. Dotted arrows show optional interactions.

See "Genesys Solutions and Components Required for Voice Callback" on page 36 for a discussion of the ways in which these components interact in VCB.

Note: Framework is only partially represented in these figures—in the database, DB Server, T-Server, and Stat Server. Because all Genesys products function only within the Framework environment, its presence is pervasive, though somewhat transparent in these figures.

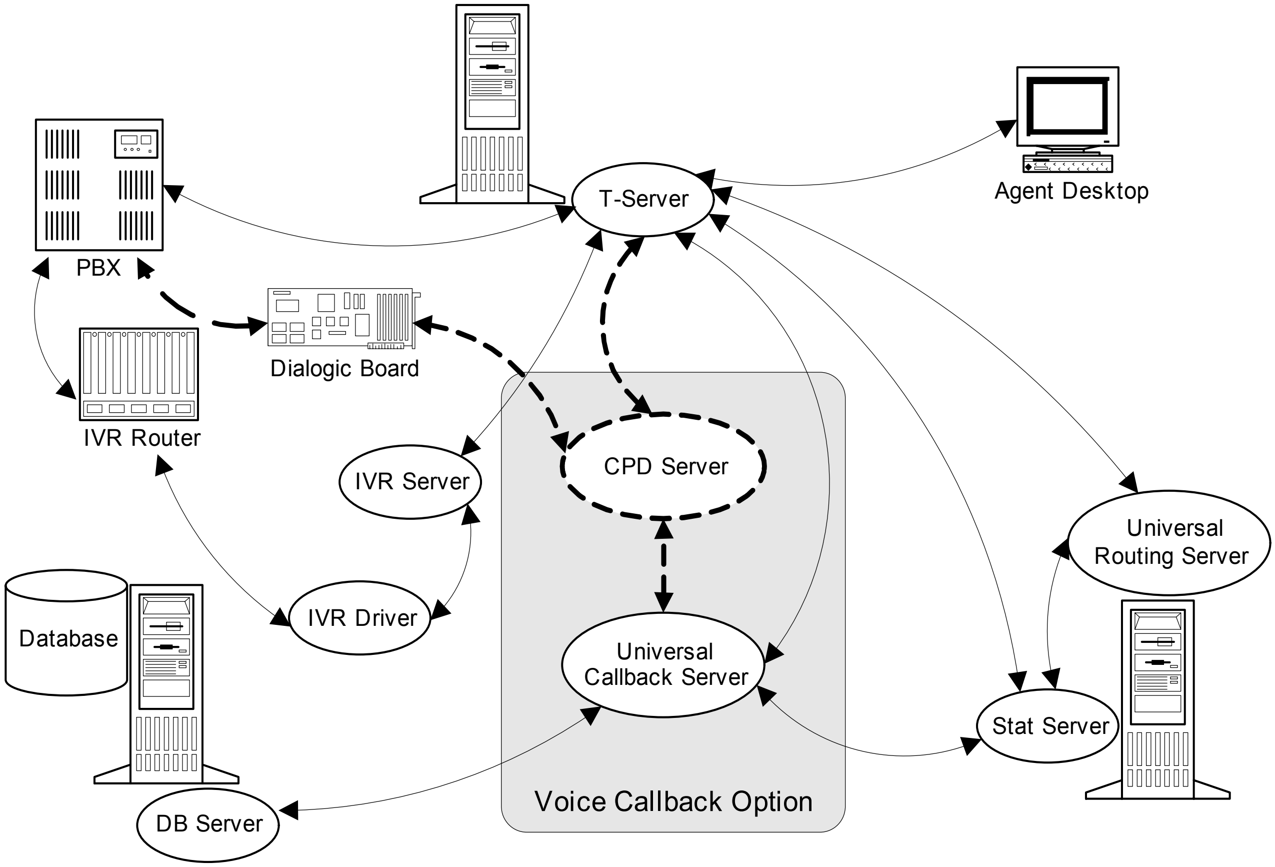


Figure 2: Interaction Between Components in IVR Behind--the-Switch Mode

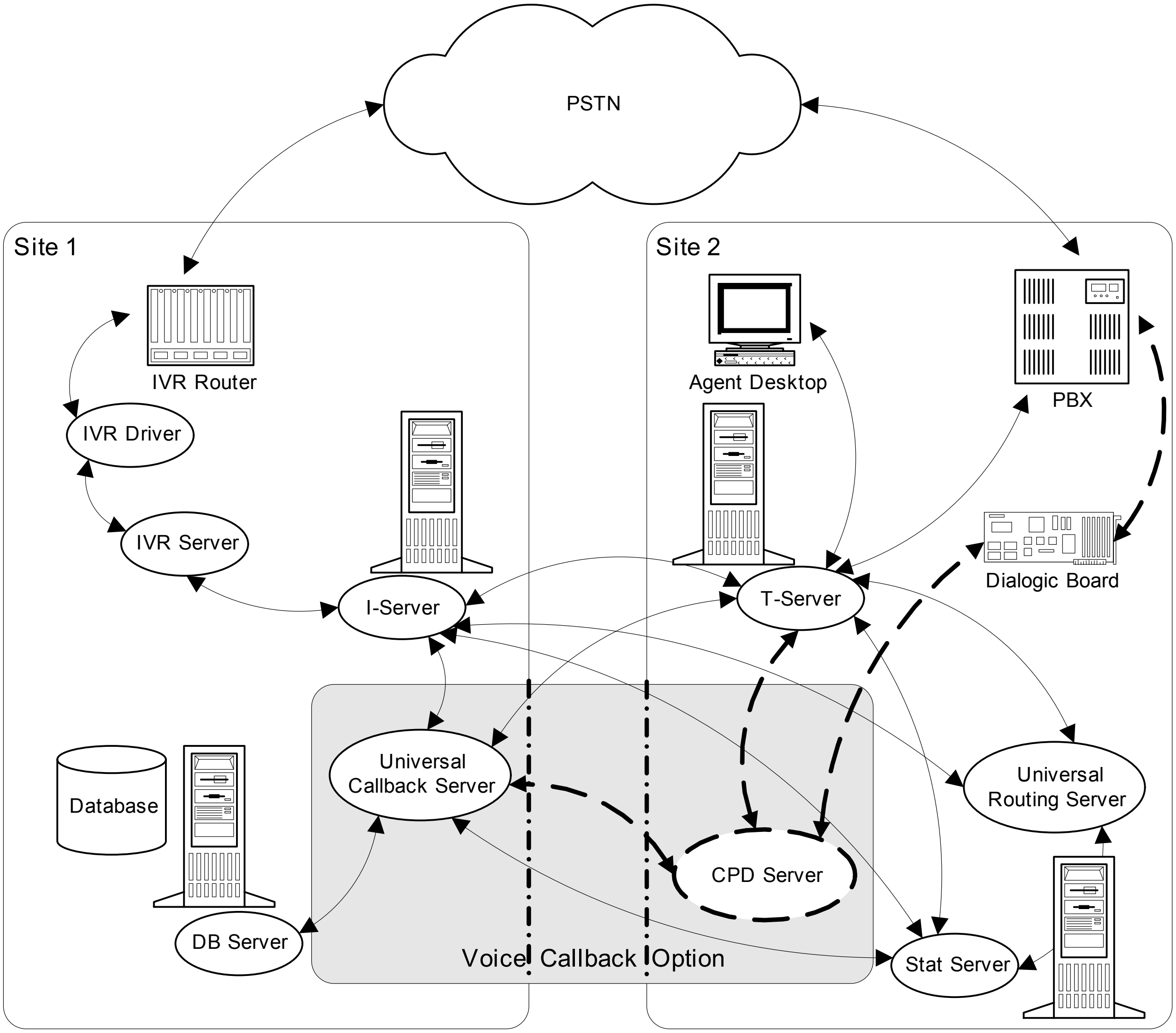


Figure 3: Interaction Between Components in IVR In-Front-of-the-Switch Mode

Voice Callback and Multi-Channel Routing

Figure 4 on page 35 shows callback requests between Web MCR components.

Note: The Web interface architecture includes interactions with the Universal Contact Server. The Web interface and the ability to place callback interactions into contact history are independent functionalities.

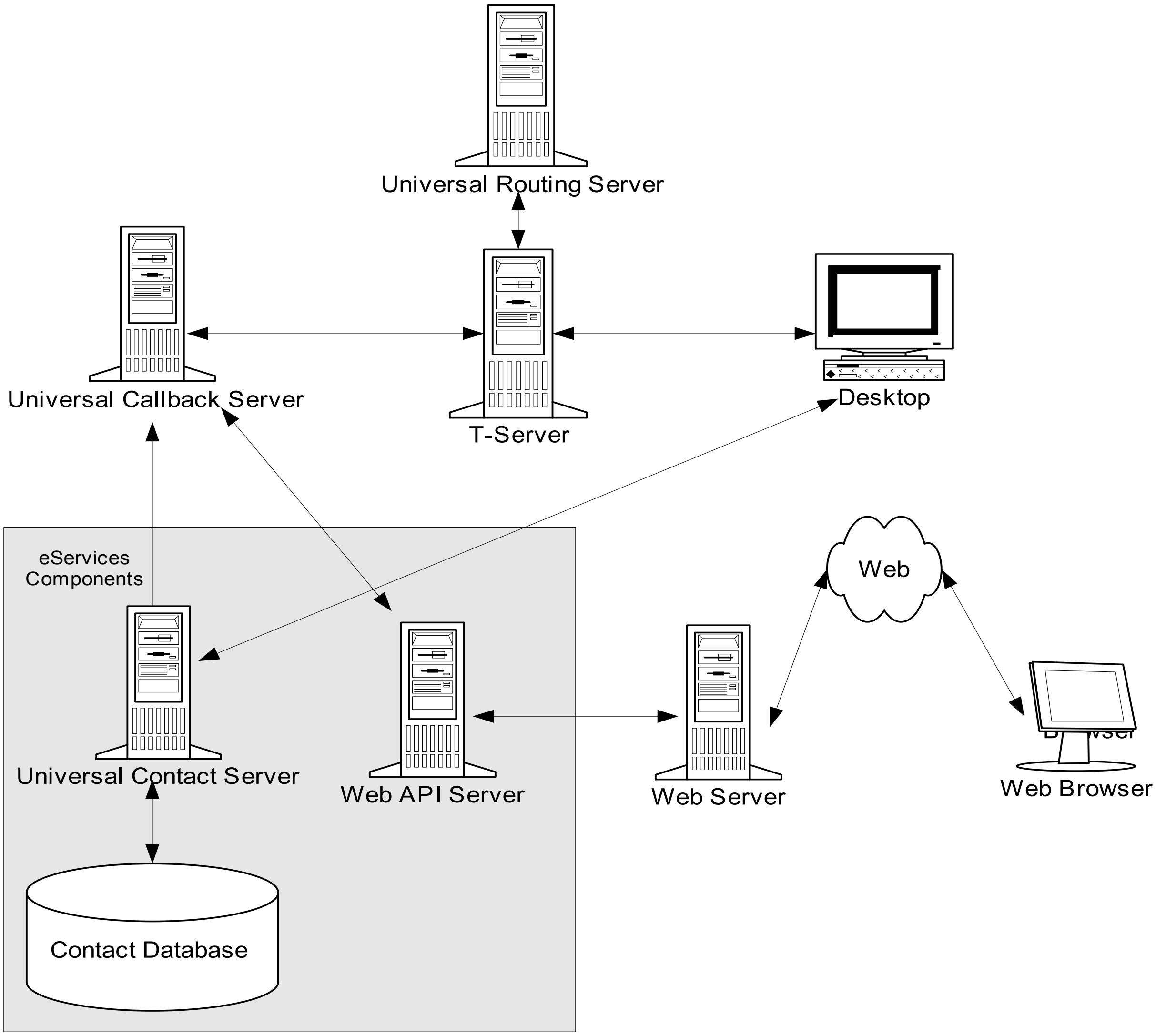


Figure 4: Callback Requests Between Web MCR Components

Genesys Solutions and Components Required for Voice Callback

Voice Callback utilizes the technology of other Genesys solutions. This section describes, in more detail, the applications and components that are required for Voice Callback and discusses specific ways in which they enable VCB functionality. Solutions and components are discussed in this order:

- Framework
- Stat Server (described under Framework Services Layer)
- Universal Routing: Enterprise Routing (ER) and Network Routing (NR)
- IVR
- Agent desktop
- CPD Server
- Universal Callback Server
- Web API Server
- Universal Contact Server

The software is discussed in this order to allow a clearer understanding of Voice Callback. You do not have to install and configure these components in this exact order, except for Framework and then Universal Routing which you install and configure first.

You can decide when to add the Voice Callback option of Universal Routing, as long as the other software required for Voice Callback functionality is included in the overall system configurations at your contact center. The Universal Callback Server is discussed last in this section because of its dependence on the other VCB applications/components. Having first clarified how these other components support and enable Voice Callback functionality, the role of the Universal Callback Server in relation to them should become clear.

Framework

Genesys Framework is a mandatory part of any Genesys-based, interaction-management system. Framework provides functions required for the normal operation of any Genesys solution or option.

Figure 5 represents the layers of Framework that support Voice Callback.

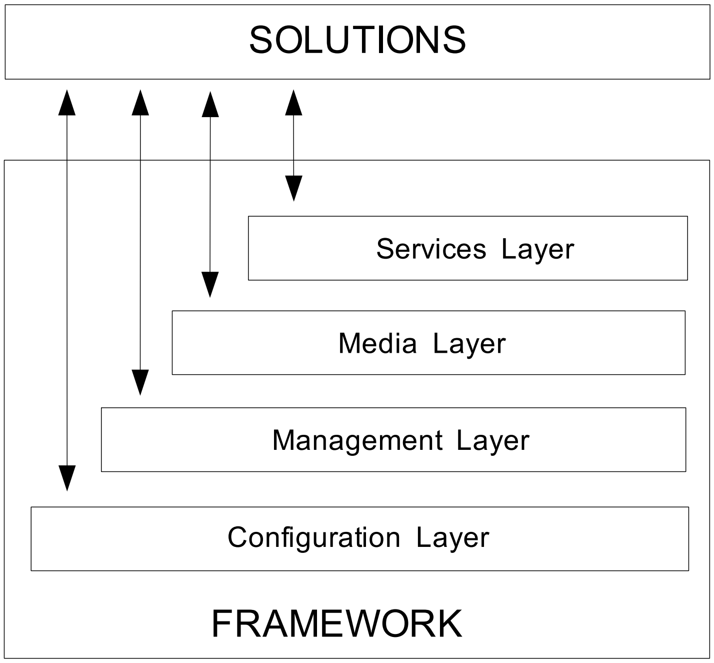


Figure 5: Framework Architecture Underlying VCB

The Configuration Layer includes these components:

- Configuration Server
- DB Server
- Configuration Manager
- Configuration Database

The Management Layer (optional) includes these components:

- Local Control Agent
- Message Server
- Solution Control Server
- Solution Control Interface

The Media Layer includes:

- Enterprise T-Servers and/or
- Network T-Servers

The Service Layer includes:

- Stat Servers
- DB Servers

Framework Deployment Guide describes Framework architecture in detail, layer by layer, including each layer's components and functions. The following examples show some of the ways in which VCB depends on the Framework layers.

Configuration Layer

This Framework layer centralizes the processing and storage of all the data required for Genesys solutions to work within a particular environment.

The configuration data required to run the Universal Callback Server (the primary component of Voice Callback) and other applications/components that enable callback functionality are configured in the Configuration Manager and stored in the Configuration Database.

The "Deployment Planning Worksheet" on page 181 shows the type of configuration data required to implement VCB.

Management Layer

This Framework layer performs these types of functions:

- Automatically starts and stops the Universal Callback Server and other applications that perform functions pertaining to VCB, such as, the T-Server, Stat Server, DB Server, and CPD Server (optional).
  See the "Deployment Planning Worksheet" on page 181 for more information.
- Monitors the status of Application objects configured to support VCB. See the "Deployment Planning Worksheet" on page 181 for more information.
- Logs and signals maintenance events. The logging approach may use the Message Server, which provides centralized processing and storage of maintenance events.

Media Layer

Enterprise T-Servers and Network T-Servers comprise this Framework layer. T-Server, which is a computer-telephony integration (CTI) server, provides a transport layer for telephony information for the system. All information to and from the Agent Desktop and other servers passes through T-Server.

Voice Callback relies on T-Servers to distribute the callback-request data. User data, attached to the original inbound call, passes to the callback request (virtual call) and to the callback (outbound call). The T-Server is the messenger, so to speak, that delivers the data by means of the `UserData` attribute of a T-Server event. This data is available for routing, reporting, and Agent Desktop applications.

T-Server clients communicate primarily through the data attached to requests and responses.

In Voice Callback, requests have the key-value pair:

```
VCB_USER_EVENT_REQUEST <Request Type>
```

For example, the IVR sends this request, via T-Server, to the Universal Callback Server to determine the availability of Voice Callback service:

```
VCB_USER_EVENT_REQUEST RequestCallbackServiceStatus
```

In Voice Callback, responses have the key-value pair:

```
VCB_USER_EVENT_RESPONSE <Request Type>
```

For example, Universal Callback Server sends this response, via T-Server, to IVR to confirm the availability of Voice Callback service:

```
VCB_USER_EVENT_RESPONSE RequestCallbackServiceStatus
```

For information about predefined attributes of T-Server events and key-value pairs specific to Voice Callback, see the “Predefined Attributes” section of Chapter 4 of the *Voice Callback 7.1 Reference Manual.*

Services Layer

This Framework layer includes Stat Server(s) and DB Server(s). In Voice Callback, the Stat Server calculates the EWT for an agent to answer an incoming call and Statistical Data for real-time and historical reporting. The EWT is sent to the IVR, which informs the caller, who can then decide whether to wait on hold or to request a callback (`ASAP` or `Scheduled`).

Genesys recommends using one DB Server for the Configuration Database and another DB Server for the `Calling List` table and possibly for the `Black List` table, which provides storage of data regarding callback requests.

In the “Deployment Planning Worksheet” beginning on page 181, see “Stat Server(s)” on page 184 and “DB Servers” on page 184.

These examples reveal some of the functions that Framework performs for Voice Callback. For more information about Framework architecture, see *Framework Deployment Guide*.

Universal Routing: Enterprise Routing and Network Routing

Depending on the size of your contact center, you can use either Enterprise Routing (ER) or Network Routing (NR) to perform these types of functions for Voice Callback:

- Obtain the EWT from Stat Server
- Check callback-service availability if EWT requires the callback option
- Route `ASAP` callbacks and `Scheduled` callbacks
- Route callback requests by such criteria as:
  - Queue-by-queue selection
  - Call-by-call selection
  - Date/time selection

  See also “Routing Strategies” below and “Routing Targets” on page 40.
- Track an agent’s real-time availability.
- Prioritize callback requests.
- Support a multi-site environment.

Both ER and NR enable fast and flexible customer service by means of routing strategies.

Routing Strategies

A *strategy* is a set of decisions and instructions that tell Enterprise Routing and Network Routing components how to handle and where to direct interactions in different circumstances. Strategies use various criteria to route interactions—for example, agent availability, customer information such as the account number or the type of account, arrival date/time of a call, or a combination of different criteria. Routing strategies enable a contact center to offer the callback option based on queue-by-queue selection, call-by-call selection, or hour/day/time selection. (For more information about these criteria, see “Contact Center Service Options” on page 19.)

To achieve VCB objectives, the Universal Routing Server (URS) utilizes routing strategies to request the EWT, to determine the call’s position in the queue, and to assign priority to callback requests.

VCB requests (`ASAP` callback and `Scheduled` callback; ACD Queue or Routing Point in `Autodial` mode) are submitted to the routing engine to be delivered to agents. Based on routing rules for specific business needs and objectives, the URS routes an interaction to an appropriate resource in the shortest possible time. Requests are routed to the most qualified agent (who is best skilled to handle the particular call), according to the routing strategy applied to the original inbound call.

The routing strategies result in fewer transfers among agents, shorter interaction-handling times, increased agent efficiency, and improved customer satisfaction.

For more information about developing routing strategies, see “Designing Routing Strategies” on page 134. You can use the Interaction Routing Designer (IRD)—which is installed and configured as part of the ER/NR configuration—to create, modify, load, and monitor routing strategies. For instructions on starting the IRD, see “Starting Interaction Routing Designer” in the *Universal Routing Deployment Guide*. For information about using the IRD to design routing strategies, see *Interaction Routing Designer Help.*

Routing Points

Voice Callback uses both physical Routing Points (RPs) (or Routing Queues) and Virtual Routing Points (VRPs). Virtual Routing Points are mainly used for virtual call routing. (See “Virtual Elements in Voice Callback” on page 41.)

Routing Targets

A *routing target* is a set of resources to which a call is directed. The target could be an Agent, Agent Group, Place, Place Group, ACD Queue, or RP; an agent with a particular type of knowledge or skill for customer assistance; or any combination of these. Business queues correspond to routing targets.

Virtual Elements in Voice Callback

The virtual elements of Voice Callback are best understood by way of analogy. The following analogy compares the virtual elements—virtual call, virtual routing point, virtual queue—with their physical counterparts—an actual call, a routing point, and a business queue.

Analogy

Imagine that you go to a very popular restaurant on a Saturday night. When you arrive, there is a long line of people waiting for tables. You can either stand in line with the others or add your name to a waiting list and be called when a table is available. When you hear the EWT for a table, you choose to put your name on the waiting list and leave to enjoy a view of the city.

In this analogy, your arrival at the restaurant represents an incoming call. The option to physically stand in the waiting line is akin to waiting on hold. As the line of people moves forward, the prospective diners (incoming calls) physically move to the desk where the hostess directs them to a table; the hostess at the desk is the physical routing point, and the table is the routing target. Your request to be put on the waiting list (so you can leave to enjoy the view) represents requesting a callback and then hanging up. Your name on the waiting list represents you when you are no longer physically standing in line.

The waiting list symbolizes a virtual queue, where your virtual presence (your name on a piece of paper) resides while your request (callback request/virtual call) is being processed. When your name reaches the top of the waiting list, your request is routed to a live queue. At that point, you are called back, and the whole virtual process is replaced by physical counterparts. In Voice Callback, the physical counterparts are a real call from a live agent associated with a business queue.

Interactive Voice Response

Voice Callback may use an Interactive Voice Response (IVR) product. As previously mentioned, you could use Genesys Voice Portal (GVP) or the Genesys Voice Treatment Option (VTO) to provide that functionality. See “Genesys Voice Platform Enterprise Edition” on page 42 and “Voice Treatment Option” on page 43 for more information. Any Genesys-compliant IVR product functions with VCB. Here *Genesys-compliant* means that the IVR can read and update user data attached to a call. To do this, the IVR works with Genesys IVR Drivers, or the IVR interacts with Genesys Framework through a T-Library interface.

IVR Scripts

IVR software interfaces with Genesys Framework. An IVR script, executed by the IVR Driver, can enable these VCB functions:

- Announces the EWT (calculated by the Stat Server) for a customer on hold in a queue, if the caller's expected hold time exceeds a defined threshold.
- Announces the callback-request options (ASAP or Scheduled).
- Provides automated number identification (ANI) and announces (plays back) the caller's telephone number for confirmation.
- Accepts announced ANI or customer-entered digits (CEDs) for the callback request.
- Verifies (for a Scheduled callback) that the CEDs for the date and/or time are outside the range of the estimated waiting time (EWT). If either the date or the time that the caller requests for the callback is within the configured EWT, the IVR script responds to the invalid entry with a prompt such as Try again or informs the caller, "Since your entry is within the estimated waiting time, you will be called back ASAP."
- Determines if the callback date and time are within standard operating hours (SOH). If they are outside the SOH, the IVR script prompts the user with a message such as, "Time and date are outside standard operating hours. Please enter a date and time within standard operating hours."
- Retains customer data from the original call.
- Assigns priority to callback requests.

Genesys Voice Platform Enterprise Edition

Genesys Voice Platform Enterprise Edition, when integrated with Genesys Framework, provides voice processing for call routing and voice self-service at your contact center. In addition to traditional IVR functionality, this product provides an open framework for integrating Voice Extensible Markup Language (VoiceXML) and speech technologies into your contact center environment.

You can position Genesys Voice Platform Enterprise Edition as a solution in-front-of- or behind-of- a premise-based switch configuration.

- In an *in-front-of-the-switch* configuration, calls terminate on a Voice Communications Server (VCS) and do not pass through an onsite ACD Queue. The VCS uses a Dialogic circuit board that connects to T1 or E1 line(s) that carry the voice calls.
- In a *behind-the-switch* configuration, calls pass through an ACD Queue first and are then sent to the Voice Communications Server (VCS). In this configuration, the VCS has a Dialogic circuit board that connects to the premise-based switch.

VCS parses, interprets, and executes the VoiceXML commands in the XML documents served by the voice applications. VCS also integrates with text-to-speech (TTS) and automatic speech recognition (ASR) software.

For more information, see *Genesys Voice Platform Enterprise Edition Getting Started Guide*.

Voice Treatment Option

Genesys Voice Treatment Option (VTO) also functions as an IVR and is fully integrated with Genesys Framework.

VTO is a client-server application that works with Enterprise Routing (ER) solution to apply treatments to incoming calls. These treatments are designed to collect information prior to routing, identify the appropriate queue for a call, and direct the call to that queue.

Voice Treatment Option works with Enterprise Routing (ER) and T-Server to provide two services:

- Pre-routing Information Collection—VTO enables ER to collect data, such as account numbers or menu preferences, from callers. This information is attached to the call so that ER can use it to make call-processing decisions.
- Queuing Call Treatments—VTO helps ER define voice treatment for, and apply voice treatments to, queued calls. Multiple voice treatments can be combined with other call-handling actions (that is, transfers, holds, timeouts, and terminations) and implemented in any sequence you choose.

VTO Components

- Voice Treatment Server (VTServer)
- Voice Treatment Manager (VT Manager)

VT Server executes voice treatment scripts. VT Manager is the graphical user interface used to build and maintain the scripts. The building blocks for each script are objects representing actions, which are stored in the Configuration Database. The finished script is also stored as a Configuration Database object.

Script Types

The VT Manager defines four functional script types:

Simple Queuing Provides basic queuing features such as music treatments and voice messages that play until an agent becomes available to take the call.

Enhanced Queuing Used with more complex queue-wait and queue-position voice types. Any queuing script that includes queue-wait or queue-position voices is considered an `Enhanced Queuing` script.

Service Selection Enables callers to choose from a set of specific options. Callers make choices using the touchtone keypads on their telephones.

Data Collection Comprised of three voice files:

- One that prompts the caller to enter the desired data
- Another that replays the data to the caller
- A third that requests confirmation of the caller’s entry

Any script that has at least one Data Collection voice is considered a `Data Collection` script.

The caller input is attached to the call, so it is available to agents later in the call. The information is automatically stored and made available for subsequent processing by the VT Manager or other Genesys applications.

Chapter 7, "Designing IVR Scripts and Routing Strategies," on page 125 includes a thorough discussion of IVR logic and IVR scripts.

Agent Desktop

Genesys Agent Desktop (GAD) is the recommended agent desktop for VCB. This out-of-the-box agent desktop can perform these VCB functions:

- Distinguish a callback request from a new inbound call
- Display callback requests, including required customer data
- Enable the agent to perform actions such as these:
    - Preview a callback request on the GUI
    - Dial out for the callback
    - Reschedule a callback request
    - Apply a call result to an unsuccessful callback attempt, which, in turn, prompts a treatment
    - Cancel a callback request
    - Reject a callback request
    - Create new callback request
    - Get information on existing callback request(s)
    - Communicate with Contact Database

Note: GAD automatically manages VCB functionality if the Universal Callback Server is available and running on the network.

You can use a desktop toolkit to incorporate the VCB functions into your existing agent desktop applications or into another (third-party) agent desktop application. For more information, please see *Genesys Interaction SDK Developer's Guide*.

CPD Server for Autodialing Mode

Voice Callback offers the `Autodial` mode as an option for dialing callback requests. The system places the call automatically at the appropriate time. The CPD Server listens to the progress of the call—for example, ringing, busy signal, no answer—and then connects the call to an agent when the server detects a customer on the line.

CPD Server uses an Intel Dialogic board to perform call progress detection. For more information about the CPD Server, see "CPD Server Dialogic Hardware Setup" on page 159.

CPD Proxy

The CPD Proxy is the load-distribution component of CPD Server. It distributes calls among various CPD Servers for optimal load-balancing and utilization of system resources when the volume of calls is particularly heavy at call centers. CPD Proxy also notifies clients about changes in hardware resources due to configuration or license modifications or "out of service" events caused by hardware problems. Generally, CPD Proxy can be used in any situation to improve overall system stability and load distribution.

Web API Server

Web API Server is a Web Application (servlet) and is collection of common Genesys codes and values that the http Web server (for example, Apache) and Web application container (for example, Tomcat) use in order to handle the requests that Web-based Genesys components submit. It interacts with Universal Callback Server in order to allow the submission of callbacks from the Web page and to perform some management functions (search for callback by phone, cancel callback, get detailed information about callback, or request estimated waiting time).

See Figure 4 on page 35 on operations within MCR components.

Universal Contact Server for Contact History

Functionality that depends on the Contact Database is optional for the callback operation. Universal Contact Server allows VCB automatically to:

- Register callback interactions in contact history. This is applicable to all callbacks regardless of callback request origination (Web, IVR, Agent Desktop).
- Create new contacts in contact database if existing contact is not found for callback interaction.

Note: VCB is able to operate without the Contact Database.

Universal Callback Server

The preceding sections in this chapter describe the ways in which Voice Callback depends on Framework, Universal Routing (ER and NR), an IVR product, an Agent Desktop, Web API Server, and Universal Contact Server.

The remainder of this chapter discusses the functional relationship of Universal Callback Server to the other components that enable VCB. Universal Callback Server, as the primary component of Voice Callback, does the following:

- Processes callback orders.

- Maintains a list of ordered callbacks.
- Creates virtual calls (placeholders for callback requests).
- Requests Universal Routing Server to provide callback destination and then awaits response.
- Delivers callback and attached data to agent.
- Processes callback-failure treatments.
- Resolves duplicate callback requests.
- Handles exceptions to the Universal Routing Server's default response to a routing strategy error.

Communication Among Voice Callback Components

This section discusses communication between VCB components and some of the specific tasks that Universal Callback Server performs.

Universal Callback Server and IVR Communication

It is important to recall here that the VCB Client-Server protocol is based on T-Server user events, which are identified by key-value pairs:

- Requests have this key-value pair:
  `VCB_USER_EVENT_REQUEST` *`<Request Type>`*
- Responses have this key-value pair:
  `VCB_USER_EVENT_RESPONSE` *`<Request Type>`*

The Universal Callback Server does not interact directly with IVR; however, by means of T-Server events (requests and responses) with attached data, Universal Callback Server does communicate with IVR to accomplish tasks such as these:

- Provide licensing control
- Provide information about Voice Callback service availability to IVR
- Receive and process callback requests from IVR
- Resolve duplicate callback requests
- Receive and process management requests from IVR and send corresponding data back to IVR
- Schedule a callback request, including:
  - Requesting EWT from Stat Server and uses that data to process a `Scheduled` callback request
  - Verifying that a requested callback date and time are within standard operating hours (SOH)

- Verifying that a requested callback date and time are not within the EWT

By means of specific key-value pairs attached to the T-Server's `UserData`, IVR is able to specify the type of request to Universal Callback Server.

To send a callback request to Universal Callback Server, IVR delivers the original call (with specific key-value pairs in the `UserData`) to a Routing Point that Universal Callback Server monitors.

Sample Call Flow Involving IVR

The following explains a sample call flow for an incoming call that results in a scheduled callback request. Universal Callback Server delivers the callback request data to the agent desktop in `Preview` dialing mode. The agent reviews the data and dials the callback telephone number.

1. Incoming call arrives at the contact center (PBX).
2. IVR application collects the customer's account number.
3. IVR sends a routing request to the routing engine (Universal Routing Server in this case).
4. Router predicts the EWT for the call, based on the Stat Server's calculation.
5. If the predicted EWT is over a predefined threshold, IVR announces the caller's options: `ASAP` callback, `Scheduled` callback, or `Wait on Hold`.
6. If the caller requests a callback (`ASAP` or `Scheduled`), IVR collects the caller's telephone number. If the callback request is the `Scheduled` type, IVR also asks when (the preferred date/time) the caller would like to receive a callback from an agent.
7. Customer hangs up.
8. Universal Callback Server sends the callback request to URS in the form of a T-Server event with attached data. It also sends the callback to URS in advance for estimated waiting time.

Note: This is applicable for scheduled callbacks only because the customer does not request the time for ASAP callbacks.

9. URS selects an appropriate agent and informs Universal Callback Server.
10. Universal Callback Server sends the callback request data in `Preview` dialing mode to the agent's desktop.
11. Agent reviews the callback-request and dials the telephone number.

Callback Request Submission Scenarios

Universal Callback Server supports several different types of callback request submissions that are discussed in this section:

- Submission using Universal Callback Server client-server protocol: see Figure 6.
- Submission on Routing Point triggered by `EventRouteRequest` or `EventPartyChanged`, IVR behind-the-switch-configuration: see Figure 7 on page 51.
- Submission on the Routing Point triggered by `EventTreatmentEnd`, IVR-behind-the-switch-configuration: see Figure 8 on page 53.
- Submission in IVR-in-front-of-the-switch configuration: see Figure 9 on page 54.
- Submission using the Web Interface: see Figure 4 on page 35.

Submission Using Universal Callback Server Client-Server Protocol

In this scenario, a CTI-enabled IVR or a routing strategy executed by the Universal Routing Server or Agent Desktop application performs all VCB Service-related tasks through Universal Callback Server client-server protocol. See Figure 6.

.

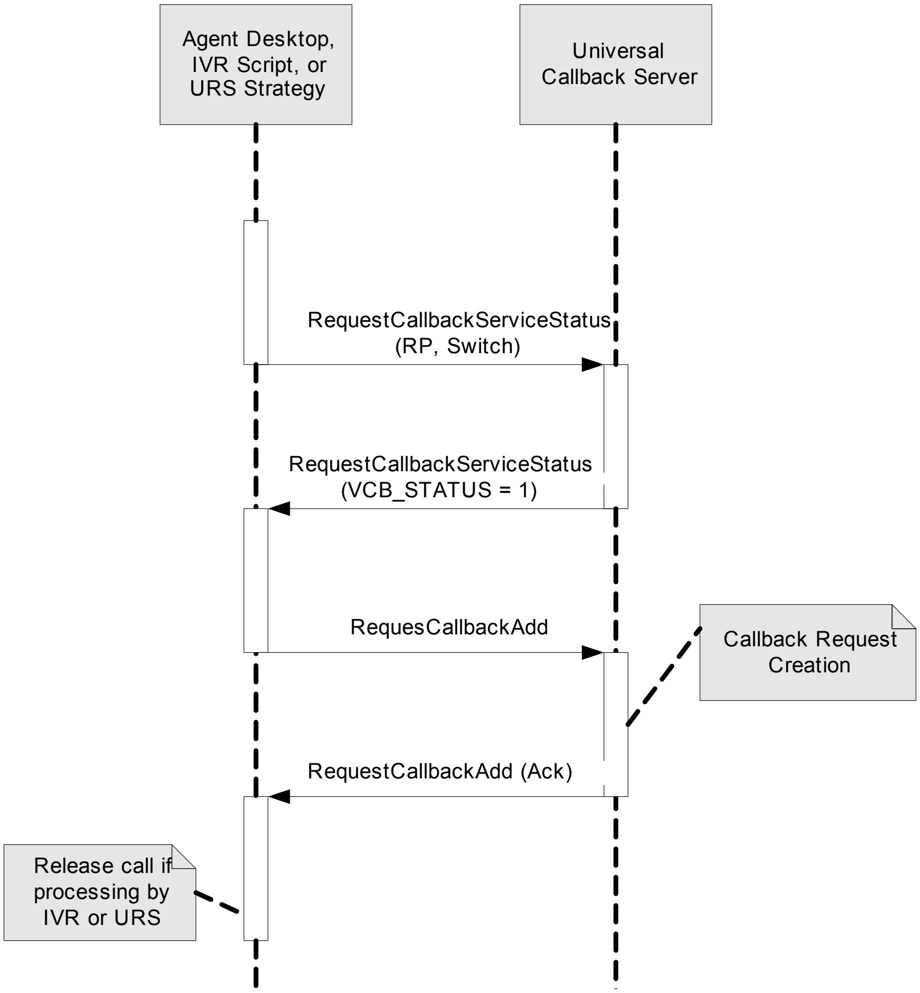

Agent Desktop, IVR Script, or URS Strategy
Universal Callback Server
RequestCallbackServiceStatus (RP, Switch)
RequestCallbackServiceStatus (VCB_STATUS = 1)
RequesCallbackAdd
Callback Request Creation
RequestCallbackAdd (Ack)
Release call if processing by IVR or URS

Figure 6: Submission Using Universal Callback Server Client-Server Protocol

Submission on Routing Point Triggered by EventRouteRequest or EventPartyChanged

This kind of callback request submission on the Routing Point (IVR-behind-the-switch configuration) can be used in the following deployment scenario:

- IVR transfers the inbound call to the Routing Point on the premise switch to check callback service availability (optional step).
- Universal Callback Server confirms its service availability on the given Routing Point.
- Provided that all targets are not available, the routing strategy that is loaded on this Routing Point transfers the call to the IVR for IVR treatment.
- IVR offers a callback to the customer and collects callback-related data. IVR then attaches all the collected data to the call and transfers the call to the Routing Point.
- Universal Callback Server processes the callback request and confirms its successful submission.

- Provided that all targets are not available, the routing strategy that is loaded on this Routing Point transfers the call to the IVR for IVR treatment.
- IVR confirms to the customer that a successful callback submission has occurred and then releases the call.

Note that in order for Universal Callback Server to react on EventRouteRequest (in case of single-step transfer from IVR) or EventPartyChanged (in case of mute or two-step transfer from IVR), the following needs to be done:

- Universal Callback Server's option urs_treatment_mode should be set to transfer. (See *Voice Callback Reference Guide* for further information on this option.)
- Call which arrives to the Routing Point needs to have callback-specific key-value pairs attached in its user data. (See "Services Layer" on page 39 and "Universal Callback Server and IVR Communication" on page 46.)

See Figure 7 on page 51.

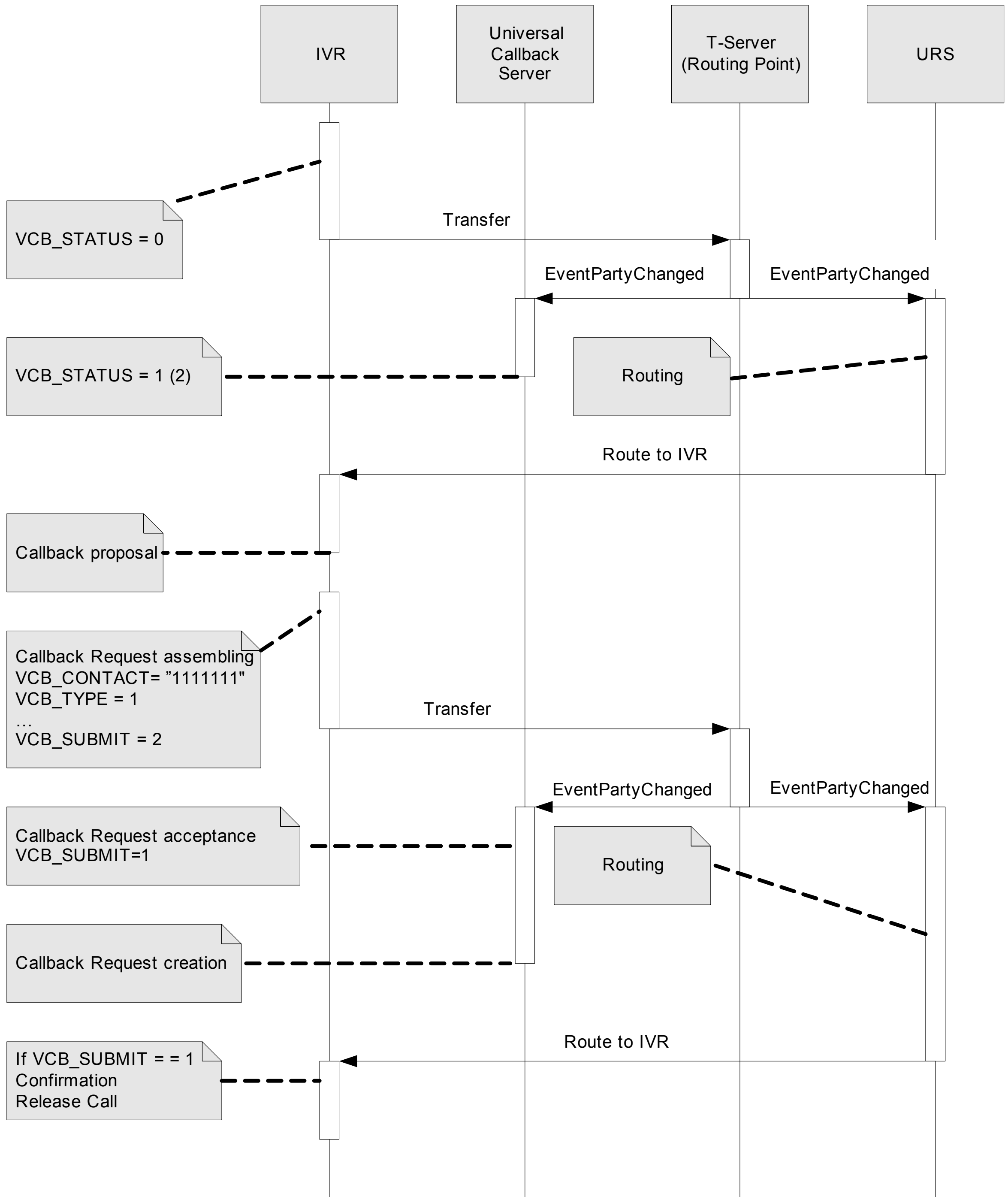


Figure 7: Callback Request Submission on Routing Point Triggered by EventPartyChanged (or EventRouteRequest)

Submission on Routing Point Triggered by EventTreatmentEnd

This type of submission is very similar to the one described above. The only difference is the way that a call is delivered to the Routing Point. In this scenario, the call is passed to the Routing Point as a result of a specific IVR treatment: therefore, an Event that Universal Callback Server expects is

`EventTreatmentEnd`. This kind of callback request submission on the Routing Point (IVR-behind-the-switch configuration) can be used in the following deployment scenario:

- A call arrives to the Routing Point. No targets are available for routing, so URS applies treatment of type IVR to it.
- IVR applies a treatment to the call and passes the call back to the Routing Point to check callback service availability (optional step).
- Universal Callback Server confirms its service availability on a given Routing Point.
- Provided that all targets are not available, the routing strategy that is loaded on this Routing Point passes the call to the IVR for IVR treatment again.
- IVR offers a callback to the customer and collects callback-related data. IVR then attaches all the collected data to the call and passes the call to the Routing Point.
- Universal Callback Server processes the callback request and confirms its successful submission.
- Provided that all targets are not available, the routing strategy that is loaded on this Routing Point passes the call to the IVR for IVR treatment.
- IVR confirms to the customer that a successful callback submission has been made and then releases the call.

Note that in order for Universal Callback Server to react on EventTreatmentEnd, the following needs to be done:

- Universal Callback Server's option `urs_treatment_mode` should be set to treatment. (See *Voice Callback Reference Guide* for further information about this option.)
- Call which arrives at Routing Point needs to have callback-specific key value pairs attached in its user data (See "Services Layer" on page 39 and "Universal Callback Server and IVR Communication" on page 46.)

See Figure 8 on page 53.

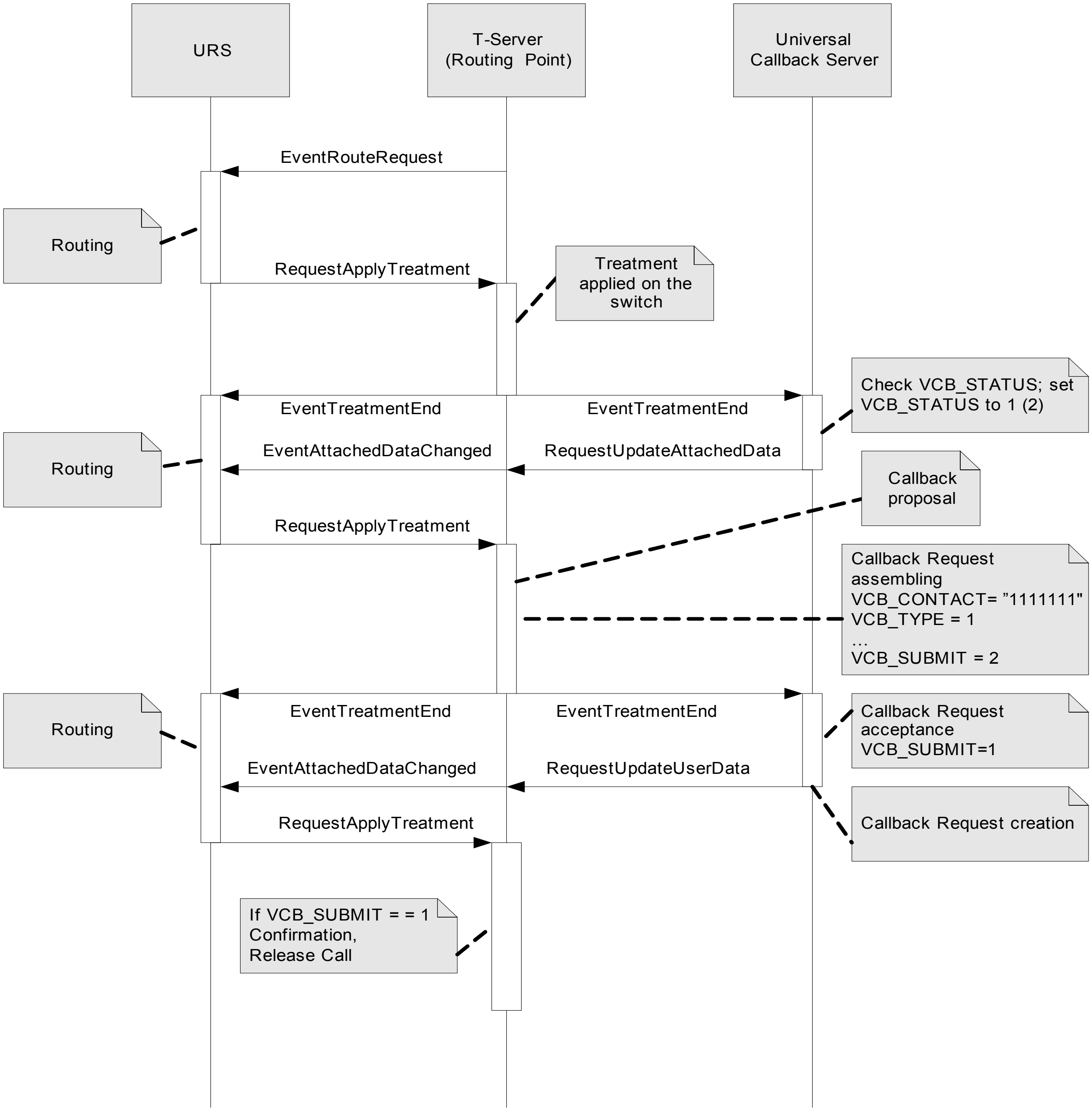


Figure 8: Callback Request Submission on Routing Point Triggered by EventTreatmenEnd

Submission in IVR In-Front-of-the-Switch Configuration

This scenario is not very different from the submissions in IVR-behind-the-switch configurations. Universal Callback Server relies on the same telephony events to trigger the callback submission. However, due to the nature of IVR in-front-of-the-switch configuration, the Virtual Routing Point is used instead of the Routing Point as the DN which Universal Callback Server monitors for telephony events which trigger callback submission. See Figure 9 on page 54.

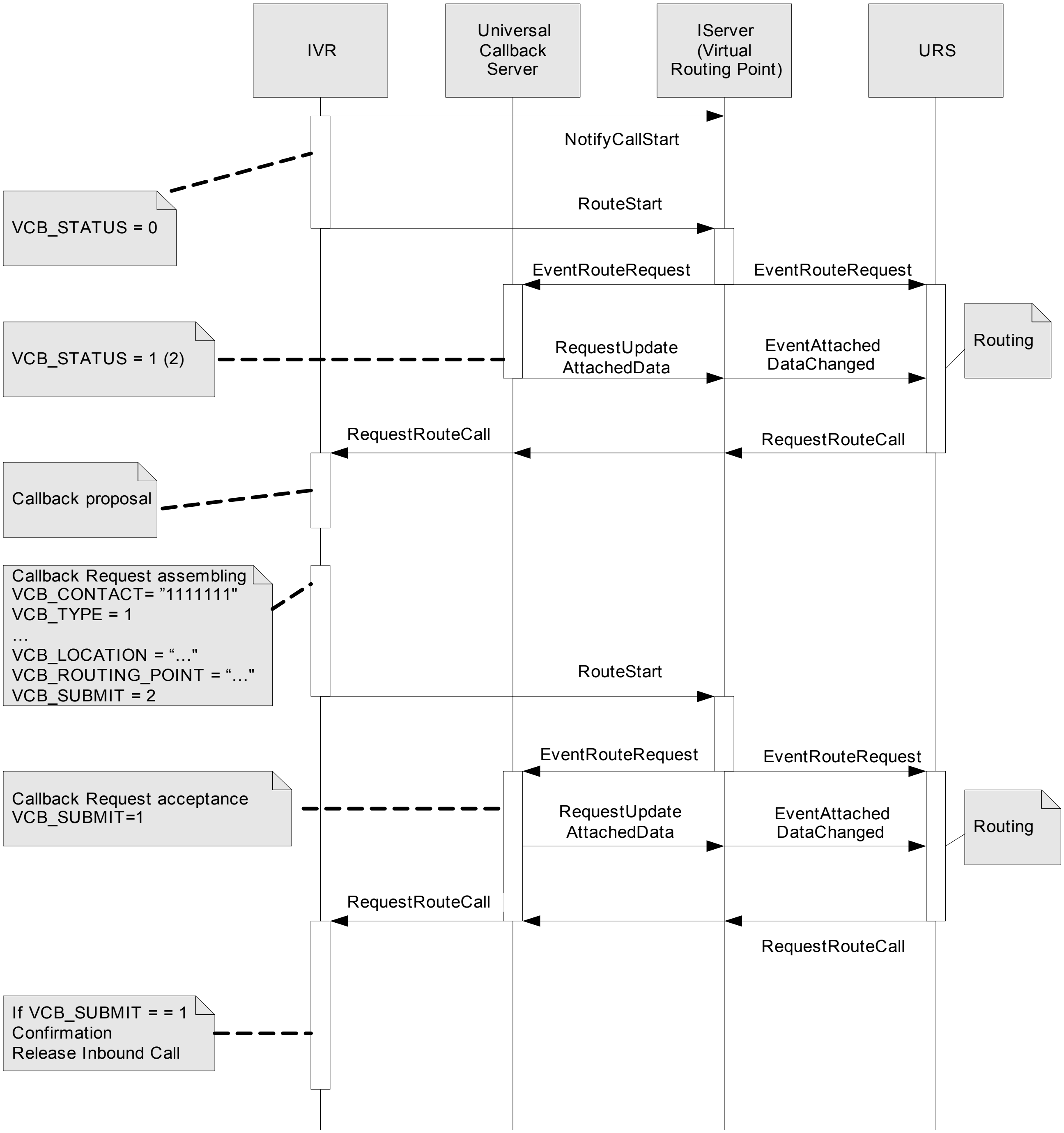


Figure 9: Callback Request Submission on Virtual Routing Point in IVR In-Front-of-the-Switch Mode

Universal Callback and Web API Server Communication

The following section describes the communication between the Web API Server and the Universal Callback Server. Universal Callback Server communicates with Web API Server to accomplish tasks such as these:

Table 1: Supported Requests

Request	Description
Request callback	Submit request ASAP or schedule callback from the Web. Parameters of request are: • Customer number for callback. • Desired time of callback (date, and time) (optional). If this parameter is not given, the callback attempt is executed ASAP. • Time specified to terminate attempts to call customer. • User Data entered by customer or provided by application.
Cancel callback	Cancel request; that is, remove the request from the list of requests. Parameters of request are: • Customer number or request identifier.
Get request properties	Instruct VCB Servers to find the request and respond with requested information. Parameters of request are: • Customer number or request identifier.

Table 1: Supported Requests (Continued)

Request	Description
Get statistical data	Find the request's statistical data. Parameters of the request are: • Customer number or request identifier. • Status of callback. • Number of callbacks in queue. • Estimated waiting time in queue. • Position of specified callback in queue. • Estimated waiting time for specified callback.
Search callback	Find the requests with corresponding customer number in the list of requests. Parameter of the request is: • Customer number.

See Figure 10 on page 57 for Web callback protocol.

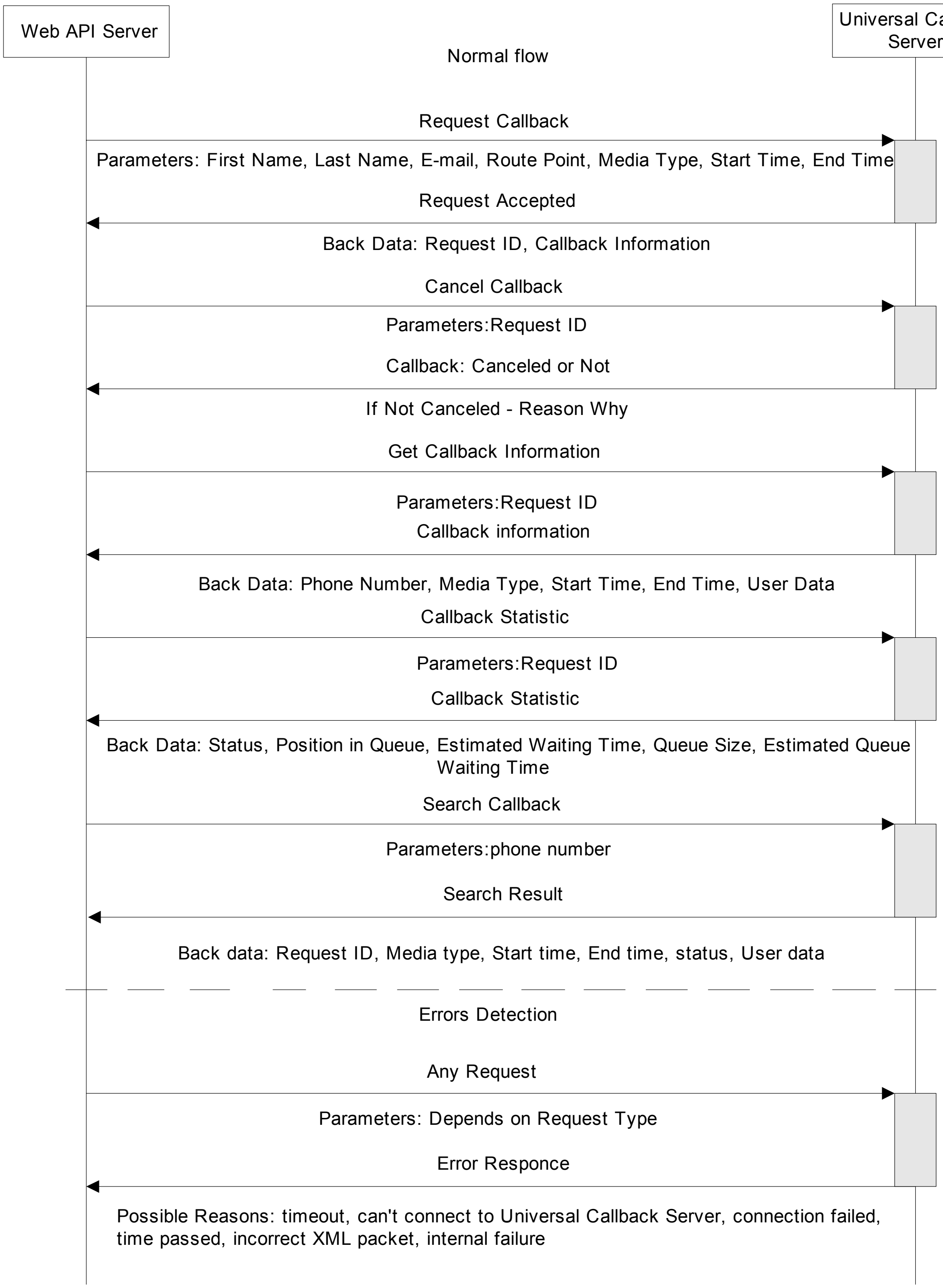

Web API Server
Universal Callback Server
Normal flow
Request Callback
Parameters: First Name, Last Name, E-mail, Route Point, Media Type, Start Time, End Time
Request Accepted
Back Data: Request ID, Callback Information
Cancel Callback
Parameters:Request ID
Callback: Canceled or Not
If Not Canceled - Reason Why
Get Callback Information
Parameters:Request ID
Callback information
Back Data: Phone Number, Media Type, Start Time, End Time, User Data
Callback Statistic
Parameters:Request ID
Callback Statistic
Back Data: Status, Position in Queue, Estimated Waiting Time, Queue Size, Estimated Queue Waiting Time
Search Callback
Parameters:phone number
Search Result
Back data: Request ID, Media type, Start time, End time, status, User data
Errors Detection
Any Request
Parameters: Depends on Request Type
Error Responce
Possible Reasons: timeout, can't connect to Universal Callback Server, connection failed, time passed, incorrect XML packet, internal failure

Figure 10: WEB Callback Protocol

Sample Call Flow Involving Web API

See *Multi-Channel Routing Web API Callback Sample*, which is a supplement to the *Multi-Channel Routing 7 Web API Client Developer's Guide* and gives information about Callback Samples that have been released with MCR 7.0.2.

Universal Callback Server and Universal Contact Server Communication

Universal Callback Server connects to Universal Contact Server in Universal Callback Server's Connections List. Universal Callback Server supports the protocol used by the Interaction Server in order to communicate with Universal Contact Server.

VCB Server creates interaction upon callback submission and closes it upon callback processing finalization (no more delivery attempts should be done).

VCB Server updates interaction status, attempted number, and call result for every callback delivery attempt.

See Figure 11 for Universal Callback Server and Universal Contact Server Interactions.

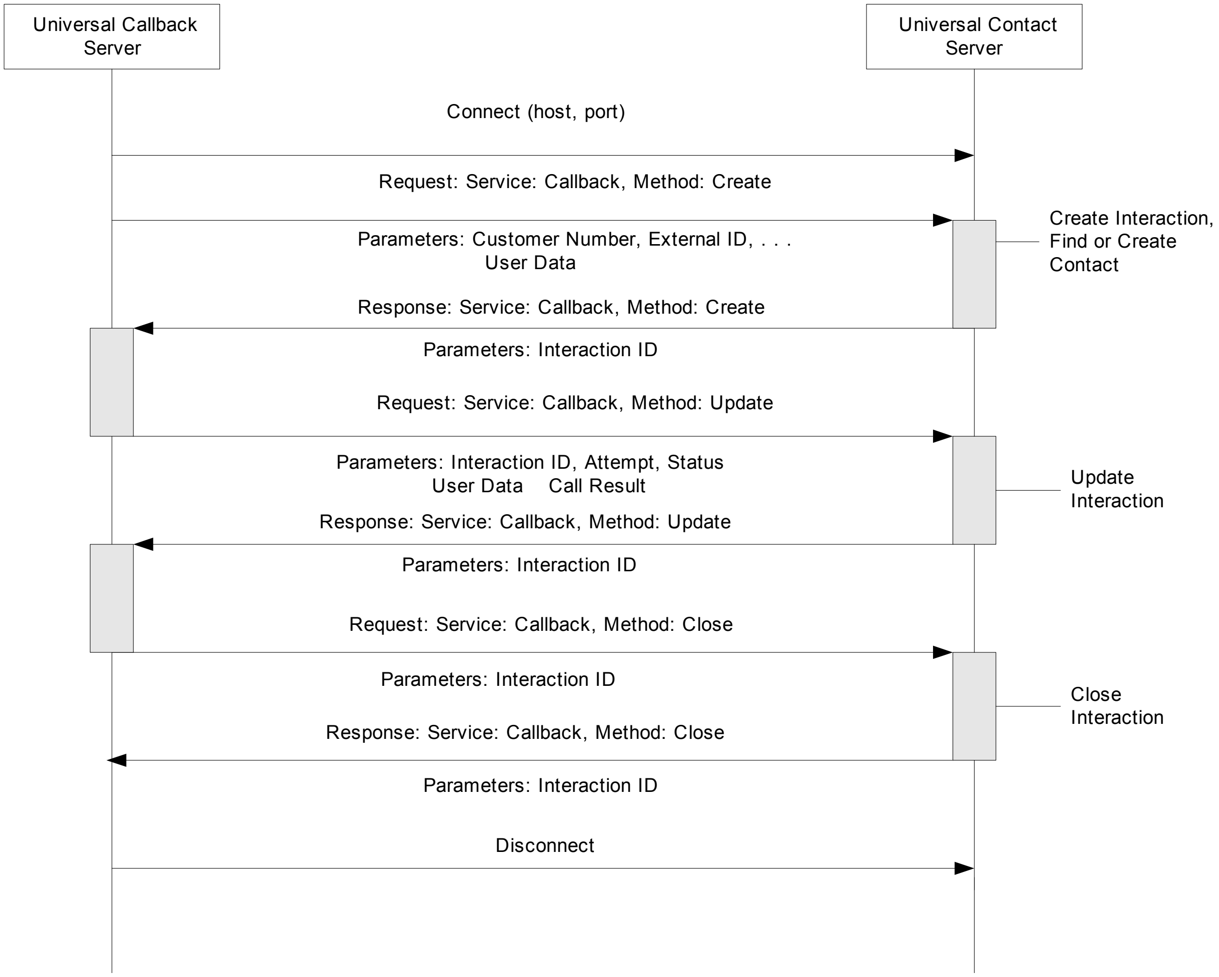

Universal Callback Server
Universal Contact Server
Connect (host, port)
Request: Service: Callback, Method: Create
Parameters: Customer Number, External ID, . . .
User Data
Create Interaction, Find or Create Contact
Response: Service: Callback, Method: Create
Parameters: Interaction ID
Request: Service: Callback, Method: Update
Parameters: Interaction ID, Attempt, Status
User Data Call Result
Update Interaction
Response: Service: Callback, Method: Update
Parameters: Interaction ID
Request: Service: Callback, Method: Close
Parameters: Interaction ID
Close Interaction
Response: Service: Callback, Method: Close
Parameters: Interaction ID
Disconnect

Figure 11: Universal Callback Server-Universal Contact Server Communication

Chapter

4 Deployment Procedures

This chapter presents the initial deployment procedures for VCB. It guides you through the process of licensing VCB and importing application templates for the Universal Callback Server and the Call Progress Detection (CPD) Server (optional). It also provides instructions for creating and manually configuring these Application objects. Finally, this chapter guides you through the VCB installation procedures.

This chapter contains the following sections:

The creation and configuration of other VCB-related objects are covered Chapter 6, "Configuration of Other Objects," on page 99.

Overview

Each of the following procedures is explained step-by-step in this chapter. The page numbers below point you to the specific sections.

Note: Before you start any of these procedures, make sure that the Configuration Manager is installed and running.

High-Level View of VCB Deployment

This high-level view of VCB deployment shows you how to:

1. Check licensing requirements: provide licenses for VCB and CPD Server (optional) (page 63).
2. Import application templates for:
   - Universal Callback Server (page 66)
   - CPD Server (optional, page 66).
3. Create and configure the Universal Callback Server Application objects from templates for:
   - Universal Callback Server Application object (page 67)
   - CPD Server Application object (optional, page 69).
4. Import application templates for:
   - Universal Contact Server (page 66)
   - Web API Server (page 66)
5. Create and configure the Web API Server and Universal Contact Server Application objects from templates for
   - Web API Server Application object (page 70)
   - Universal Contact Server Application object (page 70)

Note: If you do not have Multi-Channel Routing (MCR) components installed prior to VCB deployment, Web API Server and Universal Contact Server Application Objects have to be created from corresponding templates prior to their configuration, and templates have to be imported before that.

6. After you have configured the Application objects in Configuration Manager, install VCB on Windows or UNIX:
   - Install Universal Callback Server on Windows (page 71) or on UNIX (page 73).
   - Install CPD Server on Windows (page 72).

   If you do not use MCR Email/Web Media, install:
   - Third-Party components
   - Universal Contact Server (optional)
   - Web API Server

   See *Multi-Channel Routing Event/Media Deployment Guide.*
7. Configure these objects for VCB (page 99):
   - Database Access Point (DAP) object(s)
   - Table Access objects
   - Calling List object
   - Treatment object(s)
   - Routing Points (DN object)
   - Virtual Routing Points (DN object)
   - DN object for call termination

- Virtual Queue (DN object)
- Statistical Days
- Statistical Tables
- Time Zones

8. Design IVR scripts and routing strategies (page 139).
9. Design Web content to be used for callback-related activities (see *Multi-Channel Routing Deployment Guide).*
10. Install Dialogic boards (optional, page 153).
11. Start VCB (page 171).

Licensing

This section discusses the licensing features for VCB.

Note: Universal Callback Server includes all types of callbacks when calculating the number of callbacks allowed per license.

Types of Licenses for VCB

Preview License

- The VCB preview license limits available dialing modes to `Preview` dialing mode only; in this mode, a special User Event is delivered to agent’s desktop to signal screen pop; no actual telephony dialing occurs. `Autodial` (automatic dialing) mode is not allowed.

Full License

- The VCB full license enables both `Preview` and automatic dialing modes (with an optional CPD Server). It has no functional limitations.

  You can use either license.

Note: If you purchase Universal Callback Server, you can easily add the CPD Server by simply requesting a separate license for CPD Server.

See *Voice Callback Reference Manual* for licensing details and *Genesys 7 Licensing Guide* for system-wide licensing requirements.

Server-Side License Control

Table 2 illustrates how licensing control works in VCB Server:

Table 2: License Types

License Type	Number of Callback Requests
vcb_preview	Number of callback requests processed in Preview mode per 60 minutes (sliding window)
vcb_full	Number of callback requests processed in any mode (that is Preview and/or Auto Dialing) per 60 minutes (sliding window)

Initial License Checkout

Universal Callback Server licenses are checked out at startup. The maximum number of licenses that can be checked out is specified in the configuration.

Available licenses are decreased upon receiving a callback request or a request to block a license. This is incremented 60 minutes after the callback submission or the license lock.

Note: License Manager does not allow you to check out more than 9999 licenses per client; due to this restriction, one Universal Callback Server can handle no more than 9999 callback requests per hour. If the traffic of your anticipated callback requests is higher than this value, consider running several Universal Callback Servers and spreading the load between several Routing Points.

See *Voice Callback Reference Manual* for further information on licensing.

Primary/Backup Support

The license is checked out without dependency on primary/backup configuration. Primary and backup servers use the same license pool.

For any primary/backup support with Management Layer 7.0.1 and higher, the HA_Redundancy license for Management Layer is required. See *Framework Management Layer User’s Guide* for more information.

Licensing Database Dependencies

The main database table where all callback-related information is stored (this table is configured through the List option of Routing Point) has a field populated by the current time upon initial insertion of the callback record.

Universal Callback Server uses this field to enforce licensing after recovery on startup or a switch between primary and backup servers.

Processing Request to Work in Unlicensed Mode

Universal Callback Server produces a `GCTI_LICENSE_FAIL` licensing-violation log message with violation type in the following situation:

- If `vcb_preview` licenses are available but the callback is requested on a Routing Point (or Routing Queue) that is configured for `Autodial` mode.

See *Voice Callback Reference Manual* on how to configure `Autodial` mode on a Routing Point.

Migration

- Licenses are issued for Universal Callback Server. Calculate the number of licenses according to your configuration needs.
- At a minimum, upgrade your Genesys License Manager to version 8.3.

See *Genesys 7 Migration Guide*.

Configuration Dependencies

See *Voice Callback Reference Manual* for configuration dependencies and options.

Importing Application Templates

You need an application template for the Universal Callback Server before you can configure its Application object. This template provides most of the application's configuration options and default values.

The CPD Server is an optional VCB component. If you use this server for `Autodial` with the Call-Progress Detection feature, you will also need an application template for CPD Server 7.0.

To Locate an Existing Template

1. Open the `Environment` folder and then the `Application Templates` folder.
2. If the template is absent, import one from the `Templates` folder on the Voice Callback 7.1 CD.

The next section provides step-by-step instructions for importing application templates. From one application template, you can create as many Application objects of the same type as you need.

Procedure for Importing Application Templates

The following procedure is used to import an Application Template for Universal Callback Server and for the CPD Server:

To Import an Application Template

1. Open the Configuration Manager main window.
2. Select the Environment > Application Templates folder.
3. From the File menu, select the Import Application Template.The Open dialog box displays.
4. Click the down-arrow on the Look-In field in the Open dialog box.
5. In the dropdown list of folders, locate the appropriate Application Template file in the Templates folder.
6. Select the template file.

   The file name follows this format: <Application_Type>_<version>.apd.

   For example:

   - UCB_Server_710.apd (file name for Universal Callback Server template)
   - CPD_Server_70.apd (file name for CPD Server template)
7. Click Open on the Open dialog box to display the Properties dialog box for this template.
8. Click OK to accept the default settings in the database and close the Properties dialog box for this template.

Note: From one UCB_Server_710.apd file, you can create any number of Universal Callback Server Application objects.

From one CPD_Server_70.apd file, you can create any number of CPD Server Application objects.

Creating and Configuring Application Objects

You create Application objects from the application templates and configure them in the Configuration Manager. Your VCB Deployment-Planning Worksheet will help you configure these objects (see “Deployment Planning Worksheet” on page 181).

Application Objects

- The primary Application object for VCB is the Universal Callback Server.
- The CPD Server is used for Call Progress Detection in the Autodial mode.

Other Application objects required to run VCB include:

- T-Server(s), Stat Server(s), and DB Server(s), which you will have already installed and configured as part of Framework.
- Interaction Routing Designer (IRD) 7.0 and Universal Routing Server (URS) 7.0, which you create as part of Universal Routing.
- IVR that is Genesys-compliant.
  - Genesys Voice Portal (GVP) is only required if you choose to use this product as your IVR.
  - Voice Treatment Manager 7.0 and Voice Treatment Server 7.0 are required only if you choose to use the Voice Treatment Option (VTO) as your IVR.
- Agent desktop: Genesys Agent Desktop (GAD)

Note: The VCB deployment procedures in this guide assume that GAD or another agent desktop with VCB functions implemented by Genesys APIs.

- Web API Server: For configuration information, see the *Multi-Channel Routing Event Media Deployment Guide*.
- Universal Contact Server: For configuration information, see the *Multi-Channel Routing Event Media Deployment Guide*.

Note: Other objects that enable VCB functionality are discussed in Chapter 6, "Configuration of Other Objects," on page 99.

Configuring the Universal Callback Server Application Object

If you filled out the Deployment-Planning Worksheet in the Appendix on page 181, you should use it now for these Universal Callback Server configuration tasks. You may also want to consult "Voice Callback Configuration Options" in the *Voice Callback Reference Manual*.

To Configure Universal Callback Server Application Object

1. Open the `Configuration Manager` main window.
2. Select the `Environment > Applications` folder.
3. Choose `File > New > Application`. The `Browse` dialog box opens.
4. In the `Browse` dialog box, choose the template you imported for this application: `UCB_Server_710.apd`.
   ("Importing Application Templates" on page 65 provides instructions for importing a template.)
5. Click `OK`. The `Properties` dialog box opens.
6. In the `Name` field on the `General` tab of the `Properties` dialog box, change the default name to a specific Universal Callback Server name.

7. Select the `Tenants` tab of the `Properties` dialog box, and then click the `Add` button. The `Browse` dialog box opens.

Note: The `Tenants` tab only displays in a multi-tenant environment.

8. In the `Browse` dialog box, click the down arrow in the `Look in` field to find and select the appropriate tenant objects for this application.
9. Click `OK` on the Browse dialog box.
10. Return to the `Properties` dialog box and select the `Server Info` tab and specify these fields:
    - `Host` computer on which this server is to be installed and run.
      Your system administrator can provide the IP address, LCA port, and operating system information.

      To define a new host, see the section called "Network Objects" in the *Framework Deployment Guide*.
    - `Communication Port` that applications must use to connect to this server. Choose a unique four-digit number.
    - `Reconnect Timeout` (the default value is `10`).
    - `Reconnect Attempts` (the default value is `1`).
    - `Backup Server` (optional).

      If another server application is used as a backup for the Universal Callback Server:
      - Specify it in the `Backup Server` field.
      - Click the `down arrow` or the `Browse` button to find and select the backup server.
    - `Redundancy Type` (Hot Standby, Not Specified, and Warm Standby)
11. Select the `Start Info` tab on the Properties dialog box and define the following:
    - `Working Directory`.

      Enter temporary string. Installation will enter the command line later.
    - `Command Line.`

      Enter temporary string. Installation will enter the command line later.
    - `Command-Line Arguments.`

      You may leave this field empty. You can add these arguments during installation, after which you can add to them or change them.

      For more information about command-line parameters, see *Framework Deployment Guide*.
    - `Timeout`.

      Enter `Startup` and `Shutdown` parameters.
12. Select the `Connections` tab and add:
    - `T-Server(s).`

- Message Server.
- CPD Server(s).
- Stat Server (s).
- Universal Contact Server.

To add these components:

a. Click the Add button in the Connection Tab. The Connection Information Properties dialog box opens.

b. In the Connection Info Properties dialog box, click the Browse button to find and open the Applications folder.

c. Select the applications you wish to add and click OK.

13. Optional step:

To enable the Advanced Disconnect Detection Protocol (ADDP) between Universal Callback Server and T-Server added in Step 10, refer to the *Framework Configuration Manager Help*.

14. Select the Options tab on the Properties dialog box and specify or change, as needed, the values of the configuration options. The options are described in the *Voice Callback Reference Manual.*

If the Universal Callback Server application's Working Directory differs from the directory in which the application was originally installed:

a. Configure an option named messagefile in the Log section.

b. Specify the full path to the application-specific log-messages file (*.lms) as the option value.

Otherwise, the application is unable to generate its specific log events.

15. Click OK to accept your changes and to close the Properties dialog box for the Universal Callback Server Application object.

Configuring CPD Server Application Object

To Process Callback Requests

To process callback requests based on call results, VCB 7.1 requires Call Progress Detection (CPD) hardware from Dialogic. You must install switch-extending equipment (a Dialogic® board) to perform the callback function. For more information, see Chapter 8 on page 153.

Call Progress Detection

Before installing CPD Server, you should have all the information ready about the environment in which this server will run, including information about Configuration Server and about directory numbers (DNs) that correspond to Dialogic ports.

To Configure CPD Server

You must configure CPD Server before you install it. Follow these steps:

1. Open the Configuration Manager main window.
2. Select the Environment > Applications folder.
3. From the File menu, select New > Application. The Browse dialog box opens.

4. In the `Browse` dialog box, choose the template you imported earlier: `CPD_Server_70.apd`.
   (See page 65 for instructions on importing a template.)
5. On the `General` tab of the `Properties` dialog box, enter the CPD Server's name.
6. On the `Server Info` tab of the `Properties` dialog box, enter:
   - `Host` name of the computer where this application is installed.
   - `Port` that the application uses for client connections.
   - `Backup Server` (Optional).

   Enter the name of the backup server and adjust the `Reconnect Timeout` option in the following situation:
   - If another server is used as a backup in case the primary fails and if a Dialogic board is installed.
7. On the `Connections` tab of the `Properties` dialog box, add the T-Server(s) and the Message Server to which the CPD Server will connect.
8. Optional step:

   To enable the Advanced Disconnect Detection Protocol (ADDP) between CPD Server and T-Server added in Step 7, refer to the *Framework Configuration Manager Help*.
9. On the `Start Info` tab of the `Properties` dialog box, define `Working Directory` and `Command Line` properties. For example:

   ```
   Working Directory: C:\GCTI\cpd_server
   Command Line: cpdserver.exe
   Command-Line Arguments: -host cmhost -port 2030 -app CPDS1-L
   ```
10. On the `Options` tab of the `Properties` dialog box, change the values of the configuration options according to your configuration.

    See *Voice Callback Reference Manual* for CPD configuration options.
11. Click `OK` to accept your changes and to close the `Properties` dialog box for the CPD Server Application object.

Configuring Web API Server

Specific information on configuring Web API Server is found in *Multi-Channel Routing Event Media Deployment Guide*. Please note that the Universal Callback Server has to be added to the Web API Server `Connections` tab.

Configuring Universal Contact Server

Specific information on configuring Universal Contact Server is found in *Multi-Channel Routing Event Media Deployment Guide*.

Installing Voice Callback

This section describes the procedures for installing VCB on Windows and Unix platforms. These procedures include installing Universal Callback Server and CPD Server.

Installing Universal Callback Server on Windows

The following section describes the procedures for installing Universal Callback Server on Windows.

Installation Package

The installation package on the VCB 7.1 CD contains the `Setup.exe` file.

Setup File Location

The setup file for Universal Callback Server is located in this folder:

```
\solution_specific\universal_callback_server\windows
```

If you download VCB 7.1 from an FTP site, the installation package is contained in a ZIP file, which you can extract using the WinZip utility (`http://www.winzip.com`). After extracting the files, look for the `Setup.exe` file for the Universal Callback Server application in the `Windows` folder.

Installation Procedures

You must configure the Application object for Universal Callback Server before VCB installation begins.

To Configure Application Object

1. From the Voice Callback 7.1 CD, open the folder: `\\solution_specific\universal_callback_server\windows`
2. Locate and double-click `Setup.exe` to start installation.
3. Enter the `Host` name and `Port` for Configuration Server.
4. Enter the `User Name` and `Password` for access to Configuration Server.
5. Confirm the local computer’s host name for Universal Callback Server.
6. Select the Universal Callback Server application that you want to use from the list.
7. Specify the destination directory either by clicking `Next` to accept the directory displayed or by clicking `Browse` to specify another directory.
8. Enter the full path for the license file or click `Browse` to select the path.

   Now all Genesys servers are installed as services on Windows NT platform. Therefore, you will not be asked to install Universal Callback Server as a Windows NT service. See “Using Windows NT Service” on page 73.

9. Click Finish to complete the installation.

Installing CPD Server

If you want the Call Progress Detection feature available in the Autodial mode, you need to install CPD Server.

To Install CPD Server Application

1. From the product CD, open this folder:
   solution_specific\cpd_server\windows
2. Locate and double-click Setup.exe to start installation.
3. Choose the installation mode: CPD Server.
4. Enter the Host name and the Port for Configuration Server.
5. Enter the User Name and the Password for Configuration Server.
6. Confirm the local computer's host name for CPD Server.
7. Specify the destination directory. By default, CPD Server is installed in a directory called \GCTI\cpdserver.
8. Now all Genesys servers are installed as services on Windows NT platform. Therefore, you will not be asked to install CPD Server as a Windows NT Service. (See "Using Windows NT Service" on page 73.)
9. Click Finish to complete the installation.

Installing CPD Proxy

If the volume of call is particularly heavy at a contact center, CPD Proxy distributes calls among various CPD Servers for optimal load-balancing and utilization of system resources.

Note: CPD Proxy installation does not require Dialogic hardware to be installed on the target machine.

To Install CPD Proxy Application

1. From the product CD, open this folder:
   solution_specific\cpd_server\windows
2. Locate and double-click setup.exe to start installation.
3. Choose the Installation mode: CPD Proxy mode.
4. Enter the Host name and Port of Configuration Server.
5. Enter the User Name and Password of Configuration Server.
6. Confirm the local computer's host name.
7. To select the destination directory, either click Next to accept the default (displayed) setting, or click Browse to locate another directory.

   Do not install CPD Proxy in the same directory as CPD Server.

8. Now all servers are installed as services on Windows NT platform. Therefore, you will not be asked to install CPD Proxy as a Windows NT service. See "Using Windows NT Service" for more information.
9. Click `Finish` to complete the installation.

Using Windows NT Service

An application set up as a Windows NT Service starts automatically when the computer is started and runs in the background without a console window.

To Set Up Windows NT Service

1. When you set up an application as a Windows NT Service, InstallShield searches for other instances of the same application set up as a Windows NT Service. If there are no other instances of the same application, you will not be prompted further.
2. However, if InstallShield finds another instance, it asks if the existing instance should be replaced with the new instance.
   a. Click `Yes` to replace the existing instance with the new one,

      Or

   b. Click `No` to keep the existing application.

      If you click `No`**,** the existing application is left as it is, and a number is added to the new application name to distinguish it from the existing one.

To Stop or Modify the Startup Type

To stop or modify the startup type for an application that is running as a Windows NT Service:

1. From the `Control Panel,` click `Services`**.**
2. Right-click to select the application.
3. Click the appropriate `Command` button to change the application status or to stop the application.

Installing Universal Callback Server on UNIX

The installation procedures for VCB 7.1 on UNIX involve installing the Universal Callback Server.

To Install Universal Callback Server

1. On the VCB 7.1 CD, locate the following group of Universal Callback Server `Installation` files (for all Unix platforms):
   - `data.tar.gz`
     binaries of Universal Callback Server (the same name is used for all Unix platforms)
   - `installer.tar.gz`
     Perl interpreter and Genesys common installation modules (Perl scripts)
   - `iscript.tar.gz`
     Universal Callback Server-specific installation modules (Perl scripts)

- `install.sh`
  main installation script
- `gunzip`
- `gunzip_user_license_agreement.txt`
- `ip_description.xml`
  IP description
- `ospatchlist.txt`
  OS patch list
- `read_me.html`
  IP `ReadMe` file with link to Release Notes

2. Copy the Universal Callback Server `Installation` files into a temporary directory.
3. Locate the shell script, `install.sh`.

Note: All files require permission to execute.

4. Run this script by entering `./install.sh` at the command line.
5. When prompted, specify the host name of the computer on which you want to install Universal Callback Server.
6. Enter host, port, user name, and password for Configuration Server access.
7. From the list of Universal Callback Server-type Applications configured for this host, select a Universal Callback Server application to install.
8. Specify the destination directory into which you want Universal Callback Server installed. Create the directory if necessary.
9. Enter `32` or `64`, depending on your Unix system, when this prompt appears:

```
There are two versions of this product available:
32-bit and 64-bit. Please enter 32 or 64 to select which version
to use.
```

Notes: Genesys server installation programs no longer give the user the choice of installing a 32-bit or 64-bit executable when version 5.x of IBM's AIX UNIX operating system is detected on the target server machine. Starting with release 7.0, the installation programs automatically install a 32-bit executable without displaying a warning message. (This applies to the Genesys server components listed in Genesys 7 Supported Operating Systems and Databases.)

Therefore, 64-bit modules built on 4.3.2.0 AIX hosts cannot be used on 5.x AIX hosts since they are not compatible. To solve this problem, you must build 64-bit modules on 5.x AIX hosts.

10. Specify the full path for the `License` file when prompted.

As soon as the installation process is finished, a message appears announcing that the installation has been successful. The process creates a directory, with the name you specified during the installation, containing the Universal Callback Server installation.

Installing Web API Components

See *Multi-Channel Routing Event Media Deployment Guide* for specific information on installing Web API Server. MCR installation does not include any VCB components; it includes only MCR components required for VCB.

- Customers who have purchased MCR could install Voice Callback without the manual installation. They would use the wizard to install MCR, which includes the required Voice Callback components. The *Multi-Channel Routing Event Media Deployment Guide* describes the wizard that deploys the MCR software.
- The manual installation is used when installing VCB and certain MCR features. It enables you to deploy your VCB components, including those that would usually come with MCR, or allows you to integrate your VCB components with an existing MCR deployment you might have.

Note: If you do not have Multi-Channel Routing (MCR) components installed prior to VCB deployment, Web API Server and Universal Contact Server Application Objects have to be created from corresponding templates prior to their configuration, and templates have to be imported before that.

For VCB, the manual installation steps for the Web API components are as follows:

To Install Web API Server Component Manually

1. Create `Application` object for your Web API Server using Configuration Manager.

Note: In setting up the object in the configuration layer, set the values outlined in chapter 4 of the *Multi-Channel Routing Event Media Deployment Guide.*For specific information on installing Web API Server, consult this chapter as well.

2. From the CD, open the appropriate directory.
3. Locate and double-click on `setup.exe` and follow the instructions.

Installing Universal Contact Server

There are no specific instructions for installing Universal Contact Server for VCB.

See *Multi-Channel Routing Event Media Deployment Guide* for specific information on installing Universal Contact Server.

Installing VCB Silently

This section describes how to use InstallShield Silent for installing Voice Callback silently, that is, for performing installations that do not require the end-user to enter any settings. This section includes the following topics:

Silent Setup Overview

InstallShield Silent allows for an automated electronic software distribution, also known as a *silent setup.* InstallShield Silent only works on Windows operating systems. With InstallShield Silent, you do not have to monitor the setup or provide input via dialog boxes. Once all input is stored in a *response file,* an InstallShield Silent setup runs on its own, without any end-user interventions.

After creating your response file and performing a silent installation, you review the log file produced for a successful result code or any errors encountered.

Note: All Local Control Agent (LCA)–related functionality is disabled during a silent installation. This means that you will not be able to start and stop applications, detect their failures, or communicate their roles in the redundancy context.

Creating the Response File

The following section shows how to select setup options and automatically record the InstallShield Silent response files.

Note: Use this procedure instead of double-clicking `setup.exe` from Windows Explorer.

To Create the Response File

1. Open a console window.
2. Enter the path to the directory in which you deployed the VCB installation files.
3. Issue the following command using the `-r` command-line parameter:

   ```
   Setup.exe -r
   ```

   Issuing this command creates a response file named `setup.iss`, which is stored in the main Windows directory (`%System Root%`). For example, if you are using Windows NT, this is the `c:\winnt directory`.
4. Configure the `setup.iss` file to include the desired parameters.

Use the configured `setup.iss` file any time you must install an application with the specified parameters.

Sample Response File

The following is a sample `Setup.iss` file for VCB:

Dialog boxes count, name and order

```
[{E4C99638-8E4C-40DF-BD62-7DFA832FBE9D}-DlgOrder]
Dlg0={E4C99638-8E4C-40DF-BD62-7DFA832FBE9D}-gctiGdWelcome-0
Count=7
Dlg1={E4C99638-8E4C-40DF-BD62-7DFA832FBE9D}-gctiGdCfgSrvPrms-0
Dlg2={E4C99638-8E4C-40DF-BD62-7DFA832FBE9D}-gctiGdAppList-0
Dlg3={E4C99638-8E4C-40DF-BD62-7DFA832FBE9D}-gctiGdLicensePrms-0
Dlg4={E4C99638-8E4C-40DF-BD62-7DFA832FBE9D}-gctiGdAskPath-0
Dlg5={E4C99638-8E4C-40DF-BD62-7DFA832FBE9D}-gctiGdReadyToInstall-0
Dlg6={E4C99638-8E4C-40DF-BD62-7DFA832FBE9D}-gctiGdFinish-0
```

Welcome dialog box

```
[{E4C99638-8E4C-40DF-BD62-7DFA832FBE9D}-gctiGdWelcome-0]
Direction=1
```

Config server connection parameters

```
[{E4C99638-8E4C-40DF-BD62-7DFA832FBE9D}-gctiGdCfgSrvPrms-0]
Direction=1
Host=altair
Port=7100
User=default
Password=password
```

Application name in config server

```
[{E4C99638-8E4C-40DF-BD62-7DFA832FBE9D}-gctiGdAppList-0]
Direction=1
AppName=VCB
```

Parameters for license location (license manager server or license file)

```
[{E4C99638-8E4C-40DF-BD62-7DFA832FBE9D}-gctiGdLicensePrms-0]
```

```
Direction=1
AccessType=1
Host=rleonid6
Port=7280
File=' '
```

Location where VCB server will be installed

```
[{E4C99638-8E4C-40DF-BD62-7DFA832FBE9D}-gctiGdAskPath-0]
Direction=1
Path=C:\GCTI\UCallbackSrv
```

Confirmation dialog box that all ready for installation

```
[{E4C99638-8E4C-40DF-BD62-7DFA832FBE9D}-gctiGdReadyToInstall-0]
Direction=1
```

Option "No Reboot" selected

```
[{E4C99638-8E4C-40DF-BD62-7DFA832FBE9D}-gctiGdFinish-0]
Reboot=0
```

Play Back the Response File

To Install VCB Silently on a Local Workstation:

1. Open a console window.
2. Enter the path to the directory where you have deployed VCB installation files.
3. Launch InstallShield Silent using the following command:
   `Setup.exe -s /f1"ResponsePath" /f2"LogPath"`

   where:

   - `-s` is the silent flag.
   - `/f1"ResponsePath"` is an optional command-line parameter that specifies the full path to the response file. If you do not specify this parameter, InstallShield assigns the default file name of `setup.iss` and looks for the file in the directory containing `Setup.exe`.
   - `/f2"LogPath"` is an optional command-line parameter that specifies the full path to the log file. If you do not specify this parameter, InstallShield assigns the default file name of `setup.log` and stores the file in the directory containing `Setup.exe`.

Note: Do not use spaces between the `/f1` or `/f2` parameter and its value in double quotation marks.

The following illustrates some commands you could issue to play back the response file:

- `setup.exe -s`

  Both `/f1"*\[`*`Setup.exe directory`*`]\setup.iss"` and `/f2"*\[`*`setup.exe directory`*`]\setup.log"` are implied on a Windows NT system.

- `setup.exe -s /f1"c:\winnt\MonitorAgents.iss"`

  `/f2"*\[`*`Setup.exe directory`*`]\setup.log"` is implied on a Windows NT system.
- `setup.exe -s /f2"c:\winnt\SuzysLog.log"`

  `/f1"*\[`*`Setup.exe directory`*`]\setup.iss"` is implied on a Windows NT system.

Remote Silent Installation

Using standard operating system tools, your network administrator can design the scripts necessary to deploy the response file to any number of machines where VCB can be deployed remotely.

About the Silent Setup Log File

When the response file has completed its playback, InstallShield Silent prints installation results to the file specified by `/f2,` as shown in the previous section. This log file contains three sections:

Log File Sections

- The `[InstallShield Silent]` section identifies the version of InstallShield Silent used in the silent install and the file type as a log file.
- The `[ResponseResult]` section contains the result code indicating whether the silent install succeeded. See Table 3 for the meanings of the `ResultCode` key given in this section.

Table 3: Log File ResultCode Key

Value	Meaning
0	Success
-1	General error
-2	Invalid mode
-3	Required data not found in the `setup.iss` file
-4	Not enough memory
-5	File does not exist
-6	Cannot write to the response file
-7	Unable to write to the uninstallation log file
-8	Invalid path to the InstallShield Silent response file
-9	Not a valid list type (string or number)

Table 3: Log File ResultCode Key (Continued)

Value	Meaning
-10	Data type is invalid
-11	Unknown error during setup
-12	Dialog boxes are out of order
-51	Cannot create the specified folder
-52	Cannot access the specified file or folder
-53	Invalid option selected

The following text shows the contents of a log file for a VCB application that was successfully installed using InstallShield Silent.

```
[InstallShield Silent]
[Application]
[ResponseResult]
ResultCode=0
```

For more information on Install Shield and its silent install mode, refer to Install Shield documentation available at: http://support.installshield.com/.

Chapter

5 Other Deployment Considerations

This chapter contains deployment considerations that were not covered in Chapter 4. It has the following sections:

Automatic Call Distribution (ACD)

You can use an Automatic Call Distribution Queue to blend callbacks and live calls with a switch-based (ACD Queue and non-Genesys controlled) routing of callbacks to agents. That is, you do not need to use only Genesys-controlled routing (agent-level URS routing) in your environment.

For example, although you must use ACD Queue to choose the appropriate agent to receive a call, you can use the Genesys routing engine to park calls. See Figure 12 on page 82 for a diagram of how call distribution works.

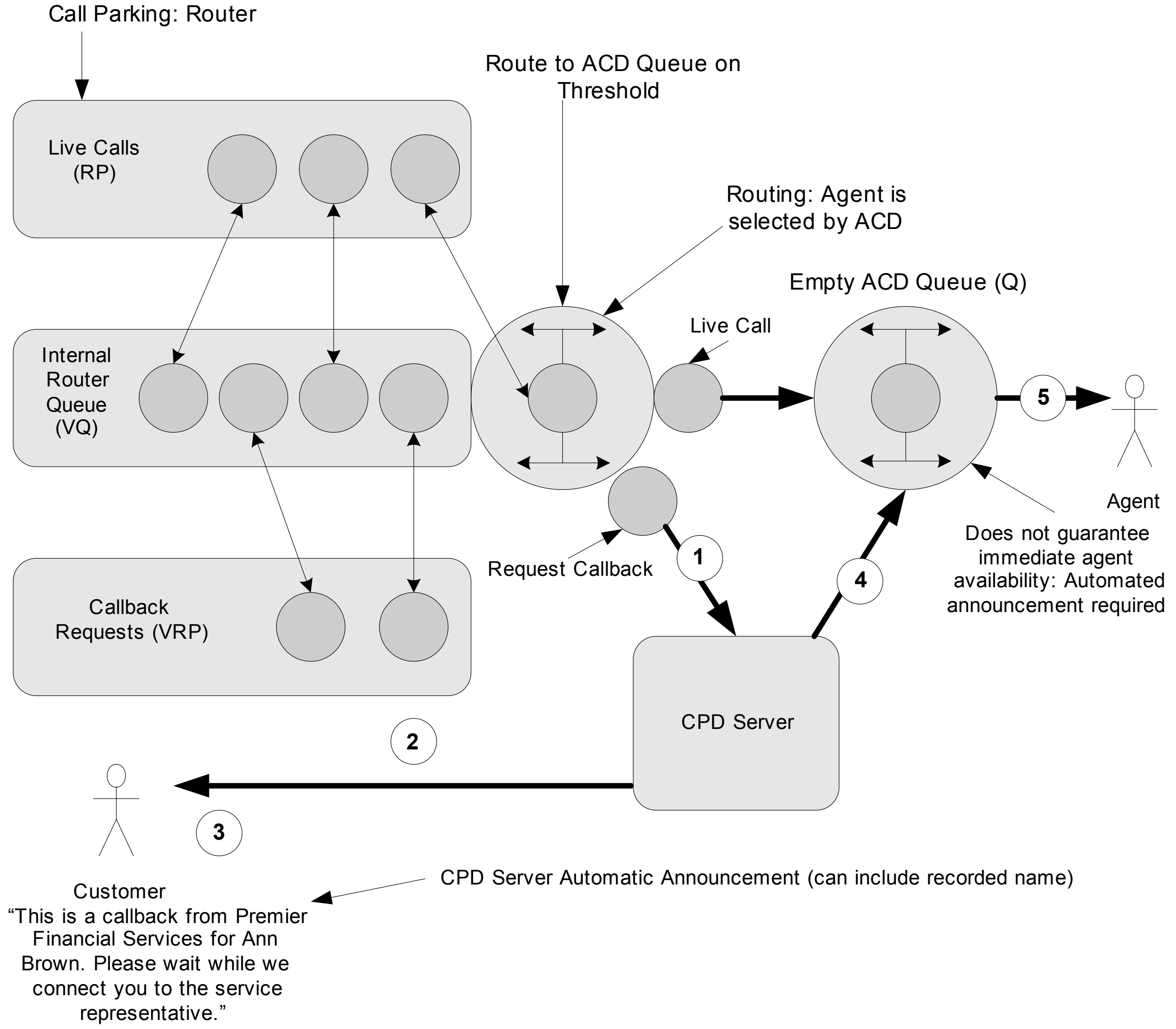

Call Parking: Router
Live Calls (RP)
Internal Router Queue (VQ)
Callback Requests (VRP)
Route to ACD Queue on Threshold
Routing: Agent is selected by ACD
Empty ACD Queue (Q)
Live Call
Request Callback
1
2
3
4
5
Agent
Does not guarantee immediate agent availability: Automated announcement required
CPD Server
Customer
"This is a callback from Premier Financial Services for Ann Brown. Please wait while we connect you to the service representative."
CPD Server Automatic Announcement (can include recorded name)

Figure 12: Automatic Call Distribution

This figure depicts these steps:

1. A request is made to dial a call as a result of callback-request processing.
2. The call is dialed.
3. An announcement is played to the party receiving the call.
4. CPD Server transfers the call to an ACD Queue.
5. The ACD Queue distributes the call to an agent.

Note: The same ACD Queue also distributes live call to the same agents.

ACD Routing of Callbacks

As Figure 12 shows, you can use an ACD Queue to distribute callbacks. In this scenario, the Universal Routing Server (URS) routes the callback to an ACD Queue and not to an agent.

Routing Principles

The Routing strategy uses the following principles:

- Chosen `Statistics` and `min/max` values define `Busy` state of target (ACD Queue).
- Router allows the use of `Statistics` value only with `ChangesBased` notification mode and `insensitivity` option set to `0`.
- Router distributes the call to an ACD Queue when the `Statistics` value reaches the value defined as a `Threshold for Target`.
- Router checks the target to determine its availability:
  - **a.** By using a 2-second timer.
  - **b.** Upon receiving `Statistics` value with `ChangesBased` notification event from Stat Server.
  - **c.** When a new call arrives to Routing Point.

    In this case, if the target is not in a `Busy` state, the new call may be distributed immediately even if some calls are already waiting on the Routing Point. To avoid this disruption in the queue, the router enforces FIFO distribution from RP (function `KeepQueue`).

Note: UCS "engaging" `Autodial` mode (see "Agent-Reserved Connection" on page 89) also works in the scenario described in Figure 12 on page 82.
When UCS uses "dial-ahead" `Autodial` mode (see"Customer-Reserved Connection Announcements" on page 94), a significant (~10 seconds) time delay occurs between the call distribution by the router (start of UCS outbound dialing) and the appearance of the answered call in the ACD Queue. As a result, the statistics on ACD Queue are not valid, which could restrict potential deployments.

Deployment for "Dial-Ahead" Autodial Mode

Deployment with ACD Queue-based call distribution and "dial-ahead" `Autodial` mode has the following requirements:

- Router uses the same Virtual Queue (VQ) for the distribution of inbound calls and virtual callback requests.
- Router uses the same ACD Queue to distribute inbound calls and outbound callbacks.
- Router parks calls on Routing Point, and uses ACD Queue only for call distribution (very low waiting time).
- CPD Server plays announcement to the customer for every outbound call that UCS dials (see "Customer-Reserved Connection Announcements" on page 94). This is done to avoid the abandonment of a call, which may

occur because of the time that ACD Queue requires for call distribution. (That time is calculated as the router's minimal average time of call distribution equal to average conversation length divided by number of logged-in agents).

Note: Implementation requires a thorough understanding of the principles for designing routing strategies and call flow.

ACD Queue Call-Parking

This scenario supports a blended queue of callback requests and live calls when both live calls and internal "engaging" calls (that is, callback requests) are parked (queued) in the ACD Queue. Figure 13 depicts ACD Queue call-parking flow.

Note: The Router is not required for:
Callback request queuing (blended with inbound calls)
Callback distribution to agent

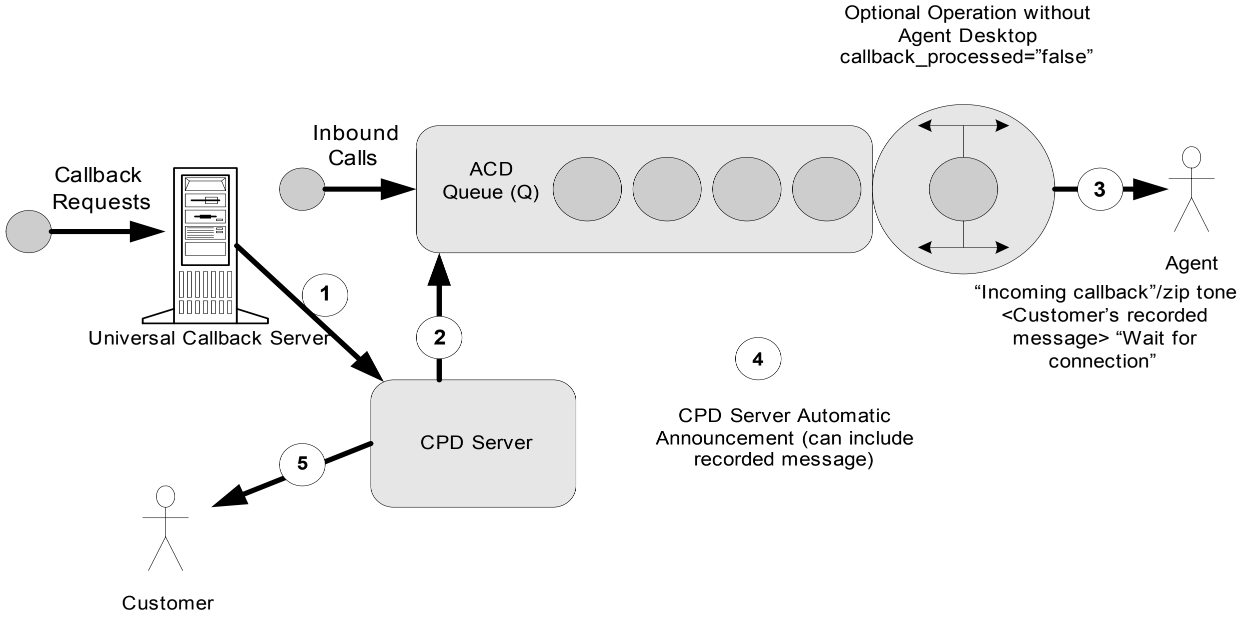

Optional Operation without Agent Desktop callback_processed="false"
Inbound Calls
Callback Requests
ACD Queue (Q)
3
Agent
"Incoming callback"/zip tone <Customer's recorded message> "Wait for connection"
1
Universal Callback Server
2
4
CPD Server Automatic Announcement (can include recorded message)
CPD Server
5
Customer

Figure 13: ACD Queue Call Parking

Diagram Explanation

This figure depicts these steps:

1. A request is made to dial a call as a result of callback-request processing.
2. CPD Server transfers "engaging" call to an ACD Queue.
3. ACD Queue distributes "engaging" call to an agent.

4. Announcement (or zip tone) is played to an agent.
5. CPD Server connects the agent with the customer.

In this scenario, Genesys-controlled call parking on the Routing Point is not required because the Automatic Call Distribution mechanism can be used instead.

“Engaging” Autodial Mode

This is implemented as a specific case of “engaging” `Autodial` mode. UCS uses the scenario described in “Agent-Reserved Connection” on page 89 in the following situations:

- If UCS is configured to work in an “engaging” `Autodial` mode.
- If UCS receives a target DN of type `ACD Queue` in the `RequestRouteCall` from the Routing Server.
- If Routing Server is not present, the default routing mechanism is used, and the default DN configured for the Virtual Routing Point is of type `ACD Queue`.

CPD Server places an internal call from the Dialogic port to the specified ACD Queue instead of the agent’s DN. This internal “engaging” call (which represents a callback request) waits in the queue along with other live calls for an available agent. ACD Queue then distributes the call to this agent. As soon as this internal “engaging” call is established on the agent’s DN, the CPD Server dials the customer who requested the callback and then connects these two calls.

Note: This implementation requires a lot of hardware resources (Dialogic ports) because a separate Dialogic port is unavailable for the duration of the “engaging” call; that is, while this call waits in the queue.

Default Callback Routing

Universal Callback Server can route callback requests to a default DN if URS is down or not available. You should configure this default DN for the Virtual Routing Point (VRP). If UCS does not receive routing instructions from the router within the specified timeout (`call_default_routing_timeout`), UCS then routes the callback to the default DN. It behaves as if it gets a response from the router with a default DN destination.

Default callback routing processes callback requests if the router (or other applications responsible for call distribution on the VRP) is unavailable.

Default Routing Processes

1. After Universal Callback Server distributes `EventRouteRequest` on the VRP, it sets the timer for the interval specified in the option `call_default_routing_timeout` (if the option is defined and has a non-zero value).

2. Upon timeout expiration, Universal Callback Server acts as if it received `RequestRouteCall` upon normal callback distribution. The DN required for further callback processing has to be defined as `DefaultDN` in VRP configuration object. If more than one `DefaultDN` is configured for VRP, Universal Callback Server uses the first one in the list. If no `DefaultDN` is configured, Universal Callback Server does not perform the default routing. It finalizes the callback request with an error and prints an error message to the log.

See the *Voice Callback Reference Manual* for more information about the `call_default_routing_timeout` option.

Operation Without Agent Desktop

UCS can function without an agent desktop for `Autodial` modes, although you lose some screen-pop enhancements as a result.

A special option is used if UCS expects notifications from the agent's desktop. See *Voice Callback Reference Manual* for more information about the `callback_processed` option.

Deployment for Different Dialing Modes

URS makes a routing decision for a virtual call, which represents a callback request. For `Autodial` mode, URS should not necessarily provide DN (position for extension) as a Target for routing. URS may route the callback to any other DN of type Routing Point or ACD Queue.

Figure 14 shows how VCB delivers a callback to a selected DN.

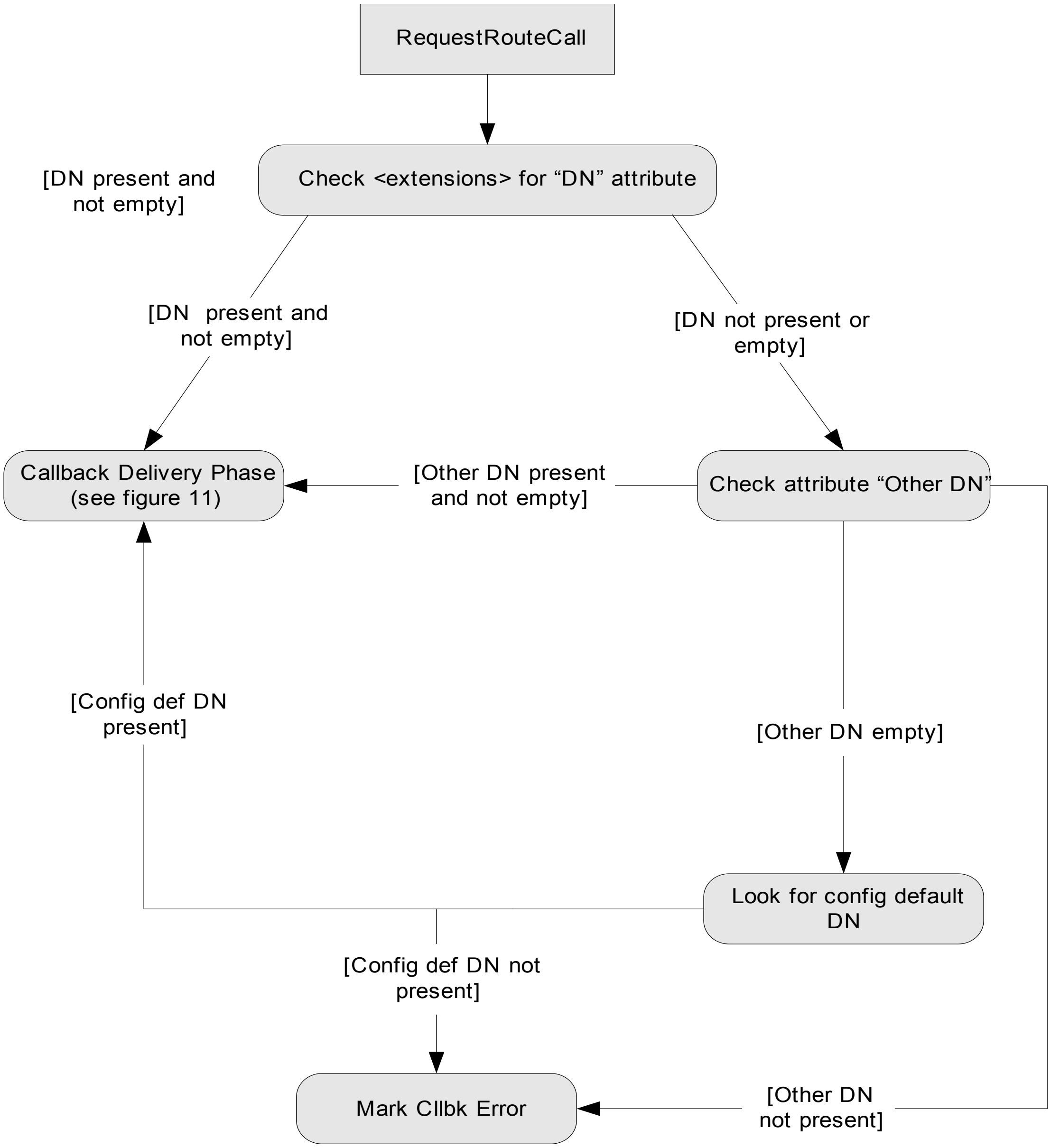

RequestRouteCall
Check <extensions> for “DN” attribute
[DN present and not empty]
[DN present and not empty]
[DN not present or empty]
Callback Delivery Phase (see figure 11)
[Other DN present and not empty]
Check attribute “Other DN”
[Config def DN present]
[Other DN empty]
Look for config default DN
[Config def DN not present]
Mark Cllbk Error
[Other DN not present]

Figure 14: Routing Callbacks: Callback Delivery Phase (1)

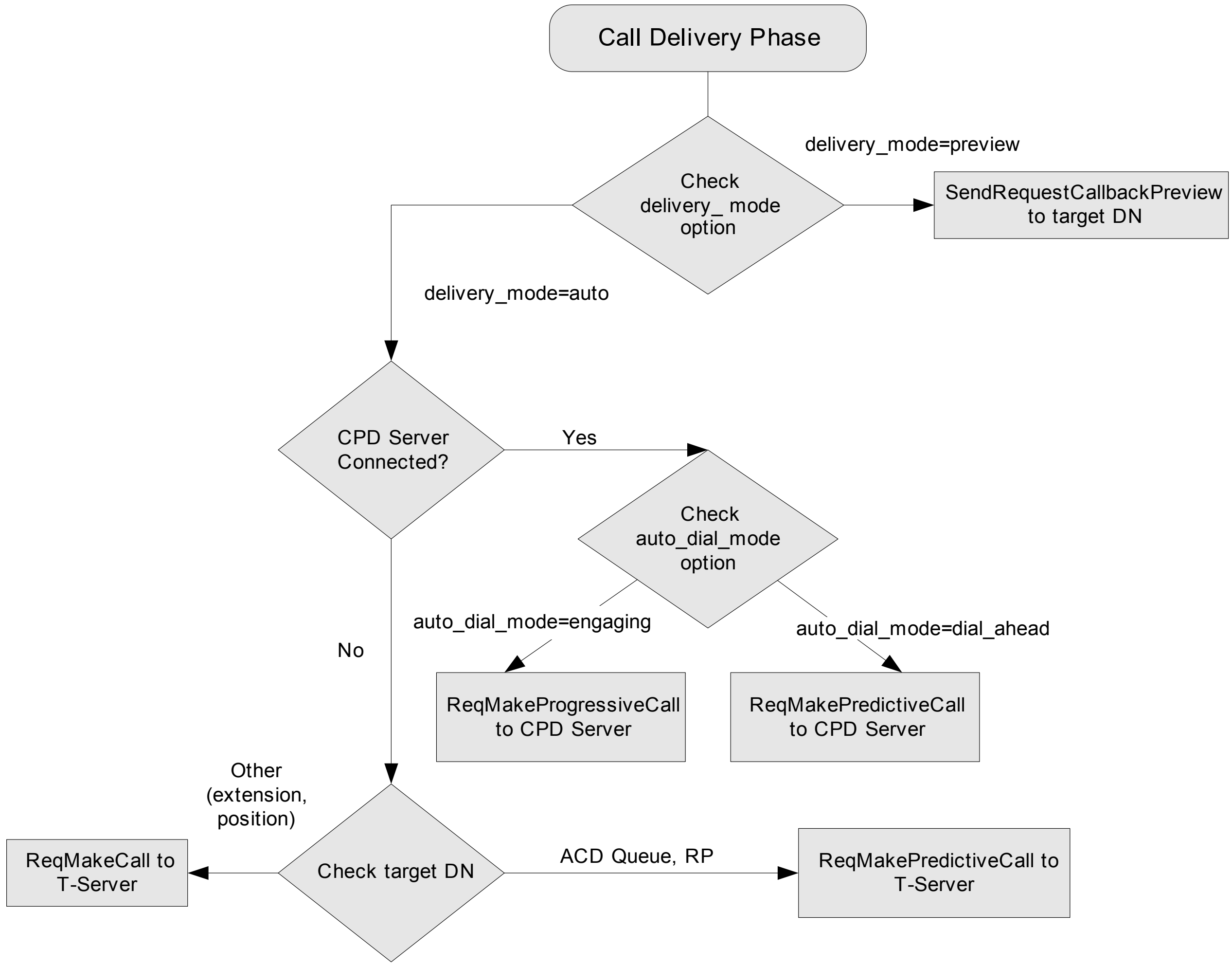

Call Delivery Phase
Check delivery_ mode option
delivery_mode=preview
SendRequestCallbackPreview to target DN
delivery_mode=auto
CPD Server Connected?
Yes
Check auto_dial_mode option
auto_dial_mode=engaging
auto_dial_mode=dial_ahead
ReqMakeProgressiveCall to CPD Server
ReqMakePredictiveCall to CPD Server
No
Other (extension, position)
Check target DN
ACD Queue, RP
ReqMakeCall to T-Server
ReqMakePredictiveCall to T-Server

Figure 15: Routing Callbacks: Callback Delivery Phase (2)

Autodialing Options

Autodial options include Agent-Reserved Connection and Customer-Reserved Connection.

Agent-Reserved Connection

This autodialing solution requires CPD Server. First, CPD Server "engages" the agent by an internal call that contains all User Data for the corresponding callback request and plays an announcement or a beep tone (if configured) to the agent. Then CPD Server dials the customer and connects these two calls when the customer answers. This ensures that the agent is available for processing the callback and also allows the agent time to prepare for a conversation with the customer.

This solution enables ACD Queue callback parking (the ability to use an internal call as a callback placeholder). ACD callback routing (ability to use ACD Queue to distribute a callback request to an agent) can be used for this mode as well as agent-level routing. See Figure 16 for Agent-Reserved-Connection flow.

Note: Currently, autodialing depends on URS's timeout, which is set in the transition_time option. If this timeout expires while CPD is in operation, the agent may receive another call from URS or receive another call from a different source regardless of this timeout.

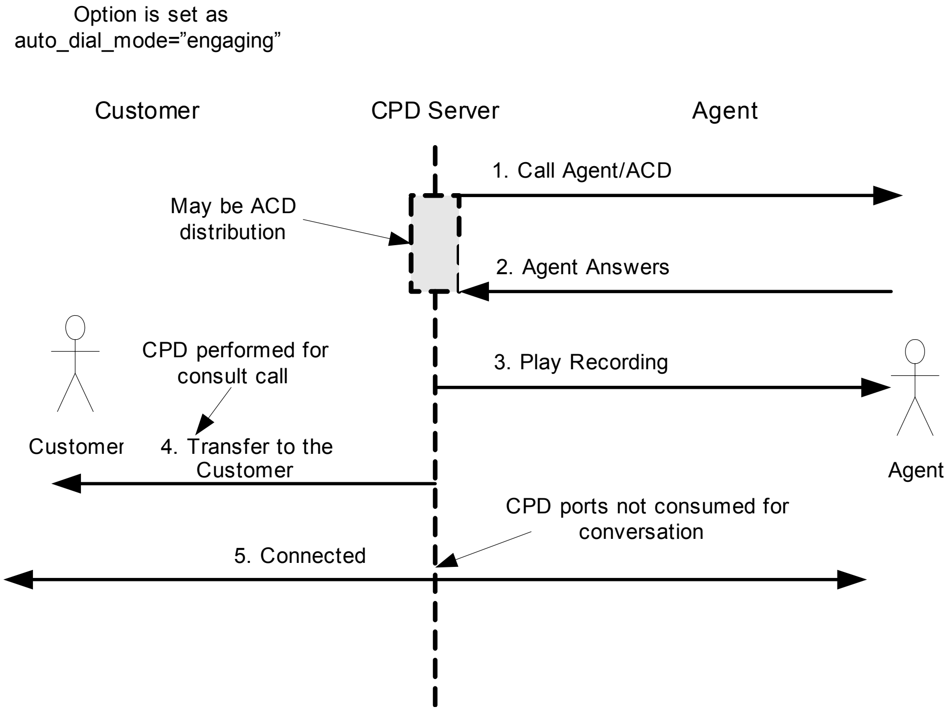


Figure 16: Agent-Reserved Connection

Diagram Explanation

Figure 16 depicts these steps:

1. CPD Server places call (internal) to the selected DN.
2. The agent answers the call.
3. CPD Server plays the customer's recorded message to the agent.
4. Dialer makes a two-step transfer to callback number.

Note: This feature may also be used as a solution for ACD-level routing support for a switch-based operation. When a callback request is routed to ACD Queue, this internal call serves as a placeholder in the ACD Queue until it is delivered to an agent.

Scenario Description

- UCS receives the destination DN and location from the Interaction Router.
- UCS determines the T-Server assigned to this location and the CPD Server servicing that location.
- UCS checks port availability on designated CPD Server.
- UCS makes a request to initiate an `Autodial` call to the CPD Server.
- CPD Server makes an internal call from the port to the agent's DN.
- After the agent answers this call, the CPD Server makes a two-step transfer of the established call to the external destination with CPD on consult call.
- In case of failure, CPD Server reports it to UCS. UCS then applies a treatment to callback request.
- Callback-request data is delivered to agent's desktop with an "engaging call."

Figure 17 on page 91 further illustrates this Agent-Reserved Connection scenario.

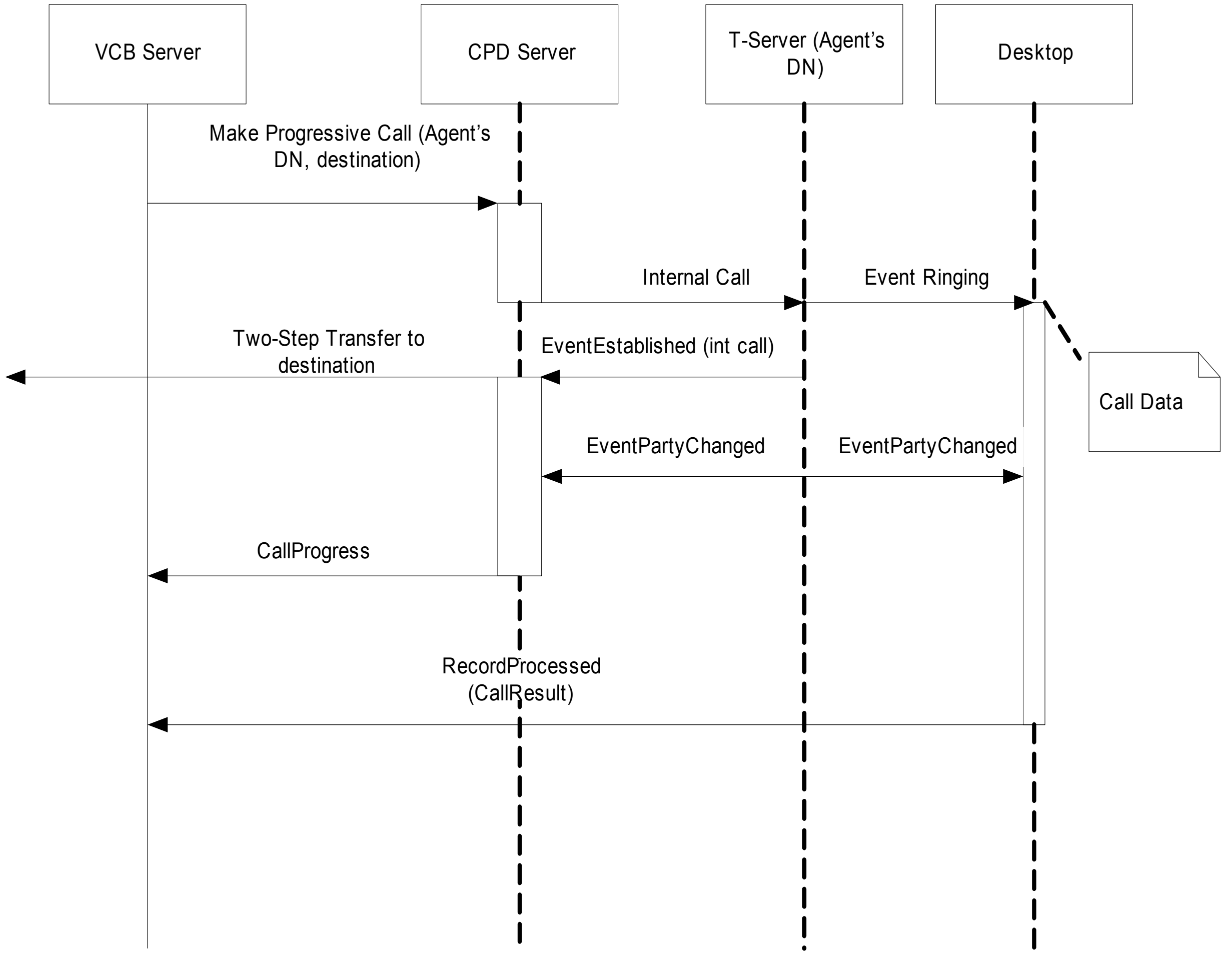

VCB Server
CPD Server
T-Server (Agent's DN)
Desktop
Make Progressive Call (Agent's DN, destination)
Internal Call
Event Ringing
Two-Step Transfer to destination
EventEstablished (int call)
Call Data
EventPartyChanged
EventPartyChanged
CallProgress
RecordProcessed (CallResult)

Figure 17: Call Flow for Agent-Reserved Connection

See *Voice Callback Reference Manual* for more information about the `auto_dial_mode` option.

Customer-Reserved Connection

This autodialing solution offers CPD Server voice-announcement capability and ACD routing capability (ability to use ACD Queue for distributing a callback request to an agent).

This solution automates callback-distribution tasks (dialing callback number, playing announcement to the customer, and reaching the right agent). See Figure 18 for the flow of a Customer-Reserved Connection.

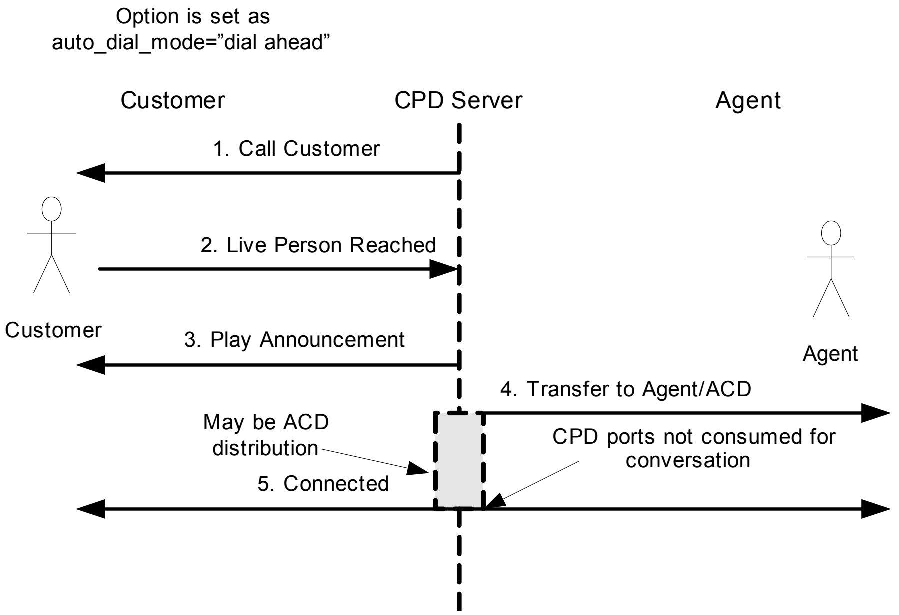
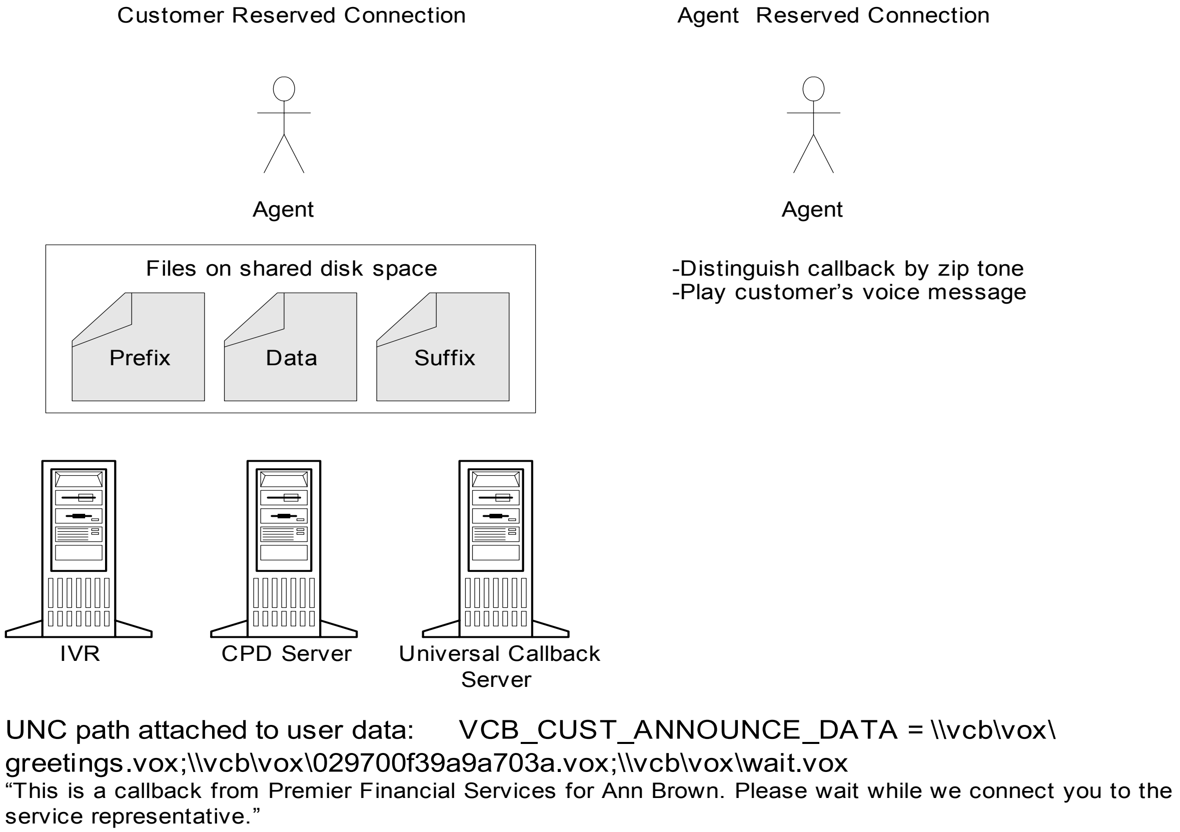


Figure 18: Customer-Reserved Connection

CPD Server Announcements

Figure 19 depicts the flow of for CPD Server announcements.


Customer Reserved Connection
Agent Reserved Connection
Agent
Agent
Files on shared disk space
Prefix
Data
Suffix
-Distinguish callback by zip tone
-Play customer's voice message
IVR
CPD Server
Universal Callback Server

UNC path attached to user data: VCB_CUST_ANNOUNCE_DATA = \\vcb\vox\greetings.vox;\\vcb\vox\029700f39a9a703a.vox;\\vcb\vox\wait.vox

"This is a callback from Premier Financial Services for Ann Brown. Please wait while we connect you to the service representative."

Figure 19: CPD Server Announcements

Announcements for Agent-Reserved Connection

This feature involves playing a predefined, configured announcement (or beep) to the agent after the "engaging" call has been made on the agent's DN. The agent is then aware that this is a callback request independent of the screen-pop (see "Agent-Reserved Connection" on page 89).

Announcement Features

- Announcement can be a beep tone or sequence of prerecorded voices.
- Universal Callback Server forms the sequence of prerecorded voices as a prefix-file full name, data-file(s) full name, and suffix-file full name, delimited by ";".
- Universal Callback Server provides the type of announcement and the full path to the voice files in case of pre-recorded announcements are provided to the CPD Server.

- If CPD Server fails to open any of the files which should be played to an agent, it prints an error message to its log and plays files that were successfully accessed (if any).

Universal Callback Server allows you to configure different announcements on any Routing Point or Application level. See Chapter 2 in the *Voice Callback 7.1 Reference Manual* for more information about the following options: `agent_announcement_type`, `agent_announcement_prefix`, `agent_announcement_data`, and `agent_announcement_suffix`.

When a callback is delivered to the agent in `Autodial` mode, a prerecorded announcement could be played to the agent after the call has been established on the agent's DN. It could be announced to the agent by voice (for example, with the recorded customer's name). The same announcement may be used for all callbacks, or the announcement can be recorded by IVR during callback submission.

Implementation on the Universal Callback Server-CPD Server side does not differ much from the beep-tone announcement. The difference is that the voice file, specified in the `agent_announcement_data` option, is recorded by IVR; the path to this file is specified in a dedicated `UserData` attribute, `VCB_RECORDED_DATA`, so that Universal Callback Server is able to extract it upon a callback submission process. This attribute is present in `UserData` attached to the call or in the Client-Server protocol request `RequestCallbackAdd`.

Universal Callback Server deletes the unnecessary voice files after completing the callback process. However, this may cause performance degradation, so this functionality may be disabled by the Universal Callback Server option `delete_ivr_announce_file`.

See *Voice Callback Reference Manual* for more information about the `delete_ivr_announce_file` option.

Customer-Reserved Connection Announcements

You can only play voice announcements to the customer.

A prerecorded voice announcement could be played to the customer after the call has been answered (See "Customer-Reserved Connection" on page 92). The same announcement may be used for all callbacks, or each callback can have its own announcement that can be recorded by IVR during callback submission.

This option allows a dialout callback (placing outbound call to customer) before the agent is available; the live customer is provided with an automated voice announcement. For example, `This is your return call from Genesys. Please wait while being transferred to an attendant.`

Announcement Features

- The predefined announcement is played to the customer after the live answer has been detected and the call has been established. This announcement is a sequence of prerecorded voices.

- Universal Callback Server forms the sequence of pre-recorded voices as prefix-file(s) full name, data-file full name, and suffix-file full name, delimited by "`;`".
- If CPD Server fails to open any of the files which should be played to the customer, it prints an error message to its log and plays files that were successfully accessed (if any).

Universal Callback Server allows you to configure different announcements on any Routing Point or Application level. See Chapter 2 in the *Voice Callback Reference Manual* for more information on the options for Announced-Reserved Customer Connection.

Black-List Blocking

Universal Callback Server can be configured to block phone numbers specified in a black list table in the database. If such phone numbers are used as a customer's contact number during callback-request submission, an error is registered.

Configuration

The table access point that contains access data for the `Black List` table is configured for Environment Tenant or for each Tenant. The table has a fixed structure. The name of the table access point is `vcb_black_list`. See *Voice Callback Reference Manual* for more information.

UCS can reread the contents of the Black List table. After new data has been read from the database, UCS starts blocking the numbers that correspond to the wildcards just being read. Time interval between rereads is controlled by the `dnc_refresh_interval` option.

Note: If table access point `vcb_black_list` is not configured at all, UCS does not check the `Black List` table upon callback submission.

Search Condition

The entry in the Black List table may be an exact phone number or search pattern.

The search pattern can include these valid wildcard characters in Table 4:

Table 4: Wildcard Characters

Wildcard Character	Description	Example
%	Any string of zero or more characters.	900%
_ (underscore)	Any single character.	___1234567 1234567 phone number within any area code.

Use the following SQL statement to make entries into the Black List table:

```
INSERT INTO <table name> VALUES ('<entry>')
```

See Table 5 for the structure of the Black List table:

Table 5: Black List Table Structure

Field Name	Type	Nullable
phone	varchar (64)	Yes

UCS performs the following actions:

- UCS checks for the presence of the Black List table in the database and creates it if it does not exist.
- UCS retrieves all records from the Black List database table on startup and on every refresh.
- UCS checks the phone number in callback during the callback-request submission phase. If the phone number matches a number or pattern in the Black List table, UCS responds with a specific error message (see Table 6):

Table 6: UCS Operations with “Black List” Table

VCB_ERROR	VCB_ERROR_CODE	VCB_ERROR_NAME (Description)
Number blocked	110	Phone Number blocked for dialing

Time Interval for Callback Processing

Time interval limitations for callback processing are set up when ordering the callback. The expiration time for the callback order can be specified for any ordering interface (IVR, Web, Agent Desktop) and is supplied in the form of start time (`VCB_DATE_TIME`) and end time (`VCB_END_DATE_TIME`) attributes. Time interval limitations take priority over any configured treatments and also ensure that the callback is not dialed outside the stipulated time window. If the specified time expires and the callback is not delivered to the caller, the callback processing is stopped, and the callback is finalized in the database with `callback_status = VcbEndTimeExpired (14)`.

The expiration time is configured either per Universal Callback Server Application or per Routing Point (option `max_proc_interval`) and applies to callback requests for which the expiration time is not specified in the callback submission request. `VCB_END_DATE_TIME` attribute (if provided) always takes precedence over `max_proc_interval` option setting. The default value for this option is `0,` which means that no expiration is enforced.

Support for Switch CPD

This capability allows UCS to utilize the Call Progress Detection (CPD) capabilities of PBX (for PBX equipped with CPD hardware) with no CPD Server required. See Figure 20 for a diagram of this environment.

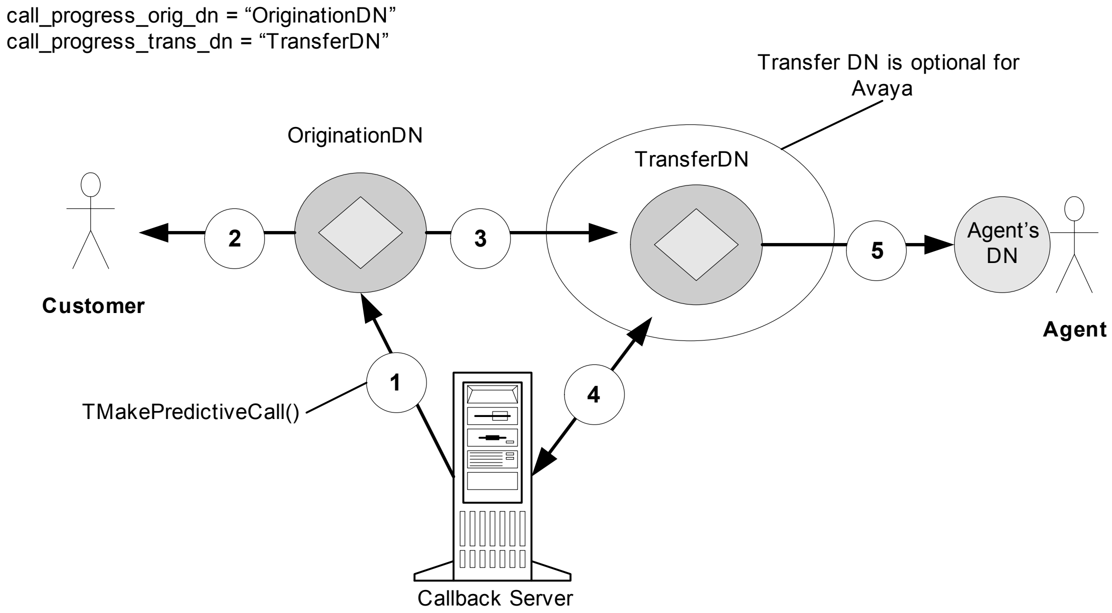

call_progress_orig_dn = “OriginationDN”
call_progress_trans_dn = “TransferDN”
Transfer DN is optional for Avaya
OriginationDN
TransferDN
Customer
2
3
5
Agent’s DN
Agent
1
4
TMakePredictiveCall()
Callback Server

Figure 20: Support for Switch CPD

Diagram Explanation

The figure depicts these steps:

1. UCS requests PBX through T-Server to make predictive call.
2. PBX makes predictive call and analyzes call result.
3. PBX transfers calls with positive CPD to `Transfer DN`.
4. UCS receives notification that a call has been dialed and (optionally) routes the call to Routing Point.
5. UCS distributes the call to an agent for processing.

Chapter

6 Configuration of Other Objects

This chapter provides instructions for the configuration of other database objects that enable VCB functionality. In addition to configuring the primary VCB applications and components, which were presented on Chapter 4, "Deployment Procedures," on page 61, you must configure other related objects in the Configuration Database to enable VCB functionality. You must configure all objects before starting VCB.

The chapter contains the following sections:

How to Use This Chapter

Before using Configuration Manager to configure objects:

1. Read *Framework Configuration Manager Help* for information on adding, deleting, and configuring objects, as well as for descriptions of elements common to Configuration Manager dialog boxes such as the `State-Enabled` check box.

   Note: Genesys strongly recommends setting the `State Enabled` check box to `on` for all objects that VCB uses.

2. For assistance with the step-by-step procedures in this chapter, refer to the Sample Deployment-Planning Worksheet you completed in “Deployment Planning Worksheet” on page 181. See also the *Voice Callback Reference Manual* for detailed information about the configurable items in this chapter.
3. See *Framework Getting Started Guide* for information about configuring Switches, Hosts, Database Access Points, Table Access, Calling Lists, Treatments, DNs, Access Groups, and Persons.
4. Configure objects in the sequence used in this chapter. (See next section.)

Recommended Object-Configuration Sequence

Because of certain dependencies, Genesys recommends the following sequence for the creation and configuration of the objects listed.

For example, you should create the:

- Table Access object before the Calling List object that will be applied to it.
- Calling List object before the treatment(s) that will be applied to the calling list.

Sequence for Configuration

1. Database Access Point(s)
2. Table Access object for Calling List
3. Table Access object for `Black List` table
4. Calling List
5. Treatments
6. Routing Points
7. Virtual Routing Points

8. DNs for call termination
9. Virtual Queues

Initial Steps in Configuration Procedure

The objects discussed in this chapter are all configured manually in the Configuration Manager.

The initial steps in the configuration procedures are the same for most of the objects being configured.

To Configure Objects

1. Open the `Configuration Manager`.
2. Select `Configuration > <tenant_name> <object name>` folder.
3. Create a new object for configuration using one of these methods:
   - From the `File` menu, select `New > <object name>`.

   or

   - Right-click the `<object_name>` folder on the left pane and select `New <object_name>` from the shortcut menu that opens.

   or

   - Select the `<object name>` directory in the right pane. Right-click and select `New > <object_name>` from the shortcut menu that opens.

   The `Properties` dialog box for the specific object opens.

Annex Tab in Properties Dialog Box

Some configurations are performed on the `Annex` tab of the `Properties` dialog box for each configurable object in the Configuration Manager. By default, the `Annex` tab is *not* shown in these `Properties` dialog boxes.

To Display the Annex Tab

1. Select the `Options` command from the `View` menu in the `Configuration Manager` main window.
2. Select the `Show Annex tab in the object properties` check box.

Database Access Point

The Database Access Point (DAP) configuration object shows the DB Server the access point to a particular database. VCB requires a DAP for the `Calling List` table and for the `Black List` table.

Calling List Table in VCB and Outbound

The `Calling List` table in VCB is not to be confused with a `Calling List` table containing dialing records, which Outbound Contact uses for its campaigns. The `Calling List` table in VCB uses a similar format as an `Outbound Calling List` table. However, the primary purpose of this table in

VCB is to provide persistent (backup) storage of callback data in the event that the system goes down and the data needs to be recovered. The VCB data might also be used for future calls.

Importing DAP Template

The DAP is not an application with its own executable (.exe) file; however, it does have a template from which to generate a number of DAP objects. The template for the DAP object comes with the Management Layer of Framework and should already be listed in the directory structure in the Configuration Manager.

If, for some reason, you do need to import the DAP template before creating and configuring specific DAP objects, follow the instructions in "Importing Application Templates" on page 65. However, modify *Step 6* as follows:

- Select the template file for the Database Access Point.
  The file name follows this format:

```
<Application_Type>_<version>.apd
Database_Access_Point_70.apd
```

Configuring DAP Object

Configure the DAP object for the VCB Calling List table in Configuration Manager.

For assistance with this section, refer the relevant section of the "Deployment Planning Worksheet" on page 181.

To Configure the Database Access Point object

1. Open the Configuration Manager main window.
2. Select the Environment > Applications folder.
3. From the File menu, select New > Application. The Browse dialog box opens.
4. In the Browse dialog box, select <template_name> and click OK. The template name pertains to Database_Access_Point_700.

   The Database Access Properties dialog box for the DAP object opens.
5. On the General tab of the Database Access Properties dialog box:
   a. Change the name in the Name field.
   b. In the Template field, click Browse to open the Browse dialog box and select the DB Server name.

Note: The name of the DAP template that you previously imported appears automatically in the Template field.

6. On the Tenants tab of the Database Access Properties dialog box, add Tenant objects if this application is operating in a multi-tenant environment.

Note: The Tenants tab only displays in a multi-tenant environment.

   a. Select the Tenants tab on the Database Access Properties dialog box and click Add. The Browse dialog box displays.
   b. In the Browse dialog box, click the down arrow to find and select the appropriate tenants for this application. (See the relevant section of "Deployment Planning Worksheet" on page 181.)

7. On the DB Info tab of the Database Access Properties dialog box, enter or select the following information:
   a. Enter the DBMS name.
      - The name of database management system is the same as the host name where your database resides.
   b. Select the DBMS type: MS SQL, Oracle, Sybase, DB2, or Informix.
   c. Enter Database Name.
   d. Enter User Name.
   e. Enter Password.

Note: The fields on the Server Info tab of the Database Access Properties dialog box are automatically populated when you select DB Server field in the General tab. The field values cannot be changed after selecting the DB Server.

8. Click OK to close the Database Access Properties dialog box for the Database Access Point.

Table Access Object

Table Access objects define the relationship between calling lists, formats, and the database. These objects describe database tables of a specified format and explain how these tables are accessed through DAPs. A Table Access object must be configured for each table in the database.

For assistance with the following step-by-step procedures, refer to:

- "Table Access Object" section in the *Voice Callback Reference Manual*
- "Deployment Planning Worksheet" on page 181.

To Create a New Table Access Object

1. In the Configuration Manager, select Configuration > <tenant_name> > Table Access folder.
2. Do one of the following:
   - Select New from the File menu.

or

- Right-click the Table Access directory in the left pane and select New.

The Table Access Properties dialog box opens.

3. On the General tab of the Table Access Properties dialog box:

a. Enter a new name in the Name field.
or
Select a name from the drop-down list for the Name field.

You must specify a value. The default value is Table Access. The name can be any number of characters, but it must be unique within the Tenant object.

Note: See the relevant section in "Deployment Planning Worksheet" on page 181.

b. Select Calling List in the Table Type field.

You must specify a value. You should always select Calling List. Once Calling List is specified, it cannot be changed.

c. Enter a brief description of the Table Access object.

The description is optional, and the default value is [Blank].

d. Select DB Access Point.

Universal Callback Server requires a DAP to access the Calling List table for VCB. There is no default value. Use the Browse button to locate an existing object.

e. The Format field must have a value in it; however, any format is acceptable. The default value is [None]. Use the Browse button to locate an existing value.

Note: VCB does not use the format specified in the Table Access object. Instead, it uses a predefined format for the persistent storage table for callback data. This format is fixed and cannot be changed.

f. In the Database Table field, enter a name for the table that Universal Callback Server creates on startup.

You must specify a value. The default value is [Blank].

The table provides persistent storage of VCB data for backup purposes and for use in future callbacks.

Note: You cannot delete a Table Access object as long as it is associated with at least one calling list.

Calling List Object

In VCB, a Calling List table provides persistent storage of callback data and other system data. This table serves as a backup database in the event of a system failure that necessitates subsequent retrieval of the data.

The Calling List table in VCB differs from a Calling List table in Outbound Contact:

- Outbound Contact *reads* data from a Calling List table.
- Voice Callback *writes* data to a persistent storage table.

The VCB Calling List table is automatically generated when the main VCB component, the Universal Callback Server, is installed. The format and the fields in this table are predefined.

- The user *does* need to define the Table Access object for the Calling List table; therefore, the configuration of the Table Access object must precede the configuration of the Calling List object.
- Genesys recommends defining calling lists according to the sequence shown in "Recommended Object-Configuration Sequence" on page 100.

For assistance with the following step-by-step procedures, refer to "Deployment Planning Worksheet" on page 181.

Creating and Configuring a Calling List Object

To Create a New Calling List Object

1. In Configuration Manager, open Configuration> <tenant_name>.
2. Right-click the Calling Lists directory. From the shortcut menu, select New, and then Calling List.

   or

   From the File menu Select New> Calling List.

   The Calling List Properties dialog box opens.

To Configure the Calling List Object

3. Define these fields on the General tab in the Calling List Properties dialog box:

   a. Name.

   This value is required, and the name of the object must be unique within the Tenant object. The default value is Calling List.

   Note: Genesys recommends that you give this list a name that identifies it as persistent (backup) data storage for VCB and distinguishes it from a regular Calling List object (as used by Outbound Contact). For example: VCB_storage.

   b. Table Access.

Table Access object to which the Calling List object refers. The Table Access object defines the format of the Calling List object and identifies the DB Server application used to access the corresponding table. Use the `Browse` button to locate an existing value.

This value is required. There is no default value.

c. `Maximum Attempts`.

The value defines the maximum number of attempts to call. This value is required. The default value is `10`. This value cannot be set to `0`.
It is not necessary to define the following fields on the `General` tab:

`Tenant`	Automatically populated by the system.
`Description`	Optional. Default is [Blank].
`Log Table Access`	No default value. Not used at this time.
`Filter`	No default. Not applicable to VCB `Calling List` table.
`Calling Time From`	Not applicable to VCB.
`Calling Time To`	Not applicable to VCB.
`Script`	Not used at this time.
`State-Enabled`	Required. A check box indicator that the customer interaction can be directed to this target. By default, this check box is selected.

Follow Steps 5, 6, and 7 if the treatments are already defined. If the treatments are not defined, see the Note below.

4. Click the `Treatments` tab on the `Calling List Properties` dialog box and then the `Add` button on that tab.

The `Browse` dialog box opens.

5. Define these fields:

`Look In`	Use the down arrow to locate and select a folder that contains treatments. When the folder name appears in the `Look In` field, the treatment names are listed in the `Browse` window.
`Object Name`	Select the appropriate treatment(s) from the list in the `Browse` dialog box.
`Object of Type`	Retain default value (`Treatment`) `for this` field.

6. Click `OK`.

Note: If the treatments are not defined yet, proceed to the next section, or see "Treatment Object" on page 107, and create the treatment(s) as instructed.
An administrator can also create a new Treatment object from `Calling Lists > Calling List Properties` dialog box > `Treatments` tab. See "Creating a Treatment in Calling List Object" on page 110.

Universal Callback Server and Databases

Universal Callback Server uses two tables (`main` and `auxiliary`) to store information about callback-request processing.

Storage Tables

- The name of the `main` table is configured in the Table Access Point of the Calling List object specified in the `List` option.
- The name of the `auxiliary` table is derived from the name of the `main` table.

Data for each callback request that Universal Callback Server processes is inserted into these tables. Information in these tables is updated throughout each callback request's life cycle. Note that Universal Callback Server has a capability of deleting the processed `Callback Requests` data from this table. This behavior is controlled by the application-level option `purge_records`. See *Voice Callback Reference Manual* for details on this option.

Final Callback Statuses

The following callback statuses are final for Universal Callback Server; that is, Universal Callback Server will not retrieve callbacks with these statuses in the database:

- `Cancelled by agent`
- `Final`
- `Error`
- `Replaced`
- `End Time Expired`

Please refer to the *Voice Callback Reference Manual* for the enumerations related to callback record statuses.

Treatment Object

Treatments tell the Universal Callback Server how to respond to unsuccessful call results—such as `Busy` or `No Answer`—for callback requests. A treatment object, for example, could specify that Universal Callback Server should redial a phone number that returns a `Busy` call result the first time the number is dialed. If no treatments are applied, the callback status is changed to `Final`, and that callback is not attempted again.

Treatment Sequences

You can define treatment sequences (sometimes called linked treatments) for the same call result by using the `Number in sequence` field in the `Treatment Properties` dialog box.

- The first treatment in the sequence must have a value of `1` in the `Number in sequence` field.

- For each subsequent treatment in the sequence, the `Number in sequence` field must be one number higher than the preceding one.

Note: The sequence will not work if the treatment sequences are broken, for example, if they are numbered `1, 2,` and `4.`

- It is possible to create two `Busy` treatments and specify the order (sequence) in which to apply them.

Example

1. Create one `Busy` treatment that has an action of `Redial` and the `Number in sequence` field set to `1`.
2. Create a second `Busy` treatment that has an action of `Retry in` and the `Number in sequence` field set to `2`.
3. Assign both `Busy` treatments to the same callback result.

Treatment Object Configuration

Treatment objects are usually created and configured in the Configuration Manager:

Configuration Location

`Configuration> <tenant_name> >Treatments` folder > `Treatment Properties` dialog box.

Note: The `Treatment Properties` dialog box, where treatment objects are created and configured, is also accessible from the `Calling List` folder in the Configuration Manager:
`Configuration <tenant_name> Calling List` folder > `Calling List Properties` dialog box > `Treatment tab >Browse` dialog box (`Add New Treatment` button) > `Treatment Properties` dialog box.

Changes to existing treatments take effect the next time the treatment is applied to a call result.

Creating Treatments

For assistance with the following step-by-step procedures, see the following:

- "Treatment Object" section in the *Voice Callback Reference Manual*.
- "Treatments" section of "Deployment Planning Worksheet" on page 181.

Creating a Treatment Object

To Create a New Treatment Object

1. In the Configuration Manager, select one of the following methods:
   a. Select `Configuration > <tenant_name> > Treatments` folder.
   b. From the `File` menu, select `New > Treatment.`

   or

a. Select Configuration > <tenant_name> > Treatments folder.

b. Right-click the Treatments folder on the left pane.

c. Select New > Treatment from the shortcut menu that opens.

or

a. Select Configuration > <tenant_name>.

b. Select the Treatment directory in the right pane.

c. Right-click and select New > Treatment from the shortcut menu that opens.

The Treatment Properties dialog box opens.

2. Define these fields on the General tab of the Treatment Properties dialog box.

Name Enter a name for the new treatment.

Tenant Search and select <tenant_name> if it is not already displayed in the Tenant field.

Description (Optional).

Call Result Click the down arrow for this field and select a call result from the drop-down list.

- See “Timing Properties Used with Apply to Record Actions” section in Chapter 6 of the *Voice Callback 7.1 Reference Manual* for assistance with the remaining fields in Step 2.

Apply to Record Click the down arrow for this field and select a treatment from the drop-down list. All treatments per record are available for selection.

Apply to Call Not applicable to VCB.

Destination DN Not applicable to VCB.

Number in Sequence Click the up or down arrow to select the sequence number for this treatment. Refer to “Treatment Sequences” on page 107 for more information.

Interval (minutes) Click the up or down arrow to select the time interval (in minutes) between applications of this treatment. The availability of this field depends on the treatment specified in the Apply to Record field.

`Increment` (minutes) Click the up or down arrow to select the amount of time (in minutes) to add to each interval between applications of this treatment.

For example, if the first `Interval` is `1 minute`, and the value in the `Increment` field is also `1 minute`, then the interval between the first and second applications of this treatment will be `one minute`, but the interval between the second and third applications of this treatment will be `2 minutes`. One minute will be added to each interval if the `Increment` value is `1 minute`.

The availability of this field depends on the treatment specified in the `Apply to Record` field.

`Cycle Attempt` Click the up or down arrow to select the number of times to repeat this treatment in the cycle of treatments. The availability of this field depends on the treatment in the `Apply to Record` field.

`Date` Click the down arrow to open the calendar and select a date for a rescheduled call. This field is available if the treatment in the `Apply to Record` field is `Reschedule`.

`Time` Click the up or down arrow to specify a time for a rescheduled call. This field is available if the treatment in the `Apply to Record` field is `Reschedule`.

3. Click `OK` if you only have one action to assign.

   or

   Click `Apply` if you want to create a treatment sequence. See “Treatment Sequences” on page 107 for more information.

Note: An administrator can also create a new Treatment object from `Calling Lists > Calling List Properties` dialog box >`Treatments` tab. See “Creating a Treatment in Calling List Object” on page 110 section.

Creating a Treatment in Calling List Object

To Create a Treatment

1. Right-click `Calling Lists` folder.
2. Select `New > Calling List` from the shortcut menu that opens.

   The `Calling List Properties` dialog box displays.
3. Click the `Treatment` tab and then the `Add` button on that tab.

   The `Browse` dialog box opens.
4. In the `Browse` dialog box, click the `Create New Object` icon.

   The `Treatment Properties` dialog box opens.
5. Define the fields on the `General` tab of the `Treatment Properties` dialog box. Refer to Step 2 in “Creating a Treatment Object” on page 108.

Applying a Treatment

The administrator may select a predefined Treatment object to apply to a calling list.

To Select a Predefined Treatment Object

1. In Configuration Manager, select `Configuration > <tenant_name> > Calling List` folder.
2. Right-click the `Calling List` folder and select `Properties` from the shortcut menu that opens. The `Calling List Properties` dialog box opens.
3. Click the `Treatments` tab in the `Calling List Properties` dialog box.
4. Click `Add`.
5. Locate and select the treatment you wish to apply to the calling list.
6. Click `OK`.

Treatment Properties

Understanding timing properties is essential when creating treatments and applying them to Calling List objects. For information about the timing properties that are required for each treatment action, see the following:

- "Timing Properties Used with Apply-to-Record Actions" section in Chapter 6 of the *Voice Callback Reference Manual.*
- "Treatment Properties" section in the *Voice Callback Reference Manual.*

Rules for Call Results

Three call-result values have special treatment rules:

- `Answer`
- `Answering Machine Detected`
- `Fax Detected`.

A complete list of call results with their descriptions is in the *Voice Callback Reference Manual*.

Treatments are normally applied to a record for unsuccessful calls.

Rules for Answer

Answer by a "live" person is a successful call result and usually does not require a treatment; however, under certain circumstances the call result `Answer` does require a treatment.

Rules for Answering Machine Detected or Fax Detected

This section explains the rules for configuring Universal Callback Server to apply a call treatment. These rules apply when a call result is `Answering Machine Detected` or `Fax Detected`.

For a call that is answered, but the call result is `Answering Machine Detected` or `Fax Detected`, the system administrator may assign the `Apply to Call` treatment and specify the `Destination DN` to which the call will be transferred. For example, the `Destination DN` might be an ACD Queue for Interactive Voice Response (IVR), such as a recorded message to be left on an answering machine. The administrator specifies the `Destination DN` in the Configuration Manager.

Adding a Destination DN

To Add a Destination DN for a Call Transfer

Add all required IVR DNs and their logins in the Configuration Manager.

1. Select `Configuration > <tenant_name> <Switches> <specific switch> DNs` folder.
2. Right-click the `DNs` folder and select `New > DN.` The `DN Properties` dialog box opens.
   a. On the `General` tab, select the `Type` field.
   b. From the drop down list, add the ACD Queue that the IVRs log in to.
3. Create a Place for each IVR DN and a shortcut to that DN.
   a. Right-click on the `Place` folder and select `New > Place`. The `Place Properties` dialog box opens.
   b. On the `General` tab, enter the place name in the `Name` field.
   c. Click `Apply.`
   d. Right-click on the new place name folder in the left frame and select `New > Shortcut to DN`. The `Browse` dialog box opens.
   e. Select the DN and click `OK`.
4. Create a Person for each IVR DN.
   a. Right click on the `Person` folder, and select `New > Person`. The `Person` dialog box opens.
   b. On the `General` tab, enter the name of the person.
   c. Select the `Is Agent` check box to configure the person as an agent.
   d. Click OK.

Note: In Step 4, the “person” is actually an IVR that emulates a person; that is, the IVR does the work of an agent. In Steps 5 and 6, the “person” (IVR) is configured as an “agent” because a `LoginID` is required for access to a DN.

5. Create a shortcut to the LoginID for that IVR Person.
   - a. On the Agent Info tab of the Person Properties dialog box, click Add ID. The Agent Login Properties dialog box opens.
   - b. Click Browse to select login. The Browse dialog box opens.
   - c. Double-click on Switches > Agent Logins.
   - d. Select agent login.
   - e. Click OK.
6. Select an entry under Agent Groups or Place Groups folder.
   - a. Right-click and select New > Shortcut to Agent. The Browse dialog box opens.
   - b. Add an Origination DN shortcut to the ACD Queue that your IVR DNs log in to.
7. Right-click Treatments folder and select New > Treatment.The Treatments dialog box opens.
   - a. On the General tab of the Treatments dialog box, create an Answering Machine treatment by creating a treatment name
   - b. Use the following parameters:

- Call Result: Answering Machine Detected
- Apply to Record: No Treatment
- Apply to Call: Transfer or Connect
- Destination DN: ACD Queue that the IVR DNs log in to

Note: Transfer and Connect are synonymous in this context. Both transfer the call to the ACD Queue for the IVR DNs.

8. Add this treatment to the calling list you are using.

Apply to Record: No Treatment

If the administrator chooses No treatment, he or she will follow the same steps as in the preceding section, “Adding a Destination DN” on page 112.

DNs for Predictive Call Origination

If delivery_mode is set to auto, Universal Callback Server can issue RequestMakePredictiveCall when it receives RequestRouteCall on a Virtual Routing Point from Universal Routing Server.

- Option call_progress_orig_dn (values true, false) is configured under Universal Callback Server section of a DN that is intended to be used as a Call Progress Origination DN.

- Option `call_progress_trans_dn` (value `<DN Name>`) under the same section shows which DN should be used as Call Progress Transfer DN in `RequestMakePredictiveCall`.

See Figure 21 for a flowchart for predictive call origination.

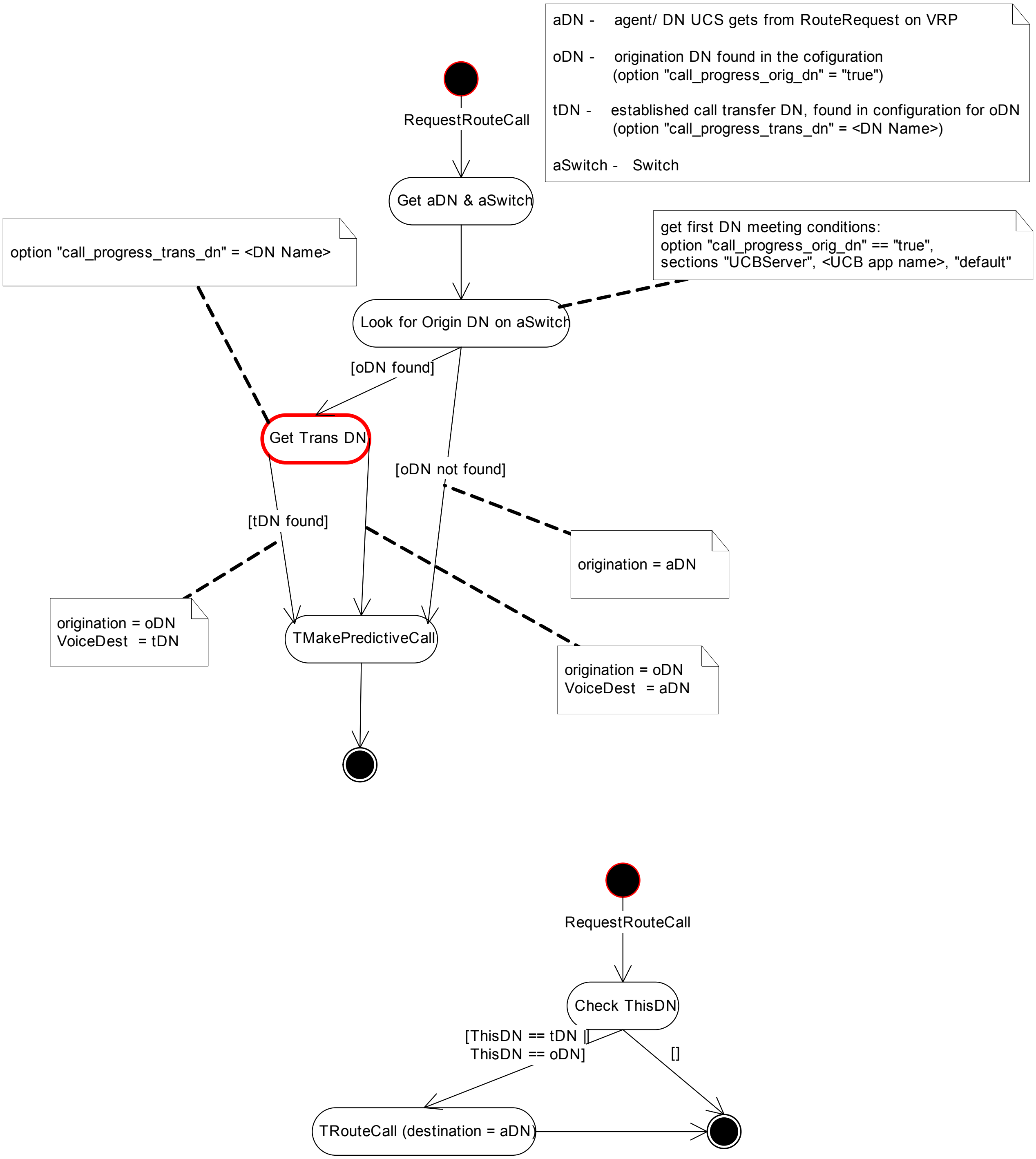

aDN - agent/ DN UCS gets from RouteRequest on VRP
oDN - origination DN found in the cofiguration (option "call_progress_orig_dn" = "true")
tDN - established call transfer DN, found in configuration for oDN (option "call_progress_trans_dn" = <DN Name>)
aSwitch - Switch
RequestRouteCall
Get aDN & aSwitch
option "call_progress_trans_dn" = <DN Name>
get first DN meeting conditions:
option "call_progress_orig_dn" == "true",
sections "UCBServer", <UCB app name>, "default"
Look for Origin DN on aSwitch
[oDN found]
Get Trans DN
[oDN not found]
[tDN found]
origination = aDN
origination = oDN
VoiceDest = tDN
TMakePredictiveCall
origination = oDN
VoiceDest = aDN
RequestRouteCall
Check ThisDN
[ThisDN == tDN ||
ThisDN == oDN]
[]
TRouteCall (destination = aDN)

Figure 21: Flowchart of Predictive Call Origination

Routing Point Objects

VCB mainly works with DNs of type `Routing Point(s)` or `Routing Queue(s)`. The relationship between Routing Point DN objects and the Universal Callback Server is created in the Configuration Manager.

For assistance with the following step-by-step procedures, refer to the following:

- *Voice Callback Reference Manual*:
  - "DN-Level Options"

    You configure the options in this section at the DN level only.
  - "Universal Callback Server Application-Level and DN-Level Options"

    You can configure the options in this section at the Application level or at the DN level.
- DNs section of "Deployment Planning Worksheet" on page 181.

To Configure the Relationship Between Routing Point and Universal Callback Server

1. In Configuration Manager, select `Configuration > <tenant_name> Switches> <specific switch> DNs` folder.
2. Select a specific DN object of `Routing Point` type on the right-hand pane.
3. Right-click the DN object and select `Properties` from the shortcut menu.

   The `Properties` dialog box displays.
4. In the `Annex` tab of the `Properties` dialog box, use the down arrow to access the drop-down menu and select the `UCBServer` folder from the list.

   If the `UCBServer` folder does not exist, click the `Create New Section/Option` icon to create the `UCBServer` folder or a folder named after a Universal Callback Server specific application.

   Note: If the section folder is named `UCBServer`, any Universal Callback Server application can use the Routing Point associated with the DN. The `UCBServer` folder must be spelled exactly like this: `UCBServer`.
   If the section folder bears the name of a particular Universal Callback Server application, only that Universal Callback Server application can use the Routing Point.

5. Double-click the `UCBServer` folder to make visible the list of options.
6. Verify or change the values of the options as needed.

   The options are listed in the "DN-Level Options" and "Voice Callback Options" sections in the *Voice Callback Reference Manual*.

   To change the value of an option

   a. Right-click the option name in the `Annex` tab.

   b. Select `Properties` from the shortcut menu that opens.

The `Edit Option` dialog box displays. You may change the value in the `Option Value` field. The `Option Name` field remains the same.

Note: If you have not yet created the Virtual Routing Point associated with this Routing Point, you will need to assign a value in the `vrp` field after you have created the VRP.

See also “Voice Callback Options” section in the *Voice Callback Reference Manual* for information about options that can be configured at either the Application level or the DN level.

7. Click `OK` in the `Edit Option` dialog box after you verify or change the value of each option.
8. Click `OK` in the `Properties` dialog box.

Virtual Routing Point Objects

VCB uses Virtual Routing Point (VRP) objects to submit callback requests. The Universal Callback Server and the URS communicate through VRPs as well as through physical Routing Points.

The VRPs are created and configured in Configuration Manager: `Configuration Manager ><Tenant> > Switches > [specific switch] > DNs` folder.

Note: For exact information on how to create Virtual Routing Point(s), see the *Framework Deployment Guide.*

The relationship between a Virtual Routing Point and the Routing Point to which it is assigned is defined when you configure the Routing Point. Refer to Step 5 in “Routing Point Objects” on page 115. Define the `vrp` option. For more information about this option, see the “DN-Level Options” section in *Voice Callback Reference Manual.*

DNs for Call Termination

VCB may use a designated DN object for call termination. In this configuration, the DN (as defined by the value of the `term_dn` option) serves as either:

- Termination point for calls that are dropped—for example, after a callback request has been submitted. (In this case, the termination DN would be of type `Position` or `Extension`.)

  or

- Distribution point from which calls are sent to other DNs to be dropped. (In this case, the termination DN would be of type `ACD Queue` or `Routing Point`.)

To Set the Option term_dn in Configuration Manager

1. Open `Configuration > <tenant_name> > Switches > <specific switch> > DNs` folder.
2. Set the `term_dn` on the `Annex` tab of the `Properties` dialog box (`UCB Server` folder or `<application name>` folder) for a DN of type `Routing Point`.

Scheduled Callback Request Processing

Estimated Waiting Time (EWT) statistics for `Scheduled` callback requests specifies the parameters required to obtain EWT for the queue involved in the distribution of `Scheduled` callbacks. The queue for the call (DN object, usually Virtual Queue) is determined when the callback is ordered.

Universal Callback Server takes into account EWT for the resource-waiting queue. This means that the `Scheduled` callback is submitted to the router in `time = scheduled time - EWT`. See Figure 22 for EWT functionality.

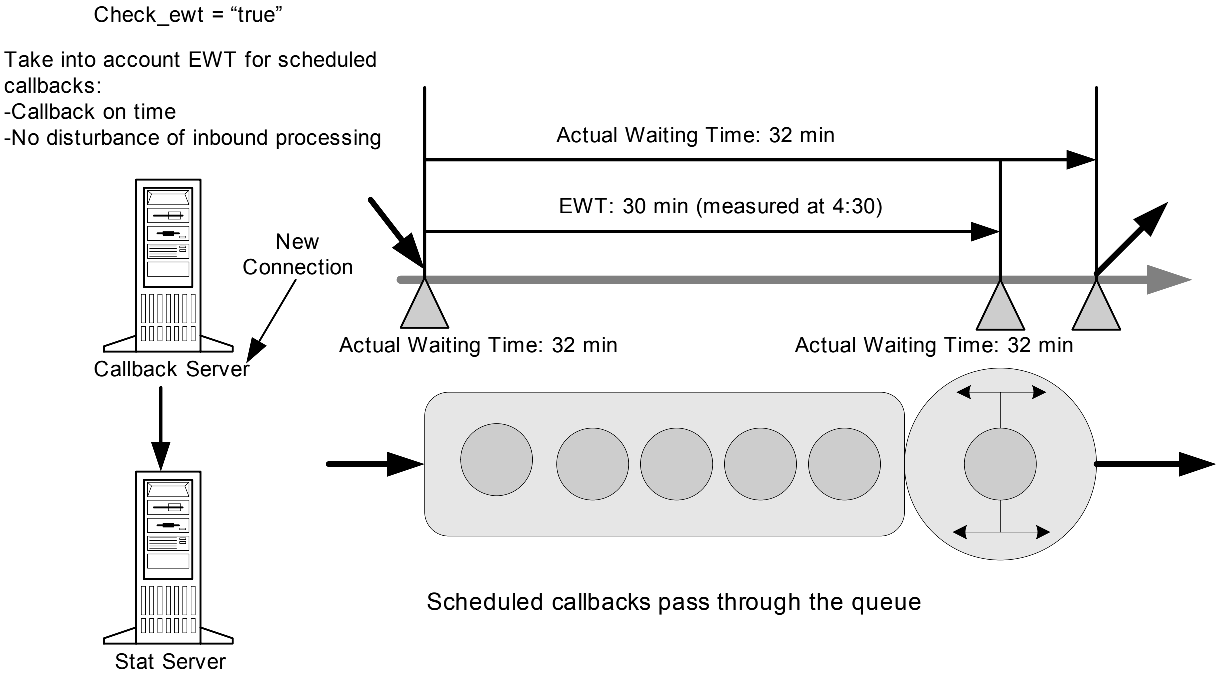

Check_ewt = "true"
Take into account EWT for scheduled callbacks:
-Callback on time
-No disturbance of inbound processing
Actual Waiting Time: 32 min
EWT: 30 min (measured at 4:30)
New Connection
Callback Server
Actual Waiting Time: 32 min
Actual Waiting Time: 32 min
Stat Server
Scheduled callbacks pass through the queue

Figure 22: Scheduled Callback Accuracy

EWT functionality is enabled or disabled by the application-level option `check_ewt`. See the *Voice Callback 7.1 Reference Manual* for more information about the `check_ewt` option.

EWT is defined as a set of parameters used to request statistics from Stat Server. Stat Server supports more than one stat type that can be used to define EWT statistics. Different EWT statistics can work for different contact centers and even different queues.

EWT statistics specifies the parameters required to obtain EWT for the queue involved in the distribution of a `Scheduled` callback.

Note: See *Framework Stat Server User Guide* for EWT statistical categories. For example:
1. Simple statistics (`CurrentMaxTime, AverageTime`).
2. Simple estimates (`EstimTimeToEndCurrentNumber`).
3. Complex EWT estimates (`ExpectedWaitTime, LoadBalance`).

Parameters for EWT Statistics

The parameters for EWT statistics are as follows:

- `Stat Server`
- `Tenant Name`
- `Switch Name`
- `ConfObject` (queue) parameters (`Name`, `Type`)
- `Statistics` parameters (`Type, Notification Mode, Time Profile, Time Range, Insensitivity`)

To Set Parameters

Use the following approach in setting them:

- Default set of parameters required to obtain EWT is configured for any Virtual Queue (VQ) assigned for callback distribution.
- Minimal set of parameters required to obtain EWT is configured in Universal Callback Server options.
- Application responsible for initial EWT retrieval and announcement (usually IVR or Router) specifies `ObjectID` in `UserData` of call or corresponding `UserEvent`, and Universal Callback Server uses EWT for this object to process a `Scheduled` callback. If object is specified in `UserData`, it overrides `Object ID` specified in Universal Callback Server options.
- All parameters that are not configured or specified in `UserData` attributes are hard coded in Universal Callback Server.
- IVR, Router, and Universal Callback Server use different Statistics to obtain EWT.

Table 7 describes this approach:

Table 7: Parameters for EWT Statistics

Parameter	Universal Callback Server section in VQ Annex	Description
StatServer Name	stat_server	If not specified, Universal Callback Server uses the first Stat Server in its Connections tab.
ObjectID (Queue) in form <DNName>@<Switch Name>	N/A	Optional. If not specified in VCN_EWT_OBJECTID attribute, then Universal Callback Server takes statistic for VRP specified in option vrp in Universal Callback Server-assigned section on RP Annex tab.
Statistic Type	stat_type	Optional. Name of Statistic Type defined in Stat Server. If not specified anywhere, Universal Callback Server uses AverDistribCallTime.
Statistic Filter	stat_filter	Optional. Name of Filter defined in Stat Server. If not specified anywhere, Filter should not be applied.
SNotificationMode	N/A	Hard coded as SChangesBasedNotification.
NotificationParam	N/A	Hard coded as 0.
TimeProfile	stat_timeprofile	Optional. Name of TimeProfile defined in StatServer. If not specified anywhere, it is not applied.
TimeRange	stat_timerange	Optional. Name of TimeRange defined in Stat Server. If not specified anywhere, it is not applied.
Tenant Name and Password	N/A	Provided by Configuration Server.

All Stat Server applications should be specified on the Universal Callback Server `Connections` tab. On startup, Universal Callback Server connects to all Stat Servers. Universal Callback Server opens required statistics only once, upon first `Scheduled` callback submission if all required parameters are specified. After this, Universal Callback Server receives modifications from Stat Server with a modified value that Universal Callback Server uses to determine the time when the `Scheduled` callback has to be submitted to URS.

All attributes that Universal Callback Server requires to obtain EWT, except `stat_server,` are configured in section `EWT` in Universal Callback Server application options. Values specified in VQ should override values specified in application.

See *Voice Callback Reference Manual* for configuration of the `stat_type`, `stat_filter`, `stat_timeprofile`, `stat_timerange`, and `stat_server` options.

Configurable Operating Hours

This feature enables contact center administrators to configure operating hours for `Scheduled` callbacks. Universal Callback Server checks scheduled time for `Scheduled` callbacks if operating hours are configured.

Configuration Location

Configure operating hours in folder `Tenant >Statistical Tables > UCBServer.`

- Statistical table `Weekdays` contains one object per standard business day.
- Statistical table `Holidays` contains one object per holiday falling on a business day.
- Statistical table `Overtime` contains one object per working day falling on a weekend.

See Figure 23 on page 121 for an example of how to configure this feature.

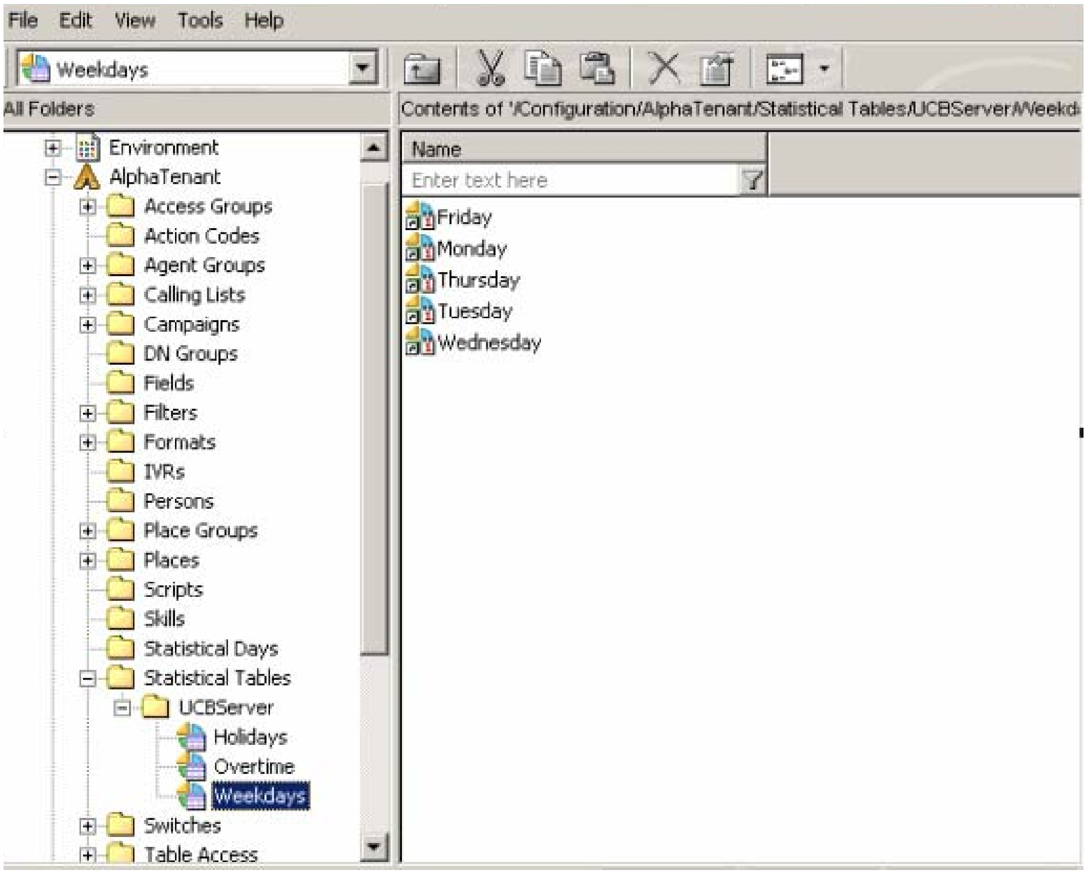


Figure 23: Configuration Manager

To Configure Operating Hours

When configuring operating hours, keep in mind the following

- Name of `UCBServer` folder and names of statistical tables `Holidays`, `Weekdays`, and `Overtime` cannot be changed.
- Statistical tables should have type `Special Day` table.
- Requirements for objects in `Weekdays` statistical table are:
  - Property `Name`—any
  - Property `isDayOfWeek`—`True`
  - Property `day`—numbers from 1 to 7 (Sunday to Saturday)
  - Properties `startTime`, `endTime`—start and end of business day in minutes from 00:00
  - Only one object per each day of week (Universal Callback Server uses data only from first one)
- Requirements for objects in `Holidays` and `Overtime` statistical tables are:
  - Property `Name`—any
  - Property `isDayOfWeek`—`False`
  - Property `day`—if property date is not specified, number of day in year when 1 is January 1

  - Property date—date in seconds from 00:00 01/011970
  - For objects in Overtime statistical table, properties startTime, endTime—start and end of business day in minutes from 00:00

- Universal Callback Server checks the sequence of scheduled date-time for Scheduled callback:
  - Checks scheduled date against objects in statistical table Holidays.
  - Checks day of week and time of day, calculated for scheduled date-time, against objects in statistical table Weekdays to determine if this is the correct business day and if the time is within business hours.
  - Checks scheduled date against objects in statistical table Overtime to determine if this is the correct business day and if the time is within business hours.

Note: If the configuration does not have a necessary object for some weekday (in Weekdays statistical table), this day is considered a holiday.

If scheduled date-time is not within business hours, Universal Callback Server rejects the scheduled callback. See Figure 24 for an example of an hours-of-operation configuration.

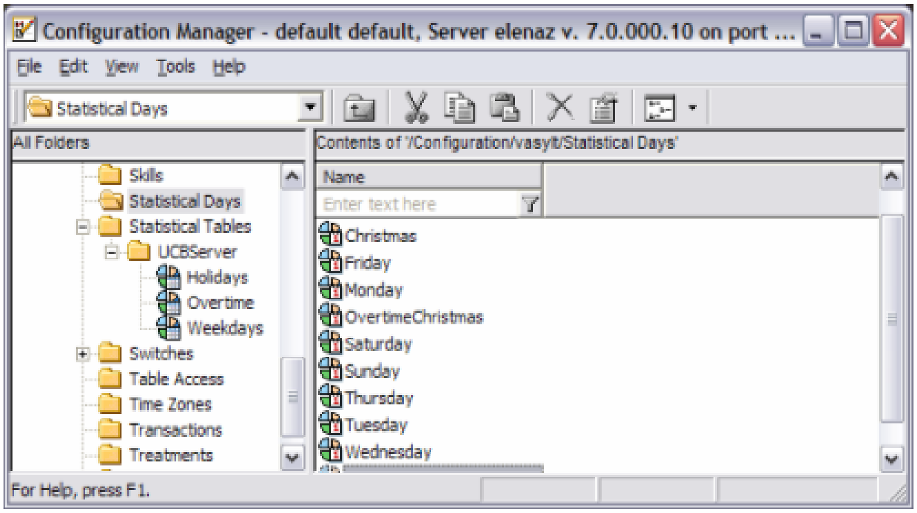


Figure 24: Hours of Operation

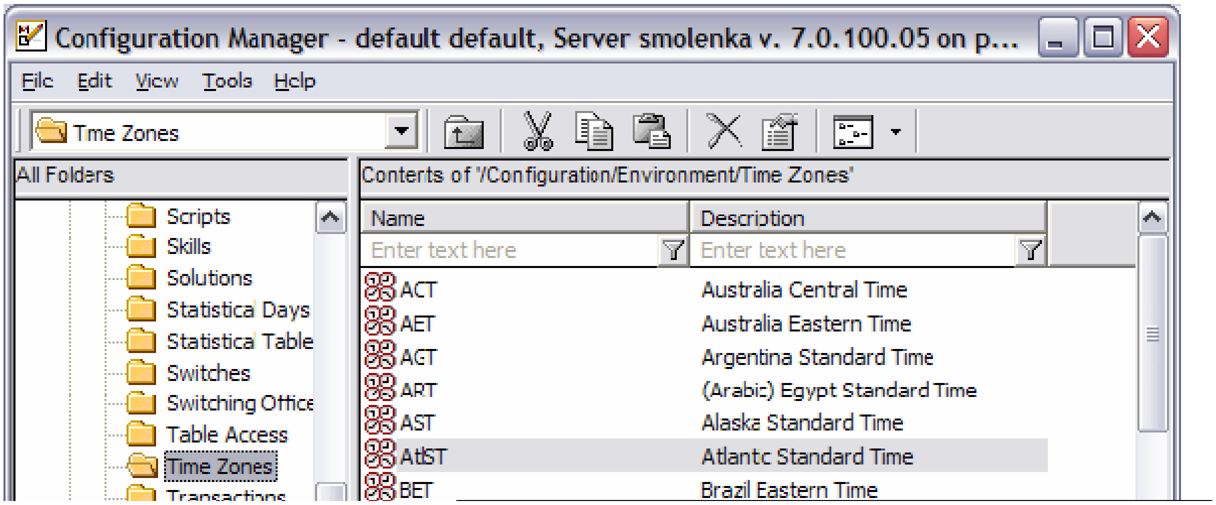
For a request with an error message, see Table 8.

Table 8: Request with Error Message

VCB_ERROR	VCB_ERROR_CODE	VCB_ERROR_NAME (Description)
Invalid attribute value	112	Scheduled Date-Time is not within business hours.

Time-Zone Support for Scheduled Callbacks

The callback requestor is responsible for putting in the proper time-zone name in the scheduled `Callback Request` attributes (see `VCB_TZ_NAME`). If the proper time zone name is specified, Universal Callback Server retrieves offset from the Configuration Server and uses it for processing that callback request. See Figure 25 for Time Zone configuration.



Contents of '/Configuration/Environment/Time Zones'

Name	Description
ACT	Australia Central Time
AET	Australia Eastern Time
AGT	Argentina Standard Time
ART	(Arabic) Egypt Standard Time
AST	Alaska Standard Time
AtlST	Atlantc Standard Time
BET	Brazil Eastern Time

Figure 25: Time Zone Support

Time zone is determined for Universal Callback Server from OS settings or configured in Universal Callback Server option.

- Caller's time zone is defaulted to Universal Callback Server time zone, but can be an optional parameter (hour difference with GMT) for `Scheduled` callback.
- The requestor desktop is responsible for putting caller's correct time zone in the parameters.
- Universal Callback Server adjusts the time for `Scheduled` callback according to time-zone difference.

If the name of time zone is incorrect, Universal Callback Server rejects this callback request with the error message shown in Table 9:

Table 9: Error Message

VCB_ERROR	VCB_ERROR_CODE	VCB_ERROR_NAME (Description)
Invalid attribute value	102	Time zone not found

Agent Reject/Timeout Process for Preview Dialing

If an agent rejects the callback preview or if the timeout has expired for accepting it, the callback is returned to the top of the queue (first in line) to be delivered to the next available agent.

- After Universal Callback Server sends the preview record to the desktop, it expects from the desktop an acknowledged response for `RequestCallbackPreview` or request `RequestCallbackReject`.
- If Universal Callback Server receives request `RequestCallbackReject`, it submits `EventError` to the URS to redistribute callback.
- If Universal Callback Server does not receive any expected response from the desktop after the timeout specified in option `call_preview_agent_ack_timeout` (`10 seconds` by default), it submits `EventError` to URS to redistribute callback.

Chapter

7 Designing IVR Scripts and Routing Strategies

This chapter describes how to design IVR scripts and routing strategies.This chapter contains the following sections:

Designing IVR Scripts

An Interactive Voice Response (IVR) product is required to enable VCB functionality. The role of the IVR in VCB is to:

- Obtain customer information.
- Offer the callback options.
- Collect callback data.
- Confirm the callback order and disconnect.

VCB works with any Genesys-compliant IVR; that is, the IVR must work with Genesys IVR Drivers to interface with Framework. You can continue to use your existing IVR or another IVR product. Genesys offers two types of IVR: Genesys Voice Portal and Voice Treatment Option.

Communication between the IVR, mediated by an IVR Driver, and the Universal Callback Server (the main component of VCB) is accomplished through an IVR script.

The following sections describe callback functions in the IVR script and rules governing the interactions between the script and the IVR Driver.

Two types of IVR configurations are discussed:

- `IVR-in-front-of-the-switch`
- `IVR-behind-the-switch`

IVR-In-Front-of-the-Switch Configuration

In an `IVR-in-front-of-the-switch` configuration, if a caller selects the callback option:

1. The live call terminates on an IVR port. In this configuration, the live inbound call never leaves the IVR port.
2. An I-Server (virtual T-Server) creates a virtual call to pass callback information between the IVR and the Universal Callback Server. That is, the callback information associated with the original call is assigned to the virtual call and passed, as attached data in a T-Server event, between the IVR and Universal Callback Server.
3. In this configuration, the IVR script simulates a call, which arrives at the IVR port and later is routed on a Routing Point (RP1).

   Both the IVR port and RP1 belong to a virtual switch and are not represented by any hardware. VCB treats a virtual call on RP1 in the same way that it treats a regular call.

Requests for IVR-In-Front-of-the-Switch Configuration

To simulate this call in your script, use the following set of requests, which are specific to an `IVR-in-front-of-the-switch` configuration:

- `NotifyCallStart`
- `WaitReply`
- `DataGetKD`
- `UDataAddKD`
- `RouteStart`
- `WaitRequest`
- `NotifyCallEnd`

EWT

The script requirements for EWT (Estimated Waiting Time), described in "VTO Scripts" on page 128, also apply to the `IVR-in-front-of-the-switch` configuration. Use the `StatPeek()` statistics function to obtain EWT in your IVR script.

IVR Behind-the-Switch Configuration

In an `IVR behind-the-switch` configuration, the live call terminates on an IVR port. In this configuration, the IVR script:

1. Plays the needed announcements.
2. Attaches the callback-related data to the call.

3. Passes that data to the Routing Point (RP1).

Strategy on RP1

Strategy on RP1 involves the following:

1. Call is distributed to a target based on business logic.
2. Treatment is set for the target in the routing strategy.
3. Router executes a busy treatment when no agents are available. Typically, the treatment returns the call to the IVR port, where IVR announces the EWT and, if appropriate, the callback options (`ASAP` or `Scheduled`).

Strategy on VRP1

Strategy on VRP1 involves the following:

1. Callback request is distributed to an agent for processing.
2. Only virtual calls are distributed by this strategy; therefore, no configuration of treatments is required.
3. Router does not apply treatments to virtual calls with `Skip_Treatments` in the attached data.

urs_treatment_mode Option

The value of the `urs_treatment_mode` option determines the *means* by which the router delivers the call to the IVR. Depending on the `Treatment` mode (compatible or regular), set the value of the `urs_treatment_mode` option to either `transfer` or `treatment`.

- If a treatment is defined in the routing strategy (compatible mode), the value of the option `urs_treatment_mode` should be `transfer`.
- If the router uses a switch treatment (regular mode), then the value of the `urs_treatment_mode` option should be `treatment`.

For more information about the `urs_treatment_mode` option, see "Configuration Options" in the *Voice Callback Reference Manual.*

Script functions

Genesys recommends that you use the special script functions supported by the Genesys IVR driver you are using. Examples of these script functions include attaching data to the call, transferring the call, and hanging up the call. The actual names and parameters of these functions depend upon the IVR type.

EWT

The script requirements for EWT, described in "VTO Scripts" on page 128, also apply to `IVR-behind-the-switch` configuration. Use the `StatPeek()` statistics function to obtain EWT in your IVR script.

Scenario

An `IVR-behind-the-switch` configuration enables the following scenario:

1. Inbound call arrives at RP1.
2. No agents are available to process this call, so the call receives an IVR treatment.
3. IVR announces the EWT to the customer and suggests that the customer selects the callback option.
4. IVR submits the callback request (the inbound call with specific attached data) to RP1.

5. Callback is scheduled. IVR confirms to the customer the `Scheduled` callback and releases the inbound call.

VTO Scripts

When creating Voice Treatment Option (VTO) scripts to use with VCB, you are likely to use these kinds of actions, which implement data exchange between Universal Callback Server and VTO:

Actions

- Is-Equal string
- Assign
- Attach data to the call info
- Hang up
- Transfer

Template for VTO script

A typical VTO script can:

1. Greet customer and/or announce the EWT in the queue.
2. Attach needed data (per VCB protocol) to check if the VCB option is available and then transfer the call to the Routing Point (RP1).
3. If attached data indicates that the VCB option is available, suggest callback to the customer.
4. Gather customer's input for callback information, attach it to the call, and transfer it to RP1.
5. If the attached data indicates that the callback has been scheduled correctly, announce this to the customer and release the call.

Note that in this scenario the call is passed several times between the VTO port and RP1, which involves significant wait times. Therefore, Genesys recommends a routing strategy in which the router distributes the call from RP1 back to the VTO port.

EWT

VCB normally works with EWT statistics in these ways:

- IVR announces the EWT to the customer.
- Customer uses the EWT to consider the callback option.
- EWT affects the customer's expectation regarding the time of the callback once the customer has chosen the callback option.

For these reasons, the VTO script should attach the EWT data to the callback request (`VCB_EWT` and `VCB_EWT_TIME` keys). Therefore, you usually need to create and use VTO scripts of the `Enhanced Queuing` type (see "Script Types" on page 43.

Use the `ExternalQueuingTime` parameter whenever possible to make the EWT statistics more accurate.

Please refer to "Designing Routing Strategies" on page 134 for information about the accuracy of the EWT calculation.

Warning! When designing a routing strategy, keep in mind that VTO does not process `EventAttachedDataChanged`. Only attached data present when the call arrives at the VTO port (`EventRinging`) is reflected in the script's variables. Data modified after the call has arrived at the port is not reflected in the variables.

Implementation of IVR Script Logic

Figure 26 shows the basic IVR logic as applied to VCB.

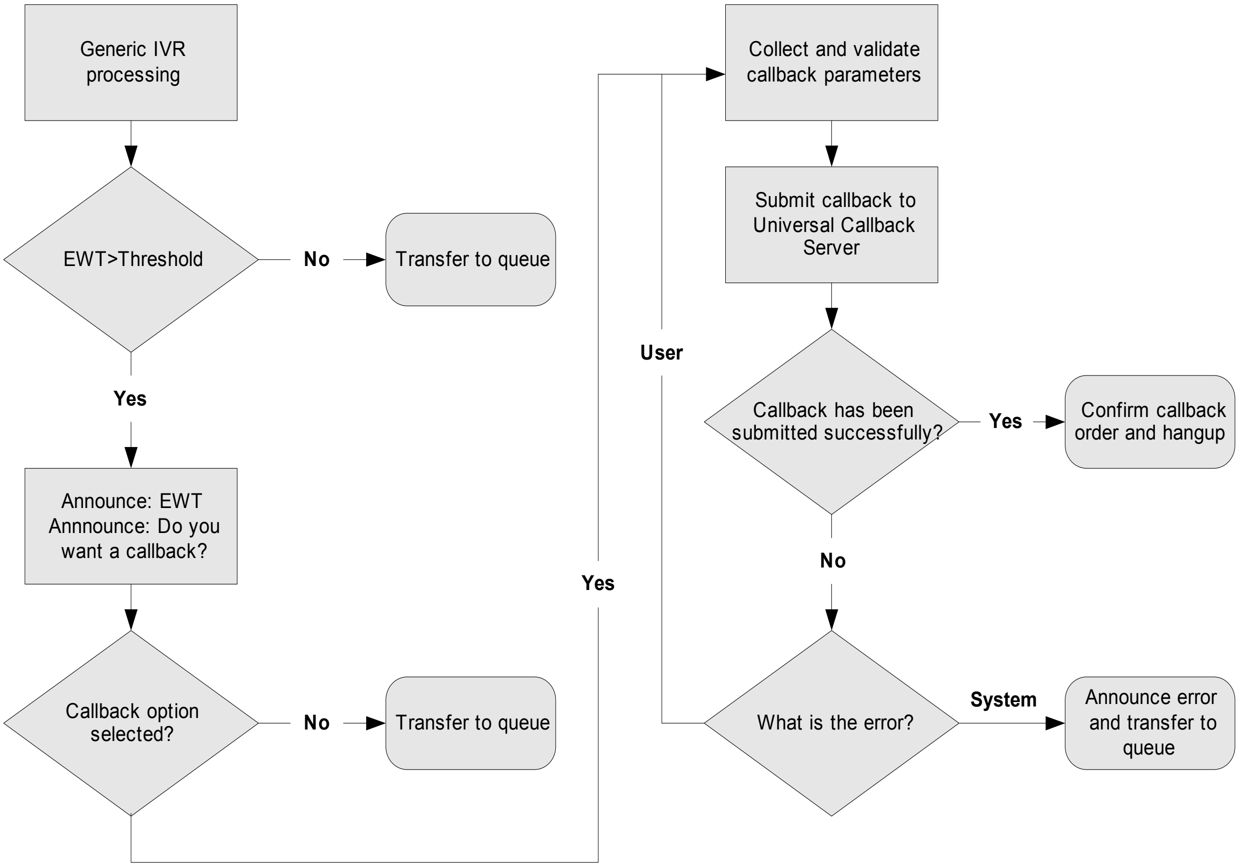

Generic IVR processing
EWT>Threshold
No
Transfer to queue
Yes
Announce: EWT
Annnounce: Do you want a callback?
Callback option selected?
No
Transfer to queue
Yes
User
Collect and validate callback parameters
Submit callback to Universal Callback Server
Callback has been submitted successfully?
Yes
Confirm callback order and hangup
No
What is the error?
System
Announce error and transfer to queue

Figure 26: IVR Logic Applied to VCB

The IVR logic in Figure 26 is as follows:

1. After the initial voice greeting and prompts, the IVR determines if the EWT for the caller exceeds a predefined threshold.
   - If the EWT does not exceed the predefined threshold, the call is transferred to a business queue for an agent.
2. If the EWT exceeds the predefined threshold, IVR announces the EWT to the caller and offers the callback options: `ASAP` callback or `Scheduled` callback.

- If the caller prefers to wait on hold, the call is transferred to a business queue for an agent.

3. If the caller selects a callback option (`ASAP` or `Scheduled`), IVR collects and validates the callback information.
4. IVR submits the callback request to the Universal Callback Server.
5. If Universal Callback Server accepts the callback request, IVR confirms the callback order to the caller and hangs up.
   - If a system error occurs in response to the callback request, the call is transferred to a queue.

Note: IVR and Universal Callback Server in Figure 26 do not interact directly. The interaction involves call bouncing, which means that the call bounces from the IVR port to the Routing Point and back again.

EWT Statistics

EWT statistics for `Scheduled` callback requests specify the parameters required to obtain EWT for the queue involved in the distribution of a `Scheduled` callback. The queue for the call (DN object, usually a Virtual Queue) is determined when the callback is ordered.

Universal Callback Server takes into account EWT for resource-waiting queue. This means that `Scheduled` callback is submitted to the router in the format, `time = scheduled time - EWT`.

This functionality is enabled or disabled by the application-level option `check_ewt`. See "VCB Configuration Objects" in the *Voice Callback Reference Manual* for more information about the `check_ewt` option.

Sample Implementation of IVR Script Logic

The IVR script is responsible for gathering all the required information for the callback request. Figure 27 on page 131 is a flow chart for a sample IVR script based on the IVR logic shown in Figure 26. This IVR script consists of two branches. One branch (on the right side of the figure) shows new call processing. Another branch (on the left side of the figure) shows callback request processing.

This sample script uses call bouncing: the inbound call bounces back and forth between the IVR port and the Routing Point. To minimize this bouncing between the IVR and the router, the script assumes that the routing strategy determines if the callback option is available.

Also note that in Figure 27 on page 131, the IVR has already given the initial greeting to the customer and ascertained if the EWT for the desired queue exceeds the predefined threshold. This example begins at the point when IVR determines if the call is already being processed for a callback (left branch) or if the call is new (right branch). The numbers and letters in the flow chart are explained in "Annotations" on page 132.

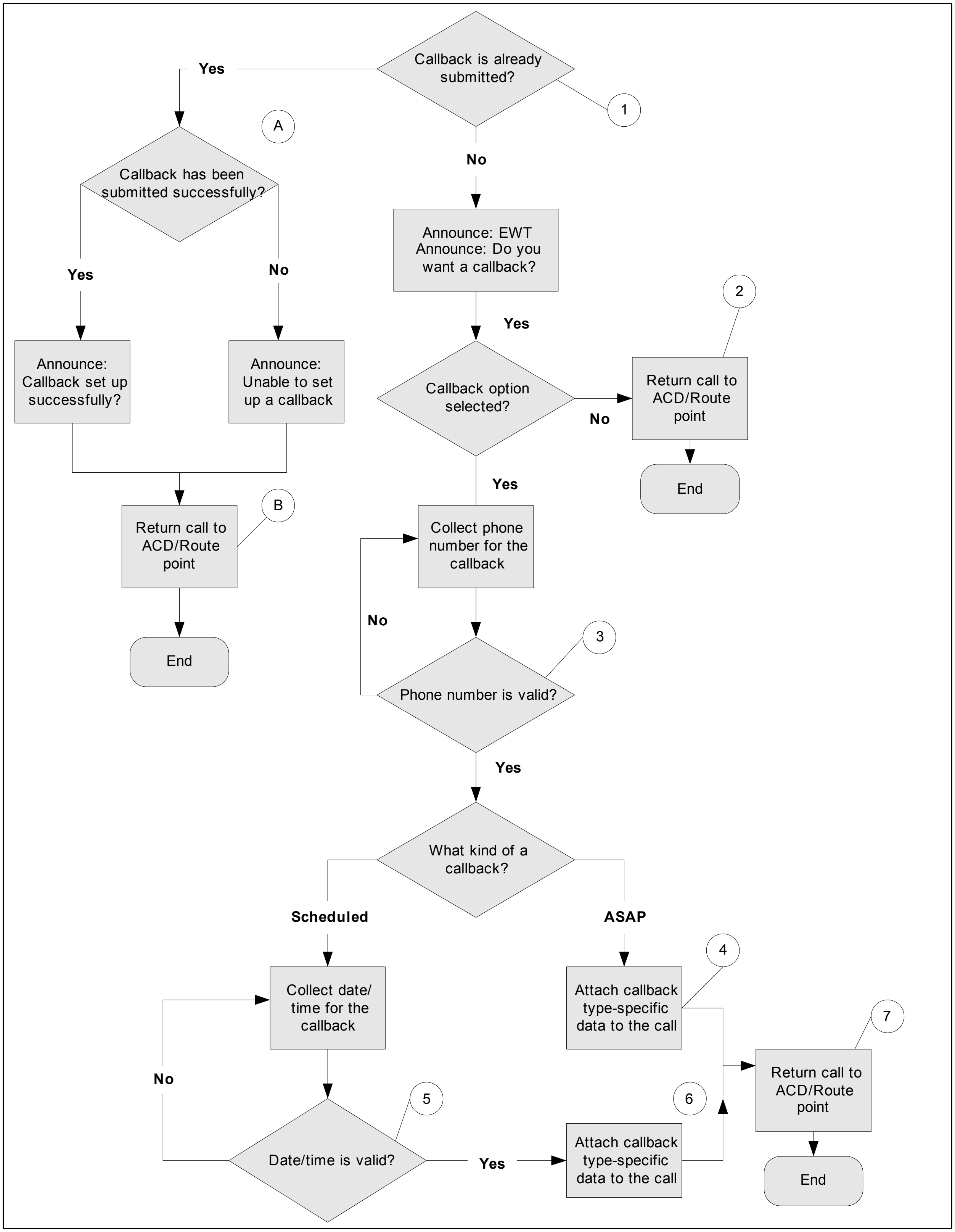

Callback is already submitted?
1
Yes
No
A
Callback has been submitted successfully?
Yes
No
Announce: Callback set up successfully?
Announce: Unable to set up a callback
Return call to ACD/Route point
B
End
Announce: EWT
Announce: Do you want a callback?
Yes
Callback option selected?
No
Return call to ACD/Route point
2
End
Yes
Collect phone number for the callback
No
Phone number is valid?
3
Yes
What kind of a callback?
Scheduled
ASAP
Collect date/ time for the callback
No
Date/time is valid?
5
Yes
Attach callback type-specific data to the call
4
Attach callback type-specific data to the call
6
Return call to ACD/Route point
7
End

Figure 27: Sample Script Implementation of IVR Logic

Annotations The numbers and letters in the following annotations correspond to the numbers and letters in Figure 27 on page 131. After number 1, the script diverges into separate branches. One branch includes numbers 2 through 7. These numbers indicate the sequence of actions that the IVR script performs to process a new incoming call. Another branch follows a different course of action (see letters A and B) to process a callback request.

1. Two types of calls are entering the IVR:
   - New incoming calls
   - Calls returned to IVR to confirm that the callback request has been submitted

   The `VCB_SUBMIT` key in the user data distinguishes the second call type.
   - If `VCB_SUBMIT` is present, then the IVR assumes that the callback request has been submitted.
   - If `VCB_SUBMIT` is absent, then IVR assumes this is a new incoming call.

Branch One: IVR Processes a New Incoming Call

2. IVR returns the call from the IVR port to the Routing Point, where the call continues to wait on hold.
3. IVR determines that the length of the phone number is valid and that the numbers adhere to the numbering plan used in the calling area.
4. Data required for `ASAP` callback is attached to the call. An `ASAP` callback request script looks like this:

```
VCB_CONTACT = <phone number>
VCB_TYPE = 1
VCB_SUBMIT = 0
VCB_EWT = <estimated wait time>
VCB_EWT_TIME = <time when EWT was calculated>
```

5. IVR checks the format of the date/time.
6. Data required for `Scheduled` callback is attached to the call. A `Scheduled` callback request script looks like this:

```
VCB_CONTACT = <phone number>
VCB_TYPE = 2
VCB_DATE_TIME = <date/time for the scheduled call>
VCB_SUBMIT = 0
VCB_EWT = <estimated wait time>
VCB_EWT_TIME = <time when EWT was calculated>
```

Note: The values of the `VCB_DATE_TIME` and `VCB_EWT` pairs in attached data must always conform to the format described in the *Voice CallBack 7.1 Reference Manual*. There is one exception: a value of `VCB_EWT` could be `0` (for example, to indicate a situation when the EWT is not available). Format conformation is also applicable for situations when VCB_DATE_TIME pair is not mandatory (for example, ASAP callbacks).

7. IVR returns the call to the Routing Point for Universal Callback Server processing. Later the call is returned to IVR as a properly submitted callback request.

Branch Two: IVR Processes a Callback Request

In Figure 27 on page 131, the IVR script begins with number 1. The processing of a callback request then branches into the flow indicated with the letters A and B.

A. IVR checks the following key-value pairs in the user data:

VCB_SUBMIT = 1 — Universal Callback Server has successfully submitted the callback request.

VCB_SUBMIT = 0 — An error occurred when Universal Callback Server attempted to submit a callback request. The callback request was unsuccessful.

B. If the callback request has been submitted successfully (VCB_SUBMIT = 1), IVR can either:

- Release the call on an IVR port.

  Or

- Pass the call back to a Routing Point to be released on a termination DN (if you have configured a termination DN using the option term_dn). For more information about the termination DN, see "DNs for Call Termination" on page 116.

Designing Routing Strategies

VCB requires routing strategies to direct call flow.

Although routing strategies differ according to the business goals of each contact center, keep the following guidelines in mid as you design routing strategies for VCB.

Consideration 1

Strategies that are loaded for Routing Point 1 (RP1) and for Virtual Routing Point 1 (VRP1) are generally two different strategies. In VCB,

- RP1 is a physical Routing Point where customers' calls wait for an agent to become available.
- VRP1 is a Virtual Routing Point where virtual callback requests reside, waiting for an agent to process them.

In some cases, the strategies for RP1 and VRP1 might be the same—for example, if the same group of agents is processing live inbound traffic and callbacks under the same business logic. However, this is not required by the Universal Callback Option.

Consideration 2

If Universal Callback Server is configured to deliver callback requests in the `Preview` dialing mode—that is, if the `delivery_mode` option is set to `preview`—then the Stat Server and, in turn, the router, cannot determine the agent's state.

For example, if the agent is busy working with a callback request or waiting for the next callback request, the agent's state is not apparent to the Stat Server and, therefore, cannot be communicated to the router by means of T-Server events. The VCB desktop protocol is based on `User` events, which Stat Server does not process by default.

To Signal Agent's Status to Stat Server

Use the agent desktop application to transform desktop protocol events to telephony- or agent-related events. For example, distribute an `EventAgentNotReady` each time an agent receives a callback, and distribute an `EventAgentReady` each time an agent finishes processing the callback request.

Consideration 3

It is very important to provide the customer with an accurate EWT because this affects his or her decision to request a callback or to wait on hold.

The strategy on RP1 is designed so that the EWT calculated by Stat Server is as accurate as possible. To achieve this, use targets with clear target-selection rules:

- If you think in terms of `If-Then` statements, a routing strategy with nonapparent target selection rules has a number of `Then` statements following an `If` statement; a routing strategy with clear target selection rules has one or very few `Then` statements following the `If` statement.
- If you use clear target-selection rules, you can avoid the situation in which the customer is quoted an EWT calculated for Target 1, and then his or her inbound call or callback request is directed to Target 2 where the EWT is totally different. (This might occur as a result of a change to the attached data, which then requires redirecting the call or the callback request.)

Consideration 4

This consideration applies only to multi-site deployment, that is, when calls and callback requests are routed from one location (site 1) to another (site 2). In such a configuration, the strategy which is loaded to the Virtual Routing Point (VRP1) should always call the function `SetTranslationOverride()` before distributing a virtual call.

The reason for this is that the Inter Server Call Control (ISCC) cannot route the virtual call, which represents the callback request, between multiple sites.

As a result, `AttributeLocation` in `RequestRouteCall` should always contain the name of the local switch so that the ISCC mechanism is not involved in routing. Universal Callback Server can extract the correct location (switch name) from the `AttributeExtensions` of the `RequestRouteCall.` Universal Callback Server does not depend on the correct location (switch name) in `AttributeLocation`.

The following usage of the function `SetTranslationOverride()` ensures that the local switch name is always used in `RequestRouteCall`, regardless of the actual (possibly remote) location of the Target DN:

```
SetTranslationOverride
['','[DEST.DN]', '[SOURCE.SWITCH]', RouteTypeUnknown, '','','']
```

For further information, see the *Universal Routing Reference Manual* on the number translation and `SetTranslationOverride()` function description.

Routing Strategy Logic

Figure 28 depicts how a routing strategy works:

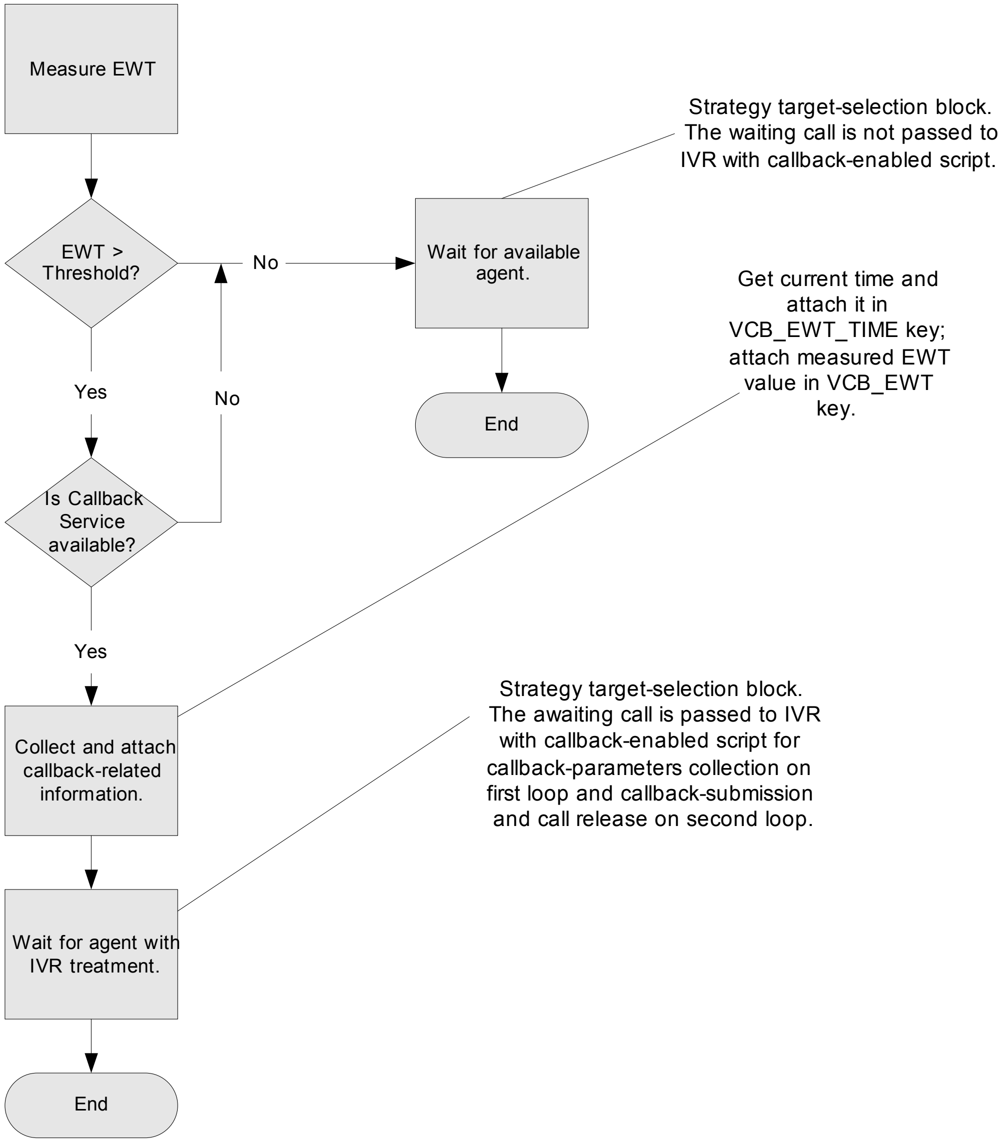

Measure EWT
EWT > Threshold?
No
Yes
Is Callback Service available?
No
Yes
Wait for available agent.
End
Collect and attach callback-related information.
Wait for agent with IVR treatment.
End
Strategy target-selection block. The waiting call is not passed to IVR with callback-enabled script.
Get current time and attach it in VCB_EWT_TIME key; attach measured EWT value in VCB_EWT key.
Strategy target-selection block. The awaiting call is passed to IVR with callback-enabled script for callback-parameters collection on first loop and callback-submission and call release on second loop.

Figure 28: Routing Strategy Logic

Note: The VCB 7.1 CD contains a sample routing strategy.

Preprogrammed VCB + IVR Bundle

VCB+IVR bundling contains preprogrammed IVR scripts that contain the procedures show discussed below.

VCB IVR Interface Operation

This section describes the requirements for VCB processing for IVR, which may be used as a template for developing IVR scripts development. (Some of these may be implemented outside of IVR by a routing strategy). VCB requires IVR to be programmed; the following steps are included in the VCB+IVR bundling functionality. See Chapter 8.

VCB Processing for IVR

1. Obtain caller's information to make a routing decision: target the call to a particular agent.
2. Obtain Estimated Waiting Time (EWT) for the call from Stat Server.
3. Make a decision to offer callback based on EWT and configured threshold.
   a. Check callback-service availability before offering it. It may be that the service is down or license capacity has been exceeded.
   b. Offer callback before call goes to waiting queue. The caller can then choose to go to a live queue (wait online or go to callback queue) or wait offline.
   c. Announce EWT when offering callback before call goes to queue.
   d. Offer callback multiple times when call is waiting in live queue online. This allows the caller the option to switch from online waiting to offline (request callback) without losing position in queue.
   e. Announce EWT after some waiting in queue online.
4. Transfer call to live queue to wait online if caller rejects callback offer.
   - Offer ASAP callback. If accepted, the caller disconnects but retains position in queue with a callback request placeholder.
   - Offer Scheduled callback for a specified time and date, a second option for callback selection.
5. Prompt caller for scheduled date and time for callback.
   a. Check scheduled time against operating hours. Make sure scheduled time is within operating hours of the contact center. Reprompt if it is not.
   b. Check scheduled time against EWT. Make sure scheduled time is after the current time + EWT.
   c. Announce and confirm scheduled date and time and get caller's confirmation.
   d. Obtain caller's time zone for Scheduled callback.

6. Use ANI for callback number. Announce caller's ANI and get confirmation that it is correct.
    - Prompt caller for callback number. If ANI is not available or caller does not confirm its usage for callback, prompt caller to enter desired callback number. Announce number and get confirmation that it is correct.
7. Record caller's name in voice file to be used for automatic announcement of callback to the caller or to be played to agent before connecting to the caller.
    - Record caller's voice message to be used for automatic announcement played to the agent before being connected to the caller.
8. Submit callback order to Universal Callback Server with all relevant callback data. Callback order is not finished until Universal Callback Server confirms the request.
9. Confirm callback order. Caller is told that the callback has been successfully scheduled before disconnecting or told why a callback has not been scheduled.
10. Allow caller to the retry callback order after failure. If the callback is not scheduled because the submitted callback number is not allowed or any other reason dependent on caller input, allow the caller to retry the callback order using different parameters.

Chapter

8 Sample Genesys Voice Portal Script

This chapter describes a sample Genesys Voice Portal (GVP) script and has the following sections:

The sample script is an example of the GVP and Genesys Universal Callback bundle. GVP sample script consists of a Genesys Studio project file `vcb_gvp_sample.vws` and a number of `VOX` files which are voice prompts used in the script. This script was developed using Genesys Studio v. 6.5.320.10 and tested with GVP VCS (Voice Communication Server) v. 6.5.310.17, VPM (Voice Portal Manager) v. 6.5.300.00 with Dialogic board DM/V480A-2T1-PCI.

Installation

The installation program copies the project file into the user-selected directory, and in this directory it creates a subdirectory `VoxFiles/en` where all voice prompts (`*.vox files`) are located.

1. Open the project file with Genesys Studio and press `<F2>` to generate `ASP` code.
2. For generated `ASP` files, select the same directory where the project file is located. This ensures that the script can locate the required `VOX` files.
3. Alternatively, you can store generated `ASP` files elsewhere and adjust the `VOXFILEDIR` script variable.

Note: Before code generation, you must modify the sample script to function properly in your environment. See "Script Customization" on page 142 for more details.

Prerequirements

To successfully use the sample script, you must install and configure the following components (please refer to the respective component's documentation for installation/configuration description).

Components

1. GVP (VCS, VPM, Genesys Studio) in either `in-front-of-the-switch` or `behind-the-switch` configuration, with `ASP` support mode.
2. IVR Server, which integrates GVP and other Genesys components.
3. Universal Routing Server (URS) and Interaction Routing Designer (IRD).
4. Universal Callback Server.
5. Microsoft SQL DBMS, or one of the other DBMSs (MS Access, Oracle, MySQL) that GVP supports.

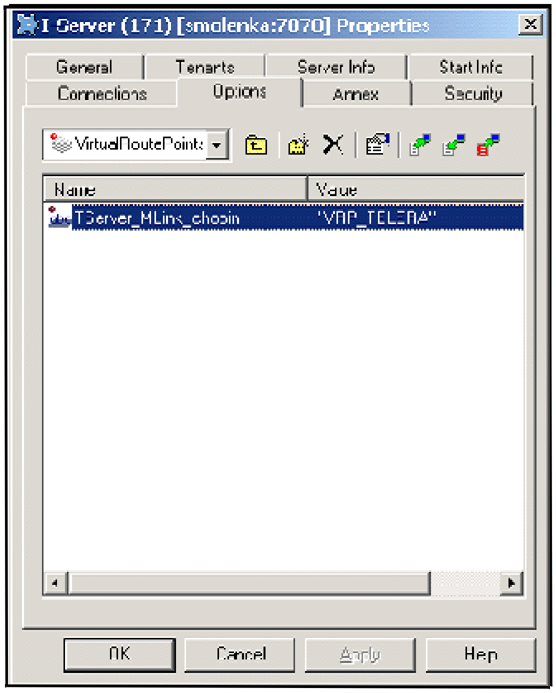
Universal Callback Configuration

Since GVP uses a treatment-based (that is, `RequestApplyTreatment -> EventTreatmentApplied -> EventTreatmentEnd`) model to communicate to URS in both `in-front-of-the-switch` and `behind-the-switch` modes. Universal Callback Server, which monitors the Routing Point, reacts on `EventTreatmentEnd`. This is configured by setting the Application-wide or Routing Point-specific option `urs_treatment_mode` to the `treatment` value.

IVR Server Configuration

Note: Regardless of the mode (`in-front-of-the-switch` or `behind-the-switch`), IVR Server is always configured with two applications in Configuration Layer: Virtual T-Server and I-Server.

To specify Routing Point(s) or Virtual Routing Point(s) on which IVR Server should register, use the option with the name of the T-Server and the value of the Routing Point(s) name(s) under the `VirtualRoutePoints` section in the T-Server application. See Figure 29 on page 141.



Name	Value
TServer_MLink_chopin	"VRP_TELERA"

Figure 29: Sample of VRP Configuration

Note: IVR Server uses register mode 11 to register on Routing Points that are configured in the way described above. This mode allows IVR Server to receive T-Requests sent on these Routing Points. Only one application at a time can register DN with mode 11. Universal Callback Server also uses register mode 11 to register on VRPs specified in `vrp` option. Therefore, you should use different VRPs for IVR Server and Callback Server.

Database

- The sample script accesses the DBMS you are using to map the phone's area code to the name of that time zone.
- This script block, `TZ_LOOKUP`, accesses MS SQL Server database and selects a time zone abbreviation from the `Timezones` table that has the area code.
- The table should be located in the `MASTER` database.

To Create This Table

To create this table, the following SQL statements could be used:

```
USE Master
go
CREATE TABLE timezones (area_code varchar (3) UNIQUE, tz_name
varchar (3))
go
```

To populate this table with the sample data, you can use the following SQL statements:

```
INSERT INTO timezones (area_code, tz_name) VALUES ('212','EST')
INSERT INTO timezones (area_code, tz_name) VALUES ('303','MST')
INSERT INTO timezones (area_code, tz_name) VALUES ('415','PST')
INSERT INTO timezones (area_code, tz_name) VALUES ('650','PST')
INSERT INTO timezones (area_code, tz_name) VALUES ('925','PST')
go
```

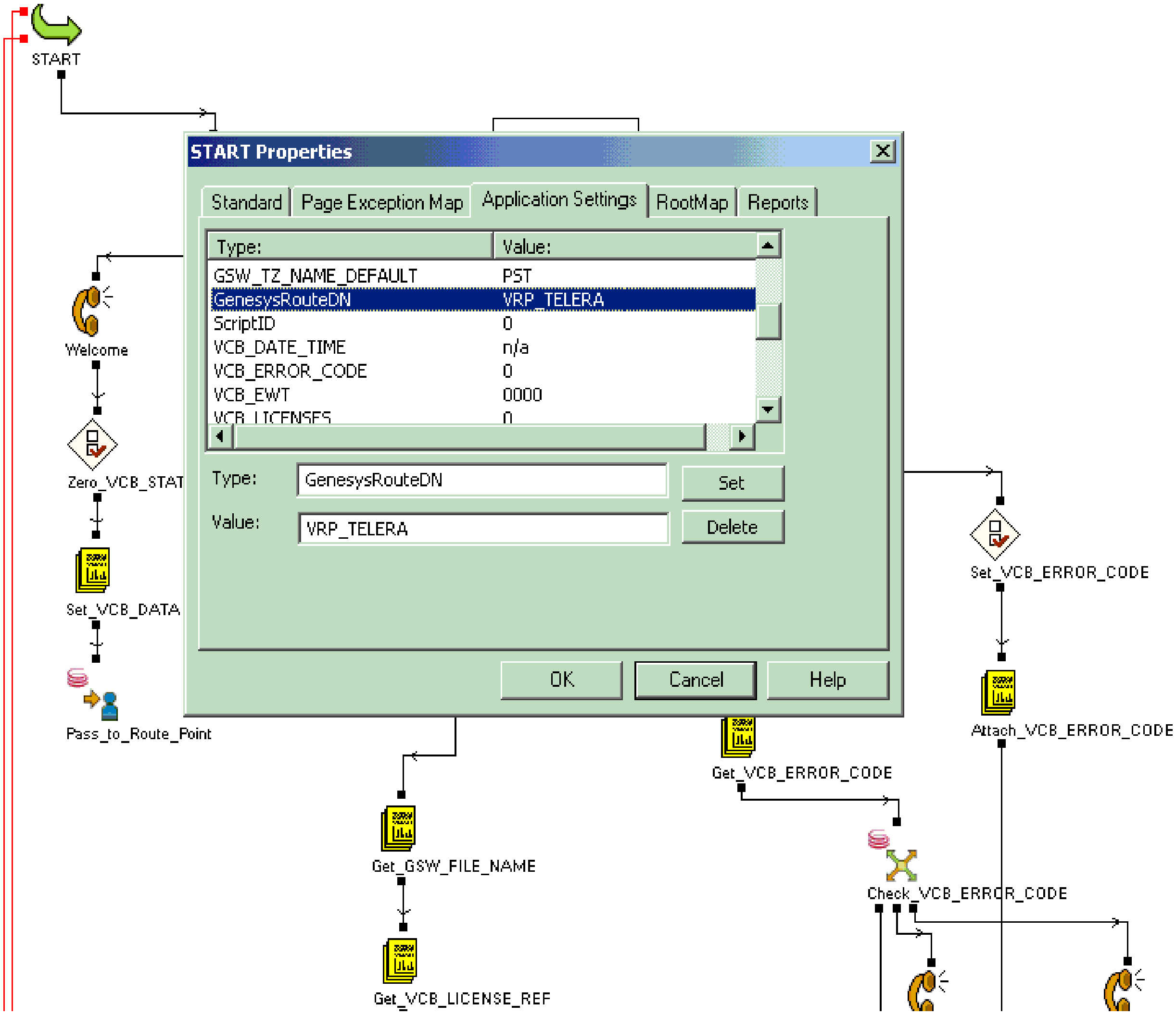
Script Customization

To customize the sample script for your environment, make the following modifications.

To Modify Sample Script

1. Right-click `START` to access the `START Properties` dialog box.
   a. Select `Application Settings` tab.
   b. Scroll down and select script variable `GenesysRouteDN` (default value `VRP_TELERA`).
   c. Make sure that variable is set to the name of Virtual Routing Point (`behind-the-switch` mode) or Routing Point (`in-front-of-the-switch` mode) where the call receives `Play application` treatment and is monitored by Universal Callback Server.

   Figure 30 on page 143 shows the `Start Properties` dialog box.



Type:	Value:
GSW_TZ_NAME_DEFAULT	PST
GenesysRouteDN	VRP_TELERA
ScriptID	0
VCB_DATE_TIME	n/a
VCB_ERROR_CODE	0
VCB_EWT	0000
VCB_LICENSES	0

Type: GenesysRouteDN

Value: VRP_TELERA

Figure 30: START Properties Dialog Box

2. Set script variable `VCB_ROUTING_POINT` (default value `VRP_TELERA`) using the process described in step 1.
3. Set the script variable `VCB_LOCATION` to the name of the switch in Configuration Layer to which the Routing Point belongs, using the same process that is described in step 1.
4. Set script variable `GSW_SHARE_NAME` to the `UNC` path where the `VOX` files with customers' names are stored.

Note: The use of forward slashes instead of back slashes in the sample script should be noted. Example: `//host_name/share_name/folder`

5. Set script variable GSW_TZ_NAME_DEFAULT to the three-letter abbreviation of the time zone used if no time zone has been provided by the customer and no time zone has been found in the database table (for example, PST).
6. Right-click script block Pass_to_Route_Point to access the Pass-to-Route-Point Properties dialog box.
   - Select Route Point tab: Route DN field should be set to the name of VRP or Routing Point. (See step 1 above.)
7. Modify script block TZ_LOOKUP to allow for a database connection.
   a. Right-click script block TZ_LOOKUP to access the TZ_LOOKUP Properties dialog box.
   b. Select the SQL_Query_Builder tab.
   c. Click Frame Query button and select your DB engine, connection parameters, table name, and column name. See “Prerequirements” on page 140.
8. Modify script block Capture_VOX_file.
   a. Right-click the script block Capture_VOX_file in order to access the Capture_VOX_file Properties dialog box.
   b. Select the Capture tab.
   c. Select the Specify capture file location check box.
   d. In the Folder field, click Browse and select the folder where the recorded files will be stored.

Note: The folder you have selected is shared, and the name of the share is set in GSW_SHARE_NAME (see step 4).
Also note that for successful VOX file creation a user with name IUSR_<IIS_host_name> should have Full Control access to this folder.

Detailed Script Description

GVP communicates to Universal Callback Server via the user data attached to each call that might require a callback request. Script variables prefixed with VCB_ are the keys in the user data with which Universal Callback Server is working. The script also employs several auxiliary internal variables used for script logics. These auxiliary variables are prefixed with GSW_.

Script Branches

The script contains five branches:

1. **Initial call processing**

   This branch contains blocks from Check_ScriptID to Pass_to_Route_Point. All initial VCB-related user data is attached to the call, and a command is issued to GVP signaling that the call is arriving to URS. This is done in the Pass_to_Route_Point block, which distributes EventRouteRequest on the

Routing Point and waits for RequestApplyTreatment or RequestRouteCall from URS.

2. **Value of EWT, Callback Server availability, and license-availability check**

   This branch determines if Callback Service is available for a specified Routing Point by analyzing the VCB_STATUS key-value pair. If VCB_STATUS indicates that service is not available, the script analyzes the VCB_ERROR_CODE key-value pair to determine the exact reason for the error.

3. **Collection of data from the caller**

   Phone number, type of callback, scheduled date and time, and other data is collected via the INPUT blocks. Note the error-handling logic built into the data-collection algorithm. Blocks that collect the date and time for the Scheduled callback expect data in the following format:

   a. Date: MMDDYYYY or MMDDYY, for example, 12122003 or 121203. If year format is YY, then year is assumed by the sample script to be 20YY (see Validate_GSW_DATE_PART block VBScript code)

   b. Time: 24HHMM, for example, 0830, 1230, 1930.

4. **Database access**

   This branch accesses the database (block TZ_LOOKUP) if previous blocks were able to extract the three-digit area code from the phone number that the customer has left for the callback.

   a. Block TZ_LOOKUP performs the following SELECT statement:

      ```
      SELECT tz_name FROM timezones WHERE area_code = <extracted area
      code>.
      ```

   b. After executing the select statement, VBScript code accesses the resulting Recordset object in the Process_Anywhere() SUB.

   c. Note the single quotes in the SELECT statement that the TZ_LOOKUP block is using:

      ```
      SELECT [timezones].[tz_name] FROM [timezones] WHERE
      [timezones].[area_code] = '[{GSW_PHONE_PREFIX}]'
      ```

      These single quotes have been added manually because the type of area_code field is VARCHAR.

5. **Recording of caller's name to the VOX file and passing the path to that file to Universal Callback Server in the VCB_RECORDERED_DATA key-value pair**

   Note the VBScript code in the Capture_VOX_File block, which sets the VOX file name to the value that URS passes to GVP in the GSW_FILE_NAME key value pair. The full VOX file name can either be passed to Universal Callback Server as a UNC path or as a local path. Most likely your CPD Server is situated on another box than GVP, and you must use the UNC path. In that case, set the local variable bUseShareName in Put_VCB_RECORDERED_DATA block to TRUE, and set the script variable

GSW_SHARE_NAME to the name of the shared resource, is assigned to the folder where captured VOX files will reside.

Note the use of forward slashes (/) instead of the back slashes (\) in the file path.

URS Strategy

Your routing strategy must use Play application block to call the GVP script. APP_ID is the parameter that is passed to GVP and is mapped to the ScriptID script variable.

1. The GVP script is expecting ScriptID = 1 each time the call returns from the Routing Point to a GVP port.
   Therefore, set APP_ID to 1 in Play application block.
2. Select English (US) locale in this block.
3. Along with ScriptID, GVP script is expecting to receive the following key-value pairs from the Router Server:
   a. GSW_FILE_NAME with the string value of the unique file name to be used to record the customer name. As you can see in Figure 31, the ConnectionID attribute of the call is used to form a unique file name for each customer interaction.
   b. VCB_EWT with the string value in the format HHMM represents EWT. The GVP script uses the VCB_EWT pair to determine if Callback Service is offered to a caller or if the call is returned to the Routing Point for holding (block IF_EWT_greater_threshold). For simplicity, the sample strategy in Figure 31 on page 147 uses constant as a value for EWT, which is set to 21 minutes. In actual strategies, this value should be obtained from the Stat Server.

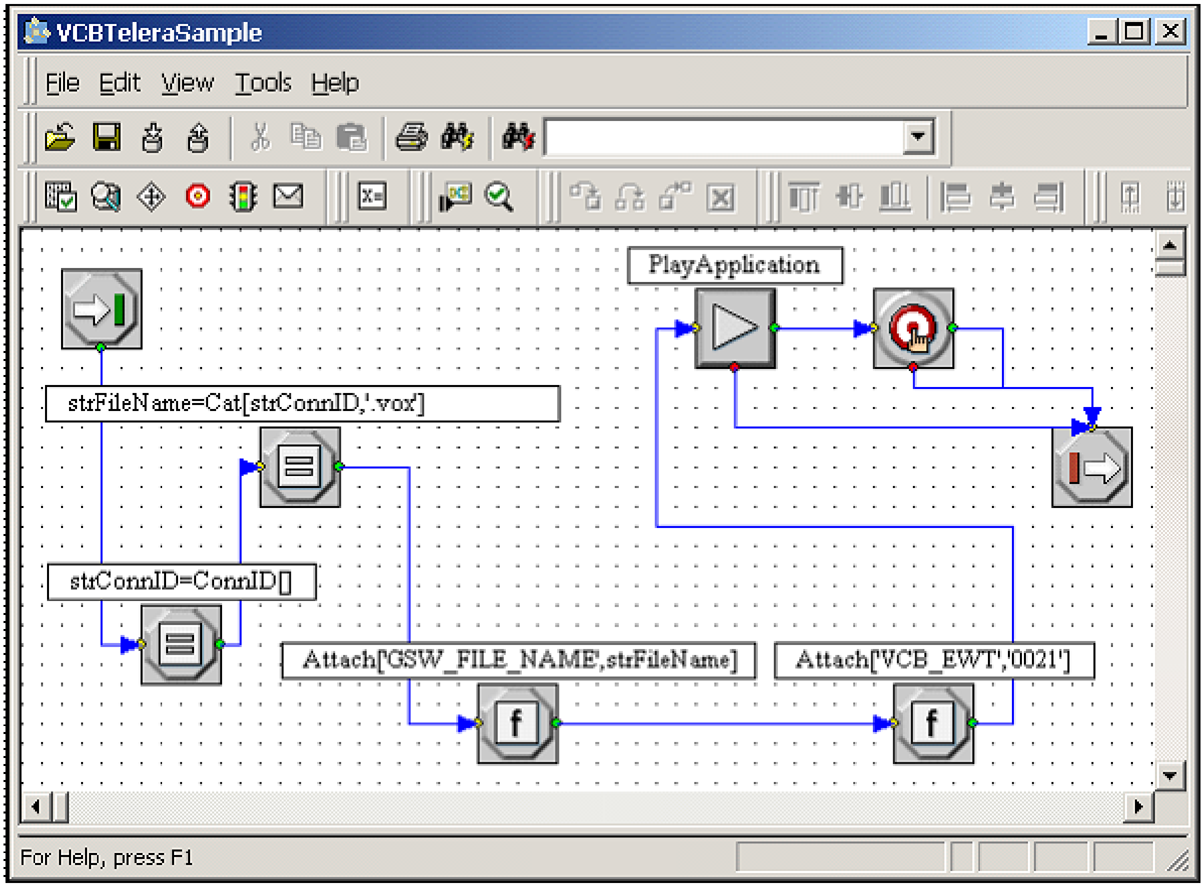

VCBTeleraSample
File Edit View Tools Help
PlayApplication
strFileName=Cat[strConnID,'.vox']
strConnID=ConnID[]
Attach['GSW_FILE_NAME',strFileName]
Attach['VCB_EWT','0021']
For Help, press F1

Figure 31: VCB Telera Sample

In addition, please note that you should load a separate strategy onto the VRP, where Universal Callback Server submits virtual calls representing callback requests (this VRP is determined by the option `vrp` in the configuration of the Routing Point). Figure 32 on page 148 illustrates a very simple strategy which you could use to pass callback requests to agents for processing.

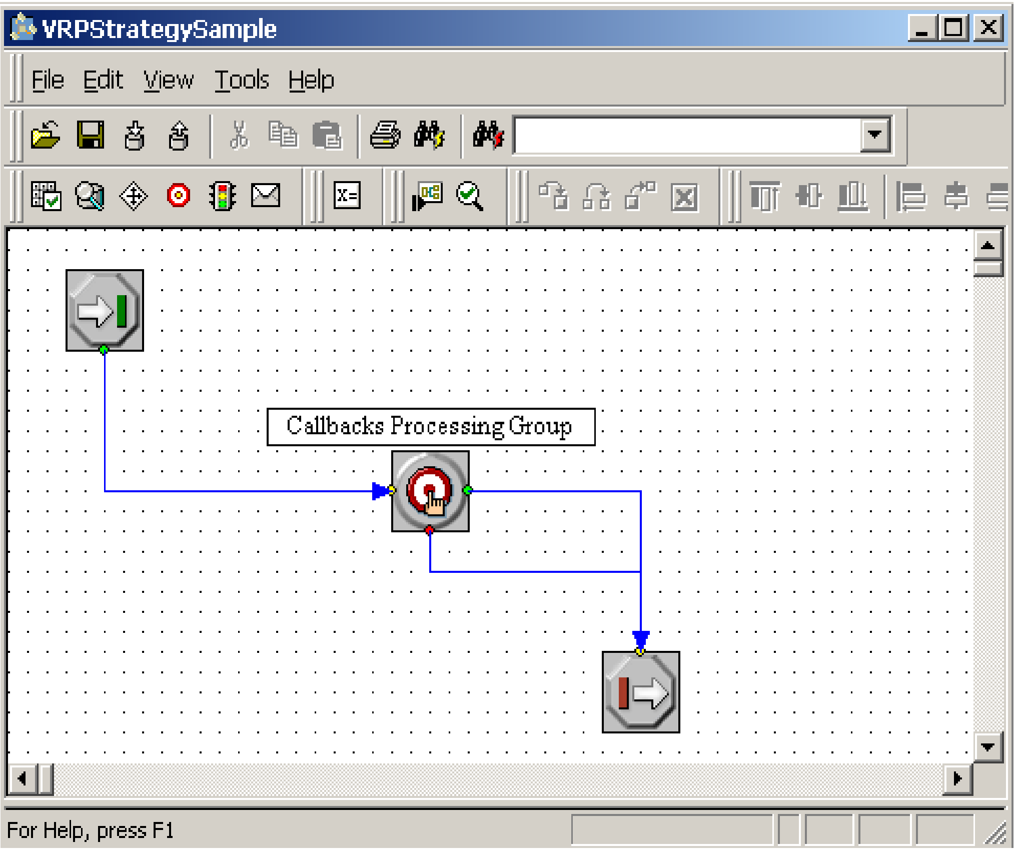


Figure 32: Routing Strategy on VRP

You do not have to load the strategy onto the VRP if you are using the default `callback routing` mode. Refer to the description of `default_routing_timeout` option for more information.

Script-Coding Recommendations

1. By default GVP assumes that the script will use ASR (Automatic Speech Recognition) resources. If ASR is not installed and/or not configured, an attempt to execute block `PROMPT` will yield the error, `Application that is not provisioned to use ASR is trying to access ASR resource`.

   To resolve this issue, specify that your script does not use ASR.

   a. Right-click `START` to access the `Start Properties` dialog box.

   b. Select the `RootMap` tab in the `Start Properties` dialog box and click `More`.

      The `Property Settings` dialog box opens.

   c. Click `Add/Edit` on the `Property Settings` dialog box.

      The `Name/Value Pair Settings` dialog box opens.

d. In the Name field on the Name/Value Pair Settings dialog box, select com.telera.speechenabled.

e. Set the Value field to false.

f. Click OK on the Name/Value Pair Settings dialog box.

You are returned to the Property Settings dialog box where you see your selection displayed.

g. Click Close.

You are returned to the Start Properties dialog box.

h. Click OK on the Start Properties dialog box.

2. Each block available in Genesys Studio contains three functions that can be used to customize the block:

- Process_Begin()
- Process_Anywhere()
- Process_End()

To access these functions:

- Click the Edit button (adjacent to label Edit custom code) on the Standard tab of the block's Properties dialog box.

To understand when these functions are called, compile the project with the selected block and then study the resulting ASP (or JSP) page source code. Process_Begin() and Process_End() are called at the beginning and the end of the page processing respectively. For the function Process_Anywhere(), check the resulting ASP (JSP) code because the actual place of the function invocation varies from block to block.

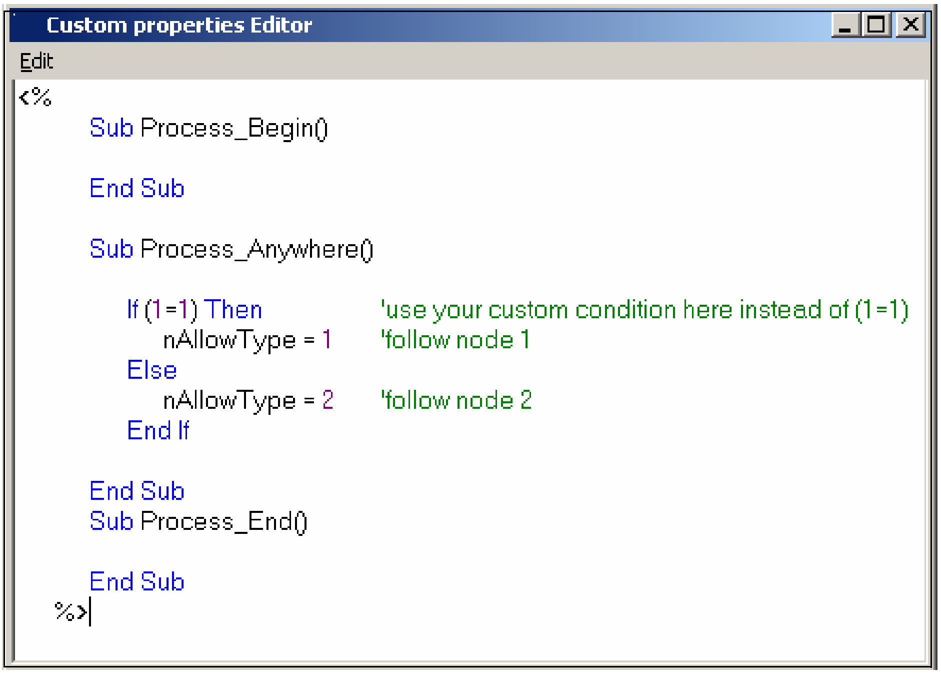
3. To implement branching in your code, you can use the following two approaches:

a. Use the Branching block, which allows you to branch based on the specific values of the variable.

Example:

- If my_flag = 0, go to outport 1.
- If my_flag = 1, go to outport 2.
- If my_flag = 2, go to outport 3.
- Otherwise, go to outport Continue.

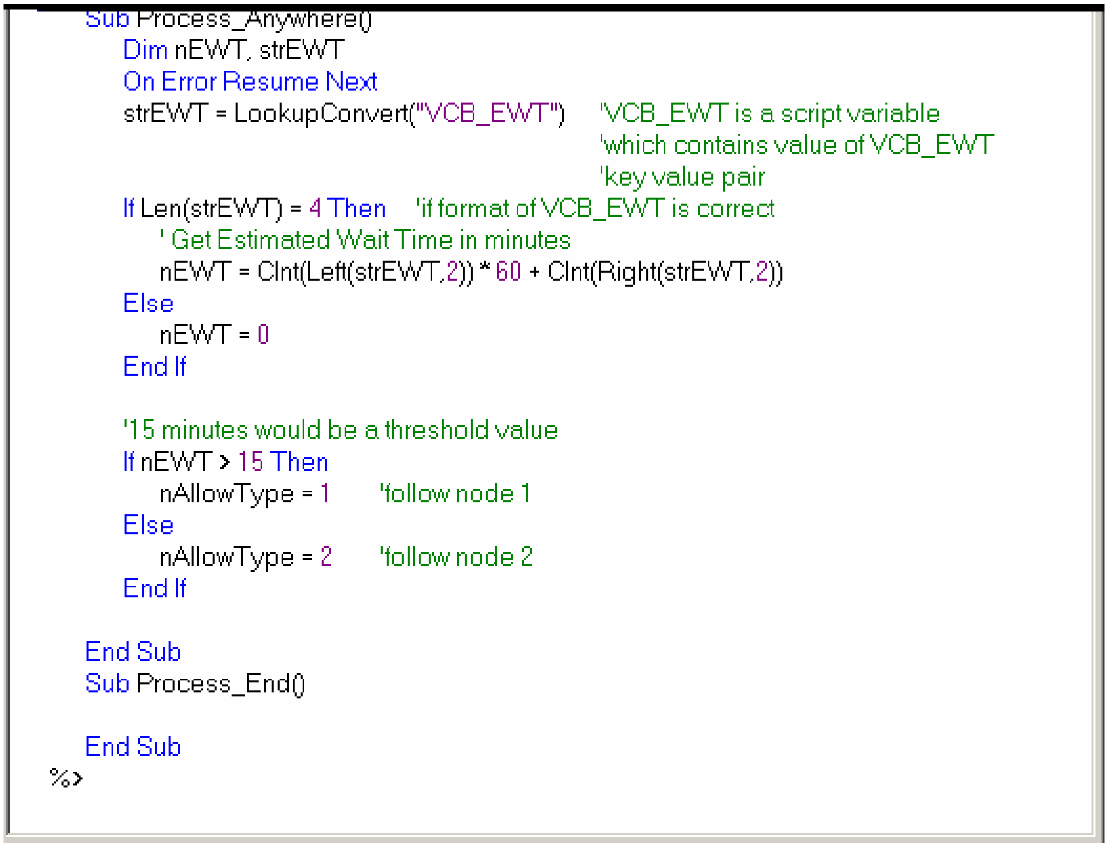
b. Use the CODE block to have more control over your branching logic (for example, if you need to branch based on Boolean expressions other than equality). To do this, specify as many outports as you need in the block's Properties dialog box. Then write your own code in the Custom Code section, using nAllowType variable as a branch selector. See the example code in Figure 33 on page 150.



```
<%
      Sub Process_Begin()

      End Sub

      Sub Process_Anywhere()

         If (1=1) Then           'use your custom condition here instead of (1=1)
            nAllowType = 1       'follow node 1
         Else
            nAllowType = 2       'follow node 2
         End If

      End Sub
      Sub Process_End()

      End Sub
   %>
```

Figure 33: Sample VB Script code in CODE block

4. To obtain the key-value pair from the user data of the call being processed, define the script variable using the same name as the key in the key-value pair. Use block `Start > Application Properties` tab for that. Once you have defined the variable, use the `UserData` block (operation `get`) to assign value from the key-value pair to the variable. To manipulate this variable's value in code, use the `LookupConvert()` method. See Figure 34 on page 151 for an example.



```
    Sub Process_Anywhere()
        Dim nEWT, strEWT
        On Error Resume Next
        strEWT = LookupConvert("VCB_EWT")     'VCB_EWT is a script variable
                                              'which contains value of VCB_EWT
                                              'key value pair
        If Len(strEWT) = 4 Then    'if format of VCB_EWT is correct
            ' Get Estimated Wait Time in minutes
            nEWT = CInt(Left(strEWT,2)) * 60 + CInt(Right(strEWT,2))
        Else
            nEWT = 0
        End If

        '15 minutes would be a threshold value
        If nEWT > 15 Then
            nAllowType = 1      'follow node 1
        Else
            nAllowType = 2      'follow node 2
        End If

    End Sub
    Sub Process_End()

    End Sub
%>
```

Figure 34: Example of LookupConvert() function usage

Important The approach, which is demonstrated in the IVR sample described, is not the only possible implementation of the Universal Callback Server and the GVP bundle. Since this approach can be used for both `in-front-of-the-switch` and `behind-the-switch` modes, it assumes that when a call arrives at the IVR ports it remains there during all stages of the callback-request processing. An alternative approach involves “bouncing” the call between the IVR port and the Routing Point. In such a scenario, the call first arrives at the Routing Point (or ACD Queue), and then it is distributed to the IVR port. With all the required callback-request data collected and attached to the call, the IVR script transfers the call back to the Routing Point, where the Universal Callback Server processes the call. Then the call is passed back to the IVR port from the Routing Point for the callback-request submission. Such a scenario is more typical for a `behind-the-switch` configuration. If you intend to use this “bouncing” approach, please keep in mind the following:

1. Because the call is physically passed to the Routing Point, Universal Callback Server, which monitors the Routing Point, reacts on `EventRouteRequest` (or `EventPartyChanged` in the case of a two-step or mute transfer to the Routing Point). This is configured by setting the Application-wide or the Routing Point-specific `urs_treatment_mode` option to the `transfer` value.

2. To pass the call to the Routing Point from the IVR script, use block `CONNECT` (which performs CTI transfer) instead of the `RoutePoint` and `Treatment_Result` pair of blocks.
3. The IVR script cannot branch based on the `ScriptID` and should use some other variable initialized from the `UserData` of the call instead.

Additional Resources

- *Genesys Voice Portal Genesys Studio Developers Guide*
- *Genesys Voice Portal Reference Manual*
- *Genesys Studio Help System* (press F1 in Studio for this help)
- *Technical Bulletin: GVP and URS Interactions*
- *Universal Routing Reference Manual*
- *Voice Callback Reference Manual*
- *IVR Interface Option IVR Server System Administrators Guide*
- VB Script documentation at `http://www.microsoft.com/scripting.`
- ASP documentation at `http://msdn.microsoft.com/library/default.asp?url+/nhp/default.asp?contentid=28000522.`

Chapter

9 Installing Dialogic Boards

This chapter describes how to install Dialogic boards and contains the following sections:

Introduction to Dialogic Boards

If you have CPD Server installed, you must first set up Dialogic boards and then install a hardware card and a software driver. The sections below show where to find a list of supported Dialogic boards. See “Supported Types of Dialogic Board” on page 167.

For more information about Dialogic boards and supporting software, contact the Dialogic vendor for your company or visit the Intel support website noted in the next section.

Support for Dialogic Boards and Software

Support for installing or configuring Dialogic boards or driver software is available through your Dialogic vendor or Intel Support at this website: `http://developer.intel.com/design/telecom/support/`.

Setting Up Dialogic Boards

To Install Dialogic boards

1. Make sure the operating system is installed (Windows 2000, Windows XP, or Windows NT 4.0 and Service Pack 4).
2. Install Dialogic SR 5.1.1.
3. Install Service Pack 1 (SP1).
4. During installation, select drivers, firmware, and configuration files in the custom package.
5. Install Dialogic boards (see the Note below about installing Dialogic boards).
6. Start Dialogic Configuration Manager.

   Dialogic Configuration Manager should automatically detect the board and locate valid `address/port/interrupt` values for the particular machine.
7. Start Dialogic drivers:

   SR5.1.1

   Verify the software release number for the Dialogic drivers. See "Support for Dialogic Boards and Software" on page 153.
8. Use the Dialogic application provided with the Dialogic board to test each port for `off-hook`, dialing a telephone number, and then `on-hook`.
9. Make sure that each port can complete this test before installing and configuring CPD Server.

When installing Dialogic boards, make sure that the board number set by the switch on the board is unique inside each board type. If you have two T1 boards, the first one must be 0, the next 1, and so on. Make sure the SCBus/CTBus cable connects the boards.

Note: Error messages display in the Windows NT `Event Viewer` if errors occur during startup of the Dialogic drivers. Restart Dialogic Configuration Manager or reinstall the Dialogic boards. If these steps do not solve the problem, contact Intel support at `http://developer.intel.com/design/telecom/support/.`

Configuring Dialogic Boards

Outbound Contact Configuration Wizard places the information about installed Dialogic boards and resources in Configuration Layer. You can use the Configuration Manager to change the configurations as needed. See “Configuration Structure for a Dialogic Board” on page 157.

Frequently Used Configurations

Table 10 lists frequently used configurations for Dialogic boards. Support for the DM3 families of boards is implemented.

Note: Driver names are given below for a single board configuration. For multiple boards, the corresponding board numbers must be unique; that is, the driver names must be sequentially numbered.

Table 10: Dialogic Board Configurations

Board	Type	Configuration
D/160SC-LS D/160JCT-LS (also for 6.5.2)	Virtual Board	4 LSI (dxxxB1...dxxxB4)
	Virtual Device	4 channels on each board (dxxxBxC1...dxxxBxC4)
	Voice Virtual Board	4 voice boards (dxxxB1...dxxxB4)
	Voice Virtual Device	4 channels on each board (dxxxBxC1...dxxxBxC4)
D/41ESC	Virtual Board	4 LSI (dxxxB1...dxxxB4)
	Virtual Device	4 channels on each board (dxxxBxC1...dxxxBxC4)
	Voice Virtual Board	4 voice boards (dxxxB1...dxxxB4)
	Voice Virtual Device	4 channels on each board (dxxxBxC1...dxxxBxC4)
D/41EPCI	Virtual Board	4 LSI (dxxxB1...dxxxB4)
	Virtual Device	4 channels on each board (dxxxBxC1...dxxxBxC4)
	Voice Virtual Board	4 voice boards (dxxxB1...dxxxB4)
	Voice Virtual Device	4 channels on each board (dxxxBxC1...dxxxBxC4)

Table 10: Dialogic Board Configurations (Continued)

Board	Type	Configuration
D/240SC-T1 D/240SC-T1	Virtual Board	1 DTI board (dtiB1)
	Virtual Device	24 channels on each board (dtiBxT1...dtiBxT24)
	Voice Virtual Board	6 voice boards (dxxxB1...dxxxB6)
	Voice Virtual Device	4 channels on each board (dxxxBxC1...dxxxBxC4)
D/240SC-2T1 D/240JCT-T1	Virtual Board	2 DTI boards (dtiB1, dtiB2)
	Virtual Device	24 channels on each board (dtiBxT1...dtiBxT24)
	Voice Virtual Board	6 voice boards (dxxxB1...dxxxB6)
	Voice Virtual Device	4 channels on each board (dxxxBxC1...dxxxBxC4)
D/480SC-2T1 D/480JCT-2T1	Virtual Board	D/480SC-2T1
	Virtual Device	2 DTI boards (dtiB1, dtiB2)
	Voice Virtual Board	24 channels on each board (dtiBxT1...dtiBxT24)
	Voice Virtual Device	12 voice boards (dxxxB1...dxxxB12), 4 channels on each board (dxxxBxC1...dxxxBxC4)
D/300SC-E1 D/300JCT-E1	Virtual Board	1 DTI board (dtiB1) 4 channels on boards 1 to 7
	Virtual Device	30 channels (dtiBxT1...dtiBxT30)
	Voice Virtual Board	8 voice boards (dxxxB1...dxxxB8)
	Voice Virtual Device	2 channels on board 8
D/300SC-2E1 D/300JCT-2E1	Virtual Board	2 DTI boards (dtiB1, dtiB2)
	Virtual Device	30 channels on each board (dtiBxT1...dtiBxT30)
	Voice Virtual Board	8 voice boards (dxxxB1...dxxxB8)
	Voice Virtual Device	4 channels on boards 1 to 7, 2 channels on board 8

Table 10: Dialogic Board Configurations (Continued)

Board	Type	Configuration
D/600SC-2E1 D/600JCT-2E1	Virtual Board	2 DTI boards (dtiB1, dtiB2)
	Virtual Device	30 channels on each board (dtiBxT1...dtiBxT30)
	Voice Virtual Board	16 voice boards (dxxxB1...dxxxB16)
	Voice Virtual Device	4 channels on boards 1 to 7 and 9 to 15; 2 channels on boards 8 and 16

Configuration Structure for a Dialogic Board

The structure of Dialogic board configuration is represented by a folder tree under the `Switch/DN` folder in Configuration Manager. This tree has the following folders:

Folders

- `<location>`

  This folder is taken from the `location` option (`default` section) and serves as the board root folder. When you delete a board, the Wizard deletes this folder and everything under it. The current Wizard implementation does not allow the deletion of only a part of the board; the whole board is deleted.

- `regular`

  This folder contains the channel-DN configuration for dialing callbacks.

- `recording`

  This optional folder contains the channel-DN configuration for CPD recording on the regular DNs. Recording is used to tune up Voice/AM detection. Do not use it in regular work because it consumes half of the voice resources. The Wizard creates this folder but does not configure it.

- `dxxxB<n>`

  This is the root folder for an analog virtual (logical) board. It contains voice resources (channels). It may have one or more voice resources.

- `dxxxB<n>C<m>`

  This is a voice resource (channel). In the configuration, it links a Dialogic voice resource and a DN on the switch.

- `dtiB<n>`

  This folder contains one or more digital resources and serves as the root channel for a network (T1) virtual board.

- `dtiB<n>C<m>`

  This is a digital (network) resource. In the configuration, it links a Dialogic digital (network) resource and a DN on the switch.

Note: Folders for non-analog line types representing `dxxx` channels do not have to include real DNs. CPD Server 7.0 works without DNs in `dxxx` folders and uses CPD on non-analog lines. Thus, multiple licenses are not required to allow for two DNs per channel and a subsequent increase in dialing ports.

Configuration of a Logical Structure

A *logical structure* means that within a configuration of boards, every board is given a unique number based on the order in which the boards are configured.

Genesys recommends a 1-to-1 relationship between the Windows NT CPU, CPD Server, and the `Location` folder. See Figure 35.

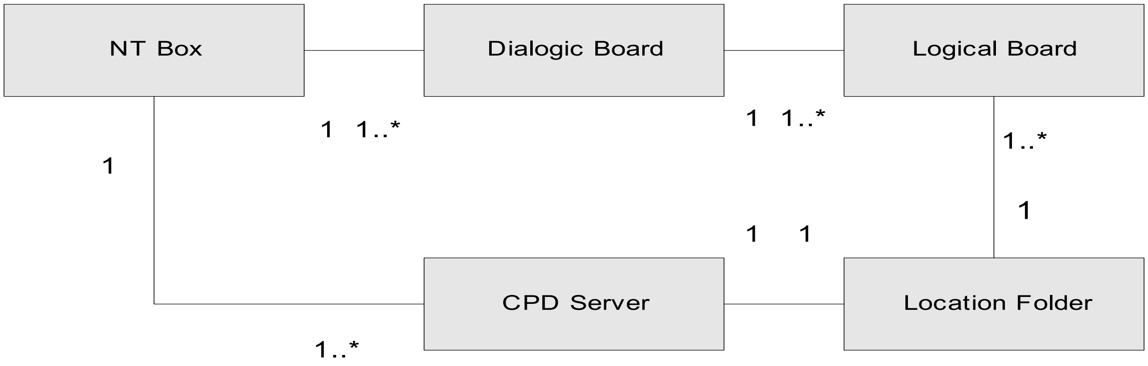


Figure 35: Logical Structure

Note: The number 1 in Figure 35 shows a 1-to-1 configuration, and 1...* represents a 1-to-more-than-1 configuration.

Windows NT Box Represents a Windows NT CPU (central processing unit of a computer running on the Windows NT operating system), where the Dialogic boards and CPD Servers are installed.

Dialogic Board Represents a hardware board installed into the Windows NT box. The box may have more than one board installed.

Logical Board A logical part of the hardware board representing four analog or digital resources (for example, `dxxxB1` and all the hierarchy belonging to it). One hardware board consists of several logical boards. One logical board belongs to only one hardware board.

CPD Server Represents a piece of software installed on the box. It can be one or more CPD Server installations.

Location Folder Represents a set of logical boards and one CPD Server. A `Location` folder can include logical boards connected to different hardware boards (see the following example).

Note: Genesys recommends that you set up a 1-to-1 association between CPD Server Applications and the `Location` folder.

Example

- The `Location` folder may have logical boards belonging to hardware boards installed in one, and only one, computer running on Windows NT.
- If you have two or more different `Location` folders, they must not use the same logical board.
- If you have more than one hardware board installed on the same computer:
  - The first board (minimal hardware ID) has `1...n` logical board numbering (`dxxxB1...dxxxBn`),
  - The second board (next ID) has `n+1...m (m > n+1)` logical board numbering (`dxxxB<n+1>...dxxxBm`), and so on.

  If a logical board is removed from a configuration, then all other logical boards in the configuration must be renumbered to preserve a logical numbering system.

CPD Server Dialogic Hardware Setup

The following procedure provides guidelines for setting up your Dialogic hardware in Configuration Manager folders.

To Set Up Dialogic Hardware

1. Select `Configuration` > `Switch DN` folder.
2. Create the `Location` folder in the `Switch DN` folder.
   a. Right-click on the `Switch DN` folder to open the `New Folder` dialog box.
   b. In the Name field on the `General` tab of the `New Folder` dialog box, enter new folder name: `Location`.
   c. Click `OK`.
3. Create new subfolders by right-clicking on the `Location` folder to access the `New Folder` dialog box.
   a. On the `General` tab of the `New Folder` dialog box, enter the names of these board-resource subfolders:
      - `recording` (containing only voice resources)
      - `regular` (containing any type of Dialogic resource)

Note: These subfolder names should be the same as the virtual board names used for the Dialogic hardware that is installed.

4. In each of these subfolders just created, create another group of board-resource subfolders named exactly the same as the individual channels of the corresponding virtual board.
5. In each channel subfolder, include the real DN (extension) or virtual DN (call processing port) that corresponds to the DN assigned to the board channel.
   - For LSI and T1/E1 ports, assign `Extension DN` type and real channel numbers that the switch can recognize.

Note: `Position` is not a valid value for CPD Server DNs.

6. Clear the `Register` check box for CPP (Call Processing Port) and select the check box for ACD Extension DNs.

Dialogic Board Configuration Examples

The following figures show examples you can use to set up Dialogic hardware in the `Switch` folder for `Analog` or `T1/E1` boards.

Analog Board

Analog Board Directory Structure

An `Analog` board configuration must have the following folders and subfolders in this type of directory structure:

- A `regular` folder
  - At least one `dxxxB<n>` folder within the `regular` folder.
    - At least one `dxxxB<n>C<n>` subfolder within the `dxxxB<n>` folder.

Figure 36 on page 161 shows the directory structure for an `Analog` configuration.

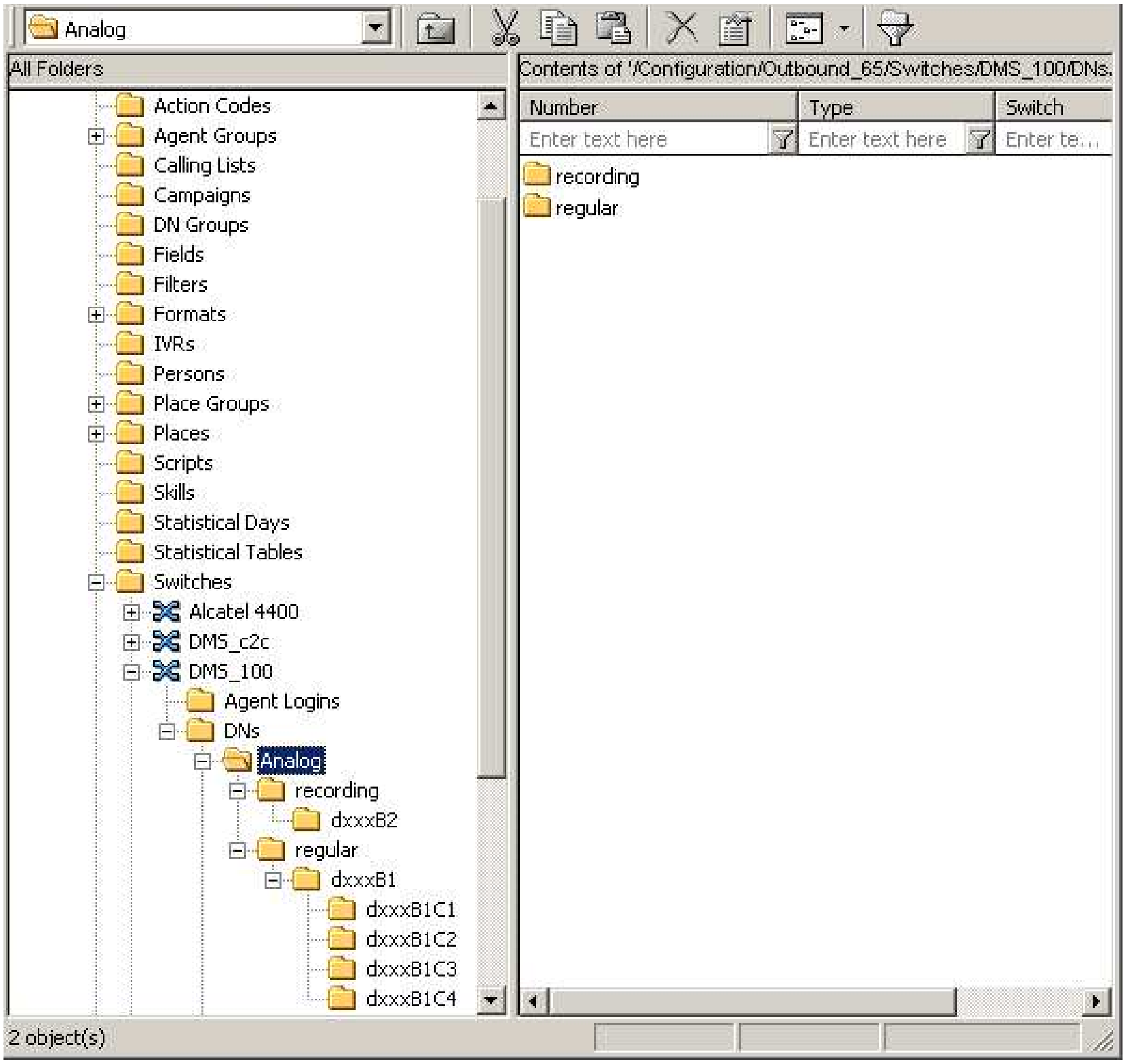
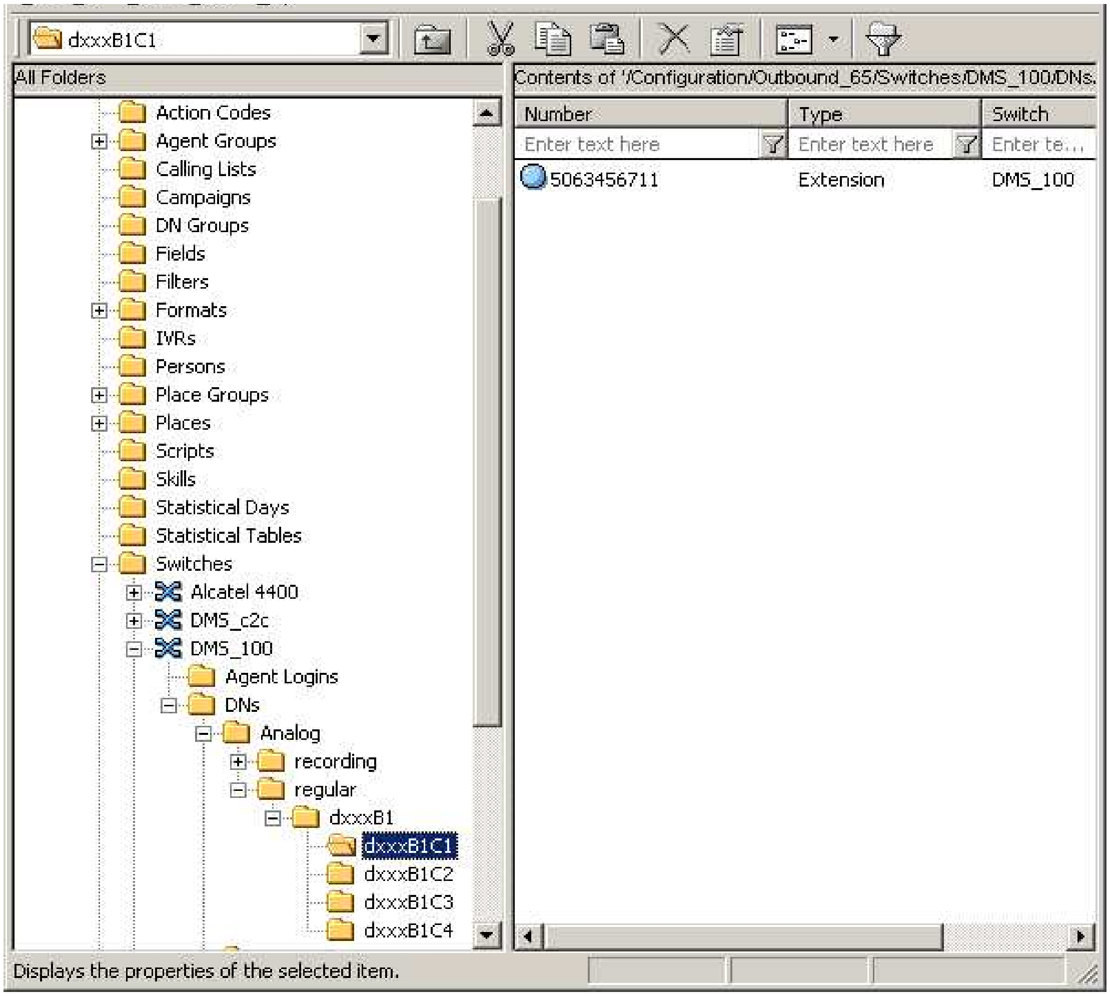


Figure 36: Analog Directory Structure

Figure 37 shows an `Extension DN` in an `Analog` configuration.



Contents of '/Configuration/Outbound_65/Switches/DMS_100/DNs.

Number	Type	Switch
5063456711	Extension	DMS_100

Displays the properties of the selected item.

Figure 37: Extension DN in Analog Configuration

For more information, see “CPD Server Dialogic Hardware Setup” on page 159.

T1/E1 Board

Sample DN Configuration for T1/E1 Board

A lineside configuration must have a least one digital and one voice logical board configured in the `regular` folder. See Figure 38.

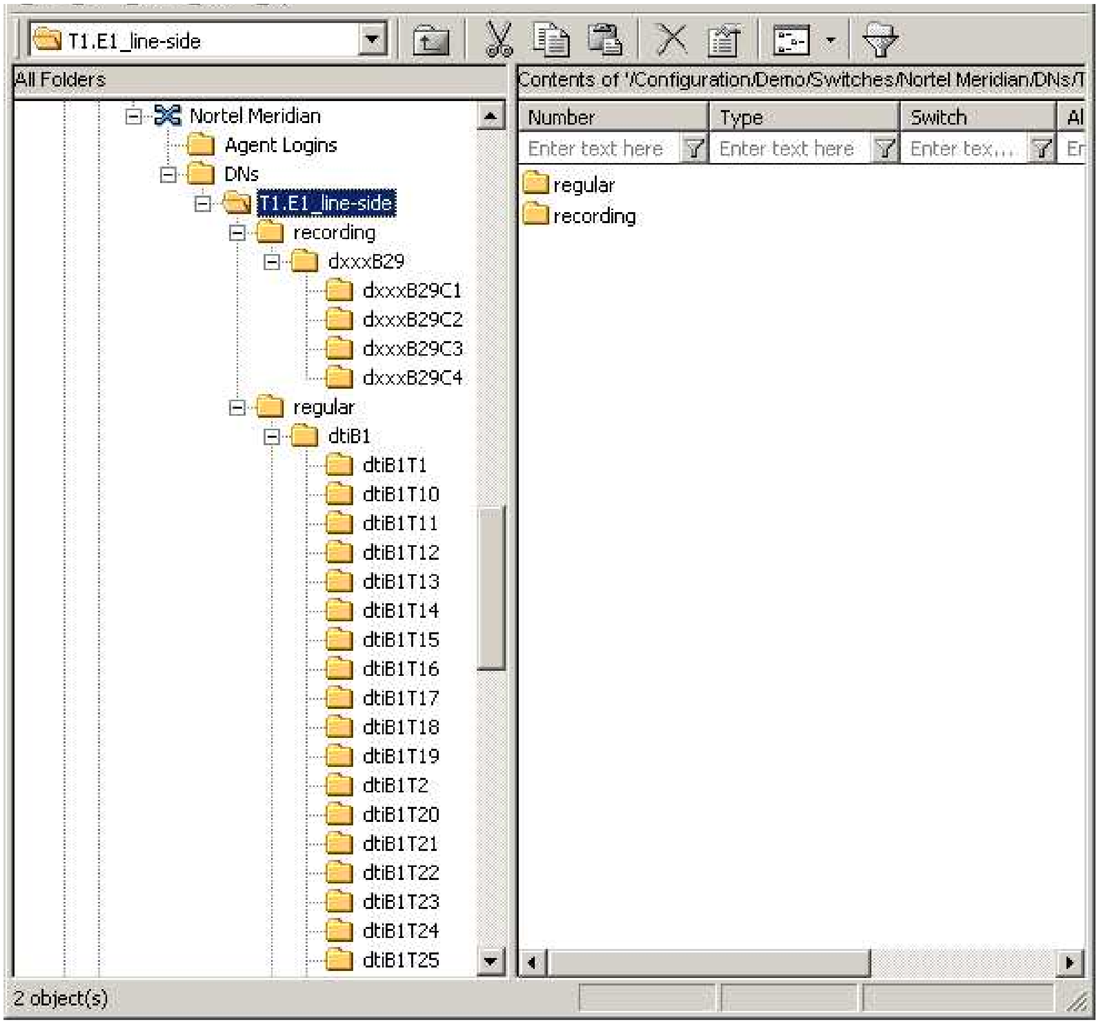
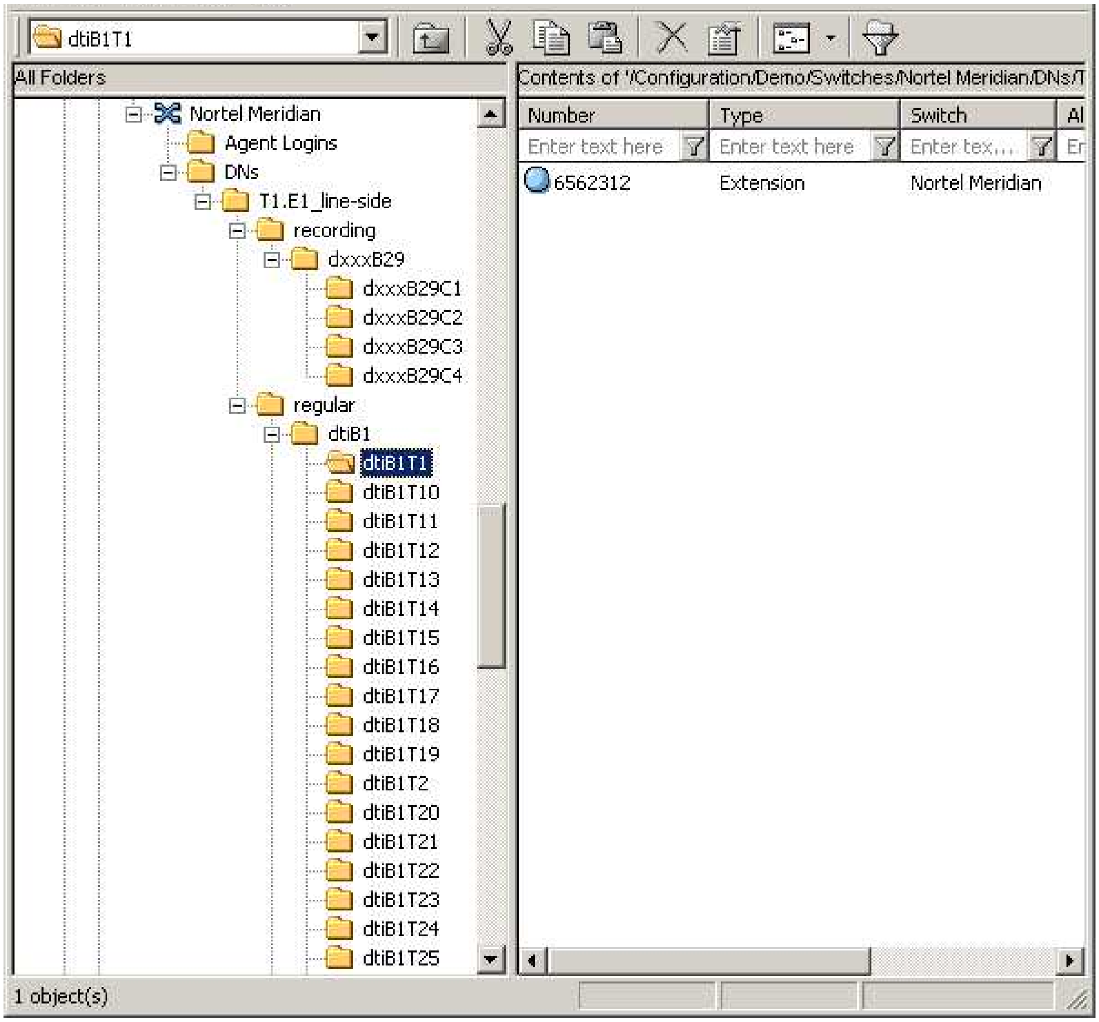


Figure 38: T1/E1 (Lineside) Directory Structure

In the lineside configuration, each network resource must have one, and only one, DN associated with it. Network resources are associated with DNs of type `Extension`.

Figure 39 shows a sample configuration for T1/E1 (lineside) DN.



Contents of '/Configuration/Demo/Switches/Nortel Meridian/DNs/T

Number	Type	Switch	Al
Enter text here	Enter text here	Enter tex...	Er
6562312	Extension	Nortel Meridian	

Figure 39: Extension DN in T1/E1 (Lineside) Configuration

Diagram of Dialogic Board Configuration

The following figures show sample configurations for `Analog` or `T1/E1` protocols.

Analog

Figure 40 shows a configuration example for an Analog board.

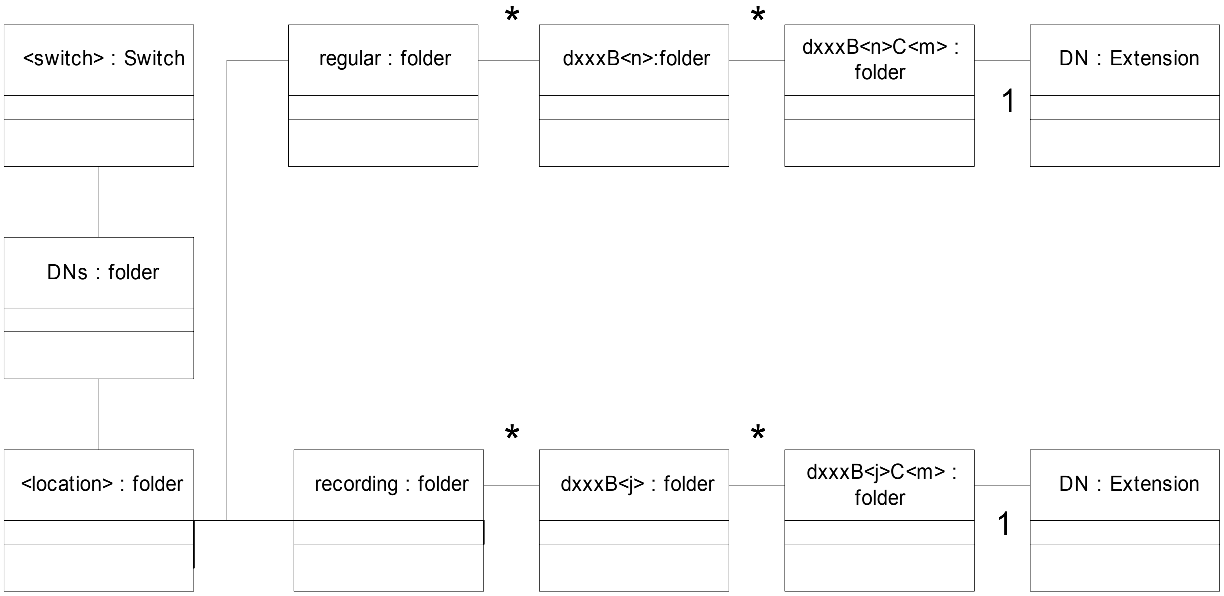


Figure 40: Configuration Example for an Analog Board

Note: The only DN type for an Analog board or a T1/E1 board is `Extension`, not `Position`.

T1/E1

A lineside configuration must have at least one digital and one voice logical board configured in the `regular` folder.

In the lineside configuration network, associate network resources with DNs of `Extension` type. Figure 41 shows a configuration example for a T1/E1 board.

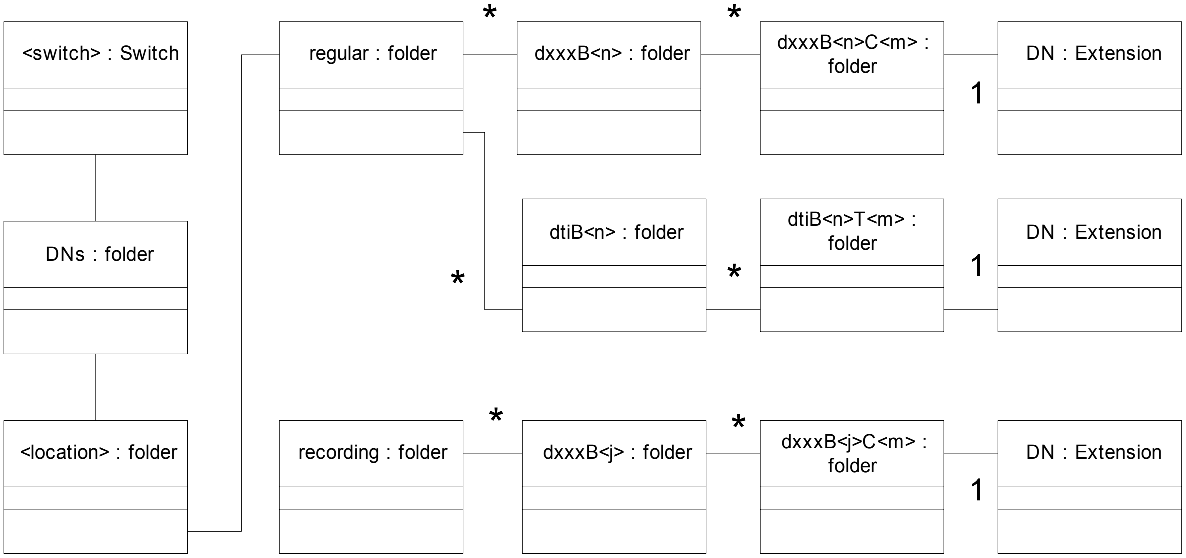


Figure 41: Configuration Example for a T1/E1 Board

Note: Associate one and only one DN with each network resource.

DNs Inside Channels Folders for Dialogic Boards

For more information about the dialogic boards and their channels, see “Supported Types of Dialogic Board” on page 167.

Configuring a Dialogic Board

To Configure a New Dialogic Board

1. In Configuration Manager, select `Configuration` > `Switch`.
2. Create a `<location>` folder under the `Switch` folder.
   a. Right-click on the `Switch` folder and select `New > Folder` to open the `New Folder` dialog box.
   b. Select the `General` tab.
   c. In the `Name` field on the `General` tab, enter the folder name,
   d. Click `OK`.
3. Under the `<location>` folder, create these folders:

   a. `recording`
   b. `engaging`
   c. `regular`

   For a description of this folder, see “CPD Server Dialogic Hardware Setup” on page 159.

4. Configure the `regular` folder.

The Wizard will not allow you to create an incomplete configuration (see “Dialogic Configuration Structure” on page 157). If you abort the configuration, the Wizard removes the incomplete configuration.

Resources Configuration Activity

The Wizard lists supported board types for selection. As more than one board may be installed, the Wizard allows selection of a few boards at a time, instead of configuring one-by-one.

When configuring a virtual board, the Wizard assigns a DN to every channel of the board. The Wizard assigns the DNs in sequence, one-by-one.

Note: Pay attention to the following constraint: A DN can belong to one, and only one, channel across the switch.

Deleting a Previously Configured Dialogic Board

If you are creating a new configuration, you must first delete the previously configured Dialogic boards.

To Delete a Configured Board

1. When the Wizard lists the configured boards, select the boards you want to delete.
2. If a board is associated with a CPD Server application, then the Wizard warns you and asks for a confirmation.
3. If the Wizard receives a confirmation, it clears the value of the `location` option in all Applications to match the board name.

Supported Types of Dialogic Board

The following list shows the supported types of Dialogic boards.

D/240SC-T1

- 1 DTI board (`dtiB<x>`), 24 channels on each board(`dtiB<x>T1...dtiB<x>T24`)
- 6 voice boards (`dxxxB<x>...dxxxB<x+5>`), 4 channels on each board (`dxxxB<x>C1...dxxxB<x>C4`)

D/240SC-2T1

- 2 DTI boards (`dtiB<x>, dtiB<x+1>`), 24 channels on each board (`dtiB<x>T1...dtiB<x>T24`)
- 6 voice boards (`dxxxB<x>...dxxxB<x+5>`), 4 channels on each board (`dxxxB<x>C1...dxxxB<x>C4`)

D/480SC-2T1

- 2 DTI boards (`dtiB<x>, dtiB<x+1>`), 24 channels on each board (`dtiB<x>T1...dtiB<x>T24`)
- 12 voice boards (`dxxxB<x>...dxxxB<x+11>`), 4 channels on each board (`dxxxB<x>C1...dxxxB<x>C4`)

D/300SC-E1

- 1 DTI board (`dtiB<x>`), 30 channels (`dtiB<x>T1...dtiB<x>T30`)
- 8 voice boards (`dxxxB<x>...dxxxB<x+7>`), 4 channels on boards `<x>-<x+6>`, 2 channels on board `<x+7>`

D/300SC-2E1

- 2 DTI boards (`dtiB<x>`), 30 channels on each (`dtiB<x>T1...dtiB<x>T30`)
- 8 voice boards (`dxxxB<x>...dxxxB<x+7>`), 4 channels on boards `<x>-<x+6>`, 2 channels on board `<x+7>`

D/300SC-2E1

- 2 DTI boards (`dtiB<x>`), 30 channels on each (`dtiB<x>T1...dtiB<x>T30`)
- 8 voice boards (`dxxxB<x>...dxxxB<x+7>`), 4 channels on boards `<x>-<x+6>`, 2 channels on board `<x+7>`

D/600SC-2E1

- 2 DTI boards (`dtiB<x>`), 30 channels on each board (`dtiB<x>T1...dtiB<x>T30`)
- 16 voice boards (`dxxxB<x>...dxxxB<x+15>`), 4 channels on boards <x>-`<x+6>` and `<x+8>-<x+14>`, 2 channels on
- boards `<x+7>` and `<x+15>`

D/41ESC

- - 1 LSI/voice board (`dxxxB<x>`), 4 channels on board

D/160SC

- - 4 LSI/voice boards (`dxxxB<x>...dxxxB<x+1>`), 4 channels on each board

Utilities

Use the following utility to resolve problems that might occur when you are using Dialogic boards.

Dialogic Configuration Manager

Dialogic Configuration Manager (DCM) is part of the Dialogic software package.

Make sure that you use this utility when you are configuring Dialogic boards for your system.

To Access DCM from Start Menu

- Select `Programs > Intel Dialogic System Software > Dialogic Configuration Manager·DCM`.

Chapter

10 Starting and Stopping Procedures

This chapter explains how to start and stop VCB with the Solution Control Interface (SCI) and how to start and stop VCB and CPD Server manually on Windows. It contains the following sections:

Preliminary Information

Prestart Information

- Before starting VCB, the following Framework components must be running:
  - Configuration Server
  - DB Server (for Configuration Server)
  - T-Server(s)
  - Stat Server(s)

  See *Framework Deployment Guide.*
- VCB is an option of Universal Routing, which includes Enterprise Routing (ER) and Network Routing (NR).
  - ER or NR must be running before you can start the Universal Callback Server, which is the main component of VCB.

  See the *Universal Routing Deployment Guide*.

- Universal Contact Server (if used) must be installed before starting VCB. See *Multi-Channel Routing Event Media Deployment Guide*.
- The installation of the next three components are optional.
  - Solution Control Server
  - SCI
  - Local Control Agent (on the host where Universal Callback Server is installed)

  However, they must be running
    - If you are using the Universal Contact Server.
    - If you want to start VCB with the Solution Control Interface (SCI).
    - If Universal Contact Server is used with Web API Server.
- Universal Callback Server
- Web API Server can be started after starting VCB. See *Multi-Channel Routing Event Media Deployment Guide*.

Note: If you will be using the CPD Server, install all Dialogic components on the same computer as CPD Server before starting CPD Server.

Using SCI to Start Servers

There is only one way to start all servers using SCI:

To Start All Servers

- Select `Start` from the `Action` menu in SCI.

To Start CPD Server Using SCI

1. Select the CPD Server component in the `Applications` folder.
2. Select `Start` from the `Action` menu.

To Start Universal Callback Server in SCI

1. Select the Universal Callback Server component in the `Applications` folder.
2. Select `Start` from the `Action` menu.

Starting Components Manually

To start VCB manually, components are started in a specific order. For information about starting the Framework components DB Server, Configuration Server, T-Server, and Stat Server, see the *Framework Deployment Guide.* For information about starting Universal Contact Server and Web API Server, see *Multi-Channel Routing Event Media Deployment Guide.*

To Start Components in a Specific Order

1. DB Server for Configuration Server
2. Configuration Server
3. T-Server
4. Stat Server
5. DB Server(s) (for `Calling List` table and `Black List` table)
6. CPD Server
7. Universal Contact Server
8. Universal Callback Server
9. Web API Server

To Start Universal Callback Server

Start VCB either from the `Start` menu, or open an MS-DOS command prompt and type the directory where the Universal Callback Server is installed, for example, `\Gcti\Voice Callback\`, and use a command line. When starting Universal Callback Server from a command line, use the following common command-line parameters:

`-host` Name of the host where Configuration Server is running

`-port` Configuration Server port

`-app` Exact name of the Application as configured in the Configuration Database

`-l` License file

For example:

```
ucbserver.exe -host ConfigServer1 -port 2020 -app UCServer1 -l
C:\flexlm\license.dat
```

To Start CPD Server

Start CPD Server either from the `Start` menu, or open an MS-DOS command prompt and type the directory where CPD Server is installed, for example, `\Gcti\cpd_server\`, and use a command line. When starting CPD Server from a command line, use the following common command-line parameters:

`-host` Name of the host where Configuration Server is running

`-port` Configuration Server port

`-app` Exact name of the Application object as configured in the Configuration Database

For example:

```
cpdserver.exe -host ConfigServer1 -port 2020 -app CpdServer1 -l
C:\flexlm\license.dat
```

Stopping Components in Order

To Stop Components in Appropriate Order

1. Universal Callback Server
2. CPD Server (if used)
3. DB Server for `Calling List` table
4. Stat Server
5. Universal Contact Server
6. T-Server
7. Configuration Server
8. DB Server for Configuration Server

Using SCI to Stop Servers

There is only one way to stop any server using SCI:

To Stop Any Server

1. Select the server in the `Applications` folder.
2. Select `Stop` from the `Action` menu in SCI.

To Stop Universal Callback Server in SCI

1. Select the Universal Callback Server component in the `Applications` folder.
2. Select `Stop` from the `Action` menu.

To stop CPD Server using SCI

1. Select the CPD Server component in the `Applications` folder.
2. Select `Stop` from the `Action` menu.

Stopping Servers Manually

This section provides the step-by-step procedures for manually stopping Universal Callback Server and CPD Server.

Universal Callback Server

To Stop Manually

1. Find the Universal Callback Server console window on the computer's desktop.
2. Click the `Close` button in the window's upper-right corner.

To Stop as a Windows NT Service

1. Select `Services` on the computer's Control Panel.
2. In the `Windows NT Services` window, select `UCS`.
3. Select `Stop`.

CPD Server

To Stop Manually

1. Find the CPD Server console window on the computer's desktop.
2. Click the `Close` button in the window's upper-right corner.

To Stop as a Windows NT Service

1. Select `Services` on the computer's Control Panel.
2. In the `Windows NT Services` window, select the `CPD Server`.
3. Select `Stop`.

Restarting VCB Components

To restart VCB components, follow the steps for starting them.

Connection and Reconnection

In VCB, the server components (Universal Callback Server and CPD Server) can be configured to support different `Standby` modes. (For more information about the `Standby` modes, refer to the *Framework Deployment Guide.)*

Three Standby Modes

- `Not specified`
- `Warm Standby`
- `Hot Standby`

In all three modes, the primary server, Universal Callback Server, maintains the current status of the callback request in the `Calling List` table associated with the `Callback Routing Point DN`.

Note: You cannot configure VCB for `Hot Standby` mode. If there is no backup server, the Management Layer tries to restart the existing server if the "Auto-Restart" option is enabled for the Universal Callback Server application. Some data in process is lost during a failure.

Not Specified

In this mode (`Not Specified`), the Management layer will not start the backup application; both primary and backup applications should be started before. The Management layer will switch the mode for the backup VCB application to primary if the primary VCB application fails.

Warning! When configuring redundant applications, do *not* select the `Not Specified` redundancy type unless using a switchover mechanism other than that provided by the Management Layer. It is acceptable, however, to leave the `Not Specified` redundancy type for nonredundant applications (that is, applications that do not have backup servers associated with them).

Warm Standby

When a switchover occurs and the primary server (Server 1) becomes the back-up server:

1. The primary server stops accepting new callback requests and remains in a `Warm Standby` mode.
2. Simultaneously, Server 2, which was the backup server, now becomes the primary server.
3. Server 2 simultaneously starts accepting new callback requests, reads the database (`Calling List` table), and, using the information in the records, maintains callback processing.

Hot Standby

Universal Callback Server and CPD Server do not support the `Hot Standby` mode, but T-Servers do.

Note: `Hot Standby` provides *high availability*, meaning that redundant configurations of a primary and a backup server provide more resource availability because they protect against hardware failures and reduce the time to return solution operations to normal.

If Universal Callback Server or CPD Server is connected to a T-Server that is in `Hot Standby` mode, there are two active T-Servers, both of which perform all functions.

- Both T-Servers start simultaneously and connect to a switch. `Hot Standby` mode guarantees the processing of interactions that are *in progress* if a failure occurs.

Note: *New* requests sent by clients *during* the failure and switchover may be lost. For complete information about T-Servers in `Hot Standby` mode, see the *Framework Deployment Guide.*

Universal Callback Server Connections

Universal Callback Server connects to the following components:

- DB Server
- CPD Server(s)
- Configuration Server
- T-Server(s)

- Stat Server(s)
- Universal Contact Server
- Message Server

To Set Up Connection to Servers

Universal Callback Server's connection to servers is set up on the Connections tab of the Universal Callback Server Application object's Properties box.

Universal Callback Server and DB Server

If the connection between Universal Callback Server and DB Server is lost:

- Universal Callback Server sends an error message and attempts to reconnect.
- Universal Callback Server stops accepting new callback requests until a successful reconnect to DB Server is made.
- If it is not possible to reconnect to the primary DB Server, then the next attempt to reconnect will be to the backup DB Server.
- Universal Callback Server connects to the DB Server and reattempts any non-completed requests.

During the reconnection or switchover to the backup DB Server, Universal Callback Server runs without manual intervention.

Universal Callback Server and T-Server

Some calls are lost while Universal Callback Server tries to reconnect to T-Server. The internal timer helps Universal Callback Server track the lost calls and mark the call records with the call result stale.

If the connection between Universal Callback Server and T-Server is lost:

- Universal Callback Server sends an error message and attempts to reconnect.
- Universal Callback Server stops accepting new callback requests and processing the existing callback requests until a successful reconnect to T-Server.
- If it is not possible to reconnect to the primary T-Server, then the next attempt to reconnect will be to the backup T-Server.
- Universal Callback Server connects to the T-Server and continues the callback processing after the switchover is complete.
- Universal Callback Server resubmits virtual calls to the backup T-Server to recreate the queue.

During the reconnect or switchover to the backup T-Server, Universal Callback Server runs without manual intervention.

Universal Callback Server and Configuration Server

If the connection between Universal Callback Server and Configuration Server is lost:

- Universal Callback Server sends an error message and attempts to reconnect.
- It continues to run and tries to reconnect to the same Configuration Server.
- If it is not possible to restore the session to the primary Configuration Server, then the next attempt to reconnect will be to the backup Configuration Server.
- There is no impact on callback processing because the configuration has been read during the initialization of Universal Callback Server.

During the reconnect or switchover to the backup Configuration Server, Universal Callback Server runs without manual intervention.

Universal Callback Server and CPD Server

If the connection between Universal Callback Server and CPD Server is lost:

- Universal Callback Server sends an error message and attempts to reconnect.
- If it is not possible to reconnect to the primary CPD Server, then the next attempt to reconnect will be to the backup CPD Server.
- If a backup CPD Server is not available, Universal Callback Server requests T-Server to dial the call (`TMakeCall` if target DN is extension or position and `TMakePredictiveCall` if target DN is Routing Point or ACD Queue).
- When Universal Callback Server is able to connect to a backup CPD Server and the switchover is complete, Universal Callback Server does not retry the interactions that the disconnected CPD Server handled. Instead, Universal Callback Server marks those callback requests with the call result `stale` when the Universal Callback Server internal timer for those requests expires.

During the reconnect or switchover to the backup CPD Server Universal Callback Server runs without manual intervention.

Universal Callback Server and Stat Server

If the connection between Universal Callback Server and Stat Server is lost:

- Universal Callback Server sends an error message and attempts to reconnect.
- It continues callback processing and tries to reconnect to the Stat Server.

- If it is not possible to reconnect to primary Stat Server, then the next attempt to reconnect will be to the backup Stat Server.
- There is no impact on callback processing during reconnects to Stat Server (for all callbacks, Estimated Waiting Time will count as zero until successful reconnection to Stat Server).
- Universal Callback Server connects to Stat Server and re-opens DataStream to transfer calculated statistics to Stat Server.

During the reconnect or switchover to the backup Stat Server, Universal Callback Server runs without manual intervention.

Universal Callback Server and Message Server

If the connection between Universal Callback Server and Message Server is lost:

- Universal Callback Server sends an error message and attempts to reconnect.
- It continues callback processing and tries to reconnect to the Message Server.
- If it is not possible to reconnect to the primary Message Server, then the next attempt to reconnect will be to the backup Message Server.
- There is no impact on callback processing during reconnects to Message Server.

During the reconnect or switchover to the backup Message Server, Universal Callback Server runs without manual intervention.

Universal Callback Server and Universal Contact Server

If the connection between Universal Callback Server and Universal Contact Server is lost:

- Universal Callback Server sends an error message and attempts to reconnect.
- It continues callback processing and tries to reconnect to the Universal Contact Server.
- If it is not possible to reconnect to the primary Universal Contact Server, then the next attempt to reconnect will be to the backup Universal Contact Server.
- There is no impact on callback processing during reconnects to Universal Contact Server.

- During the reconnect to Universal Contact Server, interaction records in Contact Server Database will not be created for new callback requests or updated for processed callbacks.

During the reconnect or switchover to the backup Universal Contact Server, Universal Callback Server runs without manual intervention.

Appendix

Deployment Planning Worksheet

This appendix contains the following sections:

Preliminary Procedures

1. Complete the Deployment-Planning Worksheet in this chapter on page 182. Refer to this worksheet as you complete the configuration procedures for VCB and its related components.
2. Keep the *Genesys Documentation Library* CD close at hand.
3. Verify that Framework is configured and installed at your contact center. For information about the configuration and installation of Framework, please refer to the *Framework Deployment Guide.*
4. Verify that Universal Routing [Enterprise Routing (ER) or Network Routing (NR)] is configured and installed at your contact center. For information about that product's configuration and installation, please refer to the *Universal Routing Deployment Guide.* See also "Designing Routing Strategies" on page 134 in this guide.
5. Verify that you have a Genesys-compliant IVR.
6. A Genesys-compliant IVR works with Genesys IVR Drivers to interface with Genesys Framework.

   For information about the configuration and installation of Genesys Voice Portal (GVP), please refer to the *Genesys Voice Portal Deployment Guide*:

   For information about the configuration and installation of Voice Treatment Option (VTO), please refer to these documents:

   - Chapter 3, *Voice Treatment Option: Voice Treatment Manager User's Guide*
   - Chapter 3, *Voice Treatment Option: Voice Treatment Server User's Guide*

“Designing IVR Scripts” on page 125 in this guide provides guidelines for IVR script design.

7. Verify that you have an agent desktop to enable VCB functionality in `Preview` mode or to enhance it in `Autodial` mode.
   - Genesys Agent Desktop (GAD) is one option for the agent desktop. For information about GAD installation and configuration, please refer to the *Genesys Agent Desktop .NET Toolkit Reference Manual* and the *Genesys Agent Desktop .NET Server Deployment Guide.*
8. Mount and open the Voice Callback 7.1 CD.

Deployment Planning Worksheet

Use the *Deployment Planning Worksheet* to record information required to configure and install the VCB components. Once completed, use it as a “look-up” sheet for important information that you need to perform the step-by-step procedures in this guide.

The “Sample Deployment-Planning Worksheet” on page 182 is provided for your convenience. Multiple boxes (cells) provide room to write additional values; for example, four boxes are provided for switch names. If you need more room for your installation information, copy these pages.

After completing the installation, keep the worksheet as a reference. If you need help from Genesys Technical Support, you can fax the worksheet to a support engineer to provide the complete system configuration.

Sample Deployment-Planning Worksheet

To assist you in filling out this worksheet, definitions of the worksheet terms follow the planning worksheet. Make copies of the Sample Deployment-Planning Worksheet for future reference.

Logging In to Configuration Manager

1. User name				
2. User password				
3. Application				
4. Host				
5. Port				

License Manager

6. Path to License Manager				

Tenant

7. Names				

Switches

8. Name(s)				
9. Type of switch				
10. Internal or external CPD capability	**Note:** Voice Callback uses the CPD capabilities of PBX (may be used in some deployment scenarios).			
11. T-Server connected to each switch (one-to-one relationship)				
12. Special DNs, if any				

DNs

Voice Callback inherits some of its DNs from Framework, Universal Routing, IVR, and customer switches.

13. DNs: ACD positions				
14. DNs: Extensions				
15. DNs: Call-processing ports (CPPs) These pertain to CPD Server (optional).				
16. DNs: Virtual Routing Points				
17. DNs: Routing Points				
18. DNs: Virtual Queues				
19. DNs: For call termination				
20. DNs: ACD Queue				

Applications

T-Server(s): Each switch you just listed requires its own T-Server.

21. T-Server name(s)				
22. Host for T-Server				
23. Port for T-Server				
24. Back-up server for T-Server				
25. Designated switch for T-Server (one-to- one relationship)				

Stat Server(s)

In VCB, IVR scripts and routing strategies use Stat Server to obtain the estimated waiting time. VCB also uses Stat Server for this purpose.

26. Stat Server name(s)				
27. Host for Stat Server				
28. Port for Stat Server				
29. Back-up server for Stat Server				

DB Servers

	for Configuration Database	**for VCB Calling List**		
30. DB Server name(s)				
31. Host for DB Server				
32. Port for DB Server				
33. Back-up server for DB Server				
34. Options (See *Voice Callback Reference Manual.)*				
35. DBMS type				

36. Name(s) of Tenants				
37. Black List				

Universal Callback Server

38. Universal Callback Server name				
39. Host for Universal Callback Server				
40. Communication port for Universal Callback Server				
41. Back-up server for Universal Callback Server				
42. Name(s) of T-Server(s) to which Universal Callback Server connects				
43. Name of CPD Server to which Universal Callback Server connects (optional).				
44. Name(s) of Stat Server(s) to which Universal Callback Server connects				
45. Name of Message Server to which Universal Callback Server connects (optional)				

CPD Server

46. CPD Server name(s)				
47. Host for CPD Server				

48. Communication port for CPD Server				
49. Back-up server for CPD Server				
50. Name of T-Server to which CPD Server connects				
51. Type of Dialogic board(s)				
52. DTI ports (DNs for Dialogic board)				

Web API Server

53. Web API Server name(s)				
54. Host for Web API Server				
55. Communication port for Web API Server				

Universal Contact Server

56. Universal Contact Server name(s)				
57. Host for Universal Contact Server				
58. Communication port for Universal Contact Server				
59. Back-up server for Universal Contact Server				

Universal Routing Server (URS)

URS is installed with the Routing solution (ER or NR). See the *Universal Routing Deployment Guide*.

Objects

Database Access Point(s)—For VCB Calling List and Black Lists

60. Database Access Point name				
61. DB Server name (See page 184.)				
62. DB Server host (See page 184.)				
63. DB Server communication port (See page 184.)				
64. DBMS name				
65. DBMS type (See page 184.)				
66. Database name				
67. User name for login to DB				
68. Password for login to DB				

Table Access

69. Table access name				
70. Table type				
71. DB Access Point name (See page 187.)				
72. Database table name				
73. Format (Use default.)				

Calling List

See “Calling List and Black List” on page 191.

74. Calling List Name	Default: Calling List			
75. Tenant				
76. Table access	See page 187.			

Note that the List option on the Annex tab of the Properties dialog box for the DN/Routing Point refers to this Calling List.

Treatments

77. Treatment name				
78. Call result				
79. Apply to call				
80. Destination DN (if "Apply to Call")				
81. Number in sequence				
82. Cycle Attempt (only for redial)				
83. Interval				
84. Increment				
85. Date/Time				

Worksheet Definitions

The "Sample Deployment-Planning Worksheet" beginning on page 182 uses the terms defined in this section.

Logging In to Configuration Manager	
	User name: UserID authorized to log in to Configuration Server
	User password: User password for UserID authorized to log in to Configuration Server
	Application: Name of Configuration Manager Application object
	Host: Host name of the computer where Configuration Server has been installed

Port: Port that the Configuration Server uses for client connections

License Manager

Path to License Manager: Write down the path to the License Manager. License Manager must be available for licensed applications.

Tenant

Name(s): If this is a multi-tenant environment, identify the tenant names to be used for Voice Callback.

Switches

As part of Framework configuration and installation, you will have already assigned agent names, agent logins, and DNs (see “DNs” on page 183) to switches.

In most systems, the telephone directly notifies the switch when the telephone set (teleset) status changes, such as when agents begin or end a call. By default, VCB registers teleset states internally. Older PBX software does not allow the teleset to notify the switch directly when the teleset status changes. If the contact center’s PBX software does not allow direct notification, you must customize the contact center’s teleset state disposition. Use configuration settings that instruct VCB to send Framework’s T-Server a notice for each teleset status change. The following refer to the capabilities of a switch.

Name(s): A switch set up during Framework configuration. Other switches can include numbers and types of all telephony devices (DNs) to be monitored and controlled through the CTI link. Identify the switch name(s) to be used for VCB. If you need new switches, add them through Framework. See the *Framework Deployment Guide* for information.

CPD capability: VCB may use the CPD capabilities of the PBX. In such a case, CPD Server is not required. If you wish to use CPD Server as a part of VCB at your contact center, collect the following information about the Dialogic boards:

- Call transferring: Does the switch have this capability?

- Call conferencing: Does the switch have this capability?

- Call forwarding: Does the switch have this capability?

- Do Not Disturb: Does the switch have this capability?

DNs

Names: Name(s) or number(s) of DNs

Types: For Routing Point, Virtual Routing Point, ACD Queue, Virtual Queue, Termination DN, Extension, ACD Position, or Voice Treatment Port

Applications (Components)

Each application name must be unique and may contain alphanumeric characters and underscores. Properties for most applications are the same; therefore, definitions are given only once for the applications (components) listed here.

T-Server name: Identify T-Server name.

Stat Server name: Identify Stat Server name.

DB Server name: Identify DB Server.

Universal Callback Server name: Identify Universal Callback Server.

CPD Server name: Identify CPD Server.

Universal Routing Server: Identify Universal Routing Server.

Server Info

Host: The host computer where your application will be running

Communication port: The port number that the application's clients will use to establish connections to the named server

Backup server: The backup server to be contacted if connection to the primary server fails

Objects

Object names: For detailed information about options, refer to the *Voice Callback Reference Manual,* Chapter 2.

Database Access Point(s)

The Universal Callback Server needs a Database Access Point for the VCB `Calling List` and `Black List`, database tables which hold persistent Voice Callback data for system backup.

Name: Name of the Database Access Point (DAP) object

DB Server: Name of an existing DB Server

Host: The host computer where the DB Server will be running

Communication port: The port number where the clients will connect to the DB Server

DBMS name: Alias of database (depends on particular RDBMS client connectivity software)

DBMS type: Select the type of database

Database name: Name of the database

User name: Login for accessing the database

Password: Password for accessing the database

Table Access

	Table Access name: Name of Table Access object
	Table Type: Select `Calling List`
	DB Access Point: Database access point through which the table can be accessed
	Database table name: Name as specified in the database

Calling List and Black List

In Voice Callback, `Calling List` and `Black List` tables are for persistent storage of callback data. These tables do not contain dialing records; however, the data serves to back up the system and may be used for future callbacks.

Decide where you will store the `Calling List` and `Black List` tables for Voice Callback. These VCB tables can be located either in the Genesys Configuration Database or in a separate database. The size of the tables and the ease of maintenance are the deciding factors for identifying the appropriate location. You can maintain very large `Calling List` and `Black List` tables in a database other than the Genesys Configuration Database.

	Calling list name: Name of calling list
	Table Access Point name: The Table Access object to which the calling list refers

Treatments

Decide on treatments to use for unsuccessful results on callbacks. For more details and examples, see “Treatments” in the *Voice Callback Reference Manual.*

	Treatment name: Name of treatment
	Call Result: The result of dialing a calling list record
	Apply to Record: The action to be applied to a record on the calling list based on the call result
	Apply to Call: The final outcome of attempts to dial a record
	Number in sequence: In a treatment sequence, each treatment contains a unique number that determines the order in which it is applied to the same call result.
	Cycle Attempt: The maximum number of consecutive times that the treatment is performed on the record.
	Interval: A time interval, in minutes, that the application waits between the first dialing attempt and the first treatment attempt or between consecutive attempts.
	Increment: Specifies the number of minutes that the application adds to the interval (see above) after the first treatment attempt and after each consecutive attempt.
	Date/Time: The date when another treatment attempt is performed

Index