GENESYS®

Voice Callback 7.1

Reference Manual

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.
http://www.genesyslab.com

Document Version: 71vcb_ref_04-2005_v7.1.001.00

Table of Contents

Chapter

1 About This Document

Welcome to the *Voice Callback Reference Manual*. This chapter has these sections:

Note: For previous releases of the *Voice Callback Reference Manual*, 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 information about the Voice Callback (VCB) option of Universal Routing, including:

- Configuration objects
- Configuration options
- Communication protocols
- Enumerations for callback record statuses
- Treatments
- Database tables used as persistent information storage
- Licensing requirements

Intended Audience

This manual is primarily intended for system administrators and agents. It assumes that you have a basic understanding of:

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

Chapter Summaries

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

- Chapter 2, VCB Configuration Objects, page 11, explains the configuration options for Universal Callback Server and for VCB-related objects.
- Chapter 3, Voice Callback Configuration Options, page 23, discusses the configuration options for Universal Callback Server and CPD Server.
- Chapter 4, Communication Protocols, page 51, summarizes the communication protocols.
- Chapter 5, Callback Enumerations, page 81, presents the enumerations for Callback record statuses.
- Chapter 6, Treatments, page 83, describes how to create treatments related to call results.
- Chapter 7, Database Tables, page 93, describes the purpose and structure of database tables used as persistent information storage by Universal Callback Server.
- Chapter 8, Licensing, page 97, describes licensing requirements for VCB.

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:

71fr_ref_03-2004_v7.1.000.00

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, 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:

- The *Voice Callback Deployment Guide* familiarizes you with the functions and features of the Voice Callback option of Universal Routing. This guide also discusses the applications/components that enable VCB functionality, and it provides a Deployment Planning Worksheet. It will help you configure and install Universal Callback Server and other components that comprise VCB.
- *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 (VT) Manager which serves as the user interface for Voice Treatment Option 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.
- *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 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 VCB Configuration Objects

This chapter describes the configuration objects you must define in the Framework Configuration Manager to enable VCB functionality. This chapter includes these sections:

General Configuration

As an option of Universal Routing, VCB works closely with Enterprise Routing (ER) or Network Routing (NR) to perform many of its functions. ER or NR must already be configured and installed in order to use the Voice Callback option.

Routing Points and Virtual Routing Points

The central component in VCB configuration is a Routing Point (RP) or Routing Queue. For VCB to function on a RP, you must, at minimum, supply values for the Calling List, and Virtual Routing Point (VRP) on the `Annex` tab of the `Properties` dialog box for an RP DN.

Figure 1 on page 12 shows a sample VCB configuration with the Routing Point as the central component.

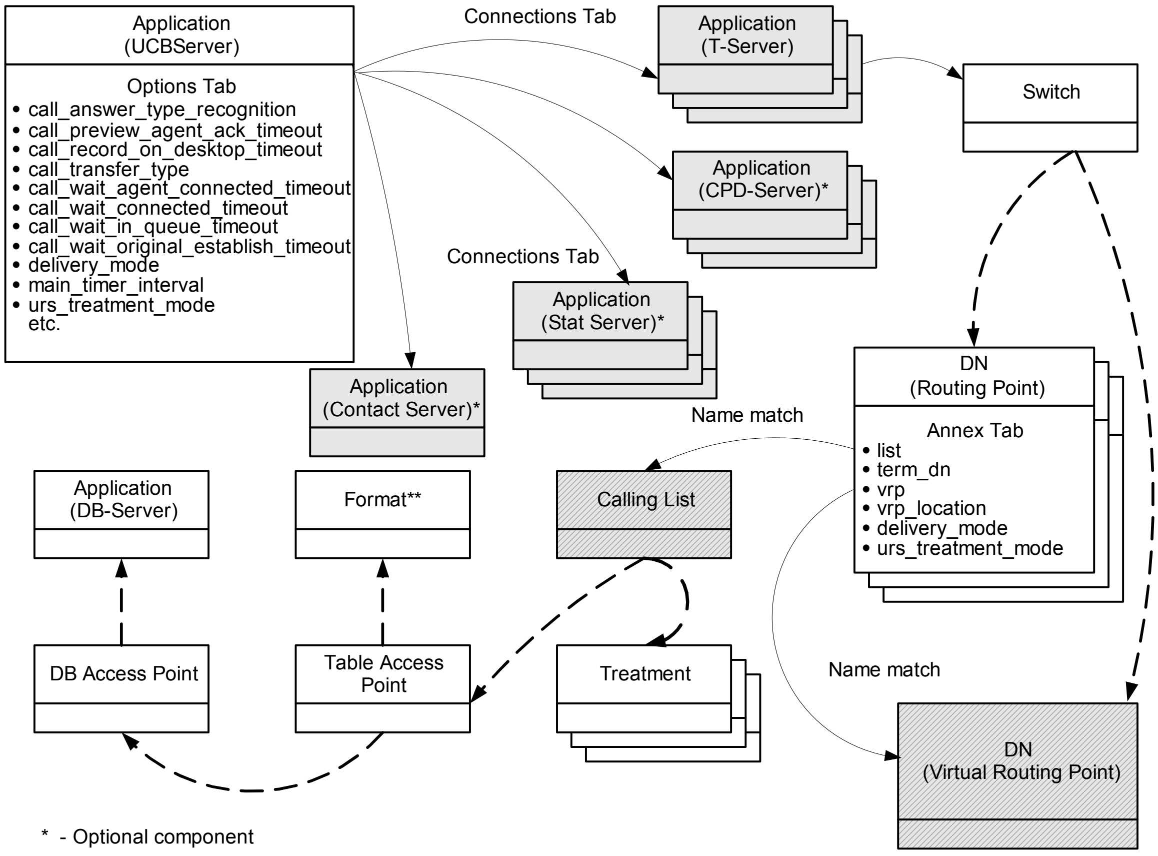


Figure 1: Sample Configuration

The options you must configure for VRPs are described in the vrp option description on page 38. For complete configuration procedures, see “Routing Points” and “Virtual Routing Points” in the *Voice Callback Deployment Guide*.

VCB and T-Server

TServer events enable the communication among the components which comprise the VCB solution. All required T-Servers must be in the Connections tab of the VCB Application object.

IVR and Agent Desktop

A Genesys-compatible Interactive Voice Response (IVR) product and an Agent Desktop product can be used to support VCB. You can find configuration information for these software products in the specific product documentation.

Universal Callback Server and VCB-Related Objects

Universal Callback Server is the main component of Voice Callback. This chapter focuses on the configuration of the Universal Callback Server and the objects that support it.

Calling List Object

In VCB, the Calling List table provides persistent storage for callback data.

The Calling List table is autogenerated when the Universal Callback Server starts. The table structure of the Calling List is hardcoded; however, since the format is a mandatory attribute of a Calling List, this field should be defined when the Calling List object is being configured. Use the default format. See "Table Access Object" section below.

For more information on persistent storage for callback data, see Chapter 7 on page 93.

Location: Calling List Object

The Calling List object is defined in this location:
Configuration Manager: <tenant_name>, Calling Lists folder

Table Access Object

Table Access objects define the relationship between calling lists, special lists, formats, and the database. They describe database tables of a specified format and explain how these tables can be accessed through Database Access Points. You can select a predefined Table Access object or create a new Table Access object. (See the *Voice Callback Deployment Guide* for these procedures.) The following section provides detailed information about the properties of the Table Access object.

In VCB, the Table Access object is used for Calling List and Black List configuration. See further information on Black List configuration in the *Voice Callback Deployment Guide*.

Table Access Object—General Tab Fields

Location: Table Access Object

Configure Table Access Object—General Tab Fields in this location:
Configuration Manager: <tenant_name>, Table Access object, Table Access Properties dialog box, General tab

Name

Default Value: `Table Access`
Valid Value: `<string>`
Changes Take Effect: Immediately

Required parameter referring to the name of the Table Access object. The length is not fixed. This value must be unique within the tenant.

Table Type

Default Value: `Unknown Table Type` (See Note below.)
Valid Value: `Calling List`
Changes Take Effect: Immediately

Required parameter referring to type of table in the database.

Note: Configuration Manager does not recognize the default value; therefore, you must select a different value for this object. Always select `Calling List`. Once you specify `Calling List`, you cannot change this value.

Description

Default Value: No default value
Changes Take Effect: Immediately

Optional parameter giving a brief description of the Table Access object.

DB Access Point

Default Value: No default value
Changes Take Effect: Immediately

Required parameter referring to the Database Access Point through which the table can be accessed. Use the `Browse` button to locate an existing object.

Format

Default Value: No default value
Changes Take Effect: Immediately

Required parameter referring to the name of the format applied to this table. Once it is specified, it cannot be changed. Use the `Browse` button to locate an existing value.

Database Table

Default Value: No default value
Valid Value: `<string>`
Changes Take Effect: Immediately

Required name of the table in the database where the calling list for VCB is stored. This table provides persistent storage for callback-related data.

State Enabled

Default Value: Check mark indicating enabled
Valid Value: Check box can be checked or unchecked.
Changes Take Effect: Immediately

Required parameter which has a check box indicating the current state of the object. For VCB to function properly, the state needs to be enabled (check box checked).

Treatment Object

A treatment is an action you can apply to any call result.

Location: Treatment Object

Configure the Treatment object in this location:
Configuration Manager: <tenant_name>, Treatment object, Treatment Properties dialog box

Treatment Object—General Tab Fields

Location: General Tab Fields

Configure Treatment Object—General Tab Fields in this location:
Configuration Manager: <tenant_name>, Treatment object, Treatment Properties dialog box, General tab

Name

Default Value: `<Treatment>`
Valid Value: `<string>`
Changes Take Effect: Immediately

Required name of the treatment. It must be unique within the tenant.

Tenant

Automatically populated by the system if the Treatment object folder is a subdirectory of `<Tenant>`.

Description

Default Value: No default value
Valid Value: `<string>`
Changes Take Effect: Immediately

Optional description of the selected treatment.

Call Result

Default Value: `Unknown Call Result`
Valid Value: See note below.
Changes Take Effect: Immediately

Note: The Configuration Manager does not recognize the default value; therefore, you must select a different value for this object. Table 1 shows the valid values for the call result.

Required parameter where you must assign an `Apply-to-Record` action to each call result for treatments to apply.

Table 1: Call Result Values

Call Result Value	Description
Abandoned	Call dropped by called party
Agent CallBack Error	N/A
All Trunks Busy	No free trunks on the switch for dialing a call
Answer	Customer was reached at the dialed phone number.
Answering Machine Detected	Answering machine was detected at the dialed phone number.
Busy	Dialed phone number was busy.
Dial Error	Hardware error from a Dialogic board
Dropped	Call dropped by calling party. Usually N/A for callback.
Fax Detected	Fax machine was detected at the dialed phone number.
General Error	General error occurs when a call is not completed, possibly caused by an invalid telephone number in the record or a wrong number according to the switch.
Group CallBack Error	N/A
No Answer	Ring without answer at destination
No Dial Tone	Absence of dial tone is based on an error returned by the Dialogic board or the Call Progress Detection (CPD) board on the switch.
No Free Port	N/A for VCB
No Port Available	No port is available to place a ‘call.’
NU Tone	This special Public Switched Telephone Network (PSTN) code is valid only in Europe.
OK	Call result is undefined.
Pager Detected	Pager was reached at the dialed phone number.

Table 1: Call Result Values (Continued)

Call Result Value	Description
Silence	Call was dialed, but there is no CPD.
SIT Detected	Any type of network tone
SIT IC (Intercept)	Only applies if the network supports this specific Standard Information Tone (SIT). Check with the switch vendor for confirmation.
SIT Invalid Number	Only applies if the network supports this specific SIT. Check with the switch vendor for confirmation.
SIT NC (No Circuit)	Only applies if the network supports this specific SIT. Check with the switch vendor for confirmation.
SIT RO (Reorder)	Only applies if the network supports this specific SIT. Check with the switch vendor for confirmation.
SIT Unknown	Only applies if the network supports this specific SIT. Check with the switch vendor for confirmation.
SIT VC (Vacant Code)	Only applies if the network supports this specific SIT. Check with the switch vendor for confirmation.
Switch Error	No dial tone received
System Error	Dialing software error from the Dialogic driver or CPD from the switch
Transfer Error	Dialer has a problem transferring calls based on the call action.
Unknown Call Result	Default value. You must change this default value and replace it with another value from this table. See Note on page 14.
Wrong Party	Call is answered but by a wrong party; this call result is sent by the Agent Desktop application and not detected by the dialer.

Apply to Record

Default Value: `Unknown Action` (See Note below.)

Note: The Configuration Manager does not recognize the default value; therefore, select a different value for this object.

Valid Values: See Table 2 on page 18.
Changes Take Effect: Immediately

Required parameter to create a treatment. Specifies the action to apply to a callback request based on the call result. Applies to all supported call results in the `Call Results Values` table except for call result `Answer`.

Table 2: Apply to Record Values (Supported by Universal Callback Server)

Apply to Record Value	Description
No Treatment	No treatment will be applied.
Redial	Redial number within a specified number of minutes (Interval) for a specified number of times (Cycle Attempt).
Retry at specified date	Record will be redialed at the specified Date and Time.
Retry in	Retry once after the specified number of minutes (Interval).
Unknown Action	Default value. You must replace this default value with another value from this table.

Note: Table 2 does not include all the values that are in the drop-down list for the `Apply-to-Record` field in the `Treatments` dialog box because some values are not applicable (not supported by VCB).

Apply to Call

Default Value: `Unknown Action Code`

Note: The Configuration Manager does not recognize the default value; therefore, you must select a different value for this object.

Valid Values: See Table 3.
Changes Take Effect: Immediately

Optional value allowing an alternate path when a dialing attempt reaches an answering machine or a FAX machine. See Chapter 6, "Treatments" and the section on "Answering Machine Detected or FAX Detected" on page 91. For all other call results, see the `Apply to Record` field information.

Table 3: Apply to Call Values (Supported by Universal Callback Server)

Apply to Call Value	Description
Connect or Transfer	Call connected to the DN defined as the Destination DN. See also Chapter 6, "Treatments"and the section on "Apply to Call Outcomes" on page 91.

Table 3: Apply to Call Values (Supported by Universal Callback Server) (Continued)

Apply to Call Value	Description
Drop	Call is dropped or disconnected.
[Unknown Action Code]	Default value. You must replace this default value with another value from this table if you are creating a treatment for the Answering Machine Detected or Fax Detected call results.

Note: Table 3 does not include all the values that are in the drop-down list for the Apply-to-Call field in the Treatments dialog box because not all values are applicable (not supported by VCB).

Destination DN

Default Value: No default value
Valid Value: <DN>
Changes Take Effect: Immediately

Required parameter for the Transfer and Connect Apply to Call outcome. DN to which the call is to be forwarded or routed. The destination DN may be any DN. You can use the Browse button to search for an existing DN.

See T-Server and Switch documentation to verify what is supported. Also consult Configuration Layer documentation.

Number in sequence

Default Value: 0
Valid Values: 1 or another positive integer
Changes Take Effect: Immediately

Required parameter describing a unique number in a treatment sequence that determines the order in which to apply the treatment. Must be set to 1 for a stand-alone treatment or for the first treatment in a sequence.

Interval (minutes)

Default Value: 0
Valid Values: 1 or another positive integer
Changes Take Effect: Immediately

Required when Apply to Record is set to Redial or Retry in. Time interval, in minutes, that the application waits between the current dialing attempt and the next treatment attempt.

Cycle Attempt

Default Value: `0`
Valid Values: `1` or another positive integer
Changes Take Effect: Immediately

Required to be other than `0` when `Apply to Record` is set to `Redial`. Maximum number of consecutive times to perform the treatment on the record.

Increment (minutes)

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

Required when `Apply to Record` is set to `Redial.` Specifies the number of minutes to add to the previous redial time interval.

Date

Default Value: `<current date>`
Valid Value: `<any date>`
Changes Take Effect: Immediately

Required if the `Apply to Record` action is set to `Retry at specified date`, the date when another treatment attempt will be performed. Click the `Date` drop-down arrow and choose a date from the calendar.

Time

Default Value: `<current time>`
Valid Value: `<any time of day>`
Changes Take Effect: Immediately

Required if the `Apply to Record` action is set to `Retry at specified date,` the time of day that another treatment attempt will be performed.

State Enabled

Default Value: Check mark indicating enabled.
Valid Values: Check box is either checked or unchecked.
Changes Take Effect: Immediately

Required parameter which has a check box indicating the current state of the object. For treatment to be applied by VCB, the state of treatment object needs to be enabled (check box checked).

Configurable Operating Hours

This functionality enables the contact center administrator to configure operating hours for `Scheduled` callbacks. Universal Callback Server checks scheduled time for `Scheduled` callbacks if operation hours are configured.

Operation hours are configured in folder `Statistical Tables/UCBServer` under `Tenant`.

- Statistical Table `Weekdays` contains one object per standard business day.
- Statistical Table `Holidays` contains one object per holiday.
- Statistical Table `Overtime` contains one object per working day that falls on weekends or for working day with after hours.

Note: Name of folder `UCBServer` and names of Statistical Tables `Holidays`, `Weekdays`, and `Overtime` cannot be changed.

- All Statistical Tables should have type `Special Day Table`.

Requirements for Objects in "Weekdays" Statistical Table

- 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 will use data only from first one)

Requirements for Objects in "Holidays" and "Overtime" Statistical Table

- Property `name`: any
- Property `isDayOfWeek: false`
- Property `day`, if Property `date` is not specified: number of day in year where 1 is Jan 1
- Property `date`: date in seconds from 00:00 01/01/1970.
- For objects in `Overtime` Statistical Table, properties `startTime` and `endTime` mean start and end of business day in minutes from 00:00.

Universal Callback Server checks 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 that the business day and time is within business hours.

- Checks scheduled date against objects in Statistical Table `Overtime` to determine that this is the business day and that the time is within business hours.

If 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 request with an error message:

`VCB_ERROR`: `TimeNotInSOH`

`VCB_ ERROR_ CODE`: 112

`VCB_ERROR_NAME` (Description): Scheduled time is out of the operating hours

Chapter

3 Voice Callback Configuration Options

This chapter discusses VCB configuration options for Universal Callback Server and Call Progress Detection (CPD) Server. It contains the following sections:

Universal Callback Server Application-Level Options

This section describes Universal Callback Server Application-Level options.

- `purge_records`
- `dnc_refresh_interval`
- `hide_private_data`
- `delete_ivr_announce_file`
- `check_ewt`
- `RouteInfoKey`
- `<Determined by the value of RouteInfoKey option>`

Note: Option `RouteInfoKey` should be configured in the section named `WebAPIRouting`. Option `<Determined by the value of RouteInfoKey option>` should be configured in the section named `WebAPIProcessing`. All other options are configured in section `UCBServer`.

purge_records

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

Optional parameter that prevents performance degradation when the database tables (in which Universal Callback Server saves intermediate data related to callbacks) fill up without any restrictions. With value `yes`, Universal Callback Server deletes database records related to callback requests that are completely processed instead of updating these records. With value `no`, Universal Callback Server updates the records for completed callback requests with `record_status = 9` (Final).

dnc_refresh_interval

Default Value: `60`
Valid Value: Interval in minutes.
Changes Take Effect: Immediately

Optional parameter: Universal Callback Server uses this option to set an interval to check periodically the Black List table and update the internal hash table in accordance with changes that have been made in the table after the last check.

hide_private_data

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

Optional parameter where, if this option is set to `true` or `yes`, Universal Callback Server does not print UserData from any telephony event in its log output.

delete_ivr_announce_file

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

Optional parameter that determines whether Universal Callback Server deletes IVR announcement file recorded during callback request processing. If set to `true`, Universal Callback Server will delete the file. If set to `false`, it will leave the file in place.

Note: In order to delete the file, Universal Callback Server requires file deletion permissions.

check_ewt

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

Optional parameter: If set to `true`, then Universal Callback Server takes into account Estimated Waiting Time (EWT) when placing `Scheduled` callback requests to the resource waiting queue. If not specified or set to `false`, Universal Callback Server does not take Estimated Waiting Time (EWT) into consideration during `Scheduled` callback processing.

RouteInfoKey

Note: This option is configured in the `WebAPIProcessing` section.

Default Value: `default`
Valid Value: Any valid `UserData` keyword
Changes Take Effect: Immediately

Optional parameter that specifies the name of the key that Universal Callback Server looks for in the request for the callback coming from the Web and uses to define a Routing Point.

<Determined by the value of RouteInfoKey option>

Note: This option is configured in the `WebAPIRouting` section.

Valid Value: String of the following format: `<dn>@<switch>`

Example of RouteInfoKey option configuration:

WebAPIProcessing

RouteInfoKey = RouteInfo

WebAPIRouting

default = 4137@Switch1

alternate = 4138@Switch1

live = 5364@Switch2

If pair RouteInfo = default arrives or no pair at all, routing takes place at Switch1, DN 4137

If pair RouteInfo = alternate arrives, then routing takes place at Switch 1, DN 4138

If pair RouteInfo = live arrives, then routing takes place at Switch 2, DN 5364

Universal Callback Server Application- and DN-Level Options

You can define most of the options that control VCB either at the Universal Callback Server Application level or at the Routing Point level. If you define them at the Universal Callback Server Application level, the following options apply to all Routing Points in the VCB configuration. If you define them at the Routing Point level, the options apply to a specific Routing Point.

- call_answer_type_recognition
- call_preview_agent_ack_timeout
- call_record_on_desktop_timeout
- call_transfer_type
- call_wait_agent_connected_timeout
- call_wait_connected_timeout
- call_wait_in_queue_timeout
- call_wait_original_establish_timeout
- callback_processed
- delivery_mode
- main_timer_interval
- max_proc_interval
- urs_treatment_mode
- auto_dial_mode
- call_default_routing_timeout
- public_network_access_code
- agent_announcement_data
- agent_announcement_type
- agent_announcement_prefix
- agent_announcement_suffix
- cust_announcement_prefix
- cust_announcement_data
- cust_announcement_suffix
- stat_type
- stat_profile
- stat_filter
- stat_timerange

call_answer_type_recognition

Default Value: no_am_detection
Valid Values: no_progress_detection, no_am_detection, positive_am_detection, full_positive_am_detection, accurate_am_detection, telephony_preset

The values and their descriptions are described below.

no_progress_detection Disables Call Progress Detection. Call will be transferred as soon as it is established.

no_am_detection Disables answering machine detection; however, detection of all other devices is still enabled.

positive_am_detection Enables standard answering machine detection (Positive Answering Machine [PAM] mode).

full_positive_am_detection Means that full positive answering machine detection (Full Positive Answering Machine [FPAM] mode) should be used.

accurate_am_detection Enables or disables Call Progress Detection (CPD) based on analysis of the whole greeting.

telephony_preset Equivalent to the value no_am_detection.

Changes Take Effect: Immediately

Optional parameter that specifies call progress detection configuration for CPD Server.

call_preview_agent_ack_timeout

Default Value: 10 (seconds)
Valid Value: 1...(seconds)
Changes Take Effect: Immediately

Optional parameter specifying maximum time that Universal Callback Server waits for an acknowledgment or rejection message from the agent desktop regarding the preview callback request. After the timeout, the callback request is redistributed.

call_record_on_desktop_timeout

Default Value: 60 (minutes)
Valid Value: Any positive integer
Changes Take Effect: Immediately

Optional parameter specifying maximum time in minutes during which an agent can hold a preview or autodialed record on the desktop. After timeout elapses, the callback request is marked with a status Error and call result Stale.

call_transfer_type

Default Value: `one_step`
Valid Values: `one_step, two_step`
Changes Take Effect: Immediately

Optional parameter that specifies the transfer type CPD Server uses for the call.

- With value `one-step,` CPD (Call Progress Detection) Server translates a `one_step` value as either a mute transfer or a single-step transfer.
- With value `two_step`, CPD Server uses a two-step transfer to overcome some switch transfer issues, such as *ring splash* on a Meridian switch.

Ring splash occurs when the dialer transfers a call to an agent, and the caller hears a ringback before hearing the agent answer. For some switches, this ringback cannot be turned off by a command in the CTI link or on the switch setting.

To overcome this difficulty, Genesys uses a two-step transfer.
In a *two-step transfer*,

1. The switch puts the original leg of the call on hold and initiates the transfer.
2. The caller hears either a moment of silence or music if the trunk or the queue has *`Music-on-Hold`* turned on. (See Note below.)
3. The second leg of the call is to dial a consult call to the agent's DN.
4. The transfer is completed when an agent answers the call.

Note: Genesys recommends turning off the *`Music-on-Hold`* on the trunk or queue when you use a two-step transfer.

call_wait_agent_connected_timeout

Default Value: `60 (sec)`
Valid Values: `1...(sec)`
Changes Take Effect: Immediately

Optional parameter referring to timeout-in-seconds between making a call and receiving `EventEstablished` on an Agent DN or, in the case of negative voice detection such as an answering machine, the timeout between making a call and receiving the `EventReleased` when the call is dropped.

call_wait_connected_timeout

Default Value: `120 (seconds)`
Valid Values: `0-600 (seconds)`
Changes Take Effect: Immediately

Optional parameter referring to length of waiting time (`timeout`) between the first ring and the determination that the called party is not answering. This definition pertains to analog and lineside DNs.

call_wait_in_queue_timeout

Default Value: `1200 (seconds)`
Valid Values: `0. . .(seconds)`
Changes Take Effect: Immediately

Optional parameter that defines how long (in seconds) a virtual call may stay in the virtual queue before it is resubmitted to the queue. Upon expiration of this timeout, Universal Callback Server deletes the original virtual call and creates a new one. This prevents virtual calls from being stuck in the queue if any component participating in virtual call distribution fails. The callback status remains in the `VcbRecStatQueued(3)` state.

call_wait_original_establish_timeout

Default Value: `4 (seconds)`
Valid Values: `0...(seconds)`
Changes Take Effect: Immediately

Optional parameter referring to timeout (in seconds) between receiving an off-hook response from PSTN (Public Switched Telephony Network) and `EventEstablished` from T-Server.

callback_processed

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

If this optional parameter is set to `true`, then Universal Callback Server in `Autodial` mode waits for `RequestCallbackProcessed` from the agent to finalize the callback processing; if it is set to `false`, Universal Callback Server relies on telephony events (`EventReleased` on agent's DN) to finalize the callback processing automatically.

Note: In `preview` mode, `RequestCallbackProcessed` from agent's DN is always awaited, regardless of the option settings.

delivery_mode

Default Value: `preview`
Valid Values: `preview, auto`
Changes Take Effect: Immediately

Optional parameter that tells Universal Callback Server in which form to deliver a callback to an agent.

- With `preview` value, Universal Callback Server delivers a callback request to an agent as a preview record and leaves the dialing up to the agent.
- With `auto` value, Universal Callback Server dials the callback automatically in one of the following ways:
    - As a direct call from the agent's DN to the customer's phone.

- As a predictive call that uses switch call progress detection capabilities.
- Through the CPD Server where CPD Server is responsible for call progress detection and further call handling.

main_timer_interval

Default Value: `2 (seconds)`
Valid Values: `1...20 (seconds)`
Changes Take Effect: Immediately

Optional parameter that controls how often to check the call timeouts in the queue. Genesys recommends that the value should not be modified; instead, keep the default value setting for this option.

max_proc_interval

Default value: `0`
Valid value: `0·3600`. Represents maximum duration (in minutes) of Callback Processing, starting from submission moment for `ASAP` callback or from scheduled time for `Scheduled` callbacks. When this timeout elapses, callback processing is stopped, and it is finalized in database with `callback_status=VcbRecStatEndTimeExpired(14)`.

Value `0` means that there is no expiration interval applicable.

Changes Take Effect: For callbacks submitted after the change was made

Note: Callback processing expiration time can be submitted from Web API (attributes `starttime` and `endtime`) from IVR or Agent Desktop (attribute `VCB_END_DATE_TIME`) and also configured in `max_proc_interval` option. The value configured in this option is used if there is no such information for a specific callback submission request. It is applicable to processing all callback requests, independently of their origin (Web, IVR, Agent Desktop).

urs_treatment_mode

Default Value: `transfer`
Valid Values: `transfer, treatment`
Changes Take Effect: Immediately

Optional parameter controlling the way a call suitable for a callback request arrives at a Universal Callback Server Routing Point.

- With `transfer` value, the call arrives via `EventRouteRequest/EventPartyChanged`.
- With `treatment` value, the call arrives via `EventTreatmentEnd`.

This option also determines the way the router delivers a call to a Routing Point.

For more information about the latter, see the “IVR Behind-the-Switch Configuration” section in the *Voice Callback Deployment Guide*.

call_default_routing_timeout

Default Value: `0`
Valid Value: Number of seconds.
Changes Take Effect: Immediately

Optional parameter: If you set this option to a non-zero value, Universal Callback Server will wait for a specified amount of seconds for Routing instructions from the Router and then will apply default routing (as configured in VRP) to the current callback request.

public_network_access_code

Default: `<empty string>`
Valid Value: Any character string
Changes Takes Effect: Immediately

Optional parameter. This string is added as a prefix to each phone number that UCS sends to a specific switch. This number specifies the PSTN access code for the switch to which the T-Server is connected. When an access code is added as a value, the system always places the access code in front of the dialed phone number. For example, if you enter the number 9 into the value field, the prefix 9 is always dialed before each phone number is dialed.

If you use Dialogic and CPD Server, then you can add Dialogic dialing control parameters as a prefix to the dialed numbers. The following symbols are available:

- `'L'`--Wait for the local dial tone before dialing.
- `'I'`--Wait for the international dial tone before dialing.
- `'X'`--Wait for the special dial tone before dialing.
- `','`(comma)--Pause two seconds.

For example, if the `public_network_access_code` is defined as ‘L9,’ the dialer will wait for a dial tone, dial 9, pause for two seconds, and then dial the number.

Note: Remember that Dialogic dialing control symbols work only if the CPD Server dials the calls through the Dialogic card (that is, CPD Server option `tscall= false).`

auto_dial_mode

Default Value: `dial_ahead`
Valid Values: `dial_ahead, engaging`
Changes Take Effect: Immediately

Optional parameter in which `dial_ahead` value means Customer Reserved Connection, and `engaging` value means Agent Reserved Connection. Universal Callback Server receives DN and Location (Switch) from URS.

Table 4 describes Universal Callback Server behavior depending on the type of DN it receives from URS, the Autodial mode that is configured, and the availability of CPD Server.

Table 4: Autodial Mode

DN Type	CPD Server Available?	auto_dial_mode	Actual Mode
Not CDN	Yes	dial_ahead	dial_ahead
Not CDN	Yes	engaging	engaging
Not CDN	No	dial_ahead	dial_ahead (TMakeCall)
Not CDN	No	engaging	Error message
CDN	Yes	dial_ahead	dial_ahead
CDN	Yes	engaging	engaging
CDN	No	dial_ahead	dial_ahead (TMakePredictiveCall)
CDN	No	engaging	Error message

Simple Announcement for Reserved Agent Connection

Options defining a simple announcement may be configured in the Universal Callback Server-assigned section in the RP annex or in the Universal Callback Server application object.

- If `agent_announcement_type= tone`, CPD Server plays the beep tone with parameters specified in the `agent_announcement_data` option.
- If `agent_announcement_type= voice`, CPD Server plays the sequence of recorded voice files specified in these options:
- `agent_announcement_prefix`
- `agent_announcement_data`
- `agent_announcement_suffix`

Note: Each option specifying a recorded voice file can consist of more than one file name. File names are separated by semicolons (;).

agent_announcement_type

Default Value: none

Valid Values: `tone, voice, none`
Changes Take Effect: Immediately

Optional parameter: If set to `tone`, CPD Server will play a tone to an agent. If set to `voice`, CPD Server will play voice file to an agent. If not specified, Universal Callback Server will not request CPD Server to play an announcement to the agent.

agent_announcement_prefix

Default Value: `<empty string>`
Valid Value: String that contains full path to voice file.
Changes Take Effect: Immediately

Optional parameter that specifies the file(s) played to an agent.

agent_announcement_data

Default Value: `<empty string>`
Valid Value: String that contains beep tone parameters or full path to voice file(s) (depends on announcement type specified in option `agent_announcement_type`). Parameters are separated by semicolons (;).
Changes Take Effect: Immediately

Optional parameter that specifies which file(s) are played to an agent. Beep tone parameters are set like this:

<frequency of tone 1 in Hz>;

<frequency of tone 2 in Hz>;

<amplitude of tone 1 in dB>;

<amplitude of tone 2 in dB>;

<duration of tone in 10 ms units>;

agent_announcement_suffix

Default Value: `<empty string>`
Valid Value: String that contains full path to voice file(s) separated by semicolons.
Changes Take Effect: Immediately

Optional parameter that provides the suffix for the sequence of voice files.

Announced Reserved Customer Connection

CPD Server plays sequence of recorded voice files specified in options:

- `cust_announcement_prefix`
- `cust_announcement_data`
- `cust_announcement_suffix`

cust_announcement_prefix

Default Value: `<empty string>`
Valid Value: String that contains full path to voice file separated by semicolons.
Changes Take Effect: Immediately

Optional parameter that provides the path to the file(s) played to a customer.

cust_announcement_data

Default Value: `<empty string>`
Valid Value: String that contains full path to voice file(s) separated by semicolons.
Changes Take Effect: Immediately

Optional parameter that specifies which file(s) are played to customer.

cust_announcement_suffix

Default Value: `<empty string>`
Valid Value: String that contains full path to voice file.
Changes Take Effect: Immediately

Optional parameter that provides the suffix for the sequence of voice files.

Estimated Waiting Time (EWT) Statistics for Scheduled Callback Requests

All parameters required by Universal Callback Server to receive EWT statistics from Stat Server are configured on the configuration objects specified below. The list is given in the order of decreasing priority; that is, <DN> has the highest priority, and Application object has the lowest priority.

- <DN> - Any DN object specified in the `VCB_EWT_OBJECTID` attribute, which Universal Callback Server receives with the callback request.

  Typically the type of DN used would be `Virtual Queue`. See Table 5, “Reserved Attribute Names,” on page 54 for further details on the `VCB_EWT_OBJECTID` attribute.
- VRP - DN of type `Virtual Routing Point` that has been specified in the `vrp` option for the Routing Point for which the current callback request is being processed.
- <Application> - Universal Callback Server Application object.

Regardless of the configuration object being used, the following options are configured in a separate section named EWT.

Note: The `stat_server` option is considered only when configured at the <DN> or VRP level. It is ignored when being configured on the <Application> level.

If the `VCB_EWT_OBJECTID` attribute is present in the callback request, then Universal Callback Server would request statistics from the Stat Server for this object. Otherwise, Universal Callback Server would request statistics from the Stat Server for the Virtual Routing Point specified in the `vrp` option for the Routing Point for which the current callback request is being processed.

stat_type

Default Value: `AverDistribCallTime`
Valid Value: Any statistical type configured on corresponding Stat Server object.
Changes Take Effect: Immediately

Optional parameter. See *Framework Stat Server User's Guide*.

stat_filter

Default Value: `<empty string>`
Valid Value: Name of any Filter configured on corresponding Stat Server object.
Changes Take Effect: Immediately

Optional parameter referring to filter used by Universal Callback Server to obtain EWT for Queue that is used in processing `Scheduled` callbacks.

stat_profile

Default Value: `<empty string>`
Valid Value: Name of the `TimeProfile` configured for the corresponding Stat Server object.
Changes Take Effect: Immediately

Optional parameter referring to the time profile used by Universal Callback Server to obtain EWT for Queue that is used in processing `Scheduled` callbacks.

stat_timerange

Default Value: `<empty string>`
Valid Value: Name of the time range for sliding interval selection as configured in corresponding Stat Server, section name `TimeRanges`.
Changes Take Effect: Immediately

Optional parameter that specifies the name of the time range. If not specified anywhere, Universal Callback Server will not specify time range in requests for statistics to Stat Server.

DN-Level Options

DN-level options are stored in a section folder called either UCBServer or the name of a particular application. This folder is created in the Annex tab of the Routing Point folder, where you plan to use VCB service.

If the section folder is named UCBServer, any Universal Callback Server application may use the Routing Point being configured. If the section folder bears the name of a particular Universal Callback Server Application object, only that instance of the Universal Callback Server application may use the Routing Point. (For information about naming the section folder, see "Routing Points" section in the *Voice Callback Deployment Guide*.)

Location: DN-Level Options

You can define the options listed below *only* at the DN level.The DN-level options are defined on the Annex tab of the DN object:

Configuration Manager: <tenant_name>, Switches, <specific switch>, DNs Folder, <specific DN (RP)>, Annex tab

- channel_num
- list
- term_dn
- vrp
- vrp_location
- call_progress_orig_dn
- call_progress_trasn_dn
- stat_server

channel_num

Default Value: 40
Valid Values: 0 or any positive integer
Changes Take Effect: Immediately

Optional parameter that specifies the total number of available channels (CPD ports) on the Dialogic board(s) when using the Autodial mode. Universal Callback Server dials a call when the number of busy channels is less than the value specified for the channel_num option. For example, if you use the default value 40 for the channel_num option, Universal Callback Server will dial a call when the number of busy channels is 39 or less. A zero (0) value assigned to the channel_num option means that no CPD ports are designated for Universal Callback Server.

list

Default Value: `<empty string>`
Valid Value: `<string>`
Changes Take Effect: After restarting the application.

Required parameter giving the name of Calling List configuration option. This Calling List option is used for persistent storage of callback requests.

term_dn

Default Value: `<empty string>`
Valid Value: `<string>`
Changes Take Effect: After restarting the application.
Optional parameter giving the name of the configuration object for the DN where calls are dropped. If this option is set to some DN value, the call flow changes. Figure 2 shows the call flow for the configuration of Voice Callback when option `term_dn` is used and set to a valid DN name.

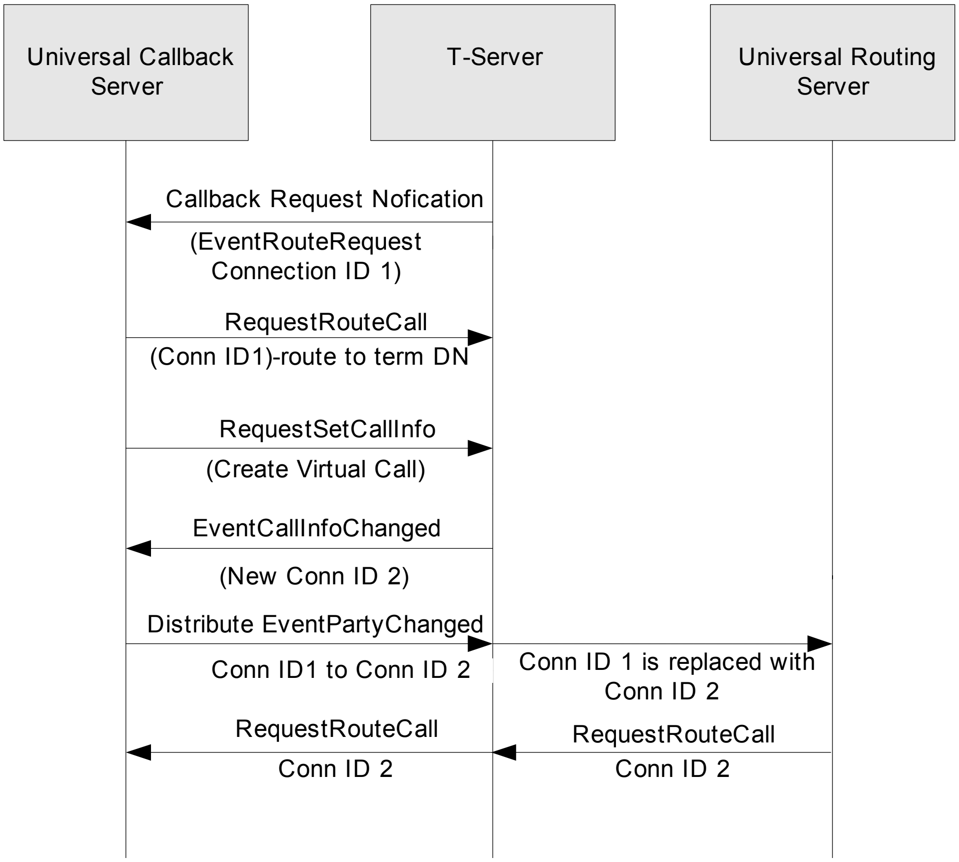


Figure 2: Callback Delivery Phase

vrp

Default Value: `<empty string>`
Valid Value: `<string>`
Changes Take Effect: After restarting the application.

Required parameter giving the name of the Virtual Routing Point configuration object. This VRP is used for routing of virtual calls which represent callback requests.

Note: You need to have created the Virtual Routing Point (VRP) before you can assign the value in the `vrp` field.

vrp_location

Default Value: `<empty string>`
Valid Value: `<string>`
Changes Take Effect: After restarting the application.

Optional parameter giving the name of Switch configuration object, where VRP specified in option `vrp` resides

call_progress_orig_dn

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

Optional parameter that points to a DN which should be used as Call Progress Origination DN in `Autodial` mode when `RequestMakePredictiveCall` is used.

Note: This option is configured under UCBServer or `<particular application name>` section in the DN intended to be used as Call Progress Origination DN.

call_progress_trans_dn

Default Value: `<empty string>`
Valid Value: `<DN name>`
Changes Take Effect: Immediately

Optional parameter that points to a DN which should be used as Call Progress Transfer DN in `Autodial` mode when `RequestMakePredictiveCall` is used.

stat_server

Default Value: `<empty string>`
Valid Value: Name of the Stat Server Application object on the Universal Callback Server `Connections` tab.
Changes Take Effect: Immediately

Optional parameter: If you set no value for this option or do not define this option, Universal Callback Server uses first Stat Server listed on its `Connections` tab. If you set value, Universal Callback Server will use this Stat Server for particular DN.

CPD Server Options

CPD Server options are located in sections on the `Options` tab of the CPD Server Application object in the Configuration Manager.

Each of these sections has its own set of configuration options:

- General
- Line-Side
- Tones

These sections include new options that allow you to configure the 7.0 CPD Server to meet the needs of your contact center.

For information about options that do not apply in release 7.1, see the section entitled "Please note that SIT tone definitions should be customized only if the default settings are inappropriate for your particular system." on page 49.

Section: General

continuous-no-signal

Default Value: `20`
Valid Value: Unsigned integer
Changes Take Effect: Immediately

Optional parameter that specifies the time, in seconds, of continuous silence (no ringback timeout).

The next two options define the prefixes in the file names for two types of voice recording files.

conversation-file-name-prefix

Default value: conv_
Valid value: An ASCII string
Changes Take Effect: Immediately

Required if voice recorded files are enabled for CPD Server: see cpd-recording option. The file names for the voice files that CPD Server creates for outbound calls are structured as follows:

```
<Prefix>_<RecRef>_<PhoneNumber>_<Date>_<Time>_<Agent/CustPortNames>
```

Each of the variables in angle brackets is replaced by an alpha/numeric string.

The prefix in the file name is set by one of two CPD Server configuration options, depending on the type of file it names (call result or conversation):

- cpd-file-name-prefix for File 1 (default is cpd_)
- conversation-file-name-prefix for File 2 (default is conv_)

  The value of the option conversation-file-name-prefix identifies File 2 (conversation). This is a sample file name for a conversation voice file:

  ```
  Conv__5551212_060703_060005_dtiB1T1_dtiB1T2.wav
  ```

cpd-file-name-prefix

Default Value: cpd_
Valid Value: An ASCII string
Changes Take Effect: Immediately

Required parameter if voice recorded files are enabled for CPD Server: see cpd-recording option. You can configure CPD Server to create two voice files for each outbound call that it dials on a Dialogic port. This option specifies the prefix in the file name for one type of recorded voice file. The value of the option cpd-file-name-prefix identifies File 1 (call result).

The following is a sample file name for a call result voice file:

```
Call_Result__5551212_060703_060000_dtiB1T1.wav
```

cpd-recording

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

Optional parameter that indicates whether to record cpd (value = yes) or not to record cpd (value = no).

destination-busy-timeout

Default Value: 2000 (ms)
Valid Values: <A numeric value representing milliseconds>
Changes Take Effect: Immediately

Optional parameter that prevents CPD Server from waiting indefinitely for the results of Call Progress Detection on a busy signal. CPD Server normally waits for the following two indicators that show a dialed number is busy before it terminates Call Progress Detection:

- Busy call result from the Dialogic board
- EventDestinationBusy from T-Server

Now, instead of CPD Server waiting indefinitely for a *Busy* call result from both of these sources, this option, destination-busy-timeout, specifies the length of time (in milliseconds) that CDP Server will wait for confirmation of the call result from the second source after the first has arrived. When the timeout expires, CPD Server accepts the *Busy* call result as correct.

line-type

Default Value: No default value
Valid Values: dm3, line-side, analogue, or isdn string
Changes Take Effect: Immediately

Required parameter that specifies the following:

- Type of line (Dialogic card) being used.
- Type of protocol used to connect the Dialogic board to the switch or PSTN: for example, dm3, analogue, line-site, or ISDN.

max-number-ports-to-record

Default Value: 0
Valid Values: 0, 1. . .k
Changes Take Effect: Immediately

Optional parameter that supports the recording of two voice files in WAVE format for each outbound call that CPD Server dials:

- File 1 contains the line recording for the Call Progress Detection stage.
- File 2 records the conversation between an agent and the called party if the call result is Answer (ASM mode only).

CPD Server records these voice files on the Dialogic ports used for outbound dialing. *Specific* ports are *not* dedicated for reporting, but the *number* of ports to use for recording is configurable. The option max-number-ports-to-record specifies the maximum number of agent ports on which to record at the same time. A value of 1 or more activates a voice file recording on that number of ports concurrently. A value of 0 (zero) deactivates the voice file recording function.

CPD Server keeps track of the ports being recorded and stops recording when the specified number of ports are engaged for this purpose. CPD Server will not record on another port until one is disengaged. The number of times that the same call transfers from Agent DN to Agent DN does not increase the number of ports being used for the recording of that call. The call remains on the same port when it passes from agent to agent.

The value of the max-number-ports-to-record is limited by two factors:

1. The number of configured recording ports.
2. The hardware limit for the number of Dialogic ports per box.

For example:

```
switch
    DNs
            'location_name'
                    regular
                    engaging
                    recording
                            dxxxB1
                                    dxxxB1C1
                                    dxxxB2C2
```

You configure the option `max-number-ports-to-record` in the CPD Server Application object in Configuration Manager.

Note: The extended port usage for recording might necessitate the allocation of more Dialogic resources, that is, an increased number of ports to be used for outbound dialing in general.

location

Default Value: No default value
Valid Value: String
Changes Take Effect: Immediately

Required parameter that specifies the name of the folder under the DN section where the configuration is stored.

off-hook-delay

Default Value: `-1`
Valid Value: Integer
Changes Take Effect: Immediately

Optional parameter that specifies the delay, in seconds, between `off-hook` and `tmakecall:`

- negative: `off-hook,` then `tmakecall`
- positive: `tmakecall,` then `off-hook`

off-hook-timeout

Default Value: `1`
Valid Value: Unsigned integer
Changes Take Effect: Immediately

Optional parameter that specifies the time to wait, in seconds, before assuming `off-hook` state of channel.

on-hook-timeout

Default Value: 1
Valid Values: Unsigned integer
Changes Take Effect: Immediately

Optional parameter that specifies the time to wait, in seconds, before assuming on-hook state of channel.

out-of-service-attempts

Default Value: 2
Valid Value: Unsigned integer; minimum 2
Changes Take Effect: Immediately

Optional parameter that specifies the number of failed attempts to use a port (channel) before CPD Server marks it out of service.

out-of-service-timeout

Default Value: 1
Valid Value: <unsigned integer with no upper limit restrictions; minimum 0>
Changes Take Effect: Immediately

Optional parameter that specifies the amount of time (in minutes) that CPD Server will wait before trying to use an out-of-service port again. If the value is 0, then CPD Server ignores out-of-service conditions and continues to attempt to use the port.

sit-detection

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

Optional parameter that defines the SIT signal parameters in the Dialogic DX_CAP data structure to enable CPD Server to identify a SIT signal more easily.

Controls SIT detection under the following conditions: SIT is disabled, and tscall is enabled to determine from the EventDestinationBusy if a number is valid.

tscall

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

Required parameter that indicates whether to send a request to T-Server (tscall = yes) or to Dialogic board (tscall = no) to make a call.

tsclear

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

Optional parameter that controls the way a call is released. If this option is set to `yes`, a `RequestClearCall` (G3 specific) is issued to release an active call.

This option is useful, but not required, if a two-step transfer fails on a G3 switch.

wait-off-hook

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

Optional parameter that indicates whether to wait (`wait-off-hook = yes`) or not to wait (`wait-off-hook = no`) for an off-hook event from T-Server.

Section: Line-Side

off-hook-bit-mask

Default Value: `a-on;b-on;`
Valid Values: String (bit value), as shown below:

- `a-off;b-on;`
- `a-off;b-off;`
- `a-on;b-off;`
- `a-on;b-on;`

Changes Take Effect: Immediately

Optional parameter that specifies the mask for a t1/e1 line that is off-the-hook.

on-hook-bit-mask

Default Value: `a-off;b-on;`
Valid Values: String:

- `a-off;b-on;`
- `a-off;b-off;`
- `a-on;b-off;`
- `a-on;b-on;`

Changes Take Effect: Immediately

Optional parameter that specifies the mask for a t1/e1 line that is on-the-hook.

rcv-idle-bit-mask

Default Value: `a-off;b-on;`
Valid Values: String:

- `a-off;b-on;`
- `a-off;b-off;`
- `a-on;b-off;`
- `a-on;b-on;`

Changes Take Effect: Immediately

Optional parameter that specifies the mask that indicates a t1/e1 line in an idle state.

snd-idle-bit-mask

Default Value: `a-off;b-on;`
Valid Values: String:

- `a-off;b-on;`
- `a-off;b-off;`
- `a-on;b-off;`
- `a-on;b-on;`

Changes Take Effect: Immediately

Optional parameter that specifies the mask that the Dialogic board sends to a switch to make a t1/e1 line idle.

Section: Tones

A tone can represent a:

- Busy signal.
- Dial tone.
- Fax machine.
- Ringback.
- Beep signal.

Within each of those broad categories of tones, there are specific types of tones, such as a local dial tone or an international dial tone. See "Tone Options" below for a list of specific tones.

Tone Options

There is a Dialogic option for each type of tone, which the Genesys configuration environment recognizes. These are the `Tone` options:

- `busy-tone-1`
- `busy-tone-2`
- `disconnect-tone`

- `extra-dial-tone`
- `fax-tone-1`
- `fax-tone-2`
- `forth-tone`
- `intl-dial-tone`
- `local-dial-tone`
- `ring-back-tone-1`
- `ring-back-tone-2`
- `sit-tone`

Tone Parameters

A series of numbers separated by semicolons represent these nine parameters for each tone:

- `frequency of first tone`
- `frequency deviation for first tone`
- `frequency of second tone`
- `frequency deviation for second tone`
- `on duration`
- `ontime deviation`
- `off duration`
- `offtime deviation`
- `repetition count`

Note: All parameters inside string values for CPD Server options are separated by semicolons.

For examples of `Tone` parameters, see the default values for the following `Tone` options.

busy-tone-1

Default value: `500;200;0;0;55;40;55;40;4;`
Valid Values: A string of numbers separated by semicolons representing the nine `Tone` parameters (see “Tone Parameters” on page 46).
Changes Take Effect: Immediately

Optional parameter that defines a template for the first busy tone.

busy-tone-2

Default Value: `500;200;500;200;5;40;55;40;4;`
Valid Values: A string of numbers separated by semicolons representing the nine `Tone` parameters (see “Tone Parameters” on page 46).
Changes Take Effect: Immediately

Optional parameter that defines a template for the second busy tone.

disconnect-tone

Default Values: `500;200;500;200;55;40;55;40;10;`
Valid Values: A string of numbers separated by semicolons representing the nine `Tone` parameters.
Changes Take Effect: Immediately

Optional parameter that defines a template for the disconnect tone.

extra-dial-tone

Default Value: `401;125;401;125;0;0;0;0;0;`
Valid Values: A string of numbers separated by semicolons representing the nine `Tone` parameters (see “Tone Parameters” on page 46”).
Changes Take Effect: Immediately

Optional parameter that defines a template for an extra dial-tone.

fax-tone-1

Default Value: `2150;150;0;0;25;-25;0;0;0;`
Valid Values: A string of numbers separated by semicolons representing the nine `Tone` parameters (see “Tone Parameters” on page 46).
Changes Take Effect: Immediately

Optional parameter that defines a template for the first FAX tone.

fax-tone-2

Default Value: `1100;50;0;0;25;-25;0;0;0;`
Valid Values: A string of numbers separated by semicolons representing the nine `Tone` parameters (see “Tone Parameters” on page 46).
Changes Take Effect: Immediately

Optional parameter that defines a template for a second FAX tone.

forth-tone

Default Value: `0;0;0;0;0`
Valid Values: A string of numbers separated by semicolons representing these `Tone` parameters: (see “Tone Parameters” on page 46).

- Frequency of first tone:
  - Frequency range: 200 Hz to 4000 Hz
  - Frequency resolution: 1 Hz
- Amplitude of first tone:

  - (E-1) -40 dBm0 to +0 dBm per tone nominal
  - (T-1) -43 dBm0 to -3 dBm per tone nominal
- Frequency of second tone:
  - Frequency range: 200 Hz to 4000 Hz
  - Frequency resolution: 1 Hz
- Amplitude of second tone:
  - (E-1) -40 dBm0 to +0 dBm per tone nominal
  - (T-1) -43 dBm0 to -3 dBm per tone nominal
- Duration: 10 millisecond increments

Changes Take Effect: Immediately

Optional parameter in which the forth-tone option turns on or off a beep signal that alerts an agent immediately before a customer is connected to that agent. When an agent is in the *engaged* mode and waiting to be connected to a customer, this low-frequency tone notifies the agent that a connection is imminent. The tone frequency, duration, and amplitude of the signal are configurable. CPD Server supports this signal on DM3 hardware only.

intl-dial-tone

Default Value: `402;125;402;125;0;0;0;0;0;`
Valid Values: A string of numbers separated by semicolons representing the nine `Tone` parameters (see "Tone Parameters" on page 46).
Changes Take Effect: Immediately

Optional parameter that defines a template for an international dial-tone.

local-dial-tone

Default Value: `400;125;400;125;0;0;0;0;0;`
Valid Values: A string of numbers separated by semicolons representing the nine `Tone` parameters (see "Tone Parameters" on page 46).
Changes Take Effect: Immediately

Optional parameter that defines a template for a local dial-tone.

ring-back-tone-1

Default Value: `450;150;0;0;130;105;580;415;0;`
Valid Values: A string of numbers separated by semicolons representing nine `Tone` parameters (see "Tone Parameters" on page 46).
Changes Take Effect:

Optional parameter that defines a template for the first ring-back tone.

ring-back-tone-2

Default Value: `450;150;450;150;130;105;58;0;415;0;`
Valid Values: A string of numbers separated by semicolons representing nine `Tone` parameters (see "Tone Parameters" on page 46).
Changes Take Effect: Immediately

Optional parameter that defines a template for a second ring-back tone.

sit-tone

Default Value: `900;1000;5;0;0;0;0;0;0;`
Valid Values: A string of numbers separated by semicolons representing `Tone` parameters.
Changes Take Effect: Immediately

Optional parameter that defines the `SIT` signal parameters in the Dialogic `DX_CAP` data structure.

Note: Please note that SIT tone definitions should be customized only if the default settings are inappropriate for your particular system.

use-busy2-as-nu-tone

optional
Default Value: `no, false`
Valid Values: `no, false, yes, true`

Optional parameter that is located in the Tones section of the CPD Server and controls the way the `Busy2` tone is detected. This options enables detection of a `NU` (Number Unobtainable) tone by using the `Busy2` tone. When this option is set to yes, the `NU` tone call result is assigned to all `Busy2` tone-detected calls.

Chapter

4 Communication Protocols

The information in this chapter is divided among the following topics:

Introduction

This chapter explains the communication protocols for VCB. There are three major originations for callback requests: IVR, Web, and Agent Desktop.

- Agent desktop software can send requests to Universal Callback Server, and Universal Callback Server can send information and acknowledgments to the agent desktop. In that case, User Events are used for information. The `Autodial` mode uses the same communication protocol.
- Information can be transmitted from the IVR, through Universal Callback Server and T-Server, to the agent desktop and back again until the transaction is complete.
- Callbacks can be ordered and processed through the Web API Server. The Web interface provides voice callback submission and management and allows users to request callbacks from the website using the Browser.

Structural Overview

The Universal Callback Server is responsible for callback request processing in a contact center. A typical Genesys system also consists of agent desktops, a T-Server, an IVR, and a Web API application.

A central component of the system is T-Server, a computer-telephony integration (CTI) server that provides a transport layer for all telephony information for the system. All information to and from the agent desktop and other servers passes through T-Server in the form of events.

VCB can process callback requests originating from the Web, using a Web interface that enables callback ordering and management via the Web API Server. See “Web API Protocol” on page 71.

Every call has associated events, all of which contain data. Once data is attached to a call, it is permanent and attached to every event associated with this call.

Event Overview

There are two types of events in the Genesys system: user and telephony.

User Events These include attached `UserData`, and are messages that provide a documented protocol of the interactions between Universal Callback Server and the agent’s desktop application.

Telephony Events These are sent by T-Server and indicate changes in the call status. Every telephony event contains `UserData` which also could be VCB-related. For example, an agent receives notice (`TEventEstablish`) from T-Server that a call has been established and receives attached data along with this event.

Events in Voice Callback

The following topics are discussed in this section:

VCB Client-Server Protocol

The client-server protocol in VCB is based on `UserEvents`, which are conveyed by T-Server(s). See Figure 3.

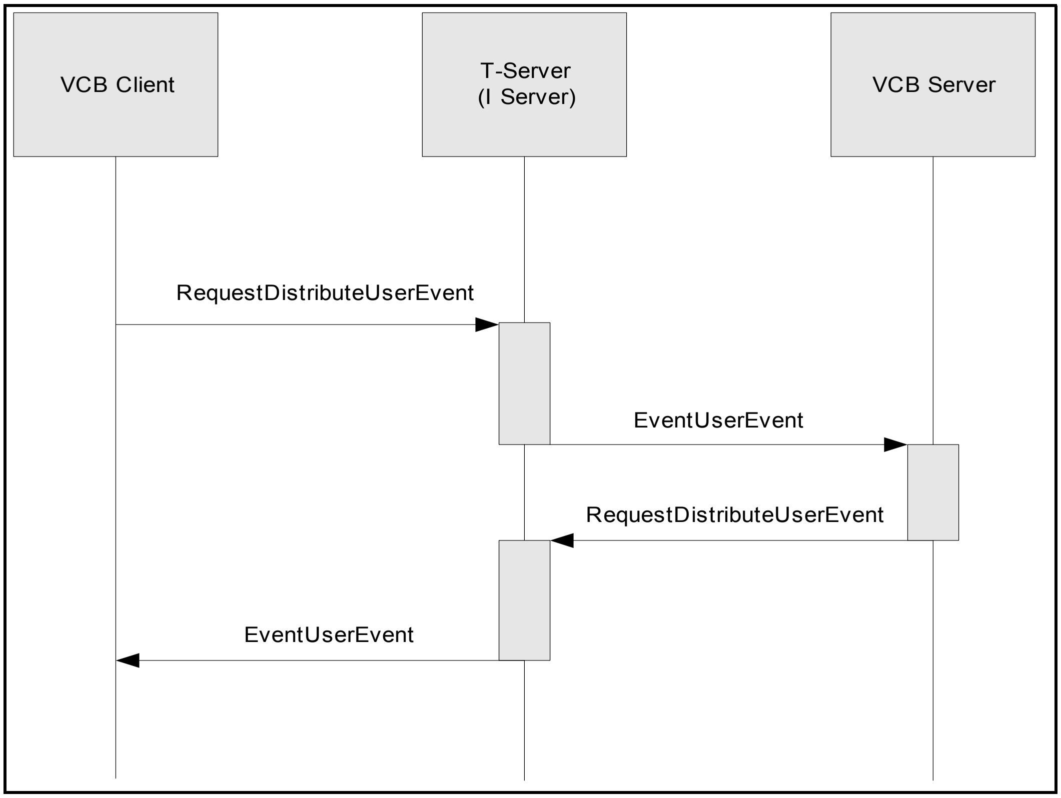


Figure 3: User Events Distribution

Characteristics of Event Structures

There are two types of user events: `requests` and `responses`.

- Requests have this Key-Value pair:

  `VCB_USER_EVENT_REQUEST <Request Type>`

- Responses have this Key-Value pair:

  `VCB_USER_EVENT_RESPONSE <Request Type>`

Predefined Attributes

Table 5 shows the reserved Attribute names for these events.

Table 5: Reserved Attribute Names

Key	Value	Type	Description
VCB_USER_EVENT_REQUEST	Request identifier for any request	String	Request (procedure) name
VCB_USER_EVENT_RESPONSE	Response identifier for any response.	String	Name of corresponding response (procedures)
VCB_MEDIA	Media type	String	"Voice" if not specified upon callback submission (Corresponds to "media" attribute in Web API callback request)
VCB_RESPONSE_TYPE	0, Acknowledge 1, SingleResult 2, MultipleResult 3, MultipleResultEnd 4, Error	Int	Type of response
VCB_ATTEMPTS	Completed number of attempts	Int	Current attempt number.
VCB_INTERACTION_ID	Unique Interaction ID in a form of Alpha-numeric string	String	Interaction ID of the interaction created by Universal Contact Server
VCB_TARGET_APPLICATION_ID	Application ID The value `0` means that the receiver of the event is unspecified. In this case, any Universal Callback Server registered on the designated DN will process the `UserEvent`.	Int	The ID of the application to which the interaction is sent

Table 5: Reserved Attribute Names (Continued)

Key	Value	Type	Description
VCB_ORIGIN_APPLICATION_ID	Application ID The value 0 means that the sender of the event is an unspecified application, but it will be processed.	Int	The ID of the application from which the interaction begins
VCB_REFERENCE_ID	Unique request identifier during a session	Int	The unique ID of the request, maintained throughout the session by the requestor
VCB_CALL_RESULT	0, Ok 3, General Error 4, System Error 6, Busy 7, No Answer 8, SIT Detected 9, Answering Machine 10, All Trunks Busy 11, SIT Invalid Num 12, SIT Vacant (Vacant Code) 13, SIT Oper Intercept 14, SIT Unknown 15, SIT No Circuit 16, SIT Reorder 17, Fax Detected 21, Abandoned 26, Dropped 27, Dropped No Answer 28, Unknown 32, Silence 33, Answer 34, NUTone	Int	

Table 5: Reserved Attribute Names (Continued)

Key	Value	Type	Description
VCB_CALL_RESULT (continued)	35, NoDialTone 36, NoProgress 37, NoRingBack 38, No Established 39, Pager Detected 40, Wrong Party 41, Dial Error 42, Call Drop Error 43, Switch Error 44, No Free Port Error 45, Transfer Error 46, Stale 47, Agent CallBack Error 48, Group CallBack Error	Int	
VCB_ERROR_CODE	Error Code. See Table 11 on page 73.	Int	Is present when `VCB_RESPONSE_TYPE = 4` ("Error")
VCB_ERROR_NAME	Error Name	String	Is present when `VCB_RESPONSE_TYPE = 4` ("Error")
VCB_CURRENT_DN	DN Number	String	DN on which the callback request is currently residing
VCB_CONTACT	String of digits	String	Contact number, such as phone number
VCB_SUBMIT	0 = Not submitted 1 = Submitted successfully 2 = Request to submit	String	If value is `2`, then IVR requests to submit a callback. If value is `1`, the callback request is successfully submitted. If value is `0`, Universal Callback Server rejected the callback request.

Table 5: Reserved Attribute Names (Continued)

Key	Value	Type	Description
VCB_TYPE	0, Unknown 1, ASAP 2, Scheduled	String	Callback Type
VCB_QUEUE	Queue name	String	Queue name used to define search
VCB_STAGE	0, Unknown 1, Queued 2, NotQueued	Int	Used to define search condition by stage of callback request processing. Queued means that callback request is in a queue waiting to be distributed.
VCB_ROUTING_POINT	Name of Routing Point associated with Voice Callback Service	String	To specify Routing Point in request to Universal Callback Server
VCB_ROUTING_POINT_DESCR	Description of Routing Point associated with Voice Callback Service	String	Description that the agent may use to assign an agent-created callback request to most appropriate queue
VCB_LOCATION	Name of Switch	String	To specify Location in request to Universal Callback Server
VCB_DATE_TIME	MMDDYYYYHHMM	String	Date/time to schedule callback
VCB_END_DATE_TIME	MMDDYYYYHHMM	String	Date/time when callback request delivery interval finishes
VCB_EWT	HHMM	String	Estimated Waiting Time (EWT) calculated for callback offering
VCB_EWT_TIME	MMDDYYYYHHMM	String	Date/Time when EWT was estimated

Table 5: Reserved Attribute Names (Continued)

Key	Value	Type	Description
VCB_EWT_OBJECTID	ObjectID in form `<DN Name>@<Switch Name>`	String	Application responsible for initial EWT retrieval and announcement (usually IVR or Router) may specify ObjectID in `UserData` of call or corresponding `UserEvent`, and Universal Callback Server will use EWT for this object to process `Scheduled` callback. If object is specified in `UserData`, it overrides Virtual Routing Point object specified in Routing Point's options.
VCB_ORIGIN	`0`, Unknown `1`, IVR `2`, Web `3`, Desktop	String	Callback origin: party that requested callback.
VCB_TZ_NAME	Time Zone name	String	Configuration Server TZ Name (usually standard three-letter abbreviation, such as PST)
VCB_STATUS	`0`, Unknown `1`, Available `2`, Not Available	String	Availability of callback service
VCB_QUERY_COUNT	`0...`	Int	Number of callback requests returned by RequestCallbackQuery
VCB_QUERY_INDEX	`1...`	Int	Index of callback request in Query Result Set
VCB_LICENSES	`0 . . . N`	String	Additional attribute that Universal Callback Server appends to `UserData` in response to `Request for Service` availability. Number of licenses currently available.

Table 5: Reserved Attribute Names (Continued)

Key	Value	Type	Description
VCB_LICENSE_REF	0	String	Optional. If Value=0, Universal Callback Server recognizes it as a request for license lock. Upon processing of Request for Service availability, Universal Callback Server updates this value by unique reference ID of license lock. If callback submission request contains that attribute with valid reference ID, this request will be accepted.
VCB_RECORDED_DATA	Full path to voice file recorded during callback offering.	String	This attribute contains full names of the voice files in .vox format, delimited by ";". Network configuration provides ability for CPD Server to access these files for read-only. CPD Server plays them all sequentially upon establishing an outbound call to customer. This attribute may present in UserData attached to call or in Client-Server protocol request, RequestCallbackAdd.
VCB_RECORD_HANDLE	Unique callback request identifier in a form of numeric string.	String	Unique callback request identifier created by Universal Callback Server.

Callback Interaction Attribute Names

Universal Callback Server works with Universal Contact Server to create, update, and close callback interactions. Table 6 shows attributes required for callback interaction.

Table 6: Callback Interaction Attribute Names

Attribute	Attribute Name	Description
Status	VCB_STATUS	See Table 12 on page 81.
CallResult	VCB_CALL_RESULT	See Table 5 on page 54.
UserData	N/A	Entire `UserData` from callback request.
StartTime	VCB_DATE_TIME	See Table 5 on page 54.
EndTime	VCB_END_DATE_TIME	Addition to 6.5-7.0
CustomerNumber	VCB_CONTACT	See Table 5 on page 54.
DN	VCB_ROUTING_POINT	Name of RP selected for this callback processing
Location	VCB_LOCATION	Name of switch selected for this callback processing
Attempt	VCB_ATTEMPTS	Current attempt number
CallbackServerID	N/A	ID of Universal Callback Server Application in configuration.
Type	VCB_TYPE	0, “Unknown” 1, “ASAP” 2, “Scheduled”
InteractionID	VCB_INTERACTION_ID	Unique ID of an interaction created by Universal Contact Server.
TenantID	N/A	ID of a Tenant Configuration Object, which belongs to the call center where callback request processing is being done.
ExternalID	VCB_RECORD_HANDLE	See Table 5 on page 54.
TimeShift	VCB_TZ_SHIFT	Time Zone shift calculated from `VCB_TZ_NAME`. See Table 5 on page 54.
Subject	VCB_SUBJECT	String specifying subject

Request Types

Table 7 shows the request types in VCB_USER_EVENT_REQUEST:

Table 7: VCB_USER_EVENT_REQUEST Types

REQUEST TYPE	Description	Action
RequestCallbackServiceStatus	Description	Client sends this request to Universal Callback Server to determine the availability of Voice Callback service. If Routing Point is not specified in this request, Universal Callback Server sends back multiple responses with information about each Routing Point configured for Callback Service.
	Universal Callback Server action	Sends back corresponding responses with requested information
	Mandatory fields	VCB_ORIGIN_APPLICATION_ID VCB_REFERENCE_ID VCB_STATUS: 0
	Additional fields	VCB_LICENSE_REF • If value = 0, Universal Callback Server recognizes it as a request for license lock. Upon processing Request for Service availability, Universal Callback Server updates this value by a unique reference ID of license lock. If callback submission request contains that attribute with valid reference ID, this request is accepted. VCB_ROUTING_POINT VCB_LOCATION

Table 7: VCB_USER_EVENT_REQUEST Types (Continued)

REQUEST TYPE	Description	Action
RequestCallbackQuery	Description	Client sends this request to Universal Callback Server to find callback requests that satisfy search conditions.
	Universal Callback Server action	Sends back corresponding response with query result (number of callback requests found) and `VCB_RESPONSE_TYPE = 0 (Acknowledge)`

	Mandatory fields	`VCB_ORIGIN_APPLICATION_ID`
		`VCB_TARGET_APPLICATION_ID`
		`VCB_REFERENCE_ID`
		At least one of the following:
		`VCB_CONTACT`
		`VCB_TYPE`
		`VCB_ROUTING_POINT`
		`VCB_LOCATION`

	Additional fields	Any fields that have to be included in search conditions
RequestCallbackQueryResult	Description	Client sends this request to Universal Callback Server to retrieve callback request by index in Query Result Set
	Universal Callback Server action	Sends callback request back to Client (`CallbackQueryResult`)

	Mandatory fields	`VCB_ORIGIN_APPLICATION_ID`
		`VCB_TARGET_APPLICATION_ID`
		`VCB_REFERENCE_ID`
		`VCB_Query_INDEX`
		At least one of the following:
		`VCB_CONTACT`
		`VCB_TYPE`
		`VCB_ROUTING_POINT`
		`VCB_LOCATION`

	Additional fields	Not specified

Table 7: VCB_USER_EVENT_REQUEST Types (Continued)

REQUEST TYPE	Description	Action
RequestCallbackCancel	Description	Client sends this request to Universal Callback Server to cancel callback request.
	Universal Callback Server action	Deletes callback request and sends back corresponding response with `VCB_RESPONSE_TYPE = 0 (Acknowledge)`
	Mandatory fields	VCB_ORIGIN_APPLICATION_ID VCB_TARGET_APPLICATION_ID VCB_REFERENCE_ID VCB_RECORD_HANDLE (or VCB_CONTACT) VCB_LOCATION and VCB_ROUTING_POINT are mandatory when RequestCallbackCancel contains VCB_CONTACT (not VCB_RECORD_HANDLE).
	Additional fields	Not specified
RequestCallbackReschedule	Description	Client sends this request to Universal Callback Server to reschedule callback request.
	Universal Callback Server action	Reschedules callback request and sends back corresponding response with `VCB_RESPONSE_TYPE = 0 (Acknowledge)`
	Mandatory fields	VCB_ORIGIN_APPLICATION_ID VCB_TARGET_APPLICATION_ID VCB_REFERENCE_ID VCB_RECORD_HANDLE VCB_CALL_RESULT VCB_DATE_TIME VCB_TZ_NAME
	Additional fields	Not specified

Table 7: VCB_USER_EVENT_REQUEST Types (Continued)

REQUEST TYPE	Description	Action
RequestCallbackAdd	Description	Client sends this request to add callback request.
	Universal Callback Server action	Creates Callback Request object
	Mandatory fields	VCB_CONTACT VCB_TYPE VCB_ORIGIN VCB_ROUTING_POINT VCB_LOCATION
	Additional fields	VCB_ORIGIN_APPLICATION_ID VCB_REFERENCE_ID VCB_TARGET_APPLICATION_ID VCB_LICENSE_REF VCB_RECORDED_DATA Any user-defined fields and default field values (if not specified in a request): VCB_DATE_TIME (if scheduled) VCB_TZ_NAME (if scheduled)
RequestCallbackPreview	Description	Universal Callback Server sends callback request to agent desktop.
	Desktop action	Sends back Acknowledge
	Mandatory fields	VCB_ORIGIN_APPLICATION_ID VCB_TARGET_APPLICATION_ID VCB_REFERENCE_ID VCB_RECORD_HANDLE VCB_CONTACT VCB_CALL_RESULT All UserData attributes from callback request.
	Additional fields	Any user-defined fields

Table 7: VCB_USER_EVENT_REQUEST Types (Continued)

REQUEST TYPE	Description	Action
RequestCallbackProcessed	Description	Agent desktop submits record-processing result to Universal Callback Server.
	Universal Callback Server action	Sends back corresponding response with `VCB_RESPONSE_TYPE = 0 (Acknowledge)` and either updates corresponding callback record in DB as final or applies treatment if specified in configuration.

	Mandatory fields	`VCB_ORIGIN_APPLICATION_ID`
		`VCB_TARGET_APPLICATION_ID`
		`VCB_REFERENCE_ID`
		`VCB_RECORD_HANDLE`
		`VCB_CALL_RESULT`

	Additional fields	Any user-defined fields
RequestCallbackReject	Description	Agent desktop sends this request to Universal Callback Server to reject a callback record.
	Universal Callback Server action	Sends `Acknowledgement` to desktop. Submits callback request to another agent.

	Mandatory fields	`VCB_ORIGIN_APPLICATION_ID`
		`VCB_TARGET_APPLICATION_ID`
		`VCB_REFERENCE_ID`
		`VCB_RECORD_HANDLE`

	Additional fields	Not specified

Response Types

The responses in VCB_USER_EVENT_RESPONSE are in Table 8.

Table 8: VCB_USER_EVENT_RESPONSE Types

Response Type	Description	Action
RequestCallbackAdd	Description	Universal Callback Server sends this response to VCB Client to confirm that the request has been processed.
	Client action	Not specified
	Mandatory fields	VCB_ORIGIN_APPLICATION_ID VCB_TARGET_APPLICATION_ID VCB_REFERENCE_ID VCB_RESPONSE_TYPE (Acknowledge)
	Additional fields	VCB_ERROR_CODE VCB_ERROR_NAME

Table 8: VCB_USER_EVENT_RESPONSE Types (Continued)

Response Type	Description	Action
RequestCallbackServiceStatus	Description	Universal Callback Server sends this response to Universal Callback Server Client to confirm the availability of Voice Callback service.
	Client action	Performs callback request processing

	Mandatory fields	`VCB_ORIGIN_APPLICATION_ID`
		`VCB_TARGET_APPLICATION_ID`
		`VCB_REFERENCE_ID`
		`VCB_STATUS: 0 or 1`
		• If value = `0`, availability of Service is not determined yet.
		`VCB_LICENSES:` Number of licenses currently available
		If `VCB_LICENSE_REF` with value `0` was in request `VCB_LICENSE_REF=<license lock refID>`
		If Routing Point was specified in request, `VCB_RESPONSE_TYPE=1 (SingleResult)`
		If Routing Point was not specified in request, `VCB_RESPONSE_TYPE=2 (MultipleResult)`
		or
		`VCB_RESPONSE_TYPE=3 (MultipleResultEnd)`
		If Routing Point was not specified,
		`VCB_LOCATION`
		`VCB_ROUTING_POINT`
		`VCB_ROUTING_POINT_DESCR`

	Additional fields	`VCB_ROUTING_POINT`
		`VCB_LOCATION`
		`VCB_ERROR_CODE`
		`VCB_ERROR_NAME`

Table 8: VCB_USER_EVENT_RESPONSE Types (Continued)

Response Type	Description	Action
RequestCallbackQuery	Description	Universal Callback Server sends this response back to Client to acknowledge query execution and to provide number of callback requests in the result set.
	Client action	Sends back RequestCallbackQueryResult to request Query Result by Index
	Mandatory fields	VCB_ORIGIN_APPLICATION_ID VCB_TARGET_APPLICATION_ID VCB_REFERENCE_ID VCB_RESPONSE_TYPE=0 (Acknowledge) VCB_QUERY_COUNT
	Additional fields	VCB_ERROR_CODE VCB_ERROR_NAME
RequestCallbackQueryResult	Description	Universal Callback Server sends this response to a Client's CallbackRequestQuery.
	Client action	Collects date
	Mandatory fields	VCB_ORIGIN_APPLICATION_ID VCB_TARGET_APPLICATION_ID VCB_REFERENCE_ID VCB_RESPONSE_TYPE=1 (SingleResult) VCB_RECORD_HANDLE VCB_CURRENT_DN VCB_QUERY_INDEX UserData
	Additional fields	VCB_ERROR_CODE VCB_ERROR_NAME

Table 8: VCB_USER_EVENT_RESPONSE Types (Continued)

Response Type	Description	Action
RequestCallbackCancel	Description	Universal Callback Server sends this response to UCB Client to confirm cancellation of callback request.
	Client action	Waits for next callback request

	Mandatory fields	`VCB_ORIGIN_APPLICATION_ID`
		`VCB_TARGET_APPLICATION_ID`
		`VCB_REFERENCE_ID`
		`VCB_RESPONSE_TYPE=0 (Acknowledge)`
		`VCB_RECORD_HANDLE`
		`(or VCB_CONTACT)`

	Additional fields	`VCB_ERROR_CODE` `VCB_ERROR_NAME`
RequestCallbackReschedule	Description	Universal Callback Server sends this response to the Client’s request for Universal Callback Server to reschedule callback request.
	Client action	Waits for next callback request

	Mandatory fields	`VCB_ORIGIN_APPLICATION_ID`
		`VCB_TARGET_APPLICATION_ID`
		`VCB_REFERENCE_ID`
		`VCB_RESPONSE_TYPE=0 (Acknowledge)`
		`VCB_RECORD_HANDLE`

	Additional fields	`VCB_ERROR_CODE` `VCB_ERROR_NAME`

Table 8: VCB_USER_EVENT_RESPONSE Types (Continued)

Response Type	Description	Action
RequestCallbackPreview	Description	Agent desktop confirms that callback request has been received and processed.
	Universal Callback Server action	Stop `call_preview_agent_ack_timeout` timer. Wait for further events from desktop.

	Mandatory fields	`VCB_ORIGIN_APPLICATION_ID`
		`VCB_TARGET_APPLICATION_ID`
		`VCB_REFERENCE_ID`
		`VCB_RESPONSE_TYPE=0 (Acknowledge)`
		`VCB_RECORD_HANDLE`

	Additional fields	Not specified
RequestCallbackProcessed	Description	Universal Callback Server sends this response to the agent desktop to confirm that the corresponding request has been processed.
	Desktop action	Wait for next callback from Universal Callback Server

	Mandatory fields	`VCB_ORIGIN_APPLICATION_ID`
		`VCB_TARGET_APPLICATION_ID VCB_REFERENCE_ID`
		`VCB_RESPONSE_TYPE = 0 (Acknowledge)`
		`VCB_RECORD_HANDLE`

	Additional fields	Not specified
RequestCallbackReject	Description	Universal Callback Server sends this response to the desktop to confirm that the corresponding request has been processed.
	Desktop action	Wait for next callback from Universal Callback Server.

	Mandatory fields	`VCB_ORIGIN_APPLICATION_ID`
		`VCB_TARGET_APPLICATION_ID`
		`VCB_REFERENCE_ID`
		`VCB_RESPONSE_TYPE = 0 (Acknowledge)`
		`VCB_RECORD_HANDLE`

	Additional fields	Not specified

Web API Protocol

The Web API Server enables Web processing of callback requests and orders. For further information on Web API protocol, see *Multi-Channel Routing Event Media Deployment Guide*.

Web API Requests and VCB Responses

Table 9 shows the mandatory and optional attributes for Web API requests. For description of protocol needed for implementation, see the diagrams in the "Architecture" chapter in the *Voice Callback Deployment Guide*.

Table 9: Web API Requests

Type	Mandatory Attributes	Optional Attributes
Request for Callback	• `refid`—reference ID, type string • `contact`—customer phone number, type string	• `media`—media type (`voice` or `voip`) • `starttime`—scheduled time in form MM/DD/YYYY HH:MM HH:MM (UTC, with Timezone Offset). If absent, callback will be processed ASAP. • `endtime`—time when callback processing should be stopped regardless of Treatments. If absent, value configured in VCB option will be used. • `kvlist`—key-value pairs entered by customer or provided by application.
Request to Cancel Callback	• `refid`— reference ID, type string • `contact`—customer phone number, type string or • `reqid`—unique ID of callback request, type string	• `kvlist`—key-value pairs entered by customer or provided by application.
Request to Search for Callback	• `refid`—reference ID, type string • `contact`—customer phone number, type string	• `reqid`—unique ID of callback request, type string • `media`—media type (`voice` or `voip`) • isdone—accomplishment status of callback request, type boolean

Table 9: Web API Requests (Continued)

Type	Mandatory Attributes	Optional Attributes
Request for Callback Information	• refid—reference ID, type string	• reqid—unique ID of callback request, type string or • contact—customer phone number, type string
Request for Statistical Information	• refid—reference ID, type string	• Contact—customer phone number, type string or • reqid—unique ID of callback request, type string • kvlist—key-value pairs entered by customer or provided by application

Table 10 shows the mandatory and optional attributes for VCB responses:

Table 10: VCB Responses

Type	Mandatory Attributes	Optional Attributes
Acknowledgement Response	• refid—reference ID, type string	• reqid—unique ID of callback request, type string
Error Response	• refid— reference ID, type string • errcode—code of error, type int	
Response for "search" Request	• refid—reference ID, type string • req—sequence of items	
Item of Response for "search" Request	• reqid—unique ID of callback request, type string • contact—customer phone number, type string • media—media type (voice or voip) • starttime—scheduled time • endtime—time when callback processing should be stopped • isdone—accomplishment status of callback request, type boolean • status—callback current status, type string	

Table 10: VCB Responses (Continued)

Type	Mandatory Attributes	Optional Attributes
Request for "getinfo" Request	• refid—reference ID, type string • reqid—unique ID of callback request, type string • contact—customer phone number, type string • media—media type (voice or voip)	• starttime—scheduled time • endtime—time when callback processing should be stopped • kvlist—key-value pairs entered by customer or provided by application
Response for "getstat" Request	• refid—reference ID, type string • queueewt—EWT for queue	

Error Codes

Table 11 shows the error codes pertaining to callback requests.

Table 11: Error Codes

VCB_ERROR	VCB_ERROR_CODE	Description
Invalid Request	100	Invalid request type
Invalid Request Data	101	Some mandatory keys are missed OR Routing information is not provided, and there is no default routing point. Error occurs when mandatory attribute (for example, VCB_CONTACT) is missing from the request.
Invalid Attribute Value	102	Received request has attribute with incorrect value, such as incorrect time format. VCB_ERROR_NAME contains the description.
Invalid Attribute Value	102	Time zone not found
Record not found	103	Error number 103 is no longer used.
Call Already Processed	104	Received request refers to callback request already processed.
DB Error	105	Cannot execute the request due to DB error
AddRecordError	106	Cannot add the record
ScheduledRecordError	108	Error reschedule a record

Table 11: Error Codes (Continued)

VCB_ERROR	VCB_ERROR_CODE	Description
Waiting for Black List Opening	109	Not connected to the database containing Black List table
Number Blocked	110	Phone number blocked for dial
No License Available	111	No license available
TimeNotInSOH	112	Scheduled time is out of the operating hours.

Desktop-Specific Interactions

The Agent Desktop Application is a particular case of a Universal Callback Server Client application. Thus, it uses VCB Client-Server protocol to interact with Universal Callback Server. These requests and their corresponding responses are desktop-specific and may not be used by other clients:

- `RequestCallbackPreview`

  Callback request data that Universal Callback Server sends to the Agent Desktop Application

- `RequestCallbackProcessed`

  Results of this callback request, which the Agent Desktop Application processes and sends back to Universal Callback Server

In the `Autodial` mode, the desktop receives the callback information as data attached to one of these two T-Server events:

- `EventDialing` (direct autodial)
- `EventRinging` (CPD autodial)

Interaction Sequences

This section provides several examples of the Universal Callback Server-Agent Desktop protocol for handling callback requests. The first three examples occur in `Preview` dialing mode, in which Universal Callback Server submits the request to the Agent Desktop. The agent then has an opportunity to review the customer's record and either to accept or reject the call prior to dialing. The last two examples demonstrate how callback requests are handled in `Autodial` mode.

- `Preview` dialing mode
  - Callback Request Accepted
  - Callback Request Rejected
  - Callback Request Query
- `Autodial` mode

- Direct Autodial
- Autodial through CPD Server

Callback Request Accepted

Figure 4 shows the protocol for a callback request that Universal Callback Server, in `Preview` dialing mode, submits to the agent desktop. The agent:

- Acknowledges the request.
- Previews the customer's record.
- Dials the number.
- Informs Universal Callback Server of the call result.

Universal Callback Server acknowledges the outcome (callback request processed or cancelled), as reported by the desktop.

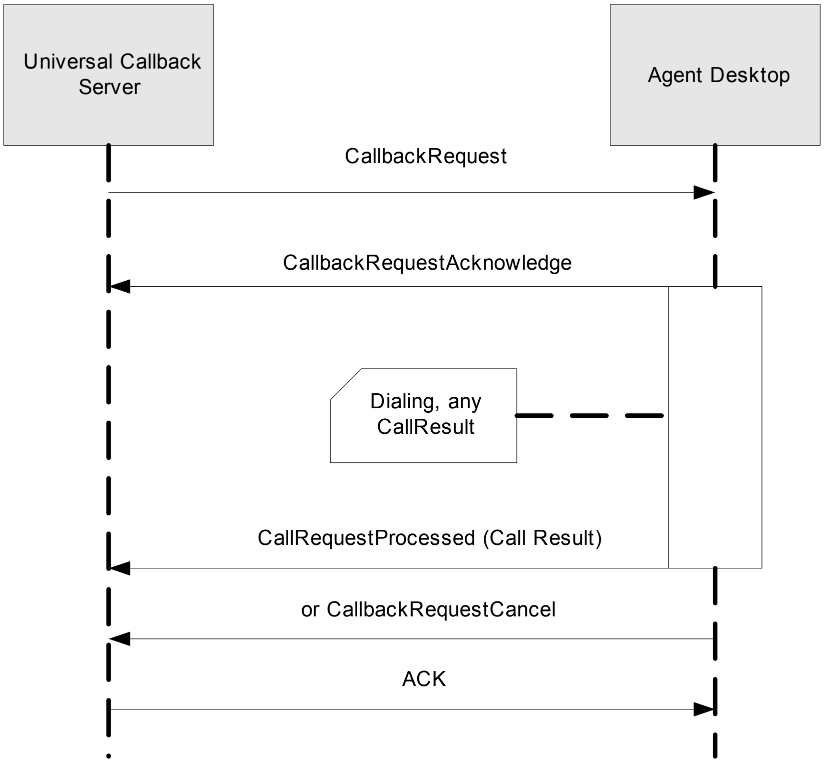

Universal Callback Server
Agent Desktop
CallbackRequest
CallbackRequestAcknowledge
Dialing, any CallResult
CallRequestProcessed (Call Result)
or CallbackRequestCancel
ACK

Figure 4: Preview Dialing Mode: Callback Request Accepted

Callback Request Rejected

Figure 5 shows the Universal Callback Server-Agent Desktop protocol for a callback request that Universal Callback Server, in `Preview` dialing mode, submits to the agent desktop, where an agent acknowledges the callback request but rejects it. Universal Callback Server acknowledges that the callback request is rejected.

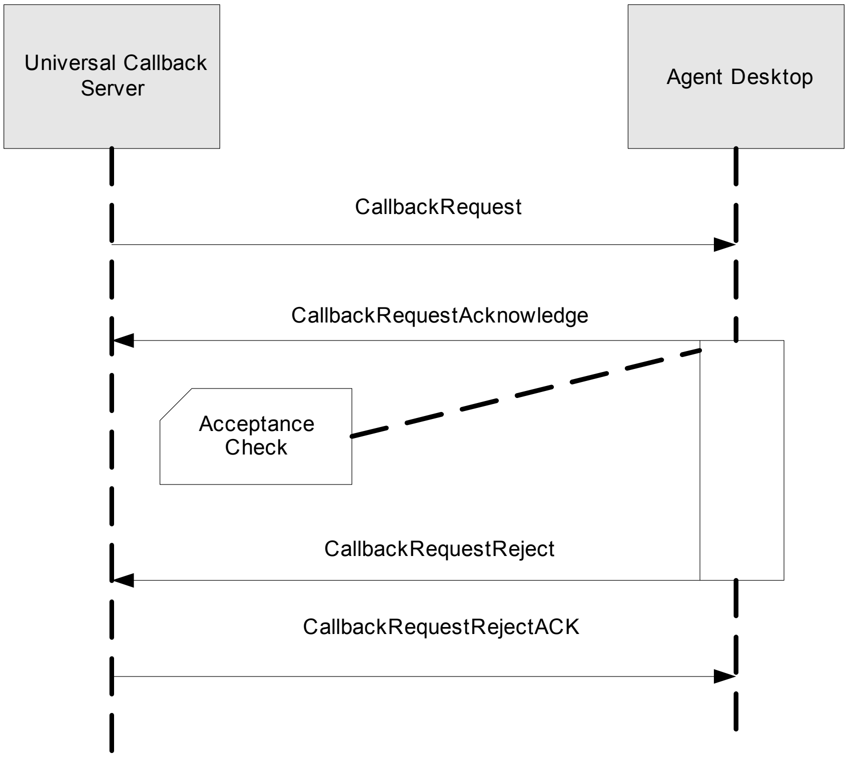


Figure 5: Callback Request Rejected

Callback Request Query

Figure 6 shows the Universal Callback Server-Agent Desktop protocol when the agent queries Universal Callback Server for a callback request, reviews the record, and then either reschedules or cancels the call.

The exchange between Universal Callback Server and the agent desktop ends when Universal Callback Server acknowledges the outcome.

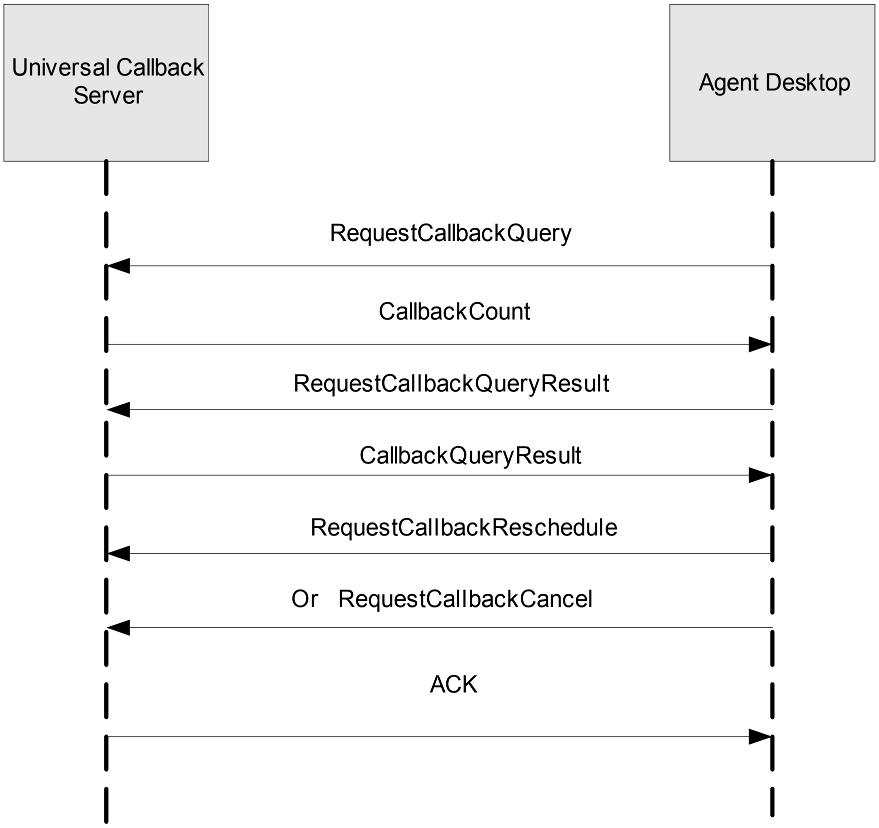


Figure 6: Callback Request Query

Direct Autodial

Figure 7 shows communication between Universal Callback Server, in direct `Autodial` mode, and the agent desktop. Here Universal Callback Server initiates the dialing with a `TMakeCall` request.

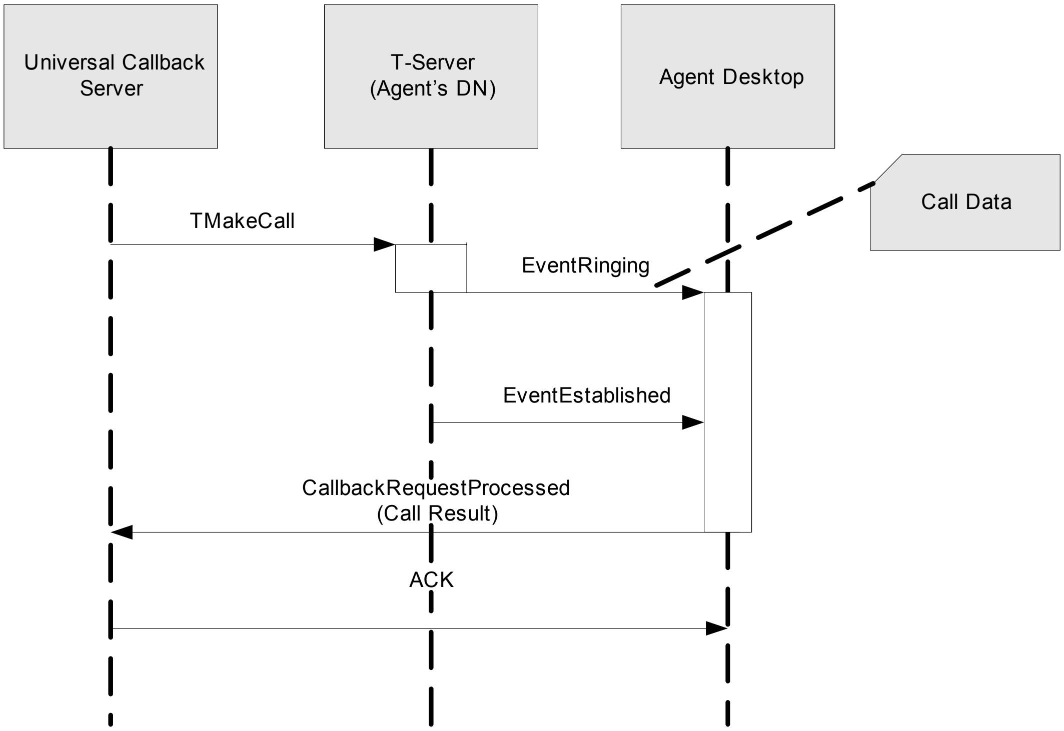


Figure 7: Autodial Mode

Autodial Through TMakePredictiveCall

In this scenario (see Figure 8), the communication between Universal Callback Server and the agent desktop is driven initially by T-Server events `EventRinging` and `EventEstablished` with call data attached. The Agent Desktop then conveys the call result to Universal Callback Server by means of `CallbackRequestProcessed`, which Universal Callback Server acknowledges.

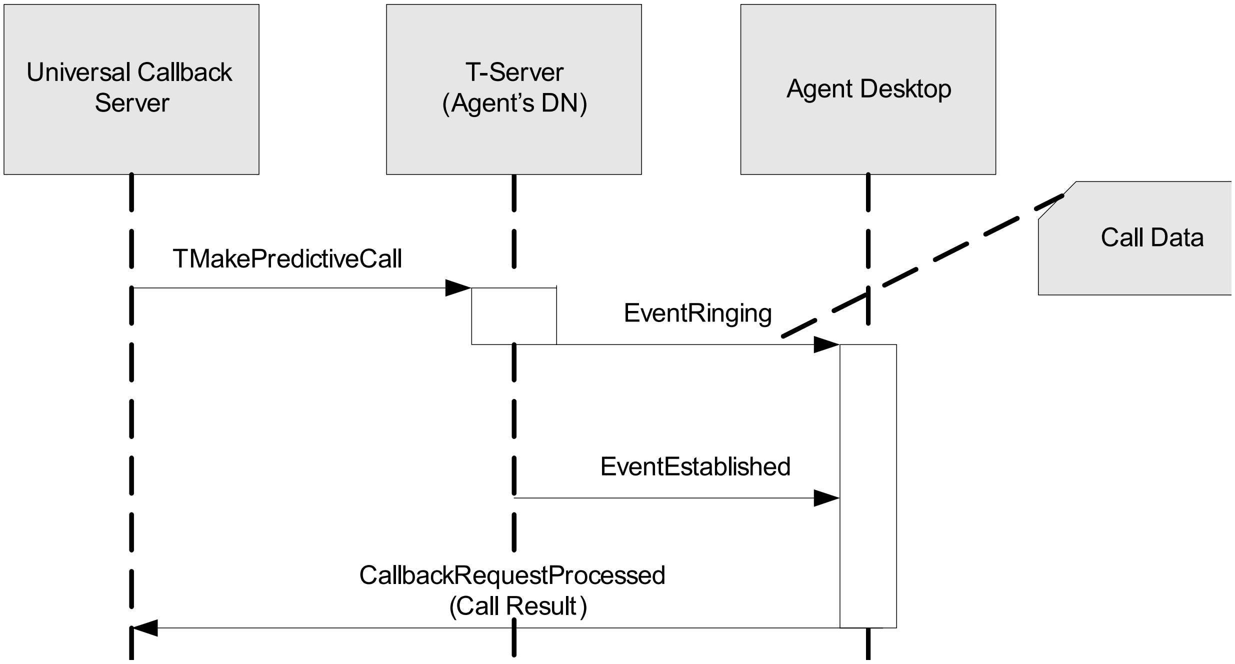


Figure 8: TMakePredictiveCall

Autodial Through CPD Server

When CPD Server is used for autodial, Universal Callback Server requests CPD Server to dial a call. CPD Server dials the call, performs call progress detection, and, in case of a successful call result, transfers the call to an agent or to the queue where the call is then delivered an agent. If the call result is not successful, CPD Server drops the call and notifies Universal Callback Server about the call result for possible treatment application. See Figure 9.

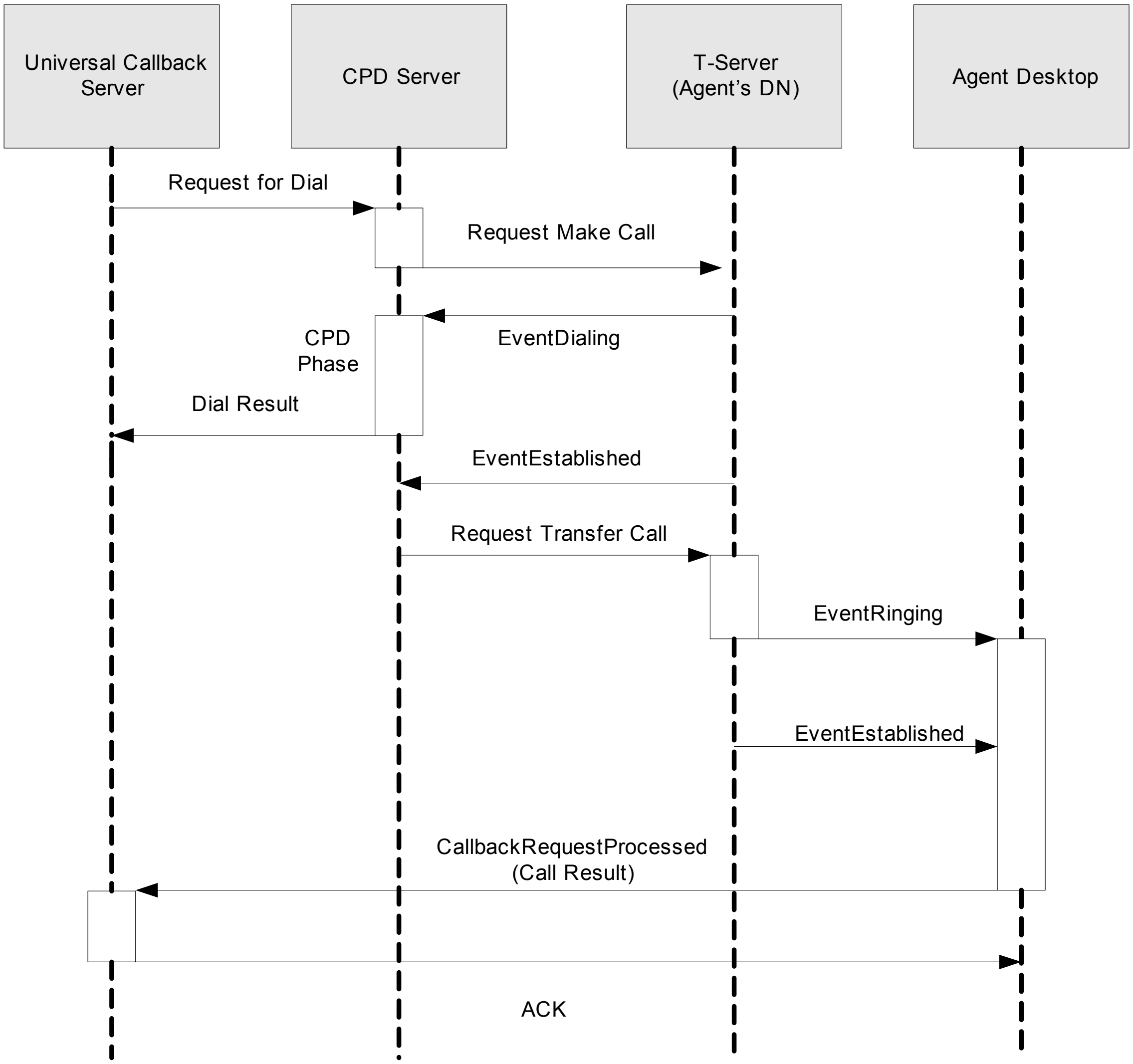


Figure 9: Autodial via CPD Server

Chapter

5 Callback Enumerations

This chapter lists the enumerations that correspond to callback statuses. This enumeration is used by Universal Callback Server when saving information regarding callback requests to the database (column name `callback_status`).

Enumerations for Callback Statuses

Table 12 presents the enumerations that correspond to callback statuses.

Table 12: Enumerations for Callback Statuses

Callback Status	Enumeration
VcbRecStatUnknown	0
VcbRecStatReceived	1
VcbRecStatTimer	2
VcbRecStatQueued	3
VcbRecStatDistributed	4
VcbRecStatSubmittedToAgent	5
VcbRecStatRejectedByAgent	6
VcbRecStatCanceledByAgent	7
VcbRecStatProcessedByAgent	8
VcbRecStatFinal	9
VcbRecStatNotQueued	10

Table 12: Enumerations for Callback Statuses (Continued)

Callback Status	Enumeration
VcbRecStatError	11
VcbRecStatReplaced	12
VcbRecStatAgentTransfer	13
VcbRecStatEndTimeExpired	14

Chapter

6 Treatments

The information in this chapter is divided among the following topics:

How Treatments Work

This section contains a basic description of treatments. The other topics covered in this chapter contain more complex treatment information for those who create and change treatments in Configuration Manager.

Treatment Object

The treatment object tells Universal Callback Server how to respond to an unsuccessful call result (a call that does not reach the intended party). For example, if the first dialing attempt has a Busy call result, a treatment can tell Universal Callback Server to resubmit the same telephone number to the queue after 10 minutes, then submit it to the queue two more times at 30-minute increments. If this is the only treatment set up for the Busy call result, Universal Callback Server redials three times before finishing the call unless another call result is received or the number of maximum dialing attempts is reached.

Treatment Sequence

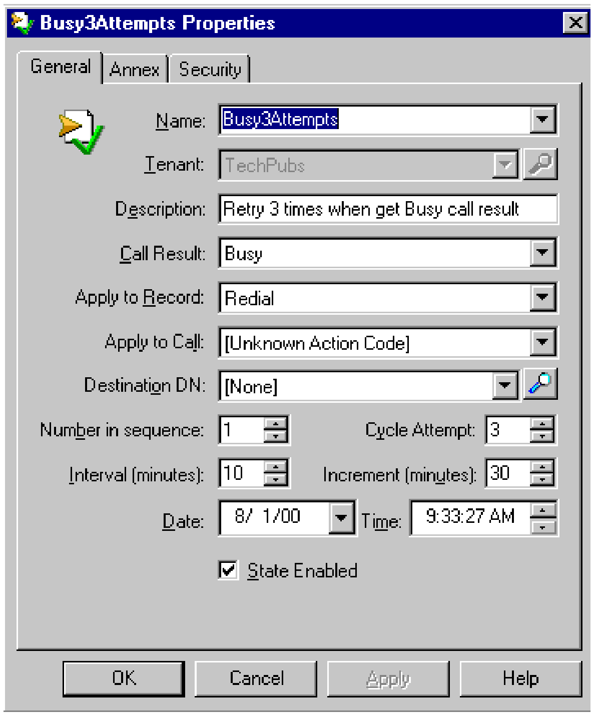
The treatment sequence (sometimes called a linked sequence) describes a series of treatment actions, each assigned a unique sequence number and applied to the same unsuccessful call result. For example, if a callback receives consecutive Busy call results, you may want to use a different treatment action each time.

After you create Treatment objects and then apply them to Calling List objects in Configuration Manager, treatments are generally handled in the following sequence:

1. Universal Callback Server applies each treatment until:
   - A successful call result is received.
   - The treatment actions are exhausted.
   - The number of dialing attempts for the callback reaches the `Maximum Attempts` value assigned in the Calling List object.
2. After a successful call result, Universal Callback Server updates the callback status to final when it receives a response from the agent desktop.
3. If the call never reaches the intended party after Universal Callback Server has exhausted all the treatment actions or has reached the maximum number of dialing attempts, Universal Callback Server:
   - Automatically sets the call result to the unsuccessful call result of the last dial attempt.
   - Sets the callback status to `Final`.
4. The administrator may assign `Transfer call` to the `Answering Machine Detected` or `Fax Detected call results.`
   Universal Callback Server transfers the call to the specified Destination DN.
5. The database stores the final outcome of the last dialing attempt for each callback.

Treatment Properties

Properties are assigned to Treatment objects in the `Treatment Properties` dialog box in Configuration Manager. (See example in Figure 11.)



Name: Busy3Attempts

Tenant: TechPubs

Description: Retry 3 times when get Busy call result

Call Result: Busy

Apply to Record: Redial

Apply to Call: [Unknown Action Code]

Destination DN: [None]

Number in sequence: 1 Cycle Attempt: 3

Interval (minutes): 10 Increment (minutes): 30

Date: 8/ 1/00 Time: 9:33:27 AM

☑ State Enabled

Figure 11: Treatment Properties Dialog Box

For configuration instructions for a Treatment object, see “Treatments” in the *Voice Callback Deployment Guide.*

Table 13 defines the primary treatment properties.

Table 13: Primary Treatment Properties

Treatment Property	Definition
Call Result	Type of response received after dialing a callback's telephone number. An unsuccessful call result (for example, `Busy` or `No Answer`) is assigned to a call that does not reach the intended party.
Apply to Record	Treatment action applied to the next dialing attempt. This value identifies the next action (for example, `Redial`) that Universal Callback Server will take when an unsuccessful call result is received.
Apply to Call	Connection required. Determine if the `Answering Machine Detected` or `Fax Detected` call result is transferred or dropped.
Number in Sequence	A unique number assigned to each treatment in a series of treatments linked to the same call result. This number sorts treatment sequences in the order they are to be applied to consecutive instances of the same call result. Creating multiple treatments for the same call result links the treatments.
Cycle Attempt	The number of times to apply a treatment action if the first dialing attempt has an unsuccessful call result
Interval (minutes)	The number of minutes between cycle attempts if the first cycle attempt has an unsuccessful call result
Increment (minutes)	The increment is added to the interval after the first cycle attempt of an unsuccessful call result. After the second, and all subsequent cycle attempts of an unsuccessful call result, the increment is added to the sum of the last cycle attempt. For example: The interval = 5 minutes The increment = 3 minutes In this scenario the first cycle attempt is 5 minutes. If the first cycle attempt is unsuccessful, the increment (3 minutes) is added to the interval (5 minutes), and the second cycle attempt would be in 8 minutes. For all subsequent unsuccessful cycle attempts, the increment (3 minutes) is added to the sum of the last cycle attempt: (8 + 3) = 11 minutes; (11 + 3) = 14 minutes; and so on.

Timing Properties

Understanding timing properties is essential when creating treatments and applying them to Calling List objects. Table 14 presents details about the timing properties that are required for each treatment action. This information is particularly relevant to the configuration instructions in the section "Treatment Object" in the *Voice Callback Deployment Guide*.

Table 14: Universal Callback Server Supported Timing Properties

Apply to Record Action	Cycle Attempt	Interval (minutes)	Increment (minutes)	Date Time
Redial	Required Set maximum number of retry attempts.	Required Set number of minutes between subsequent cycle attempts.	Required Set number of minutes added to the previous redial time interval.	Not Applicable
Retry in	Not Applicable	Required Set minutes till next (only one) attempt.	Not Applicable	Not Applicable

Treatment Sequences

Treatment sequences allow you to apply different treatments to each consecutive instance of the same call result. For example, if a `Busy` call result is received four times in sequence, you might want to apply a different action to each occurrence. Each treatment in the sequence must have an unique treatment name, an `Apply to Record` action, and an unique, consecutive sequence number. Treatments are linked by assigning the same call result to each treatment in the sequence.

Rules for Treatment Sequences

Apply these general rules to treatment sequences.

1. Always assign consecutive numbers to treatment sequences and always begin with 1. If you use nonconsecutive numbers—such as 1, 2, and 4—the treatment sequence stops at the first nonconsecutive number, which is 4 in this example. The first two treatments would apply, but not the fourth.
2. Universal Callback Server applies each treatment in sequential order until:
   - A successful call result is received.
   - A dialing attempt generates a different unsuccessful call result.
   - Or the number of dialing attempts equals the `Maximum Attempts` value assigned in the Calling List object.
3. If a treatment sequence for one call result (for example, `Busy`) is interrupted with a different call result (`No Answer`), the sequence is broken, and the number in sequence value for `Busy` resets to 1. If the `Busy` call result is received again, the treatment sequence restarts at the beginning.

Treatment Object List

Table 15 shows Treatment objects that will be used in Examples 1 and 2, which follow.

Table 15: Treatment Objects List

Treatment Object Name	Call Result	# in Sequence	Treatment Action (Apply to Record)
Busy1	Busy	1	Redial
Busy2	Busy	2	Retry in (60 min.)
AnsMach1	Answering Machine Detected	1	Retry in (60 min.)
AnsMach2	Answering Machine Detected	2	Retry at specified date
NoAnswer	No Answer	1	Retry in (60 min.)

Example 1: Treatment Sequence Exhausted

In this example, the `Maximum Attempts` value in the Calling List object is `8`, and a record is dialed five times. The results are shown in Table 16.

Table 16: Example 1, Treatment Sequence Exhausted

Call Result	Treatment Action (Apply to Record)	Reached Party?
Busy	Redial	No
Answering Machine Detected	Retry in (60 min.)	No
Busy	Redial	No
Answering Machine Detected	Retry in (60 min.)	No
Answering Machine Detected	Retry at specified date	No
Answer	No treatment	Yes

In Example 1, the call did not reach its intended party and, on the fifth dial attempt, the second number (AnsMach2 from Table 15 on page 88) in the `Answering Machine Detected` treatment sequence is executed.

Example 2: Callback Reaches Maximum Attempts Value

When the number of times the telephone number is dialed matches the `Maximum Attempts` value from the Calling List object, the specified treatment for handling the final unsuccessful call result is applied, and the call result is logged in the database.

In this example, the Calling List object's `Maximum Attempts` value is `8`. Using the same Treatment actions as in Example 1, a callback is dialed eight times, with the results shown in Table 17:

Table 17: Example 2, Callback Reaches Maximum Attempts Value

# of Attempts	Call Result	Treatment Action (Apply to Record)	Reached Party
1	Answering Machine Detected	Retry in (60 min.)	No
2	Busy	Redial	No
3	Answering Machine Detected	Retry in (60 min.)	No
4	Busy	Redial	No
5	Answering Machine Detected	Retry in (60 min.)	No
6	Busy	Redial	No
7	Answering Machine Detected	Retry in (60 min.)	No
8	Busy	Redial	No

In Example 2, the call did not reach its intended party after eight attempts. Because the `Maximum Attempts` value is 8, dialing stops and Universal Callback Server finalizes the callback request processing.

Apply to Record Actions

When creating a Treatment object in Configuration Manager, you must assign an `Apply to Record` action to unsuccessful call results—for example, `Busy` or `No Answer`. See the *Voice Callback Deployment Guide* for complete `Apply to Record` configuration instructions.

Apply to Call Outcomes

There are three `Apply to Call` treatments:

- **Drop.** This is the default if no treatment is specified. Universal Callback Server drops the call and logs the call result as `Answering Machine` or `Fax Detected` in the database.
- **Connect** or **Transfer.** This treatment can be user-assigned only to `Answering Machine Detected` or `Fax Detected` call results. Universal Callback Server transfers calls to the specified DN.

Rules for Call Results

Treatments are normally applied to callbacks for unsuccessful calls.

Two call results—`Answering Machine Detected` and `Fax Detected`—have special treatment rules. The call result `Answer` (by a person) is a successful call result and usually does not require a treatment; however, under certain circumstances `Answer` (successful call) does require a treatment. For a list of call results with descriptions, see Table 1, "Call Result Values," on page 16.

Answering Machine Detected or FAX Detected

Special 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. The administrator specifies the Destination DN in the Configuration Manager. The step-by-step configuration procedures are in "Adding a Destination DN" in the V*oice Callback Deployment Guide*.

Such a configuration allows leaving a message on the customer's Answering Machine or sending the customer a Fax as a Treatment for these two call results.

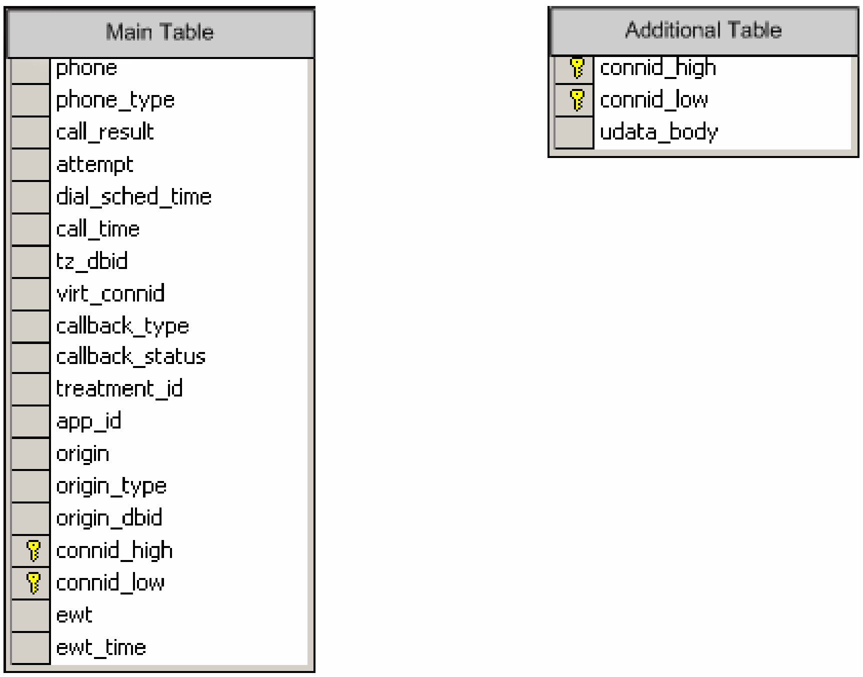
Chapter

7 Database Tables

This section describes the purpose and structure of database tables used as persistent information storage by Universal Callback Server.

Universal Callback Server uses DBMS for the purposes of persistent information storage. In the event that the system goes down and callback requests-related data needs to be recovered for further processing, Universal Callback Server on its startup gathers all the information from the database tables.

Universal Callback Server uses two database tables to store callback requests-related data. Figure 12 on page 94 shows these tables as they are presented by Microsoft SQL Server Enterprise Manager's Database Diagram Wizard.



Main Table

phone
phone_type
call_result
attempt
dial_sched_time
call_time
tz_dbid
virt_connid
callback_type
callback_status
treatment_id
app_id
origin
origin_type
origin_dbid
🔑 connid_high
🔑 connid_low
ewt
ewt_time

Additional Table

🔑 connid_high
🔑 connid_low
udata_body

Figure 12: Database Tables Diagram

Universal Callback Server stores mandatory attributes of the callback requests in the main table which is configured in CME via Calling List object. User data which might be associated with the callback is stored in the additional table, which has a column `udata_body` capable of storing binary data. The name of this additional table is formed by Universal Callback Server from the name of "main" table. The link between the data in two tables is established by (`connid_high, connid_low`) combination of columns.

Table 18 below illustrates all fields from the "main" database table, their types and descriptions.

Table 18: Fields of the "Main" Database Table

Column Name	Data Type	Description
phone	character (64)	Phone number to dial for a callback; attached in `VCB_CONTACT` attribute
phone_type	integer	Zero, not used at this time
call_result	integer	Final outcome of the callback
attempt	integer	Number of attempts made to deliver the callback

Table 18: Fields of the "Main" Database Table (Continued)

Column Name	Data Type	Description
dial_sched_time	integer	Date and time for which the callback has been scheduled: in UTC format (seconds since midnight 01/01/1970)
call_time	integer	Last date and time the callback was delivered: in UTC format
tz_dbid	integer	Configuration DBID of the time zone object; name of the time zone attached in `VCB_TZ_NAME` attribute
virt_connid	character (32)	Connection ID of the virtual call, which represents callback request
callback_type	integer	Type of requested callback (1=`ASAP`, 2=`Scheduled`)
callback_status	integer	Last status of the callback. See *Voice Callback 7.1 Reference Manual*, Enumerations chapter, Table 10.
treatment_id	integer	Zero, not used at this time
app_id	integer	Configuration DBID of the Universal Callback Server Application object which handles callback request
origin	character (220)	Name of the DN on which callback request was submitted
origin_type	integer	Callback origination (0 = `Unknown`, 1 = `IVR`, 2 = `WEB`, 3 = `Desktop`); attached in `VCB_ORIGIN` attribute
origin_dbid	integer	Configuration DBID of the DN object on which callback request was submitted
connid_high	decimal	High part of the Connection ID of the original call which requested a callback
connid_low	decimal	Low part of the Connection ID of the original call which requested a callback

Table 18: Fields of the "Main" Database Table (Continued)

Column Name	Data Type	Description
ewt	integer	Value of estimated waiting time (EWT) in seconds; attached in `VCB_EWT` attribute
ewt_time	integer	A timestamp when EWT was measured, in UTC format; attached in `VCB_EWT_TIME` attribute

Table 19 illustrates all fields from the additional database table, as well as their types and descriptions.

Table 19: Fields of the Additional Database Table

Column Name	Data Type	Description
connid_high	decimal	High part of the Connection ID of the original call which requested a callback
connid_low	decimal	Low part of the Connection ID of the original call which requested a callback
udata_body	binary object	User data associated with the callback

Chapter

8 Licensing

VCB is available by license as an option of Enterprise Routing and Network Routing solutions. See the *Voice Callback Deployment Guide* for a description of the types of licenses used by VCB. This chapter has the following sections:

Note: For deployment information on licensing, see *Voice Callback Deployment Guide*, "Deployment Procedures." For further information on licensing see the *Genesys 7 Licensing Guide*.

CPD Server requires its own license. A customer who purchases VCB can easily add the CPD Server by simply requesting a new keycode for CPD Server.

Server-Side License Control

Table 20 illustrates how license control works in the Universal Callback Server:

Table 20: License Types

License Type	What Is Controlled
`vcb_preview`	Number of callback requests processed in `Preview` mode per 60 minutes (sliding window)
`vcb_full`	Number of callback requests in `Full` mode (that is, `Preview` and/or `Auto Dialing`) per 60 minutes (sliding window)

Initial License Checkout

For each checked out license, Universal Callback Server produces GCTI_LICENSE_CHECKED_OUT log message.

Licensing Enforcement

License availability must be checked before callback submission.

Example

It is important to avoid the following situation: callback service availability is confirmed to customer by IVR, but then the attempt to place callback request is rejected due to lack of available licenses.

To Check License Availability

To resolve this issue, the following approach is used:

- Response for service availability request contains attribute VCB_LICENSES with the number of licenses currently available. In this way, the client is able to avoid overdraft.
- If request for service availability contains the attribute VCB_LICENSE_REF with value “0,” it is recognized by Universal Callback Server as a request to lock the license. In its response, Universal Callback Server updates this attribute value with a unique license reference ID when processing and then decreases the amount of available licenses.

  If a callback submission request contains VCB_LICENSE_REF attribute with a valid license reference ID, this request is accepted, and the license which is referred to in the request is consumed.

License availability is checked before callback submission. For further details see *Voice Callback Deployment Guide*, chapter on “Deployment Procedures.”

It is possible to check the availability and lock of the license before callback submission. See Table 7 on page 61 for Request CallbackServiceStatus.

Configuration Dependencies

Universal Callback Server recognizes the following license-related options; all options are specified in `License` section of the Universal Callback Server Applications object. See Table 21 for a list of license options:

Table 21: License Options

Option Name	Value	Purpose/Range/Comment
num-of-licenses	Number	Universal Callback Server checks out specified amount of licenses for corresponding feature. `Range> = 0` Where `0` means use all available licenses. If this option is not present, Universal Callback Server checks out all available licenses.
license-file	String	License server location or path to license file. If the path/server location is specified in the command line (using -l switch), this option will be ignored by license library, and command line specification will take priority.

Index