GENESYS®
AN ALCATEL-LUCENT COMPANY

Outbound Contact 7.6

Reference Manual

About Genesys

Genesys Telecommunications Laboratories, Inc., a subsidiary of Alcatel-Lucent, 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 DVD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

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

Your Responsibility for Your System's Security

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

Trademarks

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

Technical Support from VARs

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

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North and Latin America	+888-369-5555 or +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-1276-45-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp

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

Ordering and Licensing Information

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

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 76ou_ref_05-2008_v7.6.101.00

Table of Contents

Preface

Welcome to the *Outbound Contact 7.6 Reference Manual.* In this manual, we provide reference information for performing configuration and installation procedures for Outbound Contact. We also explain the options, features, and functionality of this product.

This manual is valid only for the 7.6.x releases of Outbound Contact.

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

This chapter identifies the primary audience, summarizes the content of this manual, introduces document conventions, and lists related reference and contact information:

Outbound Contact 7.6 is an automated system for creating, modifying, running, and reporting on outbound campaigns for proactive customer contact.

Outbound Contact supports automated dialing and call progress detection, so an agent is only required when a customer is connected. It also intelligently uses customer data to ensure that campaigns are contacting not only a large number of customers, but the “right” customers for your purposes.

Intended Audience

This guide is primarily intended for system engineers and other members of an implementation team who set and maintain Outbound Contact 7.6. This guide assumes that you have a basic understanding of:

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

You should also be familiar with Genesys Framework architecture and functions that support Outbound Contact 7.6.

Chapter Summaries

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

- Chapter 1, "Communication Protocols," on page 15, explains the communication protocol (requests and responses) between the Outbound Contact Server (OCS) and the Desktop during the processing of a call.
- Chapter 2, "Communication DN API," on page 95, discusses the Communication DN (CommDN) API provided by the Outbound Contact to allow third-party applications to control campaign scheduling and sequencing and to submit Do Not Call (DNC) requests.
- Chapter 3, "Defined Constants," on page 113, explains the mandatory fields, enumeration values, call results, data types, phone and record types, and record statuses in Outbound Contact.
- Chapter 4, "Recommended DBMS Optimizations," on page 129, contains information on how to maximize your DBMS for your Outbound Contact solution.
- Chapter 5, "Supported Functionality with IP Telephony," on page 133 describes how to integrate the Outbound Contact with Genesys IP products.

Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

Document Version Number

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

`ou_ref_05-2008_v7.6.101.00`

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 for emphasis, for documents' titles, for definitions of (or first references to) unfamiliar terms, and for mathematical variables.

Examples:

- Please consult the *Genesys 7 Migration Guide* for more information.
- *A customary and usual practice* is one that is widely accepted and used within a particular industry or profession.
- Do *not* use this value for this option.
- The formula, x +1 = 7 where x stands for . . .

Monospace Font

A monospace font, which looks like `teletype or typewriter text`, is used for all programming identifiers and GUI elements.

This convention includes the *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages; the values of options; logical arguments and command syntax; and code samples.

Examples:

- Select the `Show variables on screen` check box.
- Click the `Summation` button.
- In 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.

Monospace is also used for any text that users must manually enter during a configuration or installation procedure, or on a command line:

Example:

- Enter `exit` on the command line.

Screen Captures Used in This Document

Screen captures from the product GUI (graphical user interface), as 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 *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Square Brackets

Square brackets indicate that a particular parameter or value is optional within a logical argument, a command, or some programming syntax. That is, the parameter's or value's presence is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information. Here is a sample:

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

Angle Brackets

Angle brackets indicate a placeholder for a value that the user must specify. This might be a DN or port number specific to your enterprise. Here is a sample:

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

Related Resources

Consult these additional resources as necessary:

- The *Outbound 7.6 Deployment Guide,* which provides instructions for installing and configuring Outbound Contact 7.6 components by using the configuration wizards and Configuration Manager.

- *Outbound Contact Manager 7.6 Help,* which describes how to use Outbound Contact Manager.
- The *Genesys 7.6 Proactive Contact Solution Guide,* which consolidates information about the Genesys Proactive Contact solution. The Genesys Proactive Contact solution integrates Outbound Contact with Genesys Voice Platform (GVP), and provides the ability to proactively initiate and handle outbound campaign calls using GVP.
- The *Genesys Proactive Routing Solution Guide,* which provides instructions for integrating Outbound Contact with the Customer Interaction Management (CIM) Platform.
- The *Framework 7.6 Deployment Guide*, which will help you configure, install, start, and stop Framework components.
- The *Framework 7.6 Configuration Options Reference Manual,* which provides you with descriptions of configuration options for other Framework components.
- *Framework 7.6 Configuration Manager Help,* which helps you use Configuration Manager.
- The *Genesys 7 Migration Guide*, also on the Genesys Documentation Library CD, which contains a documented migration strategy from Genesys product releases 5.x and later to all Genesys 7.x releases. Contact Genesys Technical Support for additional information.
- The *Genesys 7 Events and Models Reference Manual*, which contains the T-Library API, information on `TEvents,` and an extensive collection of call models.
- The *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library CD and which provides a comprehensive list of the Genesys and CTI terminology and acronyms used in this document.
- The Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at `http://genesyslab.com/support`.

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

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

Genesys product documentation is available on the:

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

Making Comments on This Document

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

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

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

Document Change History

This section lists topics that are new in the current release of this document, or that have changed significantly from the preceding release.

New in Version 7.6.1

Table 1 provides details about what is new or has changed significantly from release 7.6 to 7.6.1 of this document.

Table 1: Document Changes

Chapter	Heading/Topic	Page	Details
Chapter 1	`ReadyTime` Request	42	To increase campaign performance in the Predictive/Predictive with seizing dialing modes for small groups, desktop sends the `ReadyTime` event to OCS, providing the estimated time (in seconds) remaining until the agent will become Ready.
	`LogOut` Request		Revised OCS Action section, indicating that if there are no `Sent`, `Dialed` and `Queued` calls, OCS sends a LogOut time equal to `0`.

Table 1: Document Changes (Continued)

Chapter	Heading/Topic	Page	Details
Chapter 5	All H.323 references and scenarios	133	Removed from this chapter, as this type of signalling/ architecture is not fully supported by Genesys applications.
	Overview chapter	133	Revised Cisco CallManager support text.
	Overview > HMP Licensing	136	New section describing CPD Server and HMP licensing.
	Outbound Contact with Cisco CallManager	146	New section describing the architecture/call flow scenarios for Outbound Contact with Cisco CallManager.

Chapter

1

Communication Protocols

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

Introduction

This chapter explains the Outbound Contact Server (OCS) and Desktop communication protocol. The desktop uses this protocol to send requests to OCS, and OCS uses it to send information and acknowledgments to the desktop and the calling list database.

This chapter also describes the overall process of transmitting information from the calling list database, through OCS and T-Server, to the agent desktop and back again until a call transaction is complete.

Note: For information about the Multimedia desktop protocols used in Push Preview dialing mode (also known as Proactive Routing Solution), see the *Genesys Proactive Routing Solution Guide.*

Event Overview

There are telephony and user events in Genesys.

- *Telephony events,* which T-Server sends, indicate changes in the call status. Every telephony event contains outbound data from the calling list database, which OCS sends to T-Server with the request to make a call. An agent receives notice (`EventEstablished`) from T-Server that a call has been established and receives attached data along with this event. Every call has approximately 10 different 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.
- *User events,* which include attached user data, are messages that provide a documented protocol of the interactions between OCS and the agent's desktop application.

Characteristics of Event Structures

There are two types of user events:

- Agent desktop request to OCS. All messages that travel from the desktop to OCS have the key `GSW_AGENT_REQ_TYPE`.
- OCS to desktop, either:
  - a response to a desktop request
  - an unsolicited notification from OCS.

  All messages that travel from OCS to the desktop have the key `GSW_USER_EVENT`.

When OCS retrieves a record, it creates a unique record identifier. (`GSW_RECORD_HANDLE`) that identifies the record. This attribute identifies the record to which attached data pertains in a user event. Any communication between the desktop and OCS concerning this record requires a key value with the GSW_RECORD_HANDLE. The key-value GSW_RECORD_HANDLE is internally generated and is not related to the RECORD_ID field of the call record.

All requests having GSW_RECORD_HANDLE as a mandatory field receive the Record Not Found response error if the record is not in the internal OCS buffers (for example, the record was already processed).

All events, whether they are from the desktop or from OCS, should have the key-value pair GSW_APPLICATION_ID <Int>, which is the OCS application ID (sometimes called the OCS DBID in the Configuration Server database). In Outbound Contact, after an agent logs in, OCS sends a campaign status notification to the agent desktop. The OCS application ID sent with this notification is attached to every request sent to OCS. Only the OCS with the matching GSW_APPLICATION_ID responds to the request.

In case the primary and backup OCS have been switched, the OCS that just became primary notifies all logged-in desktops about the change by sending them a user event with new GSW_APPLICATION_ID and the current statuses of loaded and running campaigns. Then all desktops will use the new GSW_APPLICATION_ID in their communications with the new primary OCS, but they remember for a while a previous GSW_APPLICATION_ID to let the backup OCS finish its work with the records started while it was the primary OCS.

Key-value pairs of a user event may be sent in any order. The desktop applications recognize the key-value pairs by the key and not by the sequence of the attached key-value pair.

Event Responses

When a desktop request to OCS is related to a specific record (using GSW_RECORD_HANDLE in the key-value pair), the desktop must explicitly tell OCS that it has finished with the record, using the RecordProcessed request. The RecordProcessed request signals the final transaction for the record. The only requests that do not need a RecordProcessed request are DoNotCall and RecordCancel requests related to an open record (hence sent from the desktop to OCS with GSW_RECORD_HANDLE).

With the exception of ChainedRecordRequest, OCS acknowledges all events separately, by sending either an acknowledgment, an error, or the requested data. ChainedRecordRequest is the only request that OCS returns with multiple responses. ChainedRecordRequest responds with each record in the chain, and ChainedRecordDataEnd signals the end of the user event.

Error Events and Messages

OCS sends an error event, via T-Server, when OCS cannot interpret the desktop request. The error message conveys the reason for the failure.

All error events should have the key-value pair GSW_ERROR <Error Name> in the attached data.

The key-value pairs in Table 2 should be contained in all error events.

Table 2: Error Event Attributes

Key	Type	Description
GSW_ERROR	String	Error name; See "Error Names and Codes" on page 89.
GSW_ERROR_NUMBER	Int	Mandatory

Table 94 on page 89 lists the OCS error messages sent to the desktop, their corresponding values, and possible diagnostics.

Attaching Record Information to Desktop and OCS User Events

A calling list contains two types of fields: Genesys mandatory fields and custom (user-defined) fields. The value of these fields can be attached to user events (and telephony events) as user data. The attached data is then sent as a pair, called a key-value pair.

Default Record Information

The value of certain fields from each calling list record is attached to all telephony and user events by OCS, by default. Key-value pairs might include, for example:

- phone number (key GSW_PHONE)
- chain ID of the record (key GSW_CHAIN_ID)
- call result (key GSW_CALL_RESULT)

These pairs are sent when a user event, or telephony event, is related to handling a specific calling list record. The pair with the key GSW_RECORD_HANDLE is attached to outbound-related events as a unique record identifier. Genesys recommends the following: The desktop application should not change the value of these key-value pairs (except GSW_CALL_RESULT).

Table 3 shows list of key-value pairs that OCS attaches to outbound call's user data by default.

Table 3: Default Record Information

Key	Type	Description
GSW_APPLICATION_ID	Int	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_ATTEMPTS	Int	Number of attempts for the record.
GSW_CALLING_LIST	String	Name of the calling list.
GSW_CAMPAIGN_NAME	String	Name of the campaign.
GSW_CALL_RESULT	Int	Call result enumeration
GSW_CHAIN_ID	Int	Unique Chain ID
GSW_PHONE	String	Customer's phone number.
GSW_RECORD_HANDLE	Int	Unique Record Identifier.
GSW_TZ_OFFSET	Int	Offset (time difference) in seconds between Universal Time Coordinated (UTC) and a particular time zone. It may contain different values throughout the year if Daylight Savings Time (DST) is used for the specified time zone.
GSW_CALL_ATTEMPT_GUID	String	Global unique identifier of the call processing attempt used for historical reporting.
GSW_CAMPAIGN_GROUP_NAME	String	The name of the Campaign Group.
GSW_CAMPAIGN_GROUP_DESCRIPTION	String	The description of the Campaign Group.

Send Attributes

If the value of a field is not attached by default, and you wish to include its value in the user data, then you must define the option named send_attribute. in the field configuration object which corresponds to the field of the value you want to be attached.

The value of the send_attribute option defines the key of the pair that will be attached to the user data. The value of the field is the value of the pair.

For example, a calling list might have a user-defined field for customer_name. If you want to send the content of the field customer_name (John Doe, for example) to the desktop, you would set up the option send_attribute with the `value = customer_name`. The desktop will then receive the attached data with the `key = customer_name` and the `value = John Doe`.

You can define the `send_attribute` option in the Configuration Database, on the `Annex` tab of the field configuration object. For information about how to set up an option, refer to the chapter "Outbound Contact Configuration Options" in the *Outbound Contact 7.6 Deployment Guide* for more information.

Note: The field name and the value of the `send_attribute` option generally do not need to match. They could be two different string values.

User Event Attributes

Table 4 shows a list of user event attributes that OCS uses to communicate with Stat Server for reporting purposes. The event type `GSW_STAT_EVENT` is the mandatory attribute for these events.

Table 4: User Event Attributes

Key	Type	Description
GSW_STAT_EVENT	Int	Event Type
GSW_CAMPAIGN_DBID	Long (integer)	Reference to campaign DBID of CFGCampaign object from Configuration Server
GSW_CALL_LIST_DBID	Long (integer)	Reference to calling list in campaign DBID of CfgCallList from Configuration Server
GSW_GROUP_DBID	Long (integer)	Reference to group in campaign DBID of CfgGroup
GSW_AGENT_DBID	Long (integer)	Reference to agent DBID of CFGPerson in Configuration Server
GSW_CALL_RESULT	Int	Call Result
GSW_CAMPAIGN_COMPLETE	Int	Estimated time to complete campaign

Table 4: User Event Attributes (Continued)

Key	Type	Description
GSW_LIST_COMPLETE	Int	Estimated time to complete calling list
GSW_ERROR_DESCRIPTION	String	Error description
GSW_DIAL_MODE	Int	Dial mode. Valid values are 1 - 7
GSW_APPLICATION_ID	Int	OCS application DBID Valid values begin at 101
GSW_CALLBACK_TYPE	Int	Callback Type
GSW_SCHED_REC_NUM	Int	Number of scheduled records in process. Valid values begin at 0.

Updating Genesys Mandatory Fields and Custom Fields

The desktop can use the request RecordProcessed or UpdateCallCompletionStats to modify the values in Genesys mandatory fields and custom fields. See Table 5 for modifiable mandatory Genesys fields.

Genesys Mandatory Fields

The following table contains the only Genesys mandatory fields that are modifiable by the RecordProcessed or UpdateCallCompletionStats events.

Table 5: Modifiable Mandatory Genesys Fields

Genesys Mandatory Field Name	Recommended Key for send_attribute	Type	Description
call_result	GSW_CALL_RESULT	Int	Sent to change an automatically detected call result. See Table 8 on page 28 and Table 112 on page 115.
daily_from	GSW_FROM	Int	GSW_FROM to GSW_UNTIL: Time frame when a record can be called, seconds from midnight (system or local time).

Table 5: Modifiable Mandatory Genesys Fields (Continued)

Genesys Mandatory Field Name	Recommended Key for send_attribute	Type	Description
contact_info	GSW_PHONE	String	Customer's phone number.
contact_info_type	GSW_PHONE_TYPE	Int	Customer phone type. See Table 8 on page 28.
daily_till	GSW_UNTIL	Int	GSW_FROM to GSW_UNTIL: Time until a record can be called, seconds since midnight (system or local time).

Custom Data Formats

The data type of custom fields may change as data is attached to a call; the attached data can then be sent to the desktop as user data. Integer data is sent as an integer. All other data is sent as a string.

Custom data should be formatted as shown in Table 6.

Table 6: Custom Data Formats

Data Type in Calling List	User Data Format
FLOAT	STRING
CHAR	STRING
DATETIME	STRING
INT	INTEGER
VARCHAR	STRING

Reserved Keys

The key names in Table 7 are reserved and cannot be used as the send_attribute for custom fields. The values associated with some of these keys can be changed; others cannot. The primary source of data for the values in this table is the calling list database. Values for all keys of type String are case sensitive and should appear in desktop application code exactly as shown in the Values column.

Note: In Outbound Contact, all reserved key names include the prefix `GSW_`. Do not use this prefix for custom key names that you define using `send_attribute`.

Table 7: Reserved Keys

Key	Values	Type	Description
GSW_AGENT_ID		String	Login ID of last agent who worked with the record.
GSW_AGENT_REQ_TYPE		String	Event identifier for events coming from desktops to OCS.
GSW_APPLICATION_ID	101 …	Int	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_ATTEMPTS	0…	Int	Number of attempts for the record. This key is used when a new record is added.
GSW_CALL_ATTEMPT_GUID		String	Global unique identifier of the call processing attempt used for historical reporting (same value as in the primary for all the chained records).
GSW_CALL_RESULT		Int	Call Result saved from previous call, or Call Result sent to change automatically detected call result. See Table 8 on page 28 and Table 112 on page 115.
GSW_CALL_TIME	0…	Int	System time when record was called, in seconds from 1/1/70 (GMT). This key is used when a new record is added.
GSW_CALLBACK_TYPE	Personal, Campaign	String	Type of callback an agent wants to create, either Personal or Campaign.
GSW_CALLING_LIST		String	Name of the calling list.

Table 7: Reserved Keys (Continued)

Key	Values	Type	Description
GSW_CAMPAIGN_DESCRIPTION		String	Description of campaign. Value may be an empty string.
GSW_CAMPAIGN_MODE	Power GVP, Predictive, Predictive with Seizing, Preview, Progressive, Progressive with Seizing, Push Preview	String	Campaign dialing mode. See Table 8 on page 28. The values Engaged Predictive and Engaged Progressive correspond to the dialing modes Predictive with seizing and Progressive with seizing.
GSW_CONTACT_MEDIA_TYPE	any, email, voice	String	Media type for the calling record that corresponds to the value of the contact_info_type field in the calling list.
GSW_CAMPAIGN_GROUP_DBID		String	The DBID of the campaign group.
GSW_CAMPAIGN _GROUP_NAME		String	The name of the Campaign Group.
GSW_CAMPAIGN_GROUP_ DESCRIPTION		String	The description of the Campaign Group. Value may be an empty string.
GSW_CAMPAIGN_NAME		String	The name of the campaign.
GSW_CHAIN_ATTR	AllChain, RecordOnly	String	Flag determining whether to update the record chain or just the single record.
GSW_CHAIN_ID	0…	Int	Unique chain ID.
GSW_CHAIN_N	0...	Int	Unique identifier of record in a chain.
GSW_CUSTOMER_ID		String	Customer ID that is used for requests.

Table 7: Reserved Keys (Continued)

Key	Values	Type	Description
GSW_DATE_TIME	A string represented in time in this format: MM/DD/ YY(YYYY) HH:MM.	String	Date and time of scheduled call.
GSW_ERROR	Error name	String	Error name. See Table 95 on page 91.
GSW_ERROR_DESCRIPTION	Error description	String	Error name. See Table 95 on page 91.
GSW_ERROR_NUMBER	Error Number	Int	Error code. See Table 95 on page 91.
GSW_FROM	0…	Int	GSW_FROM - GSW_UNTIL: Time frame when a record can be called, seconds from midnight.
GSW_LOGOUT_TIME	1. . .N	Int	Time remaining, in seconds, before an agent may log out after an unsuccessful logout attempt.
GSW_MESSAGE		String	DoNotCall message.
GSW_PHONE		String	Customer’s phone number.
GSW_PHONE_TYPE		Int	Customer phone type. See Table 8, “Enumeration Table,” on page 28.
GSW_RECORD_HANDLE	1…	Int	Unique Record Identifier.
GSW_RECORD_STATUS	See Table 8, “Enumeration Table,” on page 28.	Int	Status of adding record sent from a desktop.

Table 7: Reserved Keys (Continued)

Key	Values	Type	Description
GSW_RECORD_TYPE	See Table 8, "Enumeration Table," on page 28.	Int	Type of added record sent from a desktop.
GSW_SCRIPT_ID		Int	DBID of the `Script` Configuration Object.
GSW_SWITCH_DBID		Int	DBID of the `Switch` object
GSW_TREATMENT	RecordTreat Personal, RecordTreat Campaign	String	Specifies the treatment that should be applied to a record chain for RecordProcessed event.
GSW_TZ_NAME		String	Configuration Server time zone name (usually a standard three-letter abbreviation).
GSW_TZ_OFFSET	-43200 … 43200	Int	Offset (time difference) in seconds between UTC and a particular time zone. It may contain different values throughout the year if Daylight Savings Time (DST) is used for the specified time zone.
GSW_UNTIL	0… > GSW_FROM	Int	GSW_FROM - GSW_UNTIL: Time frame when a record can be called (in seconds from midnight).
GSW_USER_EVENT	Event Type, see Table 95 on page 91.	String	Event identifier for events coming from OCS to desktops.

Table 7: Reserved Keys (Continued)

Key	Values	Type	Description
InteractionType	Outbound	String	Type of the interaction that created by OCS. The value of this key is always set to `Outbound`.
InteractionSubtype	OutboundNew	String	Subtype of the interaction, that is created by OCS. The value of this key is always set to `OutboundNew`.

Genesys Enumeration Tables

Some Genesys mandatory fields in a calling list table are represented as predefined integer constants. When these fields are attached to user events or telephony events as key-value pairs, the values of these fields are sent as integers (sometimes called enumeration values or internal representations). Table 8 lists the Genesys mandatory fields that are sent as enumeration values and their corresponding descriptive strings displayed in various applications (such as Outbound Contact Manager and Configuration Manager). The desktop application should translate the enumeration value to the appropriate description when required for display.

Table 8: Enumeration Table

Genesys Mandatory Field in Calling List Table	Key	Enumeration Value	Data Type in User Event	Description
call_result	GSW_CALL_RESULT	See Table 112 on page 115 for the call result enumeration values and descriptions.	Int	Call result saved from the previous call, or the call result sent to change an automatically detected call result.
contact_info_type	GSW_PHONE_TYPE	0, No Contact Type 1, Home Phone 2, Direct Business Phone 3, Business With Ext 4, Mobile 5, Vacation Phone 6, Pager 7, Modem 8, Voice Mail 9, Pin Pager 10, E-mail Address	Int	Customer phone type.

Table 8: Enumeration Table (Continued)

Genesys Mandatory Field in Calling List Table	Key	Enumeration Value	Data Type in User Event	Description
record_status	GSW_RECORD_ STATUS	0, No Record Status 1, Ready 2, Retrieved 3, Updated 4, Stale 5, Canceled 6, Agent Error 8, Missed CallBack	Int	Status of adding record sent from a desktop.
record_type	GSW_RECORD_TYPE	0, No Record Type 1, Unknown 2, General 3, Campaign Rescheduled 4, Personal Rescheduled 5, Personal CallBack 6, Campaign CallBack 7, No Call	Int	Type of record sent from a desktop.

Attaching Script Information to OCS User Events and Telephony Events

The Configuration Object Script with the Type `Outbound Campaign` defines all of the attributes that are required by Agent Scripting.

References to this script can be defined in the Script combo box of a Campaign, Calling List, or Campaign Group Configuration Objects.

When a script is defined in either of these objects: Outbound Contact Server attaches the DBID of the corresponding Object Script to a User Data of an Outbound Call or Preview Record, as a value of a key-value pair where GSW_SCRIPT_ID is a key.

When the script is specified in multiple Outbound Objects related to a particular Record (for example, in both Campaign and Calling List), then OCS selects the script DBID in the following order:

1. Campaign Group
2. Campaign
3. Calling List

In this case, when different Scripts are specified in the Campaign and Calling List, the script DBID of the Script that is specified in the Campaign is attached to the call.

Desktop Requests and OCS Responses

The previous sections gave a general overview of the OCS/Desktop Communication protocol. The rest of this chapter describes desktop requests and the corresponding OCS responses in more detail. The topics covered include:

- Campaign status notifications
- Campaign agent assignment
- Starting Preview dialing mode
- Request preview records
- `ReadyTime` request
- Updating call results and custom fields
- Chained records
- Rejecting records
- Canceling records
- Submitting DNC requests
- The differences between canceling records and marking them DoNotCall
- Scheduling and Rescheduling records
- Adding records to the calling list
- Unsolicited notifications
- Agent logout

Finally, it provides a library of error codes and all Genesys events and event type protocols.

The general format for each event section is:

- A diagram (when appropriate) with the event sequence, conditions, and responses.
- A table that features the description, desktop action, mandatory fields, and additional fields for that event.

- Another table that shows the values and descriptions of the additional fields, gives the default values, and describes whether those keys are mandatory or optional.

It is important to note that key-value pairs can be sent in any order. That is, they may be sent in an order other than that specified in the tables in this document. Therefore, any program should have the intelligence to understand keys not by sequence, but by key name.

Note: All requests from the desktop receive the response error Invalid Request or Invalid Request Data if the request does not have all mandatory fields specified or if the mandatory fields have the wrong data.

Campaign Status Notification from OCS to Desktop

Agents receive immediate information about the active campaign at login. When an agent logs in, OCS sends notification to the desktop telling the agent if a campaign is running, the name of the campaign, and the campaign mode.

The following are notification messages from OCS to the desktop:

- `CampaignStarted`
- `CampaignLoaded`
- `CampaignUnloaded`
- `CampaignStopped`
- `CampaignModeChanged`
- `CampaignGroupAssigned`

Notification messages are sent to the agent desktop when:

- The status of a campaign changes.
- The agent logs in to a group that has a running or active (loaded) campaign associated with it.
- The agent assignment is changed.

If the primary and backup OCS switch for any reason, a new primary server sends event `CampaignStarted/CampaignLoaded` to every agent in the campaign to let the desktop know that the `GSW_APPLICATION_ID` attribute has changed. For more information, see “Characteristics of Event Structures” on page 16.

Within a given group, and simultaneously, one or all of the following can be occurring:

- One campaign is running in auto dialing mode.
- Several campaigns are running in the `Preview` or `Push Preview` dialing mode
- Several campaigns are loaded (active) within a group.

Therefore, the status of the campaign must be stated for each group, since a campaign may be started and stopped for different groups at different times.

Figure 1 shows the user events `CampaignStarted`, `CampaignStopped`, and `CampaignModeChanged`, which OCS sends to the desktop.

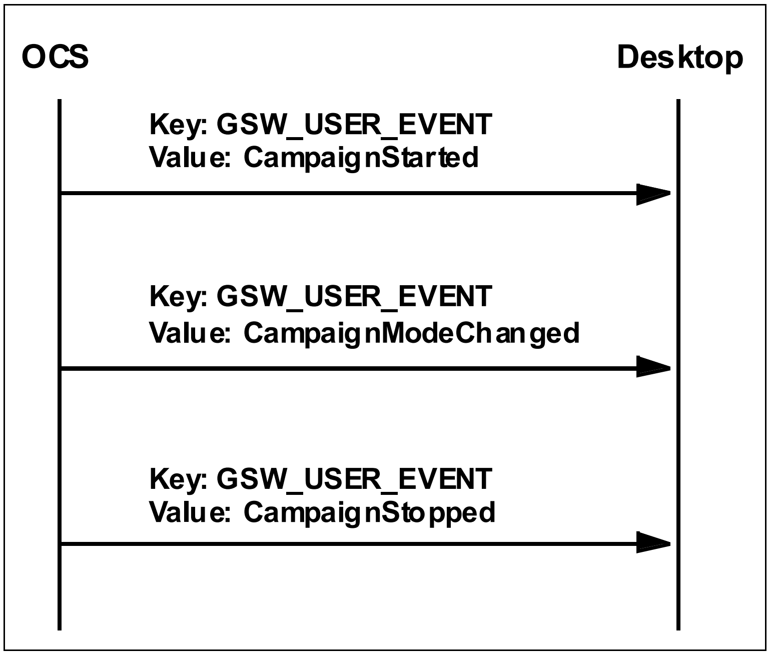

Figure 1: Campaign Status from OCS to the Desktop

CampaignStarted

OCS sends this event to the desktop when a campaign is started. Table 9 contains more information.

Table 9: CampaignStarted

CampaignStarted User Event	
Description	OCS sends this event to all logged in agents when the dialing for a campaign begins, or, as a response to an agent login when a campaign is started.
Recommended Desktop Action	The desktop should store the campaign name and OCS application ID from the attached data of this user event. The desktop can choose to display the campaign information from the attached data.

Table 10 lists the attached data for the `CampaignStarted` event.

Table 10: CampaignStarted Attached Data

Data Key	Type	Key Required	Value	Description
GSW_USER_EVENT	String	Yes	CampaignStarted	Hard coded request name
GSW_APPLICATION_ID	Int	Yes	<OCS application DBID>	DBID for OCS from Configuration DB
GSW_CAMPAIGN_ DESCRIPTION	String	Yes	<description> or an empty string	Description of campaign group.
GSW_CAMPAIGN_MODE	String	Yes	Power GVP, Predictive, Predictive with Seizing, Preview, Progressive, Progressive with Seizing, and Push Preview.	Mode in which campaign started.
GSW_CAMPAIGN_NAME	String	Yes	<Campaign name>	
GSW_CAMPAIGN _GROUP_NAME	String	Yes	<Campaign Group name>	

CampaignLoaded

OCS sends this event to the desktop when a campaign is loaded. Table 11 contains more information.

Table 11: CampaignLoaded

CampaignLoaded User Event	
Description	OCS sends this event to all logged in agents when a campaign is loaded, or, as a response to an agent login when a campaign is started.
Recommended Desktop Action	The desktop should store the campaign name and OCS application ID from the attached data of this user event. The desktop can choose to display the campaign information from the attached data.

Table 12 lists the attached data for the `CampaignLoaded` event.

Table 12: CampaignLoaded Attached Data

Data Key	Type	Key Required	Value	Description
GSW_USER_EVENT	String	Yes	`CampaignLoaded`	Hard coded request name
GSW_APPLICATION_ID	Int	Yes	<OCS application DBID>	DBID for OCS from Configuration DB
GSW_CAMPAIGN_ DESCRIPTION	String	Yes	<description> or an empty string.	Description of campaign group.
GSW_CAMPAIGN_NAME	String	Yes	<Campaign name>	
GSW_CAMPAIGN _GROUP_NAME	String	Yes	<Campaign Group name>	

CampaignUnloaded

OCS sends this event to the desktop when a campaign is unloaded. Table 13 contains more information.

Table 13: CampaignUnloaded

CampaignUnloaded User Event	
Description	OCS sends this event to all logged in agents when a campaign is unloaded.
Recommended Desktop Action	The desktop should stop sending requests to the campaign.

Table 14 lists the attached data for the `CampaignUnloaded` event.

Table 14: CampaignUnloaded Attached Data

Data Key	Type	Key Required	Value	Description
GSW_USER_EVENT	String	Yes	`CampaignUnloaded`	Hard coded event name
GSW_APPLICATION_ID	Int	Yes	<Unique ID of OCS>	
GSW_CAMPAIGN_NAME	String	Yes	<Campaign name>	
GSW_CAMPAIGN _DESCRIPTION	String	Yes	<description> or an empty string	Description of campaign group.
GSW_CAMPAIGN _GROUP_NAME	String	Yes	<Campaign Group name>	

CampaignStopped

OCS sends this event to the desktop when a campaign is stopped. Table 15 contains more information.

Table 15: CampaignStopped

CampignStopped User Event	
Description	OCS sends this event to all logged in agents when dialing for a campaign stops.
Recommended Desktop Action	The desktop should stop sending requests to the campaign.

Table 16 lists the attached data for the `CampaignStopped` event.

Table 16: CampaignStopped Attached Data

Data Key	Type	Key Required	Value	Description
GSW_USER_EVENT	String	Yes	`CampaignStopped`	Hard coded event name
GSW_APPLICATION_ID	Int	Yes	<Unique ID of OCS>	

Table 16: CampaignStopped Attached Data (Continued)

Data Key	Type	Key Required	Value	Description
GSW_CAMPAIGN_NAME	String	Yes	<Campaign name>	
GSW_CAMPAIGN _DESCRIPTION	String	Yes	<description> or an empty string	The description of the Campaign Group.
GSW_CAMPAIGN _GROUP_NAME	String	Yes	<Campaign Group name>	

CampaignModeChanged

OCS sends this event to the desktop when a campaign mode has changed. Table 17 contains more information.

Table 17: CampaignModeChanged

CampaignModeChanged User Event	
Description	Description of change sent to all logged-in agents when the dialing mode for a campaign changes from Predictive mode to Progressive mode or vice-versa.
Recommended Desktop Action	The desktop can choose to display the campaign information from the attached data.

Table 18 lists the attached data for the `CampaignModeChanged` event.

Table 18: CampaignModeChanged Attached Data

Data Key	Type	Key Required	Value	Description
GSW_USER_EVENT	String	Yes	`CampaignModeChanged`	Hard coded event name
GSW_APPLICATION_ID	Int	Yes	<Unique ID of OCS >	

Table 18: CampaignModeChanged Attached Data (Continued)

Data Key	Type	Key Required	Value	Description
GSW_CAMPAIGN_MODE	String	Yes	Power GVP, Predictive, Predictive with Seizing, Preview, Progressive, Progressive with Seizing, and Push Preview.	Mode in which campaign is currently running.
GSW_CAMPAIGN_NAME	String	Yes	<Campaign name>	
GSW_CAMPAIGN_ DESCRIPTION	String	Yes	<description> or an empty string	Description of campaign group.
GSW_CAMPAIGN _GROUP_NAME	String	Yes	<Campaign group name>	

Campaign Group Agent Assignment

OCS sends this event to the desktop when the agent has been assigned to a campaign. Table 19 contains more information.

Note: This notification is a new part of Outbound Contact functionality: agent assignment in multiple campaigns. Refer to the *Outbound Contact 7.6 Deployment Guide* for more information about this functionality.

Table 19: CampaignGroupAssigned

CampaignGroupAssigned User Event	
Description	Sent by OCS when the agent assignment has changed
Recommended Desktop Action	Process the changed campaign assignment.

Table 20 lists the attached data for the `CampaignGroupChanged` event.

Table 20: CampaignGroupAssigned Attached Data

Data Key	Type	Key Required	Value	Description
GSW_USER_EVENT	String	Yes	`CampaignGroupAssigned`	Hard coded event name
GSW_APPLICATION_ID	Int	Yes	<Unique ID of OCS>	
GSW_CAMPAIGN_NAME	String	Yes	<Campaign name>	
GSW_CAMPAIGN_ DESCRIPTION	String	Yes	<description> or an empty string	Description of campaign group.
GSW_CAMPAIGN _GROUP_NAME	String	Yes	<Campaign Group name>	

CampaignStatusRequest

OCS responds to `CampaignStatusRequest` with the same message that is delivered to the agent's desktop upon the agent's login, in the case where the agent is identified as a participant in the active/running campaign group. Possible status notification messages in a response to this request include:

- `CampaignLoaded`
- `CampaignStarted`
- `CampaignGroupAssigned`

Table 21 contains more information.

Table 21: CampaignStatusRequest

CampaignStatus Request User Event	
Description	This request queries information on campaign group(s) statuses from OCS at any arbitrary time when the agent desktop needs to synchronize with OCS on current outbound activities for the agent.
Recommended Desktop Action	Synchronize with OCS on all campaigns in which the agent participates.

Table 22 lists the attached data for the `CampaignStatusRequest` event.

Table 22: CampaignStatusRequest Attached Data

Data Key	Type	Key Required	Value	Description
GSW_AGENT_REQ_TYPE	String	Yes	`CampaignStatusRequest`	Hard coded request name
GSW_APPLICATION_ID[a]	Int	No	OCS application DBID	Target OCS application DBID

[a] GSW_APPLICATION_ID is not a mandatory attribute in the message. If it is present, it narrows the request for campaign group(s) statuses, and only the OCS application with the provided application DBID will process it. If this attribute is absent from the message, then all of the OCS applications that receive this request will process it.

Note: OCS will never reply to `CampaignStatusRequest` with an error message. It will either reply with status notification message(s) or not reply at all (for example, in the case where there are no active or running campaign groups within OCS, or the agent is unknown to OCS).

PreviewDialingModeStart Request

The `PreviewDialingModeStart` request applies to both `Preview` and `Predictive` dialing modes. It is used for receiving scheduled calls or `Preview` mode records. The `PreviewDialingModeStart` request can be activated by setting the `agent_preview_mode_start` option in the `Campaign Group` object or the OCS `Application` object in Configuration Manager. If the option is set to `true`, the desktop must send this request after an agent logs in to receive scheduled call records from OCS. If the agent wants to participate in a preview campaign, the desktop is required to send this request before sending any preview record request. Without the Preview Dialing Mode Start request, OCS ignores all preview record requests sent from the desktop. This setting and request are most often used to ensure that no rescheduled call records are sent to the desktop directly after the agent logs in.

When the option `agent_preview_mode_start` is set to `false`, OCS assumes that the agent is ready to receive any rescheduled call records. If a preview campaign is running when the agent logs in, a Preview Record Request can be sent anytime without sending a Preview Dialing Mode Start request.

PreviewDialingModeStart

The desktop sends this request to OCS when the Preview dialing mode starts. Table 23 contains more information.

Note: The `PreviewDialingModeStart` request is not required from the Agent Desktop in `Push Preview` and `Power GVP` modes, regardless of the setting for the `agent_preview_mode start` option. For information on this option, see the *Outbound Contact 7.6 Deployment Guide.*

Table 23: PreviewDialingModeStart

PreviewDIalingModeStart Request	
Description	Request to activate preview session for the agent. Needed if the `agent_preview_mode_start` option is set to `true`.
OCS Action	Link agent DN and campaign ID.

Table 24 lists the attached data for the `PreviewDialingModeStart` request.

Table 24: PreviewDialingModeStart Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	`PreviewDialingModeStart`
GSW_APPLICATION_ID	Int	Yes	Unique ID of OCS
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.

PreviewDialingModeStartAcknowledge

OCS sends this event to the desktop to acknowledge the start of Preview dialing mode. Table 25 contains more information.

Table 25: PreviewDialingModeStartAcknowledge

PreviewDialingModeStartAcknowledge	
Description	OCS accepts a desktop request to initiate preview session.
Recommended Desktop Action	The desktop can send requests to OCS and receive callbacks.

Table 26 lists the attached data for the `PreviewDialingModeStartAcknowledge` event.

Table 26: Preview Dialing Mode Start Acknowledge Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	PreviewDialingModeStartAcknowledge
GSW_APPLICATION_ID	Int	Yes	Unique ID of OCS
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign

PreviewDialingModeOver

The desktop sends this request to OCS when the Preview dialing mode is over. Table 27 contains more information.

Note: The PreviewDialingModeOver request is not required from the Agent Desktop in Push Preview or Power GVP modes, regardless of the setting for the agent_preview_mode start option. For information on this option, see the *Outbound Contact 7.6 Deployment Guide.*

Table 27: PreviewDialingModeOver

PreviewDialingModeOver User Event	
Description	Request to terminate preview session for the agent.
OCS Action	Remove the link between agent DN and campaign ID.

Table 28 lists the attached data for the PreviewDialingModeOver request.

Table 28: PreviewDialingModeOver Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	PreviewDialingModeOver
GSW_APPLICATION_ID	Int	Yes	Unique ID of OCS
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.

PreviewDialingModeOverAcknowledge

OCS sends this event to the desktop to acknowledge the end of Preview dialing mode. Table 29 contains more information.

Table 29: PreviewDialingModeOverAcknowledge

PreviewDialingModeOverAcknowledge User Event	
Description	OCS accepts a desktop request to close preview session.
Recommended Desktop Action	Desktop should disable the function for sending further requests to OCS and for receiving callbacks.

Table 30 lists the attached data for the `PreviewDialingModeOverAcknowledge` event.

Table 30: PreviewDialingModeOverAcknowledge Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	`PreviewDialingModeOverAcknowledge`
GSW_APPLICATION_ID	Int	Yes	Unique ID of OCS
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.

ReadyTime Request

This request is used to increase campaign performance in the Predictive or Predictive with seizing dialing modes for small groups. This request only applies when the `predictive_algorithm` option is set to `small_group` or `advanced_small_group` and `Overdial Rate` is used as the optimization parameter.

Through this event, the agent's desktop provides OCS with an estimate of the time (in seconds).that the agent will need to finish processing of the current outbound call and before he or she will go to the `Ready` state.

Note:

- This request was added in the 7.6.1 release.
- For more information about the `predictive_algorithm` option, see the *Outbound Contact 7.6 Deployment Guide.*

ReadyTime

The desktop sends this `ReadyTime` request to OCS, providing the estimated time (in seconds) remaining until the agent will become Ready. Table 31 contains more information.

Table 31: ReadyTime Request

ReadyTime Agent Request	
Description	Provides the time in which the agent will become Ready.
OCS Action	Use this information in the predictive algorithm when calculating the number of outbound calls to be placed in the next seconds

Table 32 lists the attached data for the ReadyTime request.

Table 32: ReadyTime Attached Data[a]

Data Key[b]	Type	Key Required	Value	Description
GSW_AGENT_REQ_TYPE	String	Yes	ReadyTime	Hard coded request name
GSW_READY_TIME	Int	Yes	<Expected time in seconds>	Expected time to go ready in seconds (N), where N >0
GSW_APPLICATION_ID	Int	Yes	<Unique ID of OCS>	Target OCS application DBID
GSW_RECORD_HANDLE	Int	Yes	<Unique record handle>	Record handle for the record currently on the agent’s desktop[c]

a. ReadyTime can only be applied to a record currently being processed by the agent.

b. ReadyTime supports only those key-value pairs listed in this table. Any other pairs will be ignored by OCS.

c. The record handle is a mandatory attribute because it identifies the record currently being process by the agent for OCS.

ReadyTimeAcknowledge

OCS sends this event to the desktop to acknowledge the ReadyTime request.or sends an error (see “ReadyTime Error”). Table 33 contains more information.

Table 33: ReadyTimeAcknowledge

ReadyTimeAcknowledge User Event	
Description	OCS acknowledges receiving the event to the desktop.
Desktop Action	Ensure the record gets finalized and the agent goes Ready after the communicated period of time elapses.

Table 34 lists the attributes for the `ReadyTimeAcknowledge` event.

Table 34: ReadyTimeAcknowledge Attributes

Data Key	Type	Key Required	Value	Description
GSW_USER_EVENT	String	Yes	`ReadyTimeAcknowledge`	Hard coded event
GSW_APPLICATION_ID	Int	Yes	<Unique ID of OCS>	Originator OCS application DBID
GSW_RECORD_HANDLE	Int	Yes	<Unique record handle>	Record handle value, as passed in the `ReadyTime` request

ReadyTime Error

If OCS is not able to properly process the `ReadyTime,` one of the errors in Table 35 is returned.

Table 35: ReadyTime Error Codes

Error Code	Error Description	Returned When:
101	Invalid request	Campaign Group dialing mode is not Predictive (Predictive ASM)
102	Invalid request; attribute is not found	A mandatory attribute is missing from the request
103	Invalid request data; bad attribute value	Time to go ready is less than or equal to 0
104	Agent not found	This agent is unknown to OCS

Table 35: ReadyTime Error Codes (Continued)

Error Code	Error Description	Returned When:
112	No call found for the record handle	The record handle is invalid
120	Duplicate request is not allowed	The `ReadyTime` request is submitted more than once for the same record handle

Preview Record Request and Acknowledgment

The desktop can send a preview record request after receiving the event CampaignStarted, with the additional key-value `GSW_CAMPAIGN_MODE` set to `Preview`. The desktop can then begin working in `Preview` dialing mode. OCS has the option of setting `PreviewDialingModeStart` as either `true` or `false`. When set to `true`, OCS waits for the `PreviewDialingModeStart` request from the desktop before allowing the agent to issue a `PreviewRecord` request. When set to `false`, the desktop can send a `PreviewRecord` request without sending the `PreviewDialingModeStart` request for receiving scheduled calls or preview records.

PreviewRecordRequest

The desktop sends this request to OCS to request preview records.Table 36 contains more information.

Table 36: PreviewRecordRequest

PreviewRecordRequest	
Description	Request to send preview record.
OCS Action	Conditionally sends acknowledgment, depending on setting of the `PreviewDialingModeStart` event. See “PreviewDialingModeStart Request” on page 39.

Table 37 lists the attached data for `PreviewRecordRequest`.

Table 37: PreviewRecordRequest Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	`PreviewRecordRequest`
GSW_APPLICATION_ID	Int	Yes	Unique ID of OCS
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.

PreviewRecord

The following event is sent by OCS to the desktop in response to a request for records in the `Preview` dialing mode. Table 38 contains more information. See “Reserved Keys” on Page 22 and “Genesys Enumeration Table” on page 28 for predefined attribute values.

Table 38: PreviewRecord

PreviewRecord User Event	
Description	Preview record to dial.
Recommended Desktop Action	Perform “Call Work” (the agent performs work associated with the call, such as dialing or updating a record).

Table 39 lists the attached data for the `PreviewRecord` event.

Table 39: PreviewRecord Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	`PreviewRecord`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_ATTEMPTS	Int	No	Number of attempts for the record. This key is used when a new record is added.
GSW_CALL_RESULT	Int	Yes	Call Result set by dialer or saved from previous call. (See the “Genesys Enumeration Table” on page 28.)
GSW_CALL_TIME	String	No	System time when record was called, in seconds from 1/1/70 (GMT).

Table 39: PreviewRecord Attached Data (Continued)

Data Key	Type	Key Required	Description
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_CHAIN_ID	Int	Yes	Unique chain ID
GSW_FROM	Int	Yes	GSW_FROM - GSW_UNTIL: Time frame when a record can be called, seconds from midnight.
GSW_PHONE	String	Yes	Phone number to dial.
GSW_PHONE_TYPE	Int	No	Customer phone type (See the "Genesys Enumeration Table" on page 28).
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.
GSW_TZ_OFFSET	Int	Yes	Offset (time difference) in seconds between UTC and a particular time zone. It may contain different values throughout the year if Daylight Savings Time (DST) is used for the specified time zone.
GSW_UNTIL	Int	No	Time until, seconds since midnight.
GSW_CONTACT_MEDIA_TYPE	String	Yes	Describes the media type used for contact.
GSW_CAMPAIGN_GROUP_NAME	String	Yes	The name of the Campaign Group.
GSW_CAMPAIGN_GROUP__DESCRIPTION	String	Yes	The description of the Campaign Group. Value may be an empty string.
Custom fields	Any	No	Custom fields.

No Records Available

OCS sends this event to the desktop when there are no more Preview records to send or if OCS has not filled the buffer yet. The desktop repeats its `PreviewRecordRequest` (page 45) in a few seconds. Table 40 contains more information.

Table 40: No Records Available

No Records Available User Event	
Description	No more records in the OCS internal buffer.
Recommended Desktop Action	Try to send a request later.

Table 41 lists the attached data for the No Records Available event.

Table 41: No RecordsAvailable Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	PreviewRecordRequest
GSW_ERROR_NUMBER	Int	Yes	Error code. See "Error Names and Codes" on page 89.
GSW_ERROR	String	Yes	No Records Available.
GSW_APPLICATION_ID	Int	Yes	OCS Application DBID
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.

NoRunningPreviewCampaigns

OCS issues this message if the agent requests a record for a preview campaign that is not currently running. Table 42 contains more information.

Note: In release 7.2, this scenario resulted in a NoActivePreviewCampaign message.

Table 42: NoRunningPreviewCampaigns

NoRunningPreviewCampaigns User Event	
Description	No campaigns are running in Preview mode
Recommended Desktop Action	Try to send a request later.

Table 43 lists the attached data for the NoRunningPreviewCampaign error.

Table 43: No Running Preview Campaign Error

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	PreviewRecordRequest
GSW_ERROR_NUMBER	Int	Yes	Error code. See "Error Names and Codes" on page 89.
GSW_ERROR	String	Yes	No Running Preview Campaign
GSW_APPLICATION_ID	Int	Yes	OCS Application DBID
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.

Updating Call Results and Custom Fields

The UpdateCallCompletionStats request updates Genesys modifiable mandatory fields and custom fields in a record to OCS.

For example, in Predictive dialing mode, this request can be used to overwrite the call result detected by call progress detection when needed. Or the desktop can overwrite a call result answer with the call result wrong party. (See the list of predefined call results on Table 112 on page 115.)

This request can be sent multiple times before the RecordProcessed request is sent. Also, the record can still be canceled or rejected (using RecordCancel or RecordReject) before the final RecordProcessed request is sent. Use the UpdateCallCompletionStats when the record is still active on the agent desktop.

The RecordProcessed request signals the final transaction for the record. The request updates all fields (including call completion statistics and custom fields) in OCS and returns the record to the database.

After the request is sent, the record cannot be canceled or rejected. Use the RecordProcessed request when the agent finishes with a record and returns it to the database. Changes made to the database after the RecordProcessed request is used are final.

If you want OCS to apply a treatment to the call result entered in UpdateCallCompletionStats request, then the final RecordProcessed request for this record should contain an optional GSW_TREATMENT attribute, which has a possible value of RecordTreatPersonal or RecordTreatCampaign.

If the value of this attribute is RecordTreatCampaign, OCS will change the status of this record to Campaign Rescheduled and treat it as regular record rescheduled by treatment.

If the value of this attribute is RecordTreatPersonal, OCS will change the status of this record to Personal Rescheduled and treat it similarly to Personal CallBack.

If GSW_TREATMENT attribute is omitted in the RecordProcessed request, no treatment will be applied to the record.

Figure 2 illustrates a typical data flow when updating Call Results and Custom Fields.

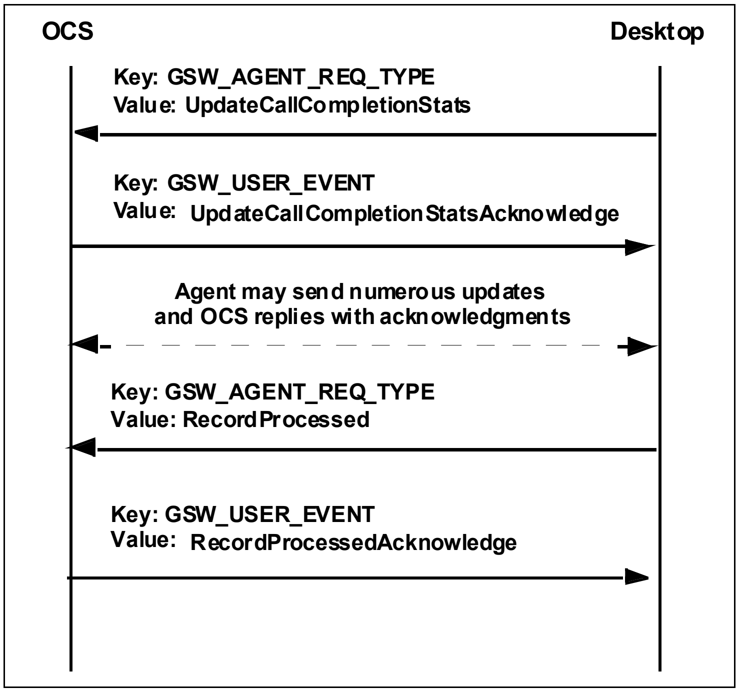
OCS
Desktop
Key: GSW_AGENT_REQ_TYPE
Value: UpdateCallCompletionStats
Key: GSW_USER_EVENT
Value: UpdateCallCompletionStatsAcknowledge
Agent may send numerous updates
and OCS replies with acknowledgments
Key: GSW_AGENT_REQ_TYPE
Value: RecordProcessed
Key: GSW_USER_EVENT
Value: RecordProcessedAcknowledge

Figure 2: Updating Call Results and Custom Fields Data Flow

UpdateCallCompletionStats

The desktop sends this request to OCS to update a record on completion of a call.Table 44 contains more information.

Table 44: UpdateCallCompletionStats

UpdateCallCompletionStats Request	
Description	Desktop sends to update record details. Intermediate update.
OCS Action	Update record fields internally; wait for next requests.

Table 45 lists the attached data for the UpdateCallCompletionStats request.

Table 45: UpdateCallCompletionStats Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	`UpdateCallCompletionStats`
GSW_CALL_RESULT	Int	No	Call Result sent to change automatically detected call result. See Table 8 on page 28 and Table 112 on page 115.
GSW_FROM	Int	No	`GSW_FROM - GSW_UNTIL`: Time frame when a record can be called, seconds from midnight.
GSW_UNTIL	Int	No	Time until, seconds since midnight.
GSW_PHONE	String	No	Customer's phone number.
GSW_PHONE_TYPE	Int	No	Customer phone type. See Table 8 on page 28.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier. Do not change this value.
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
Custom Fields	Any	No	Custom Fields.

UpdateCallCompletionStatsAcknowledge

OCS sends this event to the desktop to acknowledge a call completion notification. Table 46 contains more information.

Table 46: UpdateCallCompletionStatsAcknowledge

UpdateCallCompletionStatsAcknowledge User Event	
Description	OCS accepts a desktop request to update a record's fields.
Recommended Desktop Action	Continue "Call Work" (the agent performs work associated with the call, such as dialing or updating a record).

Table 47 lists the attached data for the `UpdateCallCompletionStatsAcknowledge` event.

Table 47: UpdateCallCompletionAcknowledge Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	`UpdateCallCompletionStatsAcknowledge` or error.
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier. Do not change this value.

RecordProcessed

The desktop sends this request to OCS to indicate that the agent has finished with a record and that it should be processed and sent to the database.

The `RecordProcessed` request is mandatory in `Preview` dialing mode and optional in the other dialing modes. When the `record_processed` option is set to `true`, it must be sent in all cases. Table 48 contains more information.

Table 48: RecordProcessed

RecordProcessed Request	
Description	Desktop sends event to indicate that record is processed. OCS should update record if it is provided.
OCS Action	Update a record and its chain in DB; use all changes made by previous requests regarding the records in the chain. If a RecordProcessed event has the GSW_TREATMENT field correctly specified, OCS applies a treatment to the record.

Table 49 lists the attached data for the `RecordProcessed` request.

Table 49: RecordProcessed Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	RecordProcessed
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CALL_RESULT	Int	No	Call Result sent to change automatically detected call result. See Table 8 on page 28 and Table 112 on page 115.
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_FROM	Int	No	GSW_FROM - GSW_UNTIL: Time frame when a record can be called, seconds from midnight.
GSW_UNTIL	Int	No	Time until, seconds since midnight.
GSW_PHONE	String	No	Customer’s phone number.
GSW_PHONE_TYPE	Int	No	Customer phone type. See Table 8 on page 28.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.
GSW_TREATMENT	String	No	Specifies the treatment type that should be applied to a record chain when RecordProcessed event is processing. Possible values are RecordTreatPersonal or RecordTreatCampaign.
Custom Fields	Any	No	Custom Fields.

RecordProcessedAcknowledge

OCS sends this event to the desktop to acknowledge a `RecordProcessed` notification. Table 50 contains more information.

Table 50: RecordProcessedAcknowledge

RecordProcessAcknowledge User Event	
Description	OCS confirms that the record has been executed.
Recommended Desktop Action	Remove the record and the chain if requested.

Table 51 lists the attached data for `RecordProcessedAcknowledge` event.

Table 51: RecordProcessedAcknowledge Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	`RecordProcessedAcknowledge`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.

Chained Records

If a customer cannot be reached at the primary contact number (for example, Home Phone), the agent may try a second, or subsequent, record in a chain of contact numbers (for example, Business Phone). For the primary contact number, the chain_n field is represented by zero or any positive number. When using the `ChainedRecordRequest,` the attached data of the request must include the initial record's `GSW_RECORD_HANDLE.`

Note: All repeated requests having the same mandatory field values (such as `GSW_RECORD_HANDLE`) receive the response error Record Not Found if the record is already processed. A desktop can send the request only once for a chain; subsequent requests are ignored to avoid multiple delivery of the same records.

The ChainedRecordRequest can be used in the Preview or Predictive dialing mode. However, In Predictive dialing mode the user should use one of the following:

- The Next In Chain treatments, to let the dialer handle the chain record automatically.
- A ChainedRecordRequest to handle chain records manually. Never use both in the same campaign. Figure 3 is an example of a typical chained record data flow.

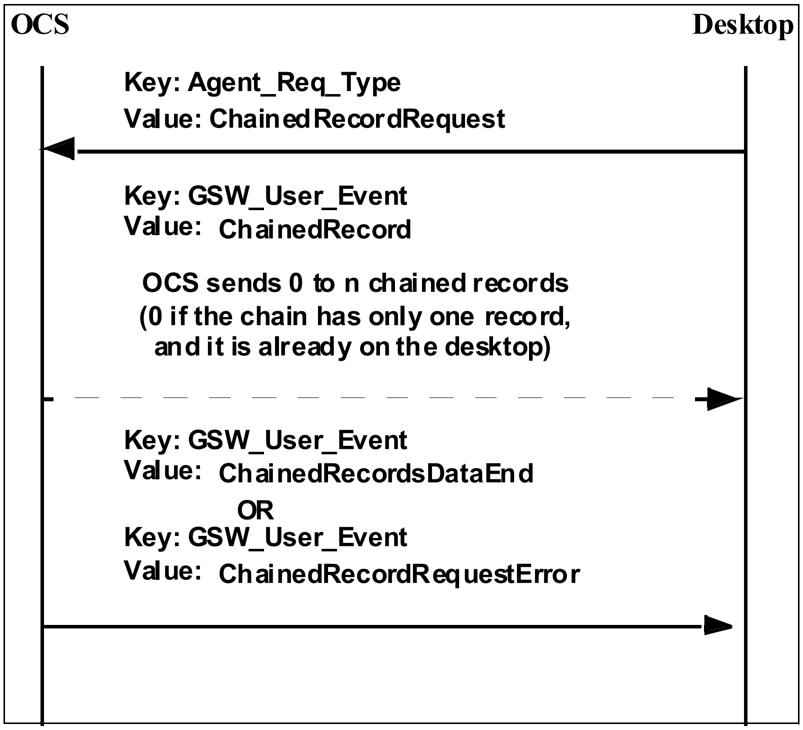
OCS
Desktop
Key: Agent_Req_Type
Value: ChainedRecordRequest
Key: GSW_User_Event
Value: ChainedRecord
OCS sends 0 to n chained records
(0 if the chain has only one record,
and it is already on the desktop)
Key: GSW_User_Event
Value: ChainedRecordsDataEnd
OR
Key: GSW_User_Event
Value: ChainedRecordRequestError

Figure 3: Chained Record Data Flow

ChainedRecordRequest

The desktop sends this request to OCS to request a record chain. Table 52 contains more information.

Table 52: ChainedRecordRequest

ChainedRecordRequest	
Description	Request to send all records from the chain defined by RecordHandle.
OCS Action	Send rest of a chain to the desktop.

Table 53 lists the attached data for ChainedRecordRequest.

Table 53: ChainedRecordRequest Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	ChainedRecordRequest
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.

ChainedRecord

The following event is sent by OCS to the desktop in response to a ChainedRecordRequest event. Non-mandatory fields should be sent only if the send_attribute option is defined. Table 54 contains more information.

Table 54: ChainedRecord

ChainedRecord User Event	
Description	Chain record delivered.
Recommended Desktop Action	Continue Call Work (the agent performs work associated with the call, such as dialing or updating a record).

Table 55 lists the attached data for the ChainedRecord event.

Table 55: ChainedRecord Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	ChainedRecord
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_ATTEMPTS	Int	No	Number of attempts for the record. This key is used when a new record is added.
GSW_CALL_TIME	String	No	System time when record was called, in seconds from 1/1/70 (GMT).

Table 55: ChainedRecord Attached Data (Continued)

Data Key	Type	Key Required	Description
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CALL_RESULT	Int	Yes	Call Result set by dialer or saved from previous call. (See the "Genesys Enumeration Table" on page 28.)
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_CHAIN_ID	Int	Yes	Unique chain ID.
GSW_FROM	Int	No	GSW_FROM - GSW_UNTIL: Time frame when a record can be called, seconds from midnight.
GSW_PHONE	String	Yes	Customer's phone number.
GSW_PHONE_TYPE	Int	No	Customer phone type (See the "Genesys Enumeration Table" on page 28).
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.
GSW_TZ_OFFSET	Int	No	Offset (time difference) in seconds between UTC and a particular time zone. It may contain different values throughout the year if Daylight Savings Time (DST) is used for the specified time zone.
GSW_UNTIL	Int	No	Time until, seconds since midnight.
GSW_CONTACT_MEDIA_TYPE	String	Yes	Describes the method of contact.
GSW_CAMPAIGN _GROUP_NAME	String	Yes	The name of the Campaign Group.
GSW_CAMPAIGN_GROUP__ DESCRIPTION	String	Yes	The description of the Campaign Group. Value may be an empty string.
Custom fields	Any	No	Custom Fields.

ChainedRecordsDataEnd

The following event is sent by OCS to the desktop when all records in a chain have been sent. Table 56 contains more information.

Table 56: ChainedRecordsDataEnd

ChainedRecordsDataEnd User Event	
Description	All chain has been delivered.
Recommended Desktop Action	Continue "Call Work" (the agent performs work associated with the call, such as dialing or updating a record).

Table 57 lists the attached data for the ChainedRecordsDataEnd event.

Table 57: ChainedRecordsDataEnd Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	ChainedRecordsDataEnd
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_CHAIN_ID	Int	Yes	Unique chain ID.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.

Rejecting Records

The term *reject* means that the agent does not want to call the record at this time.

For example, an agent might reject a record already delivered to the desktop before going on break or when leaving and logging out for the day. This is a good practice because it prevents OCS from updating these records as Stale when the stale_clean_timeout option has expired.

The rejected record is returned to the database with the following fields modified: record_type is reset to General, record_status is reset to Ready, agent_id is reset to the ID of the agent that rejected the record. It will be retrieved again with the next set of records from the database, for distribution by OCS. The agent ID will be overwritten again when the next agent receives the record.

Figure 4 shows a typical RejectRecord data flow.

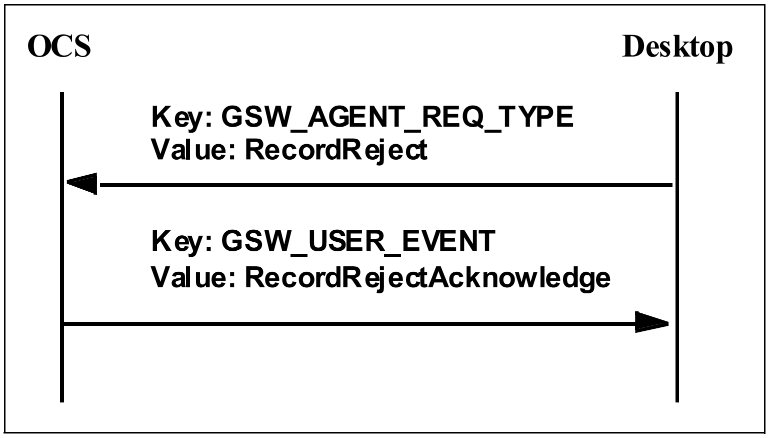

Figure 4: RejectRecord Data Flow

RecordReject

The desktop sends this request to OCS to reject a record. When a record is rejected by an agent, the `Agent ID` field of the call record is updated to that agent's ID. Table 58 contains more information.

Table 58: RecordReject

RecordReject Request	
Description	Desktop sends a request to indicate that preview record or scheduled call will not be dialed by this agent. Record should be re-sent to another agent. This is the final event for the record, which means the desktop does not need to send RecordProcessed after this request.
OCS Action	OCS marks this record, and the rest of the chain, as `general` and `ready`.

Table 59 lists the attached data for the `RejectRecord` event.

Table 59: RecordReject Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	`RecordReject`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CALLING_LIST	String	Yes	Name of the calling list.

Table 59: RecordReject Attached Data (Continued)

Data Key	Type	Key Required	Description
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.

RecordRejectAcknowledge

The following event is sent to the desktop by OCS to acknowledge a rejected record. Table 60 contains more information.

Table 60: RecordRejectAcknowledge

RecordRejectAcknowledge User Event	
Description	OCS accepts RejectRecord request.
Recommended Desktop Action	Kill the record and the chain.

Table 61 lists the attached data for the `RecordRejectAcknowledge` event.

Table 61: RecordRejectAcknowledge Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	`RecordRejectAcknowledge`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.

Canceling Records

The desktop can send a RequestRecordCancel event to notify OCS to cancel a record to be dialed by a campaign. Agents able to send this type of request include:

- Outbound agents: Those who work only in outbound campaigns. See "Example 1".
- Blended agents: Those who work simultaneously in outbound and inbound campaigns. See "Example 2".
- Inbound agents: Those who work on inbound calls. See "Example 3".

The following are three examples of record cancellations.

Example 1

1. An agent working on a campaign has a record on the desktop.
2. After reviewing the contact history of the call record, the agent decides no outbound call is required.
3. The desktop then sends a RequestRecordCancel (with GSW_RECORD_HANDLE) to OCS.
4. OCS updates the record status to cancelled. This record will no longer be handled by the campaign.

Example 2

1. An agent is working in a *blended* environment (inbound and outbound) and has outbound agent desktop at his disposal.
2. The agent accepts an inbound call and sees that there is no longer a need for an outbound call to the customer.
3. This agent then sends RequestRecordCancel (with the phone number GSW_PHONE, but without GSW_RECORD_HANDLE) to OCS by means of Outbound-Desktop protocol.
4. OCS attempts to find a record that has the matching phone number in OCS memory and in calling lists assigned to loaded campaigns.
5. If a match is found, OCS updates the record as cancelled in OCS memory (if applicable) and/or in calling lists.
6. If a match is found on other desktops within the campaign groups, OCS sends a RecordCancel notification to the desktop(s) where that record is located.
7. The agent deletes the record from the desktop application memory to ensure the record will not be dialed.

Note: A blended agent who submits a RequestRecordCancel from an agent desktop must be a member of an Agent Group or Place Group assigned to the campaign.

Example 3

1. An agent working on inbound calls only receives information that there is no need for an outbound call to a particular customer.
2. The agent sends a CM_ReqCancelRecord (with the phone number GSW_PHONE, but without GSW_RECORD_HANDLE) to OCS by means of Communication DN API. (See "Record Cancellation from a Third-Party Application" on page 107.)
3. OCS attempts to find a record that has the matching phone number in OCS memory and in calling lists assigned to loaded campaigns.
4. If a match is found, OCS updates the record as cancelled in OCS memory (if applicable) and/or in calling lists.
5. If a match is found on other desktops within the campaign groups, OCS sends a RecordCancel notification to the desktop(s) where that record is located.
6. The agent deletes the record from the desktop application memory to ensure the record will not be dialed.

Note: An inbound agent who submits a CM_ReqCancelRecord from a third-party application does not have to be a member of an Agent or Place Group.

The remaining sections pertaining to record cancellation in this chapter are applicable to OCS-Desktop protocol. For information about record cancellation from third-party applications, see Chapter 2 on page 95.

Record Cancel Requests

This section describes record cancel requests and acknowledgments.

RequestRecordCancel

The desktop sends this request to OCS to cancel a record or a chain. Table 62 contains more information.

Table 62: RequestRecordCancel

RequestRecordCancel Request	
Description	Desktop sends a request to OCS to cancel a record or a chain. The preview record or scheduled call should not be dialed. Record should not be re-sent to another agent. It should be marked in the database as `canceled`.
OCS Action	Cancel record.

Table 63 lists the attached data for the `RequestRecordCancel` request.

Table 63: RequestRecordCancel Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	`RequestRecordCancel`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance. Always required.
GSW_CALLING_LIST	String	No	Name of the calling list. Required only if `GSW_RECORD_HANDLE` is specified.
GSW_CAMPAIGN_NAME	String	No	Name of the campaign. Required only if `GSW_RECORD_HANDLE` is specified.
GSW_CHAIN_ATTR	String	No	Flag determining whether to update the record chain or just the single record. Values are `AllChain` (default) or `RecordOnly`.
GSW_PHONE	String	No	Customer’s phone number.
GSW_RECORD_HANDLE	Int	No	Unique Record Identifier.

Mandatory Fields

This statement covers two scenarios and the data key fields required for them. They vary, depending on what is specified in the event.

```
GSW_APPLICATION_ID
if (GSW_RECORD_HANDLE is specified)
    {
    GSW_CAMPAIGN_NAME
    GSW_CALLING_LIST
    }
else
{
GSW_PHONE
}
```

For example, if GSW_RECORD_HANDLE is specified, GSW_CAMPAIGN_NAME and GSW_CALLING_LIST must be specified.

If OCS receives RequestRecordCancel with the required fields for either of these two scenarios, OCS sends RecordCancelAcknowledge to the desktop. If any of the required fields for these scenarios are missing, OCS sends an error message to the desktop.

The field GSW_CHAIN_ATTR directs the update of chained records. If omitted or set with the AllChain value, all chained records are updated as Canceled; if the field has the RecordOnly value, the record with the requested GSW_PHONE is marked as Canceled, but other chained records are Updated.

Under particular conditions, records with the same chain_id are not all cancelled by RequestRecordCancel. For more information, see the “Filters that Break a Chain of Records” section in the *Outbound Contact 7.6 Deployment Guide*.

RecordCancelAcknowledge

OCS sends this event to the desktop to acknowledge a RequestRecordCancel event. Table 64 contains more information.

Table 64: RecordCancelAcknowledge

RecordCancelAcknowledge User Event	
Description	OCS accepts a desktop request to cancel a record.
Recommended Desktop Action	Remove the record and the chain from desktop.

Table 65 lists the attached data for the RecordCancelAcknowledge event.

Table 65: RecordCancelAcknowledge Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	RecordCancelAcknowledge
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance. Always required.

Table 65: RecordCancelAcknowledge Attached Data (Continued)

Data Key	Type	Key Required	Description
GSW_CALLING_LIST	String	No	Name of the calling list. Required only if GSW_RECORD_HANDLE is specified.
GSW_CAMPAIGN_NAME	String	No	Name of the campaign. Required only if GSW_RECORD_HANDLE is specified.
GSW_CHAIN_ATTR	String	No	Flag determining whether to update the record chain or just the single record. Values are AllChain (default) or RecordOnly. See also "Record Cancel for AllChain when Chain is Broken" in the *Outbound Contact 7.6 Deployment Guide*.
GSW_PHONE	String	No	Customer's phone number.
GSW_RECORD_HANDLE	Int	No	Unique Record Identifier.

Note: The mandatory fields for the RequestRecordCancel event depend on the scenario:

Scenario 1: If GSW_RECORD_HANDLE is present, then GSW_CALLING_LIST and GSW_CAMPAIGN NAME become mandatory, in addition to GSW_APPLICATION_ID.

Scenario 2: If GSW_PHONE is present, then only GSW_APPLICATION_ID is mandatory.

RecordCancel Notification

OCS sends this unsolicited notification to the desktop to cancel a record. This occurs, for example, when an inbound agent sends a CM_ReqCancelRecord from a third-party application to OCS, and OCS finds a record with the same phone number (GSW_PHONE) on another desktop. When OCS sends a RecordCancel notification to the desktop, the agent should remove the record from the desktop. Table 66 contains more information.

Table 66: RecordCancel

RecordCancel User Event	
Description	OCS sends this event to the desktop to indicate that this record should not be dialed. Applicable for preview records and scheduled calls.
Recommended Desktop Action	Delete the record if `GSW_CHAIN_ATTR= RecordOnly`. Delete the chain if the `RecordCancel` contains `GSW_CHAIN_ATTR=AllChain`.

Table 67 lists the attached data for the `RecordCancel` event.

Table 67: RecordCancel Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	`RecordCancel`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.
GSW_CHAIN_ATTR	String	No	Flag determining whether to update the record chain or just the single record. Values are AllChain or RecordOnly. (Default = AllChain)
GSW_PHONE	String	Yes	Customer's phone number.

Canceled and DoNotCall Chained Records

The `GSW_CHAIN_ATTR` key applies only to cancelled and `DoNotCall` chained records. The value of the chained record attribute determines the next action when a record that is part of a chain is marked as `Cancel` or `DoNotCall`:

- When the value is set to `RecordOnly`, only that particular record in the chain is marked with `Cancel` or `DoNotCall`.
- When the value is set to `AllChain` or is not specified, the entire chain is marked with the same `Cancel` or `DoNotCall` status as the first record.

OCS ignores the status of the GSW_CHAIN_ATTR key when processing UpdateCallCompletionStats, RescheduleRecord, and RecordProcessed requests.

If a chain of records is on an agent's desktop and a Cancel or DoNotCall by phone number or customer ID (AllChain) request is sent to OCS, OCS distributes the RecordCancel message to the desktop application.

Submitting DoNotCall Requests

The desktop can send a DoNotCall (DNC) request to OCS to prevent a record from being dialed by any campaign. Agents able to send this type of request include:

- Outbound agents: Those who work only in outbound campaigns. See "Example 1".
- Blended agents: Those who work simultaneously in outbound campaigns and on inbound calls. See "Example 2".
- Inbound agents: Those who work on inbound calls. See "Example 3".

The following are three examples of DoNotCall request handling.

Example 1

1. While an agent is working on an outbound campaign, a called party asks the agent not to call him (or her) again and wants his (or her) name or phone number removed from the contact list.
2. To accomplish this, the Agent sends a DoNotCall request (with GSW_RECORD_HANDLE) to OCS.
3. Using the GSW_RECORD_HANDLE provided, OCS identifies the record and updates the record type as NoCall.
4. OCS enters the phone number or the customer ID of this record in the gsw_donotcall_list (table).

Example 2

1. An agent is working in a *blended* environment (inbound and outbound) and has an agent desktop at his disposal. The agent accepts an inbound call from a customer who requests no contact with him (or her) in the future.
2. The desktop sends a DoNotCall request with the phone number (GSW_PHONE) or the customer ID (GSW_CUSTOMER_ID) but without GSW_RECORD_HANDLE to OCS. OCS saves the phone number in the gsw_donotcall_list (table).
3. OCS attempts to find a record that has the matching phone number in OCS memory and in calling lists assigned to loaded campaigns.
4. If a match is found, OCS updates the record as NoCall in OCS memory (if applicable) and/or in calling lists.

5. If a match is found on other desktops within the campaign groups, OCS sends a RecordCancel notification to the desktop(s) where that record is located.
6. The agent deletes the record from the desktop application memory to ensure the record will not be dialed.

Note: A blended agent who submits a DoNotCall request from an agent desktop must be a member of an Agent Group or Place Group assigned to the campaign.

Example 3

1. An agent working only on inbound calls receives a call from a customer who does not want to be contacted again.
2. The agent sends a CM_ReqDoNotCall request (with GSW_PHONE or GSW_CUSTOMER_ID) to OCS by means of Communication DN API. (See Chapter 2 on page 95.)
3. OCS saves the phone number in the gsw_donotcall_list (table).
4. OCS attempts to find a record that has the matching phone number in OCS memory and in calling lists assigned to loaded campaigns.
5. If a match is found, OCS updates the record as NoCall in OCS memory (if applicable) and/or in calling lists.
6. If a match is found on other desktops within the campaign groups, OCS sends a RecordCancel notification to the desktop(s) where that record is located.
7. The agent deletes the record from the desktop application memory to ensure the record will not be dialed.

Note: An inbound agent who submits a CM_ReqDoNotCall request from a third-party application does not have to be a member of an Agent or Place Group.

The remaining sections pertaining to DoNotCall requests in this chapter are applicable to OCS-Desktop protocol. For information about DoNotCall requests from third-party applications, see Chapter 2 on page 95.

OCS stores records marked as NoCall in the gsw_donotcall_list (one per tenant) and monitors them in the following way: When a tenant starts a campaign, OCS retrieves all records that are ready to be dialed from a calling list and checks them against the gsw_donotcall_list. If a record retrieved from a calling list matches a record marked NoCall in the gsw_donotcall_list, OCS does not dial this record, but instead returns it to the calling list and changes its record_type to NoCall.

Note: If a manual update to this gsw_donotcall_list is required, OCS must be restarted to acknowledge the changes. Most administrators choose to update the DoNotCall table (gsw_donotcall_list) during off-hour periods, so that restarting the server does not disrupt calling activities.

DoNotCall (Request)

The desktop sends this request for OCS to mark a record DoNotCall. OCS maintains the DoNotCall table (gsw_donotcall_list), which agents can update during a campaign by using this protocol. Table 68 contains more information.

Table 68: DoNotCall

DoNotCall Request	
Description	Agent requests the number or customer ID in a record not to be called again.
OCS Action	Update gsw_donotcall_list. Mark record NoCall.

Table 69 lists the attached data for the DoNotCall request.

Table 69: DoNotCall Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	DoNotCall
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CALLING_LIST	String	No[a]	Name of the calling list.
GSW_CAMPAIGN_NAME	String	No[a]	Name of the campaign.
GSW_CHAIN_ATTR	String	No[c]	Flag determining whether to update the record chain or just the single record. Values are AllChain or RecordOnly. (Default = AllChain)
GSW_MESSAGE	String	No	DoNotCall message. Message to be written in DNC log.
GSW_PHONE	String	No[b]	Customer's phone number.
GSW_CUSTOMER_ID	String	No[b]	A user-defined field in the Calling List table that serves as a customer identifier for DoNotCall requests.
GSW_RECORD_HANDLE	Int	No[a]	Unique Record Identifier.

a. If `GSW_RECORD_HANDLE` is specified, then `GSW_CALLING_LIST` and `GSW_CAMPAIGN_NAME` are required.

b. If `GSW_RECORD_HANDLE` is not specified, then either `GSW_PHONE` or `GSW_CUSTOMER_ID` must be present. See "Mandatory Fields" on page 70.

c. If the `GSW_RECORD_HANDLE` attribute is specified, then the attribute `GSW_CHAIN_ATTR = AllChain` takes effect. In this case, OCS finds the chain to which the current record belongs and updates this chain in the calling list(s) as `NoCall.` Then, it inserts all of the phone numbers in the chain into the `DoNotCall` table.
If either the `GSW_PHONE` or `GSW_CUSTOMER_ID` attribute is specified, then OCS updates the calling list(s) and inserts only the phone number/Customer ID from the request into the `DoNotCall` table. It will not insert all of the other phone numbers/Customer IDs from the chain into the `DoNotCall` table.
If the `GSW_CHAIN_ATTR = RecordOnly` attribute is specified, then only the specified record is marked as DoNotCall. All other records in the chain can be considered for dialing.

Mandatory Fields

This statement covers two scenarios and the data key fields required for them. These vary, depending on what is specified in the event.

```
GSW_APPLICATION_ID
if (GSW_RECORD_HANDLE is specified)
    {
    GSW_CAMPAIGN_NAME
    GSW_CALLING_LIST
    }
else
{
At least one from the following:
    GSW_PHONE
    GSW_CUSTOMER_ID
    }
```

For example, if `GSW_RECORD_HANDLE` is specified, `GSW_CAMPAIGN_NAME` and `GSW_CALLING_LIST` must be specified.

DoNotCallAcknowledge

This event acknowledges a DoNotCall request. Table 70 contains more information.

Table 70: DoNotCallAcknowledge

DoNotCallAcknowledge User Event	
Description	Confirmation that DoNotCall was accepted.
Recommended Desktop Action	Delete the record and the chain.

Table 71 lists the attached data for the DoNotCallAcknowledge event.

Table 71: DoNotCallAcknowledge Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	DoNotCallAcknowledge
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CALLING_LIST	String	No[a]	Name of the calling list.
GSW_CAMPAIGN_NAME	String	No[a]	Name of the campaign.
GSW_CHAIN_ATTR	String	No	Flag determining whether to update the record chain or just the single record. Values are AllChain or RecordOnly. (Default = RecordOnly)
GSW_PHONE	String	No[b]	Customer's phone number.
GSW_CUSTOMER_ID	String	No[b]	A user-defined field in the Calling List table that serves as a customer identifier for DoNotCall requests.
GSW_RECORD_HANDLE	Int	No[a]	Unique Record Identifier.

[a] If GSW_RECORD_HANDLE is specified, then GSW_CALLING_LIST and GSW_CAMPAIGN_NAME are required.

[b] If GSW_RECORD_HANDLE is not specified, then either GSW_PHONE or GSW_CUSTOMER_ID must be present. See "Mandatory Fields" on page 70.

Scheduling and Rescheduling Records

An agent can reschedule any record on the desktop. There are two methods for rescheduling records:

- Use a RecordReschedule event to reschedule a call.
- Use a ScheduledRecordReschedule event when a rescheduled call cannot be completed and must be set for another time.

A record is typically rescheduled during a call when a customer requests a callback at a certain time. The agent sends a RecordReschedule to OCS and receives a RecordRescheduleAcknowledge in return. In Outbound Desktop Protocol Version 6, there is no difference between RecordReschedule and ScheduledRecordReschedule.

If the time of the requested callback is out of the boundaries of the "daily from" - "daily till" for the record: When the call is dialed, OCS recalculates the callback time by adding an appropriate amount of time to the original value, so the callback time occurs within the boundaries.

Callbacks can be assigned to either an individual or a group. Individual or campaign group callbacks can be made in any dialing mode. In the Predictive mode, group callbacks can be dialed by OCS and are treated like any other outbound call. See the option predictive_callback in the *Outbound Contact 7.6 Deployment Guide* for more information.

If scheduling callbacks is activated on the desktop, the agent can be notified to make a scheduled call by receiving the UserEvent ScheduledCall. The agent can be either a specific agent following up on a previous call or an agent assigned to the call from a group. For example, an agent is logged in and participating in a campaign. The database indicates that a customer should be called at a certain time. When this time comes, OCS retrieves the record and attempts to locate the agent scheduled to return that call.

The agent has the option of accepting, rescheduling, or rejecting the callback. If the agent rejects a scheduled call record, it is returned to OCS with its record_type marked General and its record_status marked Ready**.** That is, this record is handled by OCS as a brand-new record, losing its scheduled call status. If rejecting a record is not desirable, use the ScheduledRecordReschedule request to reschedule the record with a different callback type or different callback time.

RecordReschedule

The desktop sends this request to OCS to reschedule a record. Table 70 contains more information.

Note: A callback is not scheduled at the time request to reschedule a record is received and acknowledged by OCS. Instead, OCS waits for the explicit RecordProcessed event from the agent's desktop to finalize the callback scheduling.

Table 72: RecordReschedule

RecordReschedule Request	
Description	Request reschedule of Preview Record, Predictive Call, or Scheduled Call.
OCS Action	Update a record chain and reschedule the record.

Table 73 lists the attached data for a `RecordReschedule` event.

Table 73: RecordReschedule Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	`RecordReschedule`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CALLBACK_TYPE	String	Yes	Type of callback an agent wants to create, either Personal or Campaign.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_DATE_TIME	String	Yes	Date and time of scheduled call.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.

RecordRescheduleAcknowledge

OCS sends this event to the desktop to acknowledge a rescheduled record.Table 74 contains more information.

Table 74: RecordRescheduleAcknowledge

RecordRescheduleAcknowledge User Event	
Description	Confirmation that record was rescheduled.
Recommended Desktop Action	Continue Call Work (the agent performs work associated with the call, such as dialing or updating a record).

Table 75 lists the attached data for a `RecordRescheduleAcknowledge` event.

Table 75: RecordRescheduleAcknowledge Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	RecordRescheduleAcknowledge
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CALLBACK_TYPE	String	Yes	Type of callback an agent wants to create, either Personal or Campaign.
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_DATE_TIME	String	Yes	Date and time of scheduled call.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.

ScheduledRecordReschedule

The desktop sends this event to OCS to reschedule a previously rescheduled record. Table 76 contains more information.

Table 76: ScheduleRecordReschedule

ScheduleRecordReschedule User Event	
Description	Request reschedule Preview Record, Predictive Call, or Scheduled Call.
OCS Action	Update a record chain and reschedule the record.

Table 77 lists the attached data for a ScheduledRecordReschedule event.

Table 77: ScheduledRecordReschedule Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	ScheduledRecordReschedule
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.

Table 77: ScheduledRecordReschedule Attached Data (Continued)

Data Key	Type	Key Required	Description
GSW_CALLBACK_TYPE	String	Yes	Type of callback an agent wants to create, either Personal or Campaign. By default, if the attribute is not specified, callback type should not be changed.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_DATE_TIME	String	Yes	Date and time of scheduled call.
GSW_PHONE	String	No	Customer’s phone number.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.

ScheduledRecordRescheduleAcknowledge

OCS sends this event to the desktop to acknowledge the rescheduling of a scheduled record. Table 78 contains more information.

Table 78: ScheduledRecordRescheduleAcknowledge

ScheduleRecordRescheduleAcknowledge User Event	
Description	Confirmation that record was rescheduled.
Recommended Desktop Action	Continue “Call Work” (the agent performs work associated with the call, such as dialing or updating a record).

Table 79 lists the attached data for a `ScheduledRecordRescheduleAcknowledge` event.

Table 79: ScheduledRecordRescheduleAcknowledge Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	`ScheduledRecordRescheduleAcknowledge`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CALLBACK_TYPE	String	Yes	Type of callback an agent wants to create, either Personal or Campaign.
GSW_CALLING_LIST	String	Yes	Name of the calling list.

Table 79: ScheduledRecordRescheduleAcknowledge Attached Data (Continued)

Data Key	Type	Key Required	Description
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_DATE_TIME	String	Yes	Date and time of scheduled call.
GSW_PHONE	String	No	Customer's phone number.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.

ScheduledCall

OCS sends this event to notify the desktop that there is a scheduled call. Nonmandatory fields are sent only if the `send_attribute` option is defined. Table 80 contains more information.

Table 80: ScheduledCall

ScheduleCall User Event	
Description	OCS sends to agent to indicate that scheduled call should be executed.
Recommended Desktop Action	Perform Call Work (the agent performs work associated with the call, such as dialing or updating a record).
Mandatory Fields	GSW_USER_EVENT GSW_APPLICATION_ID GSW_CAMPAIGN_NAME GSW_CALLING_LIST GSW_RECORD_HANDLE GSW_PHONE GSW_CALL_RESULT GSW_CALLBACK_TYPE
Additional Fields	Genesys and user-defined fields which have the `send_attribute` option configured.

Table 81 lists the attached data for a `ScheduledCall` event.

Table 81: ScheduledCall Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	`ScheduledCall`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_ATTEMPTS	Int	No	Number of attempts for the record.
GSW_CALL_RESULT	Int	Yes	Call Result set by dialer or saved from previous call. (See the “Genesys Enumeration Table” on page 28.)
GSW_CALL_TIME	String	Yes	System time when record was called, in seconds from 1/1/70 (GMT).
GSW_CALLBACK_TYPE	String	Yes	Type of callback, either Personal or Campaign.
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_CHAIN_ID	Int	Yes	Unique chain ID.
GSW_DATE_TIME	String	Yes	Date and time of scheduled call.
GSW_FROM	Int	No	GSW_FROM - GSW_UNTIL: Time frame when a record can be called, seconds from midnight.
GSW_PHONE	String	Yes	Customer’s phone number.
GSW_PHONE_TYPE	Int	No	Customer phone type (See the “Genesys Enumeration Table” on page 28).
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.
GSW_TZ_OFFSET	Int	No	Offset (time difference) in seconds between UTC and a particular time zone. It may contain different values throughout the year if Daylight Savings Time (DST) is used for the specified time zone.
GSW_UNTIL	Int	No	Time until, seconds since midnight.
Custom fields	Any	No	Custom Fields.

Adding Records to a Calling List

When a campaign is running or loaded, an agent can add both new records to the calling list and new chained records to an existing chain. The agent can add new records to a calling list if all the fields in the record are consistent with those defined in the calling list table. It is also possible to add a new record and set up this record as a personal or campaign scheduled call.

AddRecord Request

The Agent sends the `AddRecord` request to OCS to add a new record to the database. The `AddRecord` request can be used only to add records to a running or loaded campaign. Only those fields defined with the `send_attribute` option are updated using the `AddRecord` request. In addition, the agent who sends this request should belong to the campaign group that is assigned for the campaign.

To add a new record or the next record in an existing chain to a campaign's calling list, the requests UserData must include the mandatory fields (as defined in the Key Required column in Table 83). Note that when adding a new record, the GSW_RECORD_HANDLE is not a required key. Since the record is new, it has not yet been assigned a GSW_RECORD_HANDLE. Instead, GSW_PHONE is the required key in this request and is used as the identifier for the record.

If OCS receives an `AddRecord` request without the `GSW_CHAIN_ID` attribute, OCS assigns the next available `chain_id` and chain number (chain_n) with a value of 0. This creates a new chain.

If an Agent wants to add a record to an existing chain, he or she must include the attribute `GSW_CHAIN_ID` (of the existing chain) in the request's UserData. In this case, OCS assigns the next available chain number (`chain_n)` when it adds the record to the chain.

If an Agent wants to assign a specific number to a record being added to a chain, the agent must include both attributes `GSW_CHAIN_ID` and `GSW_CHAIN_N` in the request's UserData.

Table 82 contains more information.

Table 82: AddRecord

AddRecord Request	
Description	Request to add a new record to the database.
OCS Action	Verify data and create new record in the list.

Table 83 lists the attached data for an `AddRecord` request.

Table 83: AddRecord Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	`AddRecord`
GSW_AGENT_ID	String	No	Login ID of last agent who worked with the record. Optional. (Default = 0)
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_ATTEMPTS	Int	No	Number of attempts for the record. This key is used when a new record is added. Optional. (Default = 0)
GSW_CALL_RESULT	Int	No	Call Result sent to change automatically detected call result. (See the "Genesys Enumeration Table" on page 28.)
GSW_CALL_TIME	String	No	System time when record was called, in seconds from 1/1/70 (GMT). Optional. (Default = 0)
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_CHAIN_ID	Int	No	Unique chain identifier. Optional. If missing, it is assumed that a record forms a new chain.
GSW_DATE_TIME	String	No	Date and time of scheduled call. Optional. (Default = 0)
GSW_FROM	Int	No	GSW_FROM - GSW_UNTIL: Time frame when a record can be called, seconds from midnight. (Default = 28800); (28800 is 8 AM)
GSW_PHONE	String	Yes	Customer's phone number.
GSW_PHONE_TYPE	Int	No	Customer phone type (See the "Genesys Enumeration Table" on page 28). (Default = 2, DirectBusinessPhone)
GSW_RECORD_STATUS	Int	No	Status of adding record sent from a desktop (See the "Genesys Enumeration Table" on page 28). Optional. (Default = 1, ready)

Table 83: AddRecord Attached Data (Continued)

Data Key	Type	Key Required	Description
GSW_RECORD_TYPE	Int	No	Type of added record sent from a desktop. See the “Genesys Enumeration Table” on page 28. Optional. (Default = 2, general)
GSW_TZ_NAME	String	Yes	Configuration Server Time Zone Name (usually standard three-letter abbreviation.
GSW_CHAIN_N	Int	No	Unique number in a chain. Optional. If missing, the next available number is assigned.
GSW_UNTIL	Int	No	GSW_FROM - GSW_UNTIL: Time frame when a record can be called, seconds from midnight. (Default = 64800, which represents 6 PM.)
Custom fields	Any	No	Custom Fields.

AddRecordAcknowledge

OCS sends this event to the desktop to acknowledge an added record. Table 84 contains more information.

Table 84: AddRecordAcknowledge

AddRecordAcknowledge User Event	
Description	OCS sent this insert request to database.
Recommended Desktop Action	Continue session.

Table 85 lists the attached data for an `AddRecordAcknowledge` event.

Table 85: AddRecordAcknowledge Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	`AddRecordAcknowledge`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.

Table 85: AddRecordAcknowledge Attached Data (Continued)

Data Key	Type	Key Required	Description
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.

Unsolicited Notifications

Unsolicited notifications are messages that OCS sends to the agent desktop but not in response to agent requests. These are examples of unsolicited notifications:

- Scheduled record that OCS sends to the desktop without any prompting (request) from the agent.
- Notification about a cancelled record that OCS sends to a desktop other than the one that initially submitted the Cancel/DoNotCall request.
- Campaign Status and Agent Assignment notifications.

Agent Logout

Upon an agent `LogOut` request, the desktop performs the following cleanup tasks before executing the requests:

If there are existing preview records or scheduled calls, the desktop should send a `RecordReject` request to OCS, thus returning these records to the calling list table and freeing up these records for other agents to process. The `record_type` and `record_status` of the rejected records will be returned to General and Ready state.

If the agent does not perform a cleanup (reject records) before logging out, OCS, upon receiving an `LogOut` request from T-Server, returns the remaining records on the desktop to the calling list with status `updated`.

Agent Logout Protocol

The extension of the Desktop Protocol (LogOut) addresses the issue of abandoned or dropped calls as a result of this combination of circumstances:

- A significant number of Agents in a Campaign Group log out after OCS has already requested dialing of outbound calls.
- OCS is relying on the availability of these specific Agents to handle the calls dialed.

In this scenario, many of the answered calls would be abandoned or dropped due to Agent unavailability. A Desktop Protocol extension allows the Agent to notify OCS in advance about his or her intention to log out and to receive notification when log out is possible without a negative impact on outbound dialing. The protocol works like this:

- Instead of an actual logout, the agent sends a `LogOut` request to OCS to indicate his or her intention to log out. See "LogOut". After receiving the initial `LogOut` request, OCS excludes the agent from the list of available agents and stops considering him or her for dialing prediction.
- A `LogOutTime` response from OCS provides an estimated time by which the agent will be able to log out. See "LogOutTime" on page 83. In response to each of the agent's logout requests, OCS gives the agent an estimated logout time until that time expires. During this interval the agent may receive an outbound call. OCS recalculates the estimated time for each logout request.
- OCS notifies the desktop when logout is possible. The agent is able to log out when the estimated time expires or when the agent has processed the outbound call that OCS dialed in expectation of the agent's availability.

Figure 5 shows the Desktop-OCS user events (request and responses) for an agent logout.

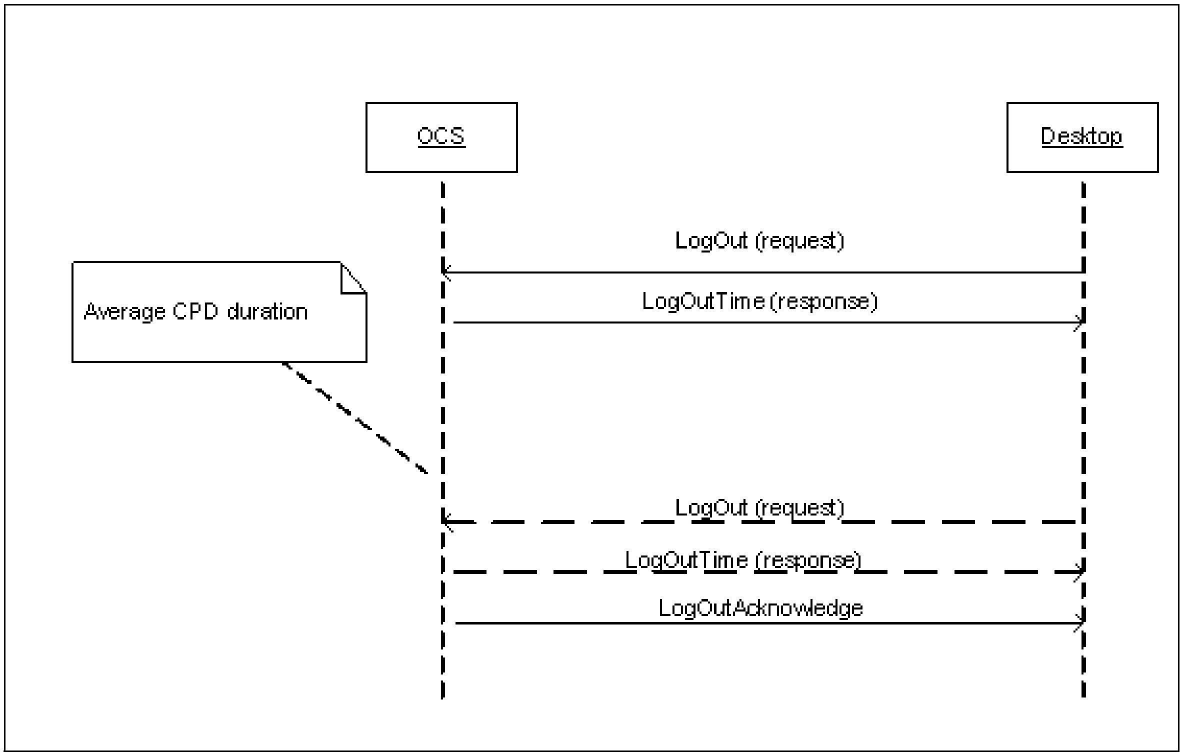

Figure 5: Logout Negotiation between Agent Desktop and OCS

LogOut

Table 86 provides information on this event.

Table 86: LogOut

LogOut Request	
Description	Request to log out.
OCS Action	OCS excludes the agent from predictive dialing. If OCS has already requested a dialer for an outbound call for which the agent is regarded as available, OCS postpones the Logout for a period of time as specified in the `call_wait_connected_timeout` option for all agents regardless of the number of `Sent` or `Dialed` calls in progress. If there are no `Sent`, `Dialed` and `Queued` calls, OCS sends a LogOut time equal to `0`.

Table 87 lists the attached data for the `LogOut` event.

Table 87: LogOut Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	`LogOut`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.

LogOutTime

OCS sends this response to the desktop for the agent's `LogOut` request. Table 88 contains more information.

Table 88: LogOutTime

LogOutTime User Event	
Description	Response to `LogOut` request
Desktop Action	Desktop displays the time remaining until it or the agent will be able to complete logout.

Table 89lists the attached data for the `LogOutTime` event.

Table 89: LogOutTime Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	`LogOut`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.
GSW_LOGOUT _TIME	Int	Yes	The time remaining before the logout will be allowed.

LogOutAcknowledge

Table 90 provides information on this event.

Table 90: LogOutAcknowledge

LogOutAcknowledge User Event	
Description	Automatic logout acknowledgement
Desktop Action	Logs agent out. Displays agent’s status change.

Table 91 lists the attached data for the `LogOutAcknowledge` event.

Table 91: LogOutAcknowledge Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	`LogOutAcknowledge`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CAMPAIGN_NAME	String	Yes	Name of the campaign.

Proactive Interaction Support

Table 92 provides information about additional data keys needed to enable proactive interaction functionality in the desktop. This feature is also known as `Push Preview` mode.

Table 92: Proactive Interaction Attached Data

Data Key	Type	Description
GSW_AGENT_ID	String	AgentID of the agent assigned to the proactive interaction record.
GSW_SWITCH_DBID	Integer	DBID of the Switch.

Table 93 provides information about identifying the media types that correspond to the `Media Type` business attribute. This defines how to contact the customer.

Table 93: Media Type Business Attribute

Data Key	Type	contact_info_type Field	Description
GSW_CONTACT_MEDIA_TYPE	string	0	any (NoContactType)
		1	voice (HomePhone)
		2	voice (DirectBusinessPhone)
		3	voice (BusinessWithExt)
		4	voice (Mobile)
		5	voice (VacationPhone)
		8	voice (VoiceMail)
		10	email (E-mail)

Caller ID Support

This feature enables OCS to distribute information required for Caller ID support to any telephony system.

Note: Currently, this feature is supported:

- By the Alcatel A4400/OXE PBX and the Avaya PBX,
- When using CPD Server in ASM mode with a trunk-side ISDN connection to the PSTN, or
- When using Outbound Notification Manager and the `Power GVP` dialing mode.

The Caller ID support features include:

- Caller ID Per Campaign

  The Caller ID can now be specified per campaign.

  This simplifies telemarketing regulation compliancy setup in certain cases, such as when a single site must transmit multiple, different Caller IDs, depending on the Outbound Campaign.

  OCS submits information if the option `CPNDigits` is configured in the corresponding Campaign/Application Configuration Object.
- Caller ID Support for ISDN Connections

  The Caller ID is transmitted to PSTN when using CPD Server with an ISDN connection (ASM mode).

  In this case, CPD Server uses the value from `CPNDigits` and `CPNPresentation` received from OCS, instead of the value specified in the `calling-party-number` and `presentation-indicator` options in the ISDN section. For other parameters, CPD Server uses the values configured in the CPDServer Options/ISDN section.
- Caller ID Support for Avaya CTI

  This capability works with the Avaya green feature, enabling the Caller ID transmission through the CTI interface.

  OCS submits information that is required to provide the Caller ID in an outbound call in the Extensions TKVList in `TMakePredictiveCall` or `SMakePredictiveCall` functional calls.

  The CPD Server submits information received from OCS as specified in the Extensions TKVList in a `TMakeCall` request. It does this without checking the `SwitchingOffice` type to determine if it is Avaya.

The Caller ID options for each campaign for CPN (Calling Party Number) are:

- `CPNDigits`
- `CPNPlan`

- CPNPresentation
- CPNScreening

For more information, see the “Outbound Configuration Options” chapter in the *Outbound Contact 7.6 Deployment Guide*.

Virtual Agent Support for Notifications

Genesys-integrated Interactive Voice Response (IVR) provides virtual agent support for IVR ports that are configured for a campaign as “virtual agents.” The virtual agent support for notifications feature includes Dialing for IVR and Blending for IVR.

This functionality simplifies the integration of Outbound Contact with IVR for outbound notifications. Agent logins and ready statuses are not required for IVR ports.

Depending on specific implementation, IVR ports can be represented by any of the following:

- Places that include one DN with the type Voice Treatment Port.
- Places that include one DN with type ACD Position
- Places that include two DNs with types "ACD Position" and "Extension"

OCS provides simplified resource availability management for IVR Groups. The IVR Group must be configured as a Group of Places with the option ivr_group=true in the Annex tab.

- Places in that Group may contain DNs with the type "Position", "Extension," or "Voice Treatment Port".

When OCS is processing a Campaign with the IVR Group assigned, the following guidelines apply:

- OCS does not rely on TEvents related to Agent (EventAgentLogin, EventAgentLogout, EventAgentReady, EventAgentNotReady) received on DNs with the type Voice Treatment Port associated with an IVR Group.
- If the Place includes a DN with the type ACD Position, OCS expects EventAgentLogin on that DN to associate the Place with a campaign.
- OCS considers Place available to receive an outbound Call, if there is no telephony activity in progress on the DNs included in that Place. For example, if an EventReleased was received on behalf of a previously established Call, and DNs (or Place) are enabled in Genesys Configuration.
- OCS considers the Place seized by a Call when any telephony activity is begun on at least one of the DNs included in that Place: For example, if EventRinging was received.
- OCS finalizes the Record processing immediately after release of a Call on DN.

- When a Place includes a DN with the type ACD Position, the OCS behavior on `EventReleased` is the following: OCS changes the agent's state to Ready and does not take into consideration the option `outbound_release_action`.
- The option `ivr_update_on_release` enables OCS to update the Calling List Record with values from Outbound Call UserData. If `ivr_update_on_release=true`, OCS updates Fields from Record with values from the corresponding `UserData KVPairs`, received in `EventReleased`. This is similar to the `UpdateCallCompletionStats` in UserEvent processing.
- OCS uses the same mechanism of "inbound call blending" as it uses for standard Campaigns.
- OCS does not process Desktop Protocol interactions related to Call processing on DNs associated with an IVR Group.
- OCS enables the transfer of calls from IVR Group to Places/Agents from regular (non-IVR) groups. Call records are not updated just after leaving an IVR Group. These records could be processed by agents according to Desktop Protocol.
- License control for an IVR Group is the same as for regular groups. The number of places assigned to an IVR Group is equal to the number of consumed licenses.
- A Group-Campaign with the option `ivr_group=true` is considered as an IVR on loading the campaign. After this, OCS does not take dynamic changes of the option until unloading the campaign.
- OCS enables the dynamic addition and removal of places to and from the IVR Group. Once a place with a logged in agent is removed from the group, it is no longer considered as IVR place. This place could be added to a regular group.
- OCS does not support `IVR Campaign_Group` in ASM dialing mode.

Note: Only "IVR behind the Switch" deployment is supported. Requirements for Outbound configuration and Call distribution are the same as for a standard Campaign with Agent or Place Group.

The options for the IVR features are:

- `ivr_group`
- `ivr_update_on_release`

For more information on these options, see the “Outbound Contact Configuration Options” chapter in the *Outbound Contact 7.6 Deployment Guide*.

Personalized Ring Tone Support

CPD Server utilizes the event flow patterns specific for personalized ring tone services to correctly detect the call results when dialing to the numbers that use these services. When using this feature, the dialer hears a custom music or voice message instead of a ring tone or busy signal.

This provides robust call progress detection for the numbers using personalized ring tone services.

The options for setting this feature are:

- `cpd-if-established`
- `pre-connect-cpd-priority`
- `post-connect-cpd-priority`

For information about these options, see the "Outbound Contact Configuration Options" chapter in the *Outbound Contact 7.6 Deployment Guide.*

Outbound Contact 7.6 Library

The following section describes:

- Error names and codes.
- All events and event type protocols.

Error Names and Codes

Table 94 displays error names and their corresponding codes for error conditions that occur while using communication protocols.

Table 94: Error Names and Codes

GSW_ERROR	GSW_ ERROR_ NUMBER	Description
Invalid Request	101	Received request has the wrong request type.[a]
Attribute Not Found	102	Mandatory attribute cannot be found.
Invalid Attribute Value	103	Attribute has the wrong value
Agent Not Found	104	OCS cannot find an appropriate agent to process the request
Campaign Group Not Found	105	Specified campaign group was not found.
No Active Campaigns	106	Cannot execute request—no campaign was loaded.

Table 94: Error Names and Codes (Continued)

GSW_ERROR	GSW_ ERROR_ NUMBER	Description
No Running Preview Campaigns	107	Cannot execute preview record request—no preview campaign was started.
No Records Available	108	All lists are empty, all records have been processed, or the internal buffer is empty. OCS is waiting for a new selection of records.
Record Not Found	109	OCS received a request for a record that does not exist or that has already been processed.
Invalid Time	110	Received time does not meet the request conditions (for example, reschedule in the past).
Invalid Time Format	111	OCS cannot convert the string to a time (for example, 25/45/00).
No call found for the record handle	112	Received request refers to a record that has already been processed.
DB Error	113	Cannot execute the request due to database error.
Chained Records not found	114	Received request refers to an absent chain of records.
Record Already Exists	115	Attempted to add a record that already exists.
Add Record Error	116	Cannot add the record.
Scheduled record not found	117	Cannot reschedule a record.
Preview mode has already been started	118	Preview mode has already been started.
Preview mode has not been started	119	Preview mode has not been started.

a. When `GSW_ERROR_NUMBER = 101`, the `GSW_ERROR` message can refer to three different messages:

- `PreviewDialingModeStart is required`, means that an agent must send a `PreviewDialingModeStart` request before issuing a desktop request if the `agent_preview_mode_start` option is set to `true`.
- `There is no 'Auto' campaign started`, means that an agent is trying to perform a smart logout when there are no auto (`Predictive` mode or `Progressive` mode) campaigns started.
- `Agent smartly logged out`, means that an agent is sending requests after performing a smart logout, but there is a record currently on the desktop.

All Genesys Events and Event Type Protocols

Table 95 represents all Genesys event and event type protocols.

Note: Starting with release 7.5, only version 6 of the desktop protocol is supported.

Key:

- O > D denotes sending a message from OCS to desktop.
- D > O denotes sending a message from desktop to OCS.

Table 95: All Desktop Protocol Events and Event Type Protocols

Messages	From > To	Description and Actions
1. Notifications		
CampaignStarted	O > D	Should be sent when campaign dialing is started or resumed, or as a response to event agent login if campaign is started.
CampaignStopped	O > D	Should be sent when dialing for campaign is stopped or paused. All lists in campaign deactivated.
CampaignModeChanged	O > D	Should be sent when mode of running campaign is changed.
CampaignLoaded	O > D	Should be sent when campaign is loaded.
CampaignUnloaded	O > D	Should be sent when campaign is unloaded.
CampaignGroupAssigned	O > D	Should be sent when the agent has been assigned to a campaign group.
CampaignStatusRequest	D > O	Request for information on active/ running campaign group(s) statuses.
2. Preview		
PreviewRecordRequest	D > O	Request to send preview record.
PreviewRecord	O > D	Preview record to dial.
NoRecordsAvailable	O > D	No more records available.

Table 95: All Desktop Protocol Events and Event Type Protocols (Continued)

Messages	From > To	Description and Actions
3. Common		
UpdateCallCompletionStats	D > O	Desktop sends this event to update record details. Intermediate update.
UpdateCallCompletionStatsAcknowledge	O > D	OCS sends this event to confirm operation.
ReadyTime	D > O	Desktop sends this request to OCS, providing the number of seconds before the agent will go `Ready`.
ReadyTimeAcknowledge	O > D	OCS sends this event to the desktop to acknowledge the `ReadyTime` request.
RecordProcessed	D > O	Desktop sends this event to indicate that record is processed. OCS should update record if it is provided.
RecordProcessedAcknowledge	O > D	OCS confirms that record has been executed.
RecordReject	D > O	Desktop sends this request to indicate that the preview record or scheduled call will not be dialed by this agent. This record should be re-sent to another agent.
RecordRejectAcknowledge	O > D	
RecordCancelAcknowledge	O > D	
RequestRecordCancel	D > O	Desktop sends this request to indicate that the preview record or scheduled call should not be dialed. Record should not be re-sent to another agent. It should be marked in the database as `canceled`.
RecordReschedule	D > O	Request a reschedule of preview record, predictive call, or scheduled call.
RecordRescheduleAcknowledge	O > D	Confirmation that record was rescheduled.

Table 95: All Desktop Protocol Events and Event Type Protocols (Continued)

Messages	From > To	Description and Actions
ScheduledCall	O > D	OCS sends this event to an agent to indicate that scheduled call should be executed.
ScheduledCallAcknowledge	D > O	
4. Chained Records		
ChainedRecordRequest	D > O	Request to send all records from the chain defined by Record Handle (Unique Record Identifier).
ChainedRecord	O > D	
ChainedRecordsDataEnd	O > D	
5. DoNotCall		
DoNotCall	D > O	
DoNotCallAcknowledge	O > D	
DoNotCallError	O > D	
6. Record Cancel from OCS to Desktop		
RecordCancel	O > D	OCS sends this to the desktop to indicate that this record should not be dialed. Applicable for preview records and scheduled calls.
7. Request add record from Desktop		
AddRecord	D > O	
AddRecordAcknowledge	O > D	Phone number can be used to relate request and response.
8. Request LogOut		
LogOut	D > O	Agent’s request to log out
LogOutTime	O > D	OCS response to LogOut request
LogOutAcknowledge	O > D	Automatic logout acknowledgement

Chapter

2 Communication DN API

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

Overview

Outbound Contact provides a Communication DN (CommDN) API that allows third-party applications, such as an inbound agent desktop, to submit DoNotCall (DNC) and record cancel requests. To use the API, a custom application must be able to access Genesys T-Server and Configuration Server, both of which have an open API.

The Communication DN API also enables users to control campaigns and campaign sequences through third-party applications or scripts instead of OCM. The third-party applications (customer applications) can be GUI applications or automated scripts that perform different kinds of scheduling, sequencing, and so on. For example, scripts can be customized to do such things as stop campaigns when all the records are dialed or mark some records as `Cancelled`.

In order for OCS to process requests from a third-party application, it is necessary to set up a connection between them. You can do this in either the third-party application or OCS.

Connection using OCS Application Object

1. Create an application of a type `Third-Party Server` in Configuration Manager.
2. Add this application object to the `Connection` tab of the `OCS` application.

Connection using Third-Party Application Object

1. Create an application of a type `Third-Party Application` in Configuration Manager.
2. Add the `OCS` application object to the `Connection` tab of this application.

OCS and API Requests

OCS accepts only those API requests that come from the following sources:

- Third-party servers included in the OCS `Connections` tab
- Third-party applications that include the `OCS` application object in their `Connection` tabs.

All other requests are omitted.

Data Formats

OCS and third-party applications communicate through the Communication DN API by means of `UserEvents` (with attached user data) that are sent and received on a CommDN. The attached user data is encoded as a key-value pairs list (TKVList). Values can be either string or integer. These values are described in "User Data Enumeration Values" on page 101. The communication is based on two types of messages: Request-Response and unsolicited notification.

Protocol Sequencing

OCS uses three types of messages to communicate:

- Requests
- Responses
- Notifications

Figure 6 shows the messaging sequence of the Communication DN API protocol.

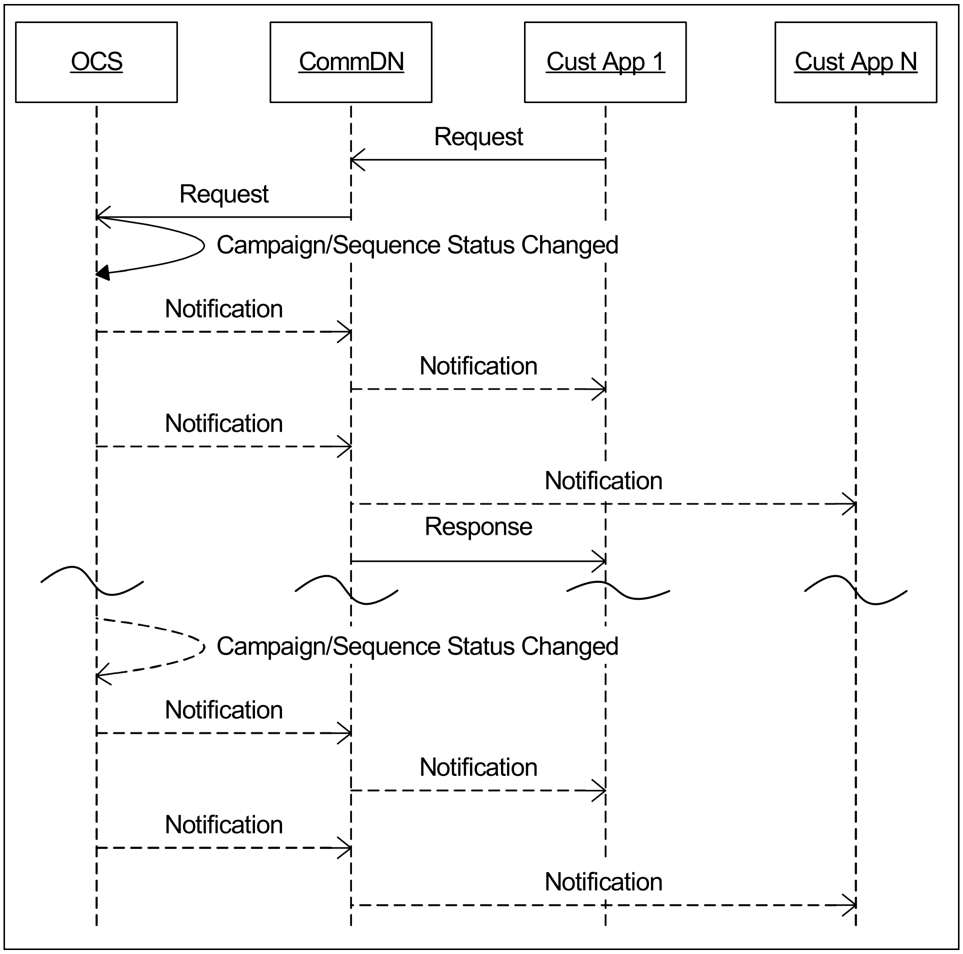

Figure 6: Protocol Sequencing for the Communication DN API

Mandatory Attributes

Requests or events sent through the CommDN must include the following mandatory attributes:

- `OriginAppDBID` (the DBID of the sender)

  If the `OriginAppDBID` in a request pertains to a third-party application, you must configure it according to the common Communication DN protocol, as explained in Chapter 1 on page 15.

- `TargetAppDBID` (the DBID of the receiver)

Table 96 shows the communication structure for the Communication DN API. If OCS receives an incorrect request or the wrong data or request sequence, it may send a `CM_EvError` event.

Table 96: Communication Structure

Request	Response/Notification	Mandatory Attributes
CM_ReqLoadCampaign	CM_EvCampaignLoaded	• CampaignDBID or Properties OCS checks CampaignDBID. If the value is 0, the request or event must have the proper schedule in the Properties attribute. • GroupDBID • DialMode • OptimizeBy • OptimizeGoal
CM_ReqUnloadCampaign	CM_EvCampaignUnloaded	• CampaignDBID • GroupDBID
CM_ReqGetCampaignStatus	CM_EvCampaignStatus	Request • CampaignDBID or Properties OCS checks CampaignDBID. If the value is 0, the request or event must have the proper schedule in the Properties attribute. If CM_ReqGetCampaignStatus CampaignDBID equals 0, OCS responds with the sequence status. • GroupDBID Response or Notification • CampaignDBID or Properties • GroupDBID • DialMode • OptimizeBy • OptimizeGoal • GroupCampStatus. If a campaign belongs to a Sequence, then the attribute GroupCampStatus status represents the part of each scheduleItem (<n> in the Sequence. See Figure 7 on page 100.

Table 96: Communication Structure (Continued)

Request	Response/Notification	Mandatory Attributes
CM_ReqSetDialingMode	CM_EvDialingModeChanged	• CampaignDBID or Properties • GroupDBID • DialMode • OptimizeBy • OptimizeGoal
CM_ReqStartDialing	CM_EvDialingStarted	• CampaignDBID or Properties • GroupDBID • DialMode • OptimizeBy • OptimizeGoal
CM_ReqStopDialing	CM_EvDialingStopped	• CampaignDBID or Properties • GroupDBID
CM_ReqDoNotCall	CM_EvDoNotCallProcessed	• Phone • CustomerID
CM_ReqCancelRecord	CM_EvRecordCanceled or CMEvError	• OriginAppDBID (the DBID of the sender) • TargetAppDBID (the DBID of the receiver) • Phone For CM_ReqCancelRecord, the TargetAppDBID may be 0, which means that all Outbound Contact Servers that monitor the communication DN will process this request and submit a response.

Special OCS Option

Usually OCS works with all existing CommDNs in the Configuration Database. You can reduce the number of CommDNs OCS uses by assigning the option outbound_contact_server to these DNs. Set this option’s value to true if you want OCS to communicate with third-party applications through a specified DN. To configure this option, see outbound_contact_server in the *Outbound Contact 7.6 Deployment Guide* for more information.

The following three examples describe how to apply the option outbound_contact_server.

1. You can set at least one CommDN to a value of `true` for this option. OCS works only with those CommDNs set to `true`. OCS disregards all CommDNs not set to `true`.

   Example 1:

   CommDN_1: `outbound_contact_server = true`
   CommDN_2: `outbound_contact_server = false`
   CommDN_3: `outbound_contact_server = undefined`

   In this configuration, OCS uses only CommDN_1.

2. You can set some CommDNs to a value of `false` and set others to a value of `undefined`. In this set up, all CommDNs with a value of `false` are excluded from OCS, while the undefined values remain available to OCS.

   Example 2:

   CommDN_1: `outbound_contact_server = false`
   CommDN_2: `outbound_contact_server = undefined`
   CommDN_3: `outbound_contact_server = undefined`

   In this configuration, OCS uses CommDN_2 and CommDN_3.

3. You can set all CommDNs to an undefined value (default value) for this option to make CommDNs available for OCS.

   Example 3:

   CommDN_1: `outbound_contact_server = undefined`
   CommDN_2: `outbound_contact_server = undefined`
   CommDN_3: `outbound_contact_server = undefined`

   In this configuration, OCS uses all CommDNs.

User Event Structure

Figure 7 shows the user event structure for communication between third-party applications and the Communication DN API.

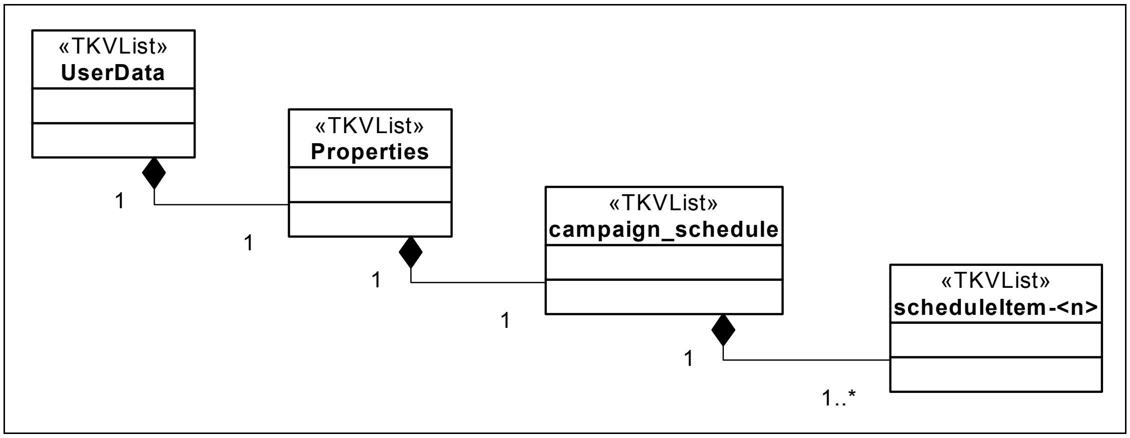

Figure 7: User Event Structure for the Communication DN API

Note: The event scheduleItem-<*n*> (<*n*> represents an integer) is formed by the prefix "scheduleItem-" and the number (converted to string), which equals 1...nItems. For more information about the user event, campaign_schedule, see Table 99 on page 102.

User Data Enumeration Values

Some of the Genesys mandatory fields are represented as predefined integer constants. When these fields are attached to user events or telephony events as key-value pairs, the values of these fields are sent as integers (sometimes also called Enumeration values or internal representations). Table 97 lists the Genesys user event attributes sent with user data through the Communication DN API.

Table 97: User Event Attributes for User Data (TKVList)

Key	Type	Description
GSW_CM_MessageType	Int	See the Enum values for GSW_CM_MessageType on page 103.
GSW_CM_AttrDialMode	Int	See the Enum values for GSW_CM_AttrDialMode on page 106.
GSW_CM_AttrOptimizeBy	Int	See GSW_CM_AttrOptimizeBy Enum values on page 106.
GSW_CM_AttrOptimizeGoal	Int	Values from 0 - 100 percent, or from 0 to <as required> seconds, represent target values for the Optimization parameter.
GSW_CM_AttrGroupCampStatus	Int	See the Enum values for GSW_CM_AttrGroupCampStatus on page page 107.
GSW_CM_AttrCampaignID	Int	Target Campaign DBID.
GSW_CM_AttrGroupID	Int	Target Group DBID.
GSW_CM_AttrError	Int	If no error, value is 0. See Enum values for GSW_CM_AttrError on page 105.
GSW_CM_AttrErrorMessage	String	String describing the error that occurred.
GSW_CM_AttrOriginAppID	Int	Application DBID.

Table 97: User Event Attributes for User Data (TKVList) (Continued)

Key	Type	Description
GSW_CM_AttrTargetAppID	Int	Application DBID.
GSW_CM_AttrProperties	TKVList	Attribute's properties.

Table 98 shows `UserEvent` attributes for `GSW_CM_AttrProperties` (TKVList).

Table 98: User Event Attribute for GSW_CM_AttrProperties (TKVList)

Key	Type	Description
campaign_schedule	TKVList	Contains information about a Campaign Sequence.
cancel_record	TKVList	Contains additional request attributes
do_not_call	TKVList	Contains additional request attributes

Table 99 shows `UserEvent` attributes for `campaign_schedule` (TKVList).

Table 99: User Event Attributes for campaign_schedule (TKVList)

Key	Type	Description
Description	String	Description of Campaign Sequence
startTime	Int	Time to start the Sequence (UTC)
nItems	Int	Number of items in the Sequence
scheduleItem-<n>	TKVList	Properties of the Sequence item <n>

Table 100 shows the `UserEvent` attributes for scheduleItem-<*n*>.

Table 100: User Event Attributes for scheduleItem-<*n*>

Key	Type	Description
stopAtTime	Int	Stop the campaign at a specified time.
stopAtContacts	Int	Stop the campaign when the predefined number of customers is contacted (number of transferred calls).
stopAtDials	Int	Stop the campaign when the specified number of dial attempts are made.

Table 100: User Event Attributes for scheduleItem-<*n*> (Continued)

Key	Type	Description
sleepBeforeNextStart	Int	The wait time, in minutes, before the start of this campaign.
campaignDBID	Int	DBID of the campaign.
dialMode	Int	Dial mode for the campaign. See the Enum values for GSW_CM_AttrDialMode on page 106.
optMethod	Int	Optimization method for the campaign. See the Enum values for GSW_CM_AttrOptimizeBy on page 107.
optMethodValue	Int	Values from 0-100 percent represent target value for the Optimization parameter.
status	Int	Status of the campaign. See the Enum values for GSW_CM_AttrGroupCampStatus on page 107.

The Enumeration (Enum) values for the user event attributes in this chapter are listed in Table 101 based on their user data type.

Table 101 displays the Enumeration values for the user data GSW_CM_MessageType, separated by responses and requests, and it includes error messages.The table also indicates that some values are *not applicable*, which means that they are not used by the CommDN API in Outbound Contact.

Table 101: Data Enumeration Values for GSW_CM_MessageType

Data	Value	Comment
Messages That OCM Uses to Communicate with OCS		
MSGCFG_NONE	0	not applicable
MSGCFG_UNKNOWN	1	not applicable
MSGCFG_ERROR	2	not applicable
MSGCFG_CLIENTREGISTER	3	not applicable
MSGCFG_DISCONNECTED	4	not applicable

Table 101: Data Enumeration Values for GSW_CM_MessageType (Continued)

Data	Value	Comment
CM_UnknownMessage	5	not applicable
Requests		
CM_ReqRegisterClient	6	not applicable
CM_ReqLoadCampaign	7	Request to load a campaign.
CM_ReqUnloadCampaign	8	Request to unload a campaign.
CM_ReqStartDialing	9	Request to start dialing a campaign.
CM_ReqStopDialing	10	Request to stop dialing a campaign.
CM_ReqSetDialingMode	11	Request to change dialing parameters for a campaign.
CM_ReqGetCampaignStatus	12	Request for campaign status.
CM_ReqCampaignRegistered	13	not applicable
CM_ReqCampaignUnregistered	14	not applicable
CM_ReqForceUnloadCampaign	29	Request to force campaign unloading.
CM_ReqCancelRecord	30	Request to cancel record from third-party application
CM_ReqDoNotCall	32	Request to add phone number of customer ID to Do-Not-Call List.
Responses		
CM_EvServerConnected	15	not applicable
CM_EvServerDisconnected	16	not applicable
CM_EvClientDisconnected	17	not applicable
CM_EvClientRegistered	18	not applicable
CM_EvCampaignLoaded	19	Acknowledge for request CM_ReqLoadCampaign.
CM_EvCampaignUnloaded	20	Acknowledge for request CM_ReqUnloadCampaign.

Table 101: Data Enumeration Values for GSW_CM_MessageType (Continued)

Data	Value	Comment
CM_EvDialingStarted	21	Acknowledge for request CM_ReqStartDialing.
CM_EvDialingStopped	22	Acknowledge for request CM_ReqStopDialing.
CM_EvDialingModeChanged	23	Acknowledge for request CM_ReqSetDialingMode
CM_EvCampaignStatus	24	Response or Notification when campaign mode is changed.
CM_EvCampaignRegistered	25	not applicable
CM_EvCampaignUnregistered	26	not applicable
CM_EvError	27	Wrong event error received.
GSW_CM_ReqCommDNGetCampaignData	28	not applicable
GSW_CM_ReqForceUnloadCampaign	29	Request to force the campaign to unload.
CM_EvRecordCanceled	31	Acknowledgement for request CM_ReqCancelRecord
CM_EvDoNotCallProcessed	33	Acknowledgement of request CM_ReqDoNotCall

Table 102 displays the Enumeration values for the user data GSW_CM_AttrError.

Table 102: Enumeration Values for GSW_CM_AttrError

Error	Value	Comment
CM_ERROR_NO	0	not applicable
CM_ERROR_SERVER_CONNECTED	1	not applicable
CM_ERROR_REGISTER_CLIENT	2	not applicable
CM_ERROR_CAMPAIGN_NOT_FOUND	3	Requested campaign not found in configuration.
CM_ERROR_CAMPAIGN_NOT_LOADED	4	Requested campaign not loaded.

Table 102: Enumeration Values for GSW_CM_AttrError (Continued)

Error	Value	Comment
CM_ERROR_CAMPAIGN_ALREADY_LOADED	5	Requested campaign already loaded.
CM_ERROR_CAMPAIGN_NOT_STARTED	6	Request to change runtime parameters for a campaign that has not started.
CM_ERROR_CAMPAIGN_ALREADY_STARTED	7	Request to start an already started campaign.
CM_ERROR_GROUP_NOT_FOUND	8	Requested group not found in configuration.
CM_ERROR_GROUP_CAMP_NOT_FOUND	9	Requested campaign is not configured for the requested group.
CM_ERROR_INVALID_PARAMETER	10	Invalid parameter in the CM_ReqSetDialingMode request.
CM_ERROR_INVALID_CAMPAIGN_MODE	11	Invalid mode is requested for running campaign.
CM_ERROR_INVALID_CAMPAIGN_SCHEDULE	12	Wrong Campaign Sequence is received.
CM_ERROR_CAMPAIGN_SCHEDULE_NOT_FOUND	13	Campaign Sequence is not found among loaded or running Sequences.
CM_ERROR_INVALID_CAMPAIGN_SCHEDULE_MODE	14	Invalid mode is requested for running Campaign Sequence.

Table 103 shows the Enumeration values for the user data GSW_CM_AttrDialMode.

Table 103: Enumeration Values for GSW_CM_AttrDialMode

Enumeration	Value	Comment
CFGDMPredict	1	Predictive mode
CFGDMProgress	2	Progressive mode
CFGDMPreview	3	Preview mode

Table 103: Enumeration Values for GSW_CM_AttrDialMode (Continued)

Enumeration	Value	Comment
CFGDMProgressAndSeize	4	Progressive with engage mode
CFGDMPredictAndSeize	5	Predictive with engage mode

Table 104 shows the Enumeration values for the user data GSW_CM_AttrOptimizeBy.

Table 104: Enumeration Values for GSW_CM_AttrOptimizeBy

Enumeration	Value	Comment
CFGOMBusyFactor	1	Optimize busy factor
CFGOMOverdialRate	2	Optimize overdial rate
CFGOMWaitTime	3	Optimize wait time

Table 105 shows the Enumeration values for the user data GSW_CM_AttrGroupCampStatus.

Table 105: Enumeration Values for GSW_CM_AttrGroupCampStatus

Enumeration	Value	Comment
CM_GCS_WaitingUnload	1	Status waiting unload
CM_GCS_UnloadInProgress	2	Status unload in progress
CM_GCS_InActive	3	Status inactive
CM_GCS_Active	4	Status active
CM_GCS_Running	5	Status running

Record Cancellation from a Third-Party Application

From a third-party application, agents who are not participating in a particular Outbound campaign may cancel a record by phone number in that campaign. An extended Communication DN Protocol for OCS gives end users this additional control over campaigns.

A custom, third-party application needs access to a Genesys T-Server and Configuration Server, both of which have an open API. Communication is conducted by the means of `UserEvents` sent and received on a Communication DN. T-Server conveys `UserData` attached to an event. The data are encoded in the key-value pairs list (TKVList).

OCS communicates with third-party applications by means of request-response.

- Request: `CM_ReqCancelRecord`
- Response: `CM_EvRecordCanceled`

The mandatory attributes are `Phone, OriginAppDBID`, and `TargetAppDBID`:

- The `OriginAppDBID` attribute is the DBID of the sender. If, in the request, the `OriginAppDBID` attribute pertains to the third-party application, this application should be configured according to the common Communication DN protocol policy.
- The `TargetAppDBID` attribute is the DBID of the receiver. Note that for `CM_ReqCancelRecord`, the value of `TargetAppDBID` may be `0`, which signifies that all OCS servers monitoring the communication DN will process this request and submit a response.

UserEvent Structure

The following depicts the event structure for the T-Server events pertaining to the cancellation of calling records from a third-party application:

```
UserEvent
        |
        UserData
                |
                "GSW_CM_MessageType" 30
                ["GSW_CM_AttrError" 0]
                "GSW_CM_AttrOriginAppID" <value>
                "GSW_CM_AttrTargetAppID" <value>
                "GSW_CM_AttrProperties"
                        |
                        "cancel_record"
                                |
                                "GSW_PHONE" <value>
                                ["GSW_CAMPAIGN_NAME" <value>]
                                ["GSW_CHAIN_ATTR" <value>]
```

The values can be of two types: String or Integer.

See Table 97, “User Event Attributes for User Data (TKVList),” on page 101 and Table 7, “Reserved Keys,” on page 23.

UserEvent Attributes

The UserEvent attributes in Table 106 pertain to the Record Cancel feature. Tables 107 and 108 provide information on GSW_CM_AttrProperties and cancel_record. Also see Table 97 on page 101.

Table 106: UserData (TKVList)

Key	Type	Description
GSW_CM_MessageType	Integer	See GSW_CM_MessageType Enum below.
GSW_CM_AttrError	Integer	0 if no error. See GSW_CM_AttrError Enum below.
GSW_CM_AttrOriginAppID	Integer	Sender's DBID
GSW_CM_AttrTargetAppID	Integer	Receiver's DBID
GSW_CM_AttrProperties	TKVList	See GSW_CM_MessageType Enum below.

Table 107: GSW_CM_AttrProperties (TKVList)

Key	Type	Description
cancel_record	TKVList	Contains additional request attributes

Table 108: cancel_record (TKVList)

Key	Type	Description
GSW_PHONE	String	Phone Number
GSW_CAMPAIGN_NAME	String	Campaign Name If specified, only records in this campaign will be canceled.
GSW_CHAIN_ATTR	String	AllChain, RecordOnly Specifies the scope of the request. AllChain is the default value

Data Enums

GSW_CM_MessageType

These data enumerations apply to the GSW_CM_MessageType for the Record Cancel feature. Tables 109 and 110 provide information on data enumerations and GSW_CMAttrError respectively. Also, see Table 101 on page 103.

Table 109: Data Enumerations

Message	Value	Description
Requests		
CM_ReqCancelRecord	30	Request to cancel records by phone.
Responses		
CM_EvRecordCanceled	31	Acknowledgement for request CM_ReqCancelRecord
CMEvError	27	An error occurred. See error codes below.

GSW_CM_AttrError

Table 110: Errors

Error	Value Type	Description
CM_ERROR_CAMPAIGN_NOT_FOUND	3	Campaign was not loaded.
CM_ERROR_INVALID_PARAMETER	10	Some parameters are invalid.

DoNotCall Requests from a Third-Party Application

DoNotCall (DNC) requests restrict the dialing of particular phone numbers or to particular customers. A field in the Calling List table, as specified by the value of the customer_id option, serves as the customer ID.

On startup, OCS reads all the records from the table referenced in the gsw_donotcall_list Table Access Point and populates separate tables in memory with the unique values from the phone and customer_id fields. DoNotCall requests from the desktop can also populate those tables.

Outbound Contact supports the submission of DNC requests from third-party applications, for example, from the desktop application of an agent handling inbound calls. OCS enables this functionality through an extension of the CommDN API. Recall that to use the API, a custom application must have access to a Genesys T-Server and Configuration Server, both of which have an open API.

The communication is performed by means of `UserEvents` sent and received on a Communication DN. All the data is sent as `UserData` attached to the event. The data is encoded in a key-value pairs list (TKVList). The values can be of two types: string or integer.

The communication between OCS and third-party applications is facilitated by a request-response system.

DNC Messages

The communication by means of T-Server events is based on request-response. They are as follows:

- Request: CM_ReqDoNotCall

  Request to add a phone number or customer ID to DoNotCall (DNC) list.
- Response: CM_EvDoNotCallProcessed

  Acknowledgement of request CM_ReqDoNotCall
- Error message: CM_EvError

  Error message sent if the request has incorrect user data.

Mandatory Attributes

The mandatory attributes of DNC messages include:

- `Phone` or `CustomerID`
- `OriginAppDBID`
- `TargetAppDBID:`

  For `CM_ReqDoNotCall,` the value of `TargetAppDBID` may be 0, which signifies that all the OCS servers monitoring the communication DN will process this request and submit a response.

UserEvent Structure

The following depicts the event structure for T-Server to convey a DNC request (CM_ReqDoNotCall) from a third-party application:

```
UserEvent
        I
        UserData
                I
                "GSW_CM_MessageType" 32
                ["GSW_CM_AttrError" 0]
```

```
"GSW_CM_AttrOriginAppID" <value=sender's ID>
"GSW_CM_AttrTargetAppID" <value=receiver's ID>
"GSW_CM_AttrProperties"
        |
        "do_not_call"
                |
                "GSW_PHONE" <value>
                ["GSW_CUSTOMER_ID" <value>]
                ["GSW_CHAIN_ATTR" <value>]
```

In this example, under `UserData`, the value of `GSW_CM_MessageType` is 32 for the request CM_ReqDoNotCall. The value would be 33 for the response/notification CM_EvDoNotCallProcessed or 27 for the error message CM_EvError, and “do_not_call” under `GSW_CM_AttrProperties` would be replaced accordingly by the proper message types.

Note: The `GSW_CUSTOMER_ID` attribute identifies the customer. The value of `GSW_CUSTOMER_ID` is a field in the Calling List table as specified by the option customer_id. At least one of these attributes— `GSW_CUSTOMER_ID` or `GSW_PHONE`—must be present.

Chapter

3 Defined Constants

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

Field Definitions and Enumeration Values

Some Genesys mandatory fields in a Calling List table are represented as predefined integer constants, called enumeration values. (See Table 8, “Enumeration Table,” on page 28.) The actual enumeration values are provided in the `cfg_locale` table in the Configuration Database.

This chapter provides field definitions. For the enumeration values, see Chapter 1 on page 15.

A calling list must contain Genesys mandatory fields and may also contain user-defined fields.

Table 111 provides a description of the Genesys mandatory fields in the Default Outbound Contact format.

Table 111: Outbound Contact Mandatory Fields in the Default Format

Column Name	Data Type	Description
record_id	integer	Unique identification number of a calling record.
contact_info	varchar(128)	Customer's contact information, phone number in the voice campaign.
contact_info_type	integer	Type of contact information, phone type in the voice campaign. See Table 116, "Contact Information Types," on page 122.
record_type	integer	Type of the record. See Table 117, "Record Types," on page 123.
record_status	integer	Current status of the record. See Table 118, "Record Statuses," on page 124.
call_result	integer	Final outcome of the record processing. See Table 112, "Call Result Types," on page 115.
attempt	integer	Number of attempts made to reach the customer.
dial_sched_time	integer	Date and time for which the processing of the record has been scheduled or rescheduled, in UTC format (seconds since midnight 01/01/1970).
call_time	integer	Latest date and time the record has been processed (dialed), in UTC format.
daily_from	integer	Earliest time of the day when a customer can be contacted (seconds since midnight).
daily_till	integer	Latest time of the day when a customer can be contacted (seconds since midnight).
tz_dbid	integer	Configuration DBID of the time zone object associated with the calling record.
campaign_id	integer	Configuration DBID of the Outbound Dialing Campaign as a part of which the record has been processed.
agent_id	varchar(32)	Login identifier of the agent who handled the record.
chain_id	integer	Unique identification number of the chain the record belongs to.
app_id	integer	Empty, not used at this time.
chain_n	integer	Unique identification number of the record within the chain.
email_subject	varchar(255)	Empty, not used at this time.
email_template_id	integer	Empty, not used at this time.

Table 111: Outbound Contact Mandatory Fields in the Default Format (Continued)

Column Name	Data Type	Description
`group_id`	integer	Empty, not used at this time.
`media_ref`	integer	Empty, not used at this time.
`switch_id`	integer	DBID of the Switch where the agent who handled the record had logged in.
`treatments`	varchar(255)	Treatments application history. For more information, see the “Treatments” chapter in the *Outbound Contact 7.6 Deployment Guide*.

Call Results

The final outcome of a call is stored in the calling list from which the call is dialed. You can view call results for records in calling lists in Outbound Contact Manager. Call results can also be selected for determining treatments. For example, if a busy signal is reached, then the record can be marked for redialing at a later time. Call results for determining treatments are selected in Outbound Contact Wizard or from the Call Result drop-down menu on the Treatment Properties window in Configuration Manager.

Note: `Unknown Call Result` is the default value. All records should be set to `Unknown Call Result` before starting a campaign.

Table 112 shows the call result types.

Table 112: Call Result Types

Call Result	Enumeration Value	Description
Abandoned	21	Call dropped by the customer. In ASM mode there is no Abandoned value. Due to the special connection required for ASM mode, a call is dropped before the customer can abandon the call.
Agent CallBack Error	47	OCS generates this call result when a call record is rescheduled according to a “personal callback” request from the desktop application, but, when the scheduled time arrives, OCS either cannot find the designated agent to receive the callback or the connection to Stat Server has been lost.
All Trunks Busy	10	No trunk is currently available to place the call.

Table 112: Call Result Types (Continued)

Call Result	Enumeration Value	Description
Answer	33	Customer was reached at the dialed phone number.
Answering Machine Detected	9	Answering machine was detected at the dialed phone number.
Bridged	31	Reserved for future use.
Busy	6	Dialed phone number was busy.
Call Drop Error	42	Error occurred while dropping the call.
Cancel Record	52	Record has been canceled.
Cleared	19	Reserved for future use.
Conferenced	2	Reserved for future use.
Consult	24	Reserved for future use.
Converse-On	30	Reserved for future use.
Covered	29	Reserved for future use.
Deafened	49	Reserved for future use.
Dial Error	41	Hardware error from a Dialogic board or from a call progress detection (CPD) board on the switch.
Do Not Call	51	Record has been marked as `Do Not Call`.
Dropped	26	Call was dropped by the dialer after being dialed. Call is dropped if the following timers expire: • `call_wait_in_queue_timeout` • `call_wait_original_establish_timeout`
Dropped on No Answer	27	Call has been dropped or released from an established three-way call before being answered.
Fax Detected	17	Fax machine was detected at the dialed phone number.
Forwarded	23	Reserved for future use.
General Error	3	General error occurs when a call is not completed, possibly caused by an invalid phone number in the record or a wrong number according to the switch.

Table 112: Call Result Types (Continued)

Call Result	Enumeration Value	Description
Group CallBack Error	48	Generated by OCS internally when a call record is rescheduled according to a “Campaign Callback” request from the desktop application; however, OCS cannot find an available agent to receive the callback record.
Held	50	Reserved for future use.
No Answer	7	Ring without answer at destination.
No Dial Tone	35	Absence of dial tone based on an error returned by the Dialogic board or the call progress detection board on the switch.
No Established Detected	38	Reserved for future use.
No Port Available	44	No port is available to place the call.
No Progress	36	Reserved for instances where the call progress detection either did not start or has been terminated due to a Dialogic hardware or CPD Server configuration error.
No RingBack Tone	37	There is no ringback tone on the called line.
NU Tone	34	A special Public Switched Telephone Network (PSTN) code valid only in Europe.
OK	0	Call result is unset; that is, the call record has not been dialed.
Overflowed	20	Reserved for future use
Pager Detected	39	Pager was reached at the dialed phone number.
Pickedup	25	Reserved for future use.
Queue Full	18	Reserved for future use.
Redirected	22	Reserved for future use.
RemoteRelease	5	Call was released in response to an error on the switch or no contact with called party.
Silence	32	Call was dialed; however, there is no call progress indication.
SIT Detected	8	Any type of network tone.

Table 112: Call Result Types (Continued)

Call Result	Enumeration Value	Description
SIT IC (Intercept)	13	Applies only if the network supports this specific SIT tone; check with the switch vendor for confirmation.
SIT Invalid Num	11	Applies only if the network supports this specific SIT tone; check with the switch vendor for confirmation.
SIT NC (No Circuit)	15	Applies only if the network supports this specific SIT tone; check with the switch vendor for confirmation.
SIT RO (Reorder)	16	Applies only if the network supports this specific SIT tone; check with the switch vendor for confirmation.
SIT Unknown Call State	14	Applies only if the network supports this specific SIT tone; check with the switch vendor for confirmation.
SIT VC (Vacant Code)	12	Applies only if the network supports this specific SIT tone; check with the switch vendor for confirmation.
Stale	46	Call result is marked as `Stale` in the following scenario: 1. The following timer has expired: `stale_clean_timeout` 2. OCS marks the call result as `Stale` if an outbound call was transferred from: • a queue to a DN which is either not registered for OCS or does not have a logged in agent. • an agent in an outbound campaign to a DN which is not registered for OCS.
Switch Error	43	No dial tone received.
System Error	4	Dialing software error from the Dialogic driver or call progress detection (CPD) from the switch.
Transfer Error	45	Dialer has a problem transferring calls based on call action.
Transferred	1	Reserved for future use.

Table 112: Call Result Types (Continued)

Call Result	Enumeration Value	Description
Unknown Call Result	28	Default. All records should be set to this call result before starting a campaign.
Wrong Number	53	Intended person cannot be reached at this number. This call result is sent by the desktop application and is not detected by the dialer.
Wrong Party	40	Call is answered but by a wrong party; this call result is sent by the desktop application and not detected by the dialer.

Note: Of those call results listed in Table 112, OCS can receive from dialing engines (for example, T-Server and CPD Server) only those that have a description, unless otherwise noted.

However, Agent Desktop can use any call result (for example, any reserved call result or call result enumeration value that is not identified in Table 112) in its communication to OCS. For example, if Agent Desktop delivers `UpdateCallCompleteStats` or `RecordProcessed` to OCS with a call result equal to `24 "Consult"` or `999,` OCS properly stores this result in the calling list and sends it to the reporting engines.

Be aware that you can apply treatments only to those call results listed in Table 112.

Call Result Mapping by OCS

In Outbound Contact, Outbound Contact Server performs mappings of the call progress and call status. When Outbound Contact Server receives call progress/call status reports from CPD Server, OCS maps them into a single Call Result. This Call Result is then stored in the Calling List Record and kept in a record history log.

Table 113 shows examples of Call Result Mappings created by OCS.

Table 113: Call Result Mappings Created by OCS

Call Status	CPD Call Result Value	Database Result
A voice, answering machine, or FAX machine answers the call, but the EventEstablished does not arrive on time. Call is released.	IAttr_CallStatus Call has been released Attr_CallProgress TransferError	Transfer Error
A voice answers the call. The call transfer starts but does not complete on time due to expiration of call_wait_in_queue timeout or another calling-party related error during the transfer. Call is dropped.	IAttr_CallStatus Call has been dropped Attr_CallProgress Answer	Dropped
A voice answers the call. The call transfer starts but does not complete on time due to called party disconnection or a called-party related error during the transfer. Call is abandoned.	IAttr_CallStatus Call has been abandoned Attr_CallProgress Answer	Abandoned
Answering machine or FAX machine answers the call. The call transfer starts but does not complete on time due to expiration of call_wait_in_queue timeout or another error during the transfer. Call is released.	IAttr_CallStatus Call has been released Attr_CallProgress AnsweringMachine/Fax	Answering Machine/FAX
Answering machine or FAX machine answers the call. Call transfer is not required. Call is released.	IAttr_CallStatus Call has been released Attr_CallProgress AnsweringMachine/Fax	Answering Machine/FAX

Table 114 contains call results that CPD Server considers to be conflicting only. They can be controlled by the `pre-connect-cpd-priority` and `post-connect-cpd-priority` CPD Server options. Refer to the *Outbound Contact 7.6 Deployment Guide* for more information about these options.

Table 114: Conflicting Call Result Mappings

Dialogic Result	T-Server Result	Possible Cause
NU Tone	EventDestinationBusy with CallState AllTrunksBusy	The wrong tone was sent by the switch or there was a detection error.
NU Tone	EventDestinationBusy with CallState Busy	The wrong tone was sent by the switch or there was a detection error.
NU Tone	EventEstablished/Answer	No real answer supervision on the last leg of the call.

Table 114: Conflicting Call Result Mappings (Continued)

Dialogic Result	T-Server Result	Possible Cause
Operator Intercept/ SIT	EventDestinationBusy with CallState AllTrunksBusy	The wrong tone was sent by the switch or there was a detection error.
Operator Intercept/ SIT	EventDestinationBusy with CallState Busy	The wrong tone was sent by the switch or there was a detection error.
Operator Intercept/ SIT	EventEstablished/Answer	No real answer supervision on the last leg of the call.
No Answer	EventEstablished/Answer	No real answer supervision on the last leg of the call.

Data Types

Data types determine the type of data that can be stored in a field. See Table 115 for data type values.

Table 115: Data Type Values

Data Type	Description
char	Character string with a fixed length
datetime	Date and time
float	Real number
int	Integer
varchar	Character string of variable length
[Unknown Data Type]	Default

Contact Information Types

Contact information types pertain to contact information. For descriptions, see Table 116.

Table 116: Contact Information Types

Phone Type	Description
Business With Extension	Office number with extension
Direct Business Phone	Office number
E-mail Address	E-mail address
Home Phone	Household number
Mobile	Mobile number
Modem	Modem number
No Contact Type	No phone type selected
Pager	Pager number
Pin Pager	Pager number with PIN
Vacation Phone	Vacation number
Voice Mail	Voice mail number

Note: For Enumeration (Enum) values that correspond to the defined constants in this table, see Table 8 on page 28.

Record Types

Record types show if a record is to be dialed, not dialed, or scheduled. See Table 117.

Table 117: Record Types

Record Type	Description
Campaign CallBack	Used by the agent to reschedule a call and have the call delivered to any agent in the campaign group when the callback occurs. See also the `predictive_callback` option.
Campaign Rescheduled	Default value for a record that has been rescheduled by a call treatment.
General	Default for dialing records.
No Call	Used to show that the record will not be called.
No Record Type	No record type selected.
Personal CallBack	Used by the agent to reschedule a call and have the call delivered to himself when the callback occurs.
Personal Rescheduled	Used by the agent to reschedule a call treatment and allow only himself to take the call when the callback occurs.
Unknown Record Type	Record type is unrecognized.

Note: For Enumeration (Enum) values that correspond to the defined constants in this table, see Table 8 on page 28.

Record Statuses

A record status shows the latest status of a record. Record statuses can be viewed for a calling list in Outbound Contact Manager. See Table 118.

Table 118: Record Statuses

Record Status	Description
Agent Error	An outbound call was distributed to a DN not monitored by OCS. OCS is not registered on that DN.
Canceled	A record has been canceled by Agent request or through the Communication DN Protocol.
Chain Ready	Reserved for future use
Chain Updated	Reserved for future use
Missed CallBack	Personal CallBack or Campaign CallBack has been missed and treatment action “Mark As Agent Error” was applied to record. Also see the `predictive_callback` option.
No Record Status	Record status is not set; that is, the call record is not ready to be dialed.
Ready	Default value; The record is ready to be dialed.
Retrieved	A record is retrieved from the database and is in the active dialing queue.

Table 118: Record Statuses (Continued)

Record Status	Description
Stale	Outbound Contact has not received acknowledgment of any user event sent to the agent's desktop application regarding this record. For all records that are considered stale (no longer useful) due to a desktop crash, and for all records that remain on the desktop without being updated in the database: OCS changes the `record_status` to `Stale` when a campaign is unloaded. The record status is marked as stale in the following scenarios: 1. If a record was dialed and not updated in the database as a result of the call progress or request from a desktop, after campaign `ForceUnload` request the record status is updated as `Stale`. 2. If a request to update a record was sent from OCS to DB Server but `DBError` was received as the response, after campaign `Unload` or `ForceUnload` request the record status is updated as `Stale`. 3. If a request to update a record was sent from OCS to DB Server but the response from it was not received, after campaign `Unload` or `ForceUnload` request the record status is updated as `Stale`.
Updated	Shows the record is updated in the database and will not be dialed again.

Note: For Enumeration (Enum) values that correspond to the defined constants in this table, see Table 8 on page 28.

Combining Record Statuses and Record Types

Record statuses and record types can be combined as shown in Table 119.

Table 119: Record Status and Record Type Combinations

Record Status	Record Type
Ready	General Campaign Rescheduled Personal Rescheduled Personal CallBack Campaign CallBack
Retrieved	General Campaign Rescheduled Personal Rescheduled Personal CallBack Campaign CallBack
Updated	General Campaign Rescheduled Personal Rescheduled Personal CallBack Campaign CallBack No Call
Stale	General Campaign Rescheduled Personal Rescheduled Personal CallBack Campaign CallBack
Canceled	General Campaign Rescheduled Personal Rescheduled Personal CallBack Campaign CallBack

Table 119: Record Status and Record Type Combinations (Continued)

Record Status	Record Type
Agent Error	General Campaign Rescheduled Personal Rescheduled Personal CallBack Campaign CallBack
Missed CallBack	Personal Rescheduled Personal CallBack Campaign CallBack

Note: For Enumeration (Enum) values that correspond to the defined constants in this table, see Table 8 on page 28.

Chapter

4 Recommended DBMS Optimizations

This chapter explains how to perform recommended DBMS optimizations. It includes these sections:

Optimizing Calling List Tables for Weight Rules

If you use a Sybase Adaptive Server as your DBMS and more than one calling list is using the same database table — that is, the calling lists refer to the same Table Access object in the configuration — Genesys recommends that you modify the default locking scheme on the database table in order for OCS to correctly maintain the weight rules for these calling lists.

Follow these procedures:

1. Stop and unload all campaigns that include calling lists referencing this database table.
2. Execute this SQL statement:

```
alter table <table_name> lock datarows
```

Note: If the table contains several thousand calling records, the above SQL statement might take some time to execute.

Temporary Tables Considerations

In the `Calling List Details` view, to enable the user to navigate in a timely manner through calling list tables with a large number of customer contacts (3 to 4 thousand or more records in the calling list table), OCM uses an auxiliary temporary table that is generated when the calling list is opened.

The suggestions below reference the different DBMS types that use temporary tables (Microsoft SQL, Sybase, Informix, and DB2).

Microsoft SQL Server

Microsoft SQL Server 7.0 and above: Temporary tables are stored in the `tempdb` database. On Microsoft SQL (MS SQL) Server 7.0 and higher, the size of tempdb is automatically enlarged by default if required.

Microsoft SQL Server 6.5 and below: Microsoft SQL Server 6.5 requires manual adjustment of the size of this database. When a large calling list is opened, the size of tempdb may become insufficient; in that case an `Out of space in tempdb` error message will be returned by the DBMS and displays on OCM's GUI. To resolve this problem and to enable OCM to process large calling lists, increase the size of the tempdb database.

Sybase

The same considerations regarding the size of tempdb as described above for Microsoft SQL Server 6.5 also apply to Sybase.

For additional details, please refer to your Sybase system documentation.

Informix

Operations on large calling lists will require sufficient space in the dbspace where temporary tables reside. You may need to increase this size. The dbspace where temporary tables are stored is defined by the `"DBSPACETEMP" ONCONFIG` parameter.

For additional details, please refer to your Informix system documentation.

DB2

Operations on calling lists require a temporary table. The DB2 engine stores this temporary table in User Temporary table space, which the database administrator (DBA) should explicitly create.

If this table space is absent, an attempt to open a calling list in OCM results in the following error message: `DBServer returned SQL error [IBM][CLI Driver][...] SQL0286N. A default table space could not be found with a page size of at least <pagesize> that authorization ID <user-name> is authorized to use.`

To resolve this problem, ensure that a table space of the correct type (User Temporary) with a page size of at least `<pagesize>` exists, and that the authorization ID `<user-name>` has `USE` privilege on this table space.

Oracle

Oracle 8.1 and below: On versions of Oracle before 8.1.7, Oracle does not use temporary tables, and OCM performs sorting in its memory when an ORDER BY clause is issued.

Calling lists containing several thousand records may require additional processing time, which may cause DB Server to force OCM to disconnect if OCM is too slow to respond. In this scenario, Genesys suggests that you add more conditions to the filter to limit the number of retrieved records (less than 2-3 thousand), or that you increase the value of the `db_timeout` option.

Maintaining Indexes for Large Calling Lists

When working with large calling lists consisting of several thousand or more records, the performance of the DBMS for outbound dialing with OCS might be affected. Follow these guidelines to improve the performance of the DMBS in this scenario.

Creating Indexes

If a dialing filter is used, Genesys recommends that you create indexes for all fields used in the dialing filter into `WHERE` and `ORDER BY` clauses in the filter. Create indexes using the available DBMS tools or using the following SQL statements:

```
CREATE INDEX <unique__index_name> ON <calling_list_table_name>
(<field_used_in_filter>)
```

Updating Indexes

If the data in a calling list table significantly changes, earlier indexes might become out of date. For example, if you create a calling list table and manually insert 10 customer contacts, then import into this table an additional 10,000 customer contacts, the indexes built on the table will become out of date.

To avoid possible performance issues, before starting a campaign, update indexes in a table that has been changed significantly. To update indexes, you should recalculate statistics in the database.

Below are recommendations for recalculating statistics for the calling list table. To recalculate statistics, follow the recommendations for your DBMS type.

Note: Statistics are updated automatically when you create or recreate an index for an existing table.

Microsoft SQL Server

Microsoft SQL Server versions 7.0 and higher automatically recalculate statistics when a table is significantly changed. The `auto update statistics` option (default = `ON`) controls recalculation.

If you are using an earlier version of SQL or if automatic statistic recalculation is switched off, you can manually recalculate statistics by executing the following statement:

```
UPDATE STATISTICS <calling_list_table_name>
```

This operation might be time-consuming depending on the size of the table.

For additional details, please refer to Microsoft SQL Sever product documentation or your Database Administrator.

Sybase

Use the following statement to manually update statistics for a calling list table:

```
UPDATE STATISTICS <calling_list_table_name>
```

This operation might be time-consuming depending upon the size of the table.

For additional details, please refer to your Sybase system documentation.

Oracle

To manually recalculate statistics for a calling list table that has been significantly changed, use the following statement:

```
ANALYZE TABLE <calling_list_table_name> COMPUTE STATISTICS
```

This operation might be time-consuming depending upon the size of the table. You can also use procedures from the DBMS_STATS package to recalculate statistics.

For additional details, please refer to your Oracle documentation or consult your Database Administrator.

Informix

Use the following statement to manually update statistics for a calling list table:

```
UPDATE STATISTICS HIGH FOR TABLE <calling_list_table_name>
```

This operation might be time-consuming depending upon the size of the table.

For additional details, please refer to your Informix documentation or consult your Database Administrator.

Chapter

5 Supported Functionality with IP Telephony

This chapter describes the IP telephony functionality that Outbound Contact supports. It contains the following sections:

Overview

This chapter describes Outbound Contact dialing scenarios that include SIP Server and Genesys Voice Platform (GVP).

- Outbound Contact Server (OCS) and SIP Server provide support for audio or audio/video outbound campaigns in both `Predictive` and `Progressive` dialing modes, in the following scenarios:
  - Using a SIP Server with a Media Gateway (MGW) that is able to perform Call Progress Analysis (CPA).
  - Using a Dialogic board in `Transfer` mode.
  - Using a Dialogic board in `ASM` (Active Switching Matrix) mode.
  - Using Dialogic's Host Media processing (HMP) software.
- OCS and T-Server for Cisco CallManager provide support for audio outbound campaigns in both `Predictive` and `Progressive` dialing modes for the following scenarios:
  - Using a Dialogic board in `ASM` (Active Switching Matrix) mode only.
  - Using HMP in ASM or Transfer modes.

- OCS and GVP provide support for self service campaigns that may or may not involve an agent. Call Progress Analysis (CPA) is provided by GVP. Only `Power GVP` mode is supported in this scenario.

Notes: Refer to the *Genesys 7 Supported Media Interfaces* guide for more information about supported media gateways.

Please consult with your media gateways provider regarding CPA availability, and for configuration information.

When using a Dialogic board, Genesys recommends that you use `Transfer` mode, which provides the most efficient usage of Dialogic resources. Contact your Dialogic card provider for further configuration information.

Universal Routing Server is not part of the following scenarios, but can be used to route a call to an agent. However, when using HMP, it is a required component.

Figure 8 provides an overview of each of these scenarios.

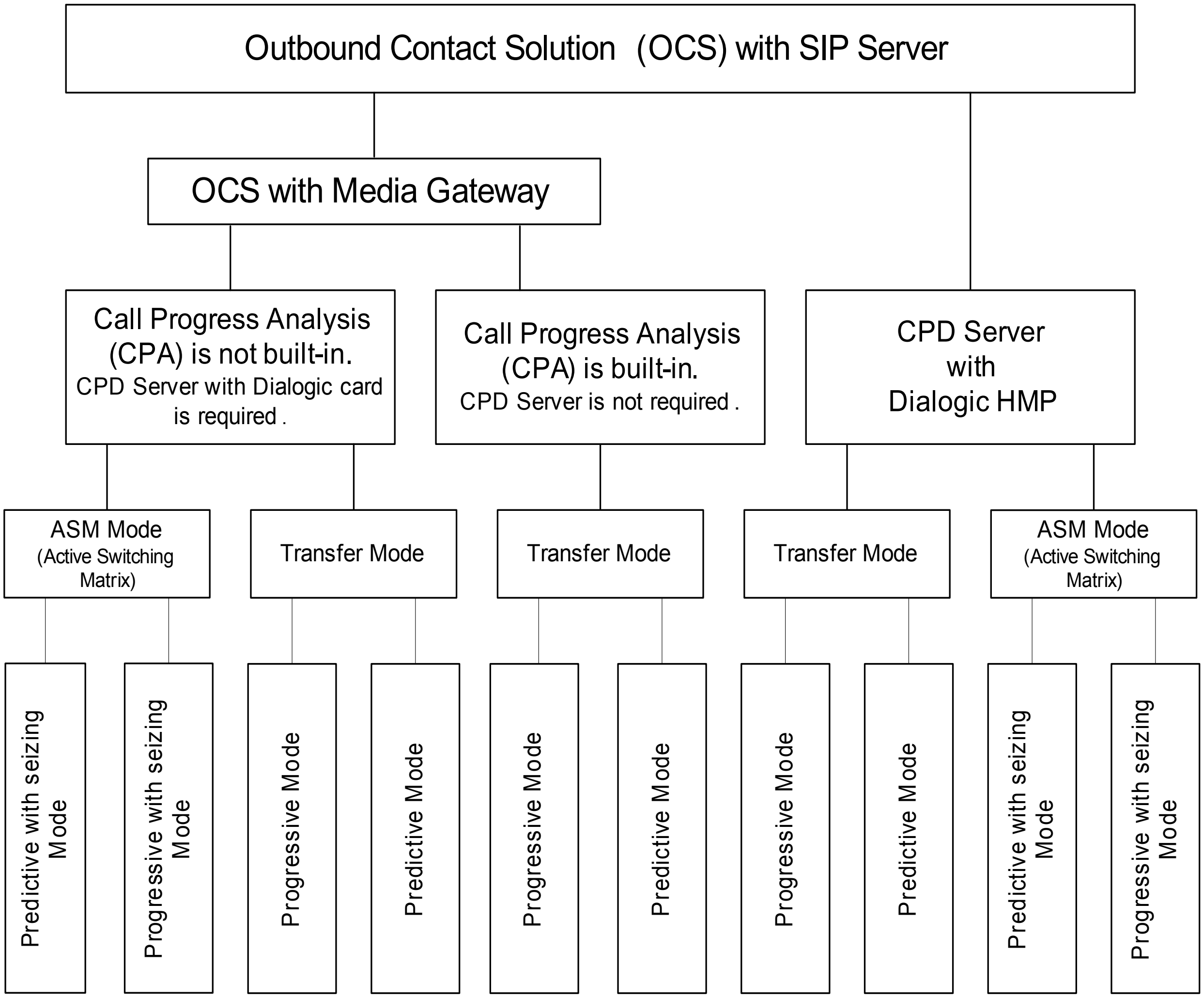
Outbound Contact Solution (OCS) with SIP Server
OCS with Media Gateway
Call Progress Analysis (CPA) is not built-in. CPD Server with Dialogic card is required .
Call Progress Analysis (CPA) is built-in. CPD Server is not required .
CPD Server with Dialogic HMP
ASM Mode (Active Switching Matrix)
Transfer Mode
Transfer Mode
Transfer Mode
ASM Mode (Active Switching Matrix)
Predictive with seizing Mode
Progressive with seizing Mode
Progressive Mode
Predictive Mode
Progressive Mode
Predictive Mode
Progressive Mode
Predictive Mode
Predictive with seizing Mode
Progressive with seizing Mode

Figure 8: Outbound Contact Solution with SIP Server Overview

The following sections describe predictive and progressive dialing mode scenarios for SIP protocols.

Notes: Each scenario describes the media flow, which can contain either audio or audio/video information. The type of media used is identified within the Session Description Protocol (SDP) parameter in SIP Server messages.

Video media flows using HMP is not supported.

HMP Licensing

CPD Server implemented a new HMP licensing schema in 7.6.1. This schema supports licensing changes for Dialogic drivers. The 7.6.1 release of CPD Server requires Dialogic HMP 3.0 for Windows SU174 or higher.

Note: CPD Server 7.6.0 is compatible with HMP SU drivers up to and including version 150.

Outbound Contact with SIP Server

The following scenarios describe how to use Outbound Contact with SIP Server:

- "Transfer Mode (MGW with CPA)" on page 136
- "Transfer Mode (MGW without CPA)" on page 138
- "ASM Mode (MGW without CPA)" on page 141

Transfer Mode (MGW with CPA)

The following scenario describes a media flow that involves a MGW (Media Gateway) with CPA (Call Progress Analysis) capabilities. The following hardware is supported in this scenario:

- AudioCodes
- Paraxip

SIP Protocol with SIP Agent Endpoint

Figure 9 illustrates a `Transfer`-mode call flow that uses the SIP protocol, and that involves an MGW with CPA and a SIP agent endpoint.

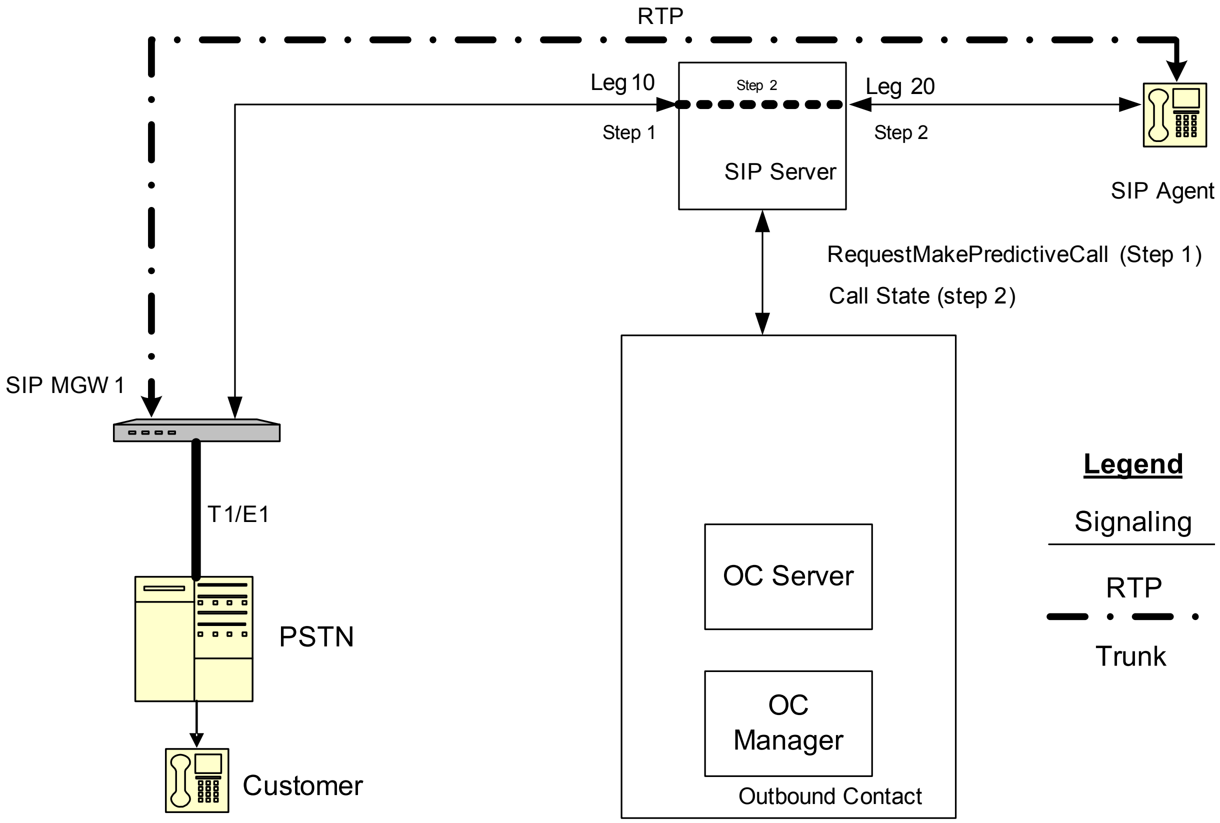

Figure 9: Transfer-Mode Call Flow (MGW with CPA)—SIP Protocol with a SIP Agent Endpoint

In this scenario, the call flow proceeds as follows:

Step 1

1. OCS sends a `RequestMakePredictiveCall` message to SIP Server. This request contains `AttributeOtherDN`, which is the customer’s DN.
2. SIP Server creates call leg 10 with MGW 1 and establishes a call with the customer DN.
3. MGW 1 performs CPA and sends the call results to SIP Server.

Step 2

4. SIP Server reports the call state to OCS.
5. SIP Server generates `EventQueued` and `RouteRequest` messages and establishes call leg 20 with a SIP agent end point.
6. All media streams will be between the SIP agent end point and the customer when SIP Server joins call leg 10 and call leg 20.

Transfer Mode (MGW without CPA)

The following scenario describes a media flow that involves an MGW without CPA abilities.

SIP Protocol with SIP Agent Endpoint

Figure 10 illustrates a `Transfer`-mode call flow in two locations that uses the SIP protocol, and that involves an MGW without CPA and a SIP agent endpoint.

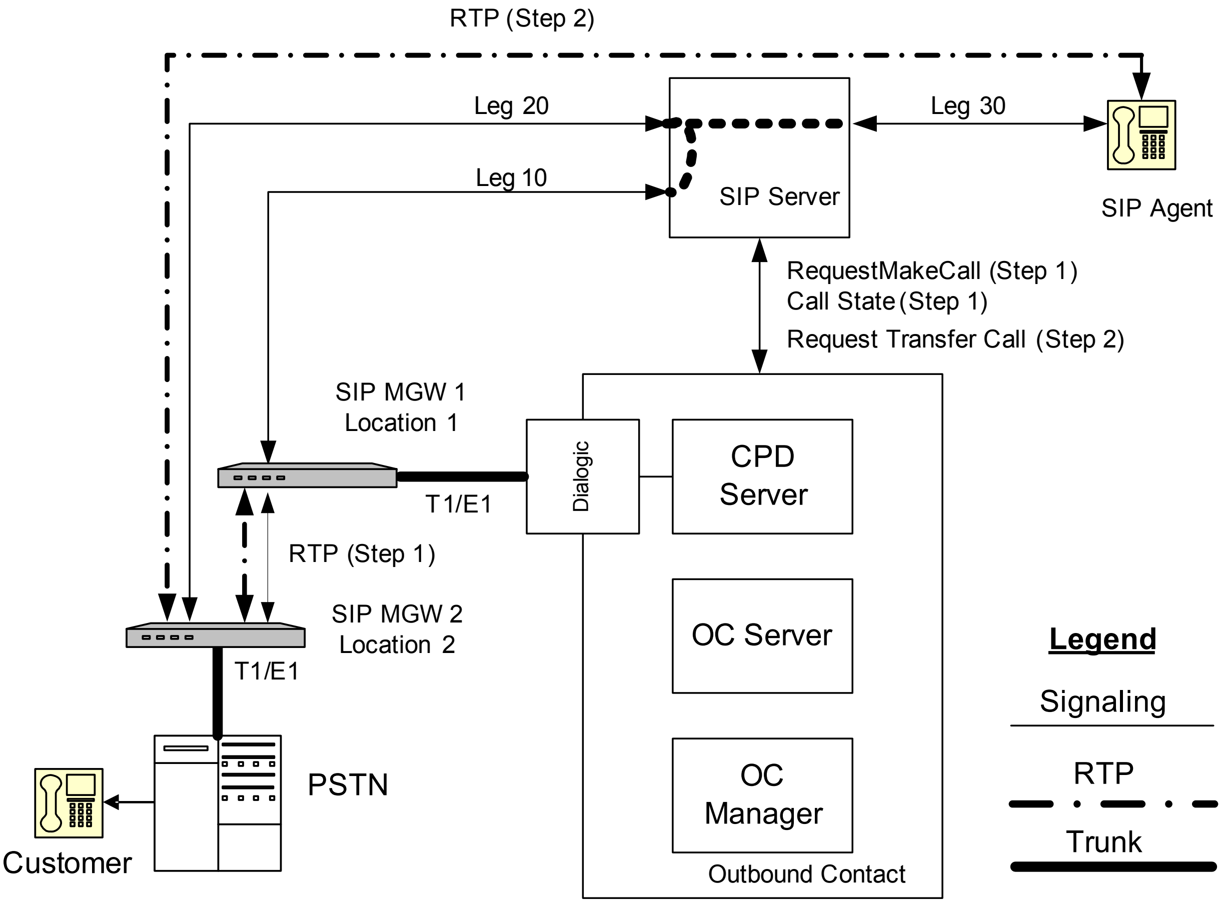

Figure 10: Transfer-Mode Call Flow (MGW without CPA) in Two Locations—SIP Protocol with a SIP Agent Endpoint

In this scenario, the call flow proceeds as follows:

Step 1

1. CPD Server places a Dialogic channel DN off hook.
2. CPD Server sends a `RequestMakeCall` message to SIP Server. This request contains `AttributeThisDN`, which is the Dialogic channel DN on MGW 1, and `AttributeOtherDN`, which is the customer DN.

3. SIP Server creates call leg 10 with the Dialogic channel DN and MGW 1 using `AttributeThisDN` in the `RequestMakeCall` message. CPD Server uses this leg to perform CPA.
4. SIP Server creates call leg 20 through MGW 2 in order to establish a call with the customer DN, using `AttributeOtherDN` in the `RequestMakeCall` message.
5. SIP Server conferences call leg 10 and call leg 20.
6. CPD Server performs CPA and sends the call result to OCS.

Step 2

7. CPD Server sends a request to SIP Server to initiate the transfer of the customer call to a SIP agent end point.
8. SIP Server creates and establishes call leg 30 with the SIP agent end point.
9. SIP Server joins call leg 20 and call leg 30. All media streams are now between the SIP agent end point and the customer.
10. CPD Server places the Dialogic channel DN on-hook when the transfer has been completed, and SIP Server issues the `EventReleased` message. This causes MGW 1 to drop call leg 10.

The Dialogic channel DN is now freed from the MGW to dial another outbound call.

Figure 11 illustrates a `Transfer`-mode call flow in one location that uses the SIP protocol in one location, and that involves a MGW without CPA and a SIP agent end point.

The MGW in this scenario must be able to support multiple T1/E1 lines and provide bridging capabilities.

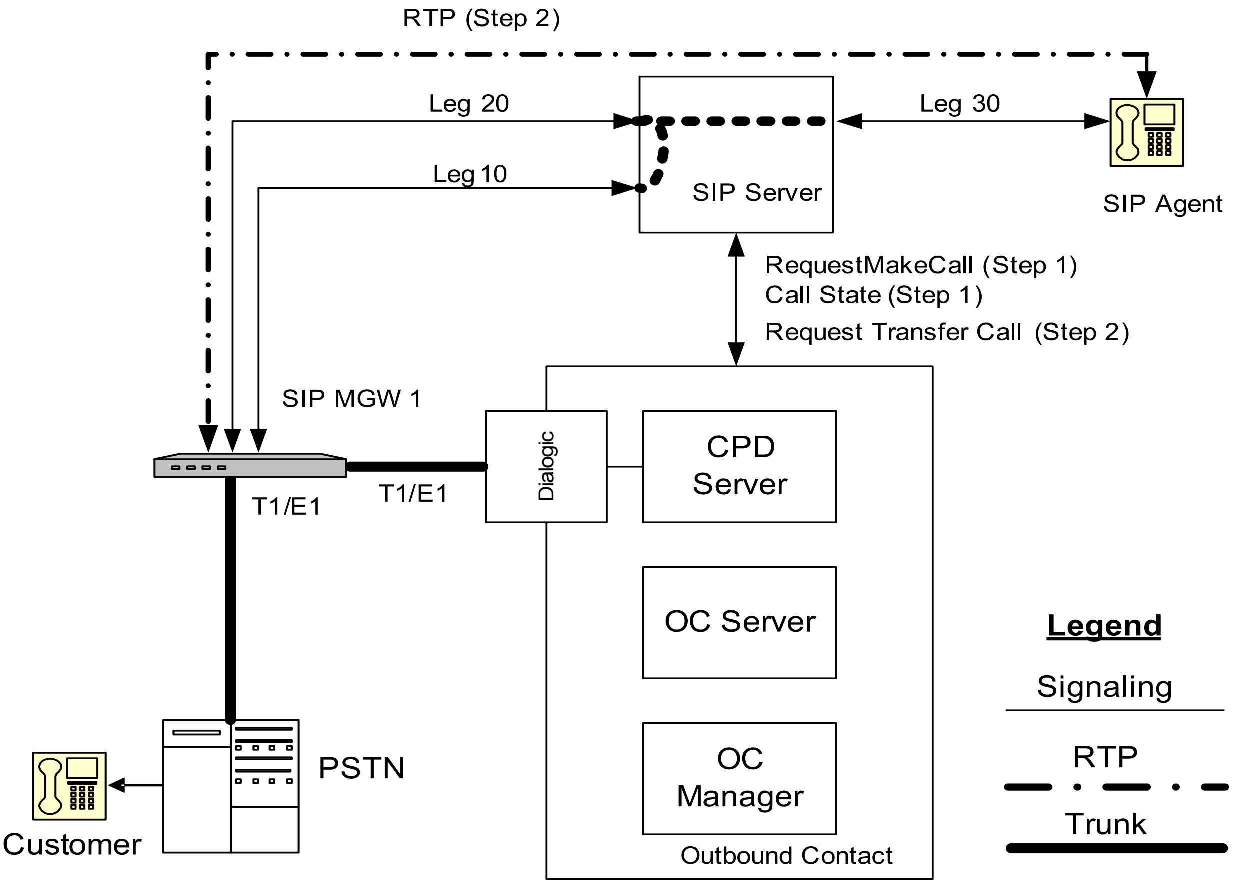

Figure 11: Transfer-Mode Call Flow (MGW without CPA) in One Location—SIP Protocol with a SIP Agent Endpoint

In this scenario, the call flow proceeds as follows:

Step 1

1. CPD Server places a Dialogic channel DN off hook.
2. CPD Server sends a `RequestMakeCall` message to SIP Server. This request contains `AttributeThisDN`, which is the Dialogic channel DN on the MGW, and `AttributeOtherDN`, which is the customer DN.
3. SIP Server creates call leg 10 with the Dialogic channel DN and the MGW using `AttributeThisDN` in the `RequestMakeCall` message. CPD Server uses this leg to perform CPA.
4. SIP Server creates call leg 20 through the MGW in order to establish a call with the customer DN, using `AttributeOtherDN` in the `RequestMakeCall` message.
5. SIP Server conferences call leg 10 and call leg 20.
6. CPD Server performs CPA and sends the call result to OCS.

Step 2

7. CPD Server sends a request to SIP Server to initiate the transfer of the customer call to a SIP agent end point.
8. SIP Server creates and establishes call leg 30 with the SIP agent end point.
9. SIP Server joins call leg 20 and call leg 30. All media streams are now between the SIP agent end point and the customer.
10. CPD Server places the Dialogic channel DN on-hook when the transfer has been completed, and SIP Server issues the `EventReleased` message. This causes the MGW to drop call leg 10.

The Dialogic channel DN is now freed from the MGW to dial another outbound call.

ASM Mode (MGW without CPA)

The following scenario describes a media flow that involves an MGW without CPA abilities, using a Dialogic card in ASM mode.

SIP Protocol with SIP Agent Endpoint

Figure 12 illustrates an `ASM`-mode call flow that uses the SIP protocol in two locations, and that involves a MGW without CPA and a SIP agent endpoint:

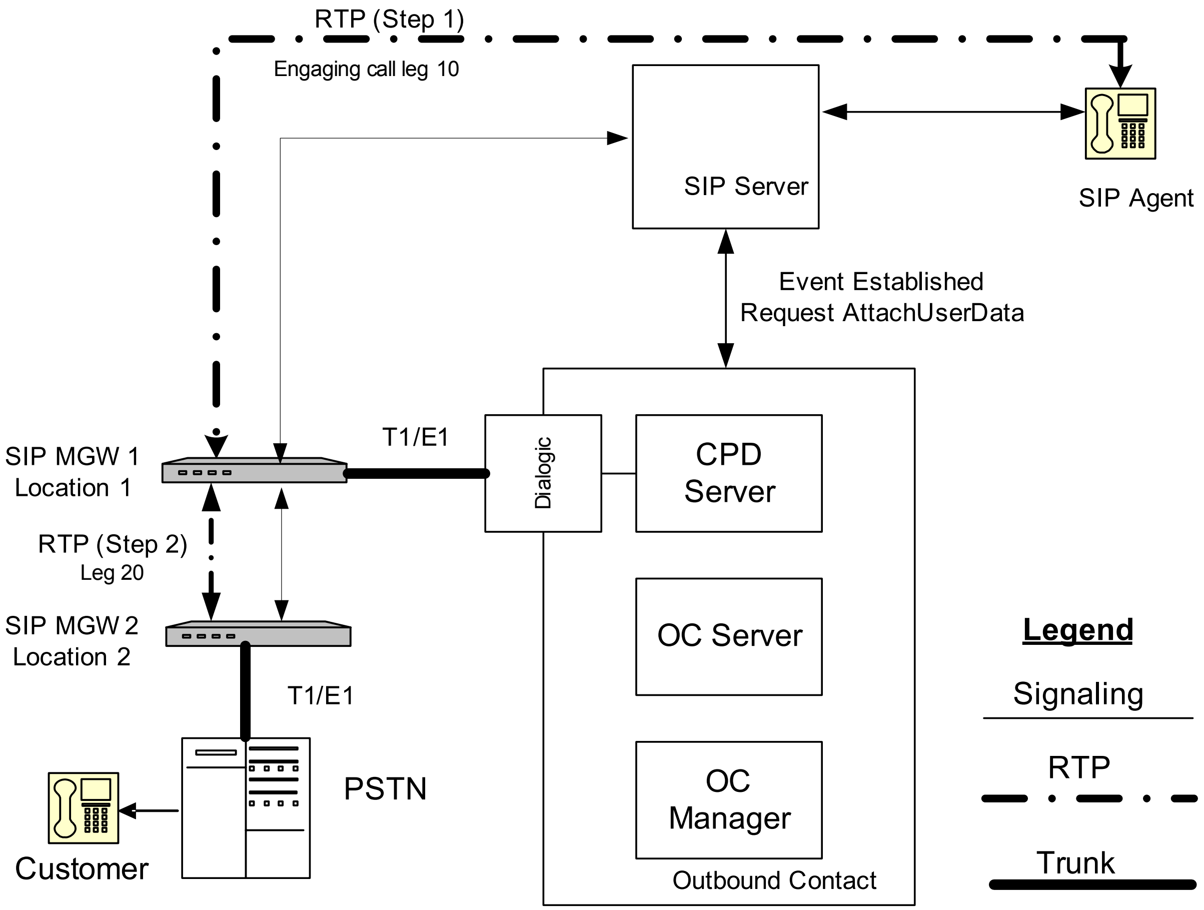

Figure 12: ASM-Mode Call Flow (MGW without CPA) in Two Locations—SIP Protocol with a SIP Agent Endpoint

In this scenario, the call flow proceeds as follows:

Step 1

1. OCS sends an engage agent request to CPD Server.
2. CPD Server instructs the Dialogic board to create an engage call (leg 10) with an available agent's queue.
3. The engage call is queued, which generates an `EventQueued` message.
4. The agent's queue diverts the engage call to an agent's desktop.
5. The agent answers the engage call, which generates an `EventEstablished` message. The agent now waits for OCS and CPD Server to generate a second call (leg 20) to a calling list number.

Step 2

6. CPD Server instructs the Dialogic board to place a call to the customer number that OCS provided.
7. If Call Progress Analysis has determined that there is a live voice, CPD Server attaches any customer data to the engage call (leg 20).
8. SIP Server delivers this data to the engaged agent's desktop as a screen pop.
9. CPD Server connects the call's internal and external leg. The call is established between the agent and the customer.
10. CPD Server informs OCS of the call result. The call is now handled as a normal outbound call.

Figure 13 illustrates an ASM-mode call flow that uses the SIP protocol in one location, and that involves a MGW without CPA and a SIP agent endpoint.

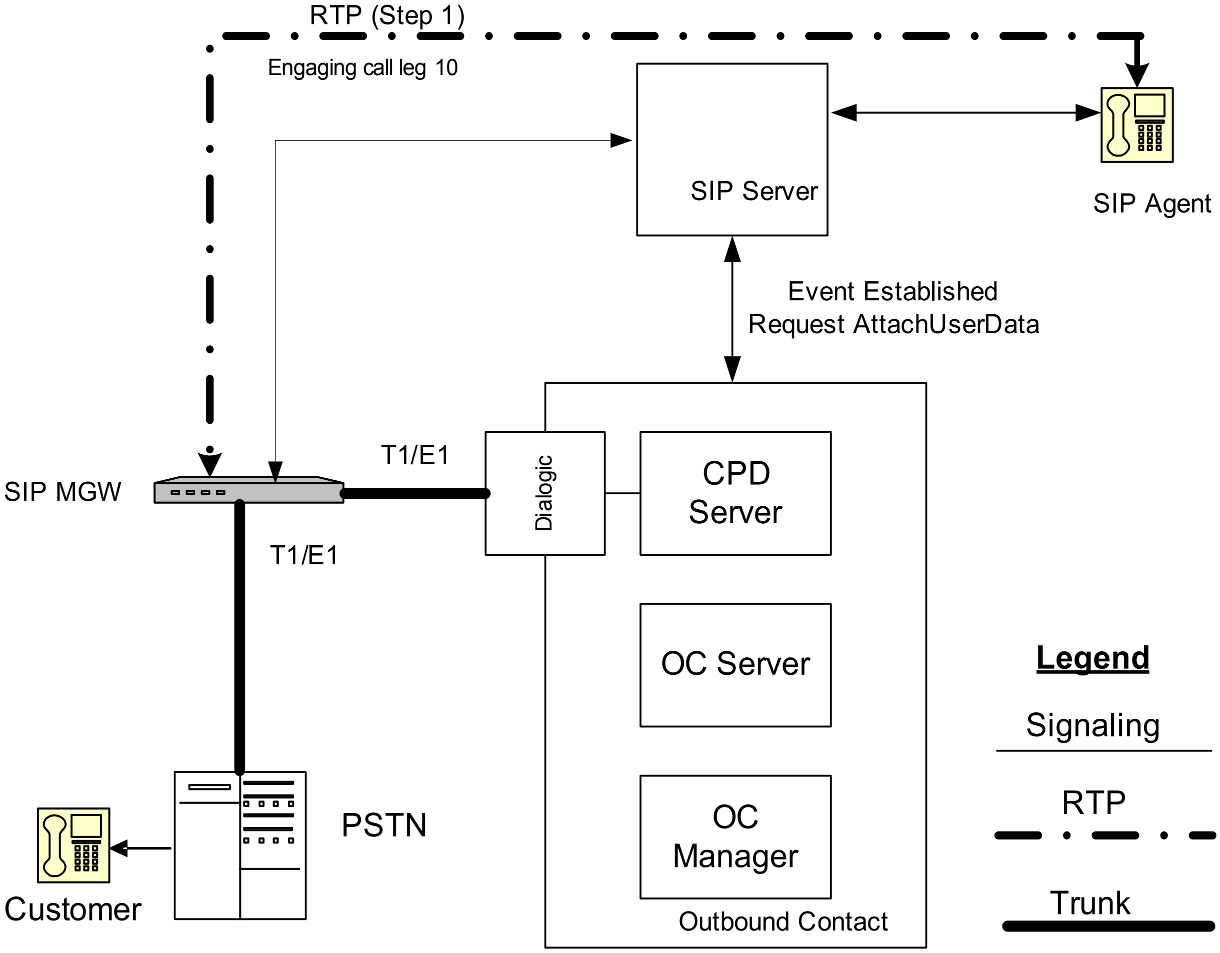

Figure 13: ASM-Mode Call Flow (MGW without CPA) in One Location—SIP Protocol with a SIP Agent Endpoint

In this scenario, the call flow proceeds as follows:

Step 1

1. OCS sends an engage agent request to CPD Server.
2. CPD Server instructs the Dialogic board to create an engage call (leg 10) with an available agent's queue.
3. The engage call is queued, which generates an `EventQueued` message.
4. The agent's queue diverts the engage call to an agent's desktop.
5. The agent answers the engage call, which generates an `EventEstablished` message. The agent now waits for OCS and CPD Server to generate a second call (leg 20) to a calling list number.

Step 2

6. CPD Server instructs the Dialogic board to place a call to the customer number that OCS provided.
7. If Call Progress Analysis has determined that there is a live voice, CPD Server attaches any customer data to the engage call (leg 20).
8. SIP Server delivers this data to the engaged agent's desktop as a screen pop.
9. CPD Server connects the call's internal and external leg. The call is established between the agent and the customer.

CPD Server informs OCS of the call result. The call is now handled as a normal outbound call.

Figure 14 illustrates an ASM-mode call flow that uses the SIP protocol and a SIP agent endpoint. HMP software is used for CPA.

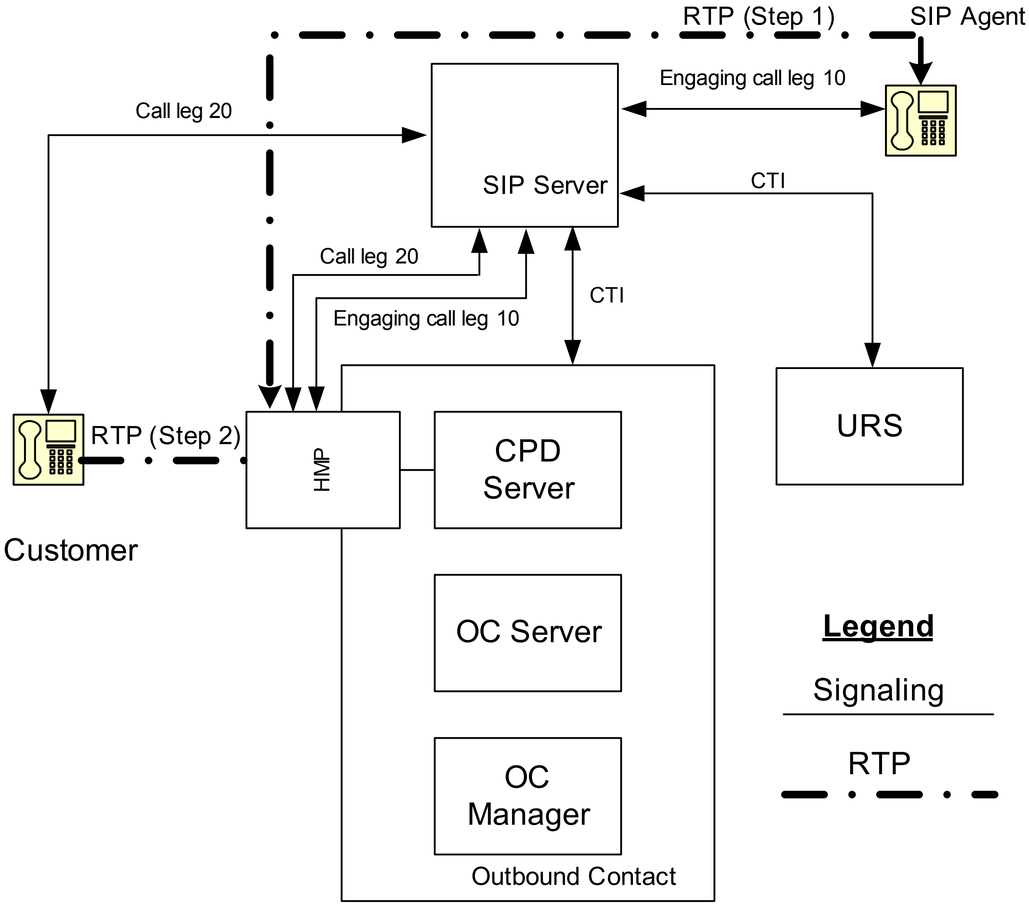

Figure 14: ASM-Mode Call Flow—SIP Protocol with a SIP Agent Endpoint

In this scenario, the call flow proceeds as follows:

Step 1

1. OCS sends an engage agent request to CPD Server.
2. CPD Server instructs HMP to create an engage call (leg 10) through SIP Server with an available agent's route point.
3. The RoutePoint strategy diverts the engage call to an agent's desktop.
4. The agent answers the engage call, which generates an EventEstablished message. An RTP Stream is opened between the Agent's Endpoint and an HMP Voice Channel.

   The agent now waits for OCS and CPD Server to generate a second call (leg 20) to a calling list number.

Step 2

5. CPD Server instructs HMP to place a call through SIP Server to the customer number that OCS provided.
6. If Call Progress Analysis has determined that there is a live voice, CPD Server attaches any customer data to the engage call (leg 10).
7. SIP Server delivers this data to the engaged agent's desktop as a screen pop.
8. HMP connects the Agent (internal) and Customer (external) call legs. RTP is established between the agent and the customer through HMP.

CPD Server informs OCS of the call result. The call is now handled as a normal outbound call.

Outbound Contact with Cisco CallManager

This section describes an ASM Mode scenario and a Transfer scenario using Cisco CallManager T-Server and agents.

ASM Mode

The following scenario describes a media flow for Outbound Contact with HMP in ASM mode and the Cisco CallManager T-Server.

Figure 15 illustrates the architecture/call flow.

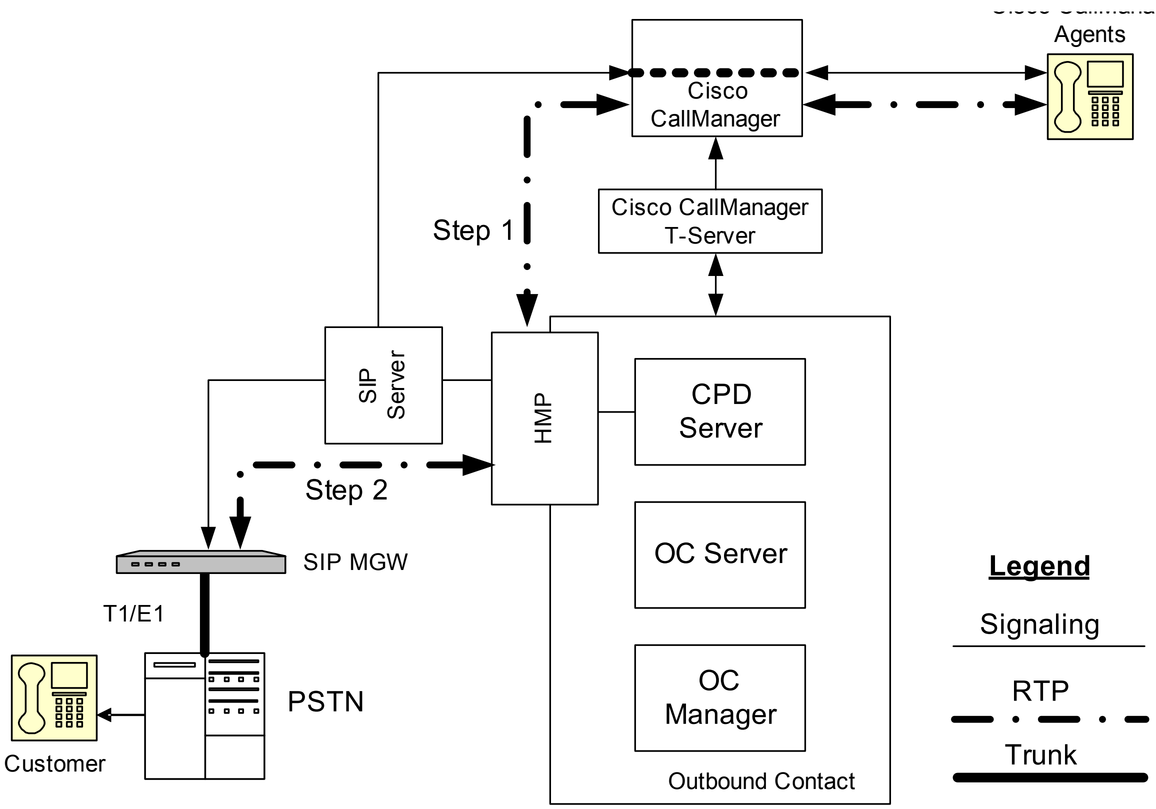

Figure 15: ASM-Mode Call Flow—Cisco CallManager

In this scenario, the call flow proceeds as follows:

Step 1: The Engage Call

1. OCS sends an engage agent request to CPD Server.
2. CPD Server places an engage call using HMP (SIP protocol) to SIP Server.
3. Using the Trunk DN configuration, SIP Server redirects the engage call to Cisco CallManager Route Point DN.
4. As a result of a IRD strategy, Universal Routing Server routes the engage call to an agent who is Ready.
5. The agent answers the call; in other words the established agent is engaged.

Step 2; The Outbound Call

6. CPD Server initiates an outbound call to SIP Server, where another Trunk points to either a Media Gateway or an IP SIP client endpoint.
7. The SIP call in initiated call.
8. Using HMP resources, CPD Server performs call progress analysis.

9. If a positive voice detection occurs, CPD Server bridges the internal leg (the engaged call) and the external leg (the outbound call).

   The call is established between the agent and the customer.
10. CPD Server informs OCS of the call result.
11. After the calls are bridged between the customer and the agent, SIP signalling occurs and RTP streams go through HMP.

Transfer Mode

The following scenario describes a media flow for Outbound Contact with HMP in the Transfer mode and the Cisco CallManager T-Server.

Figure 16 illustrates the architecture/call flow.

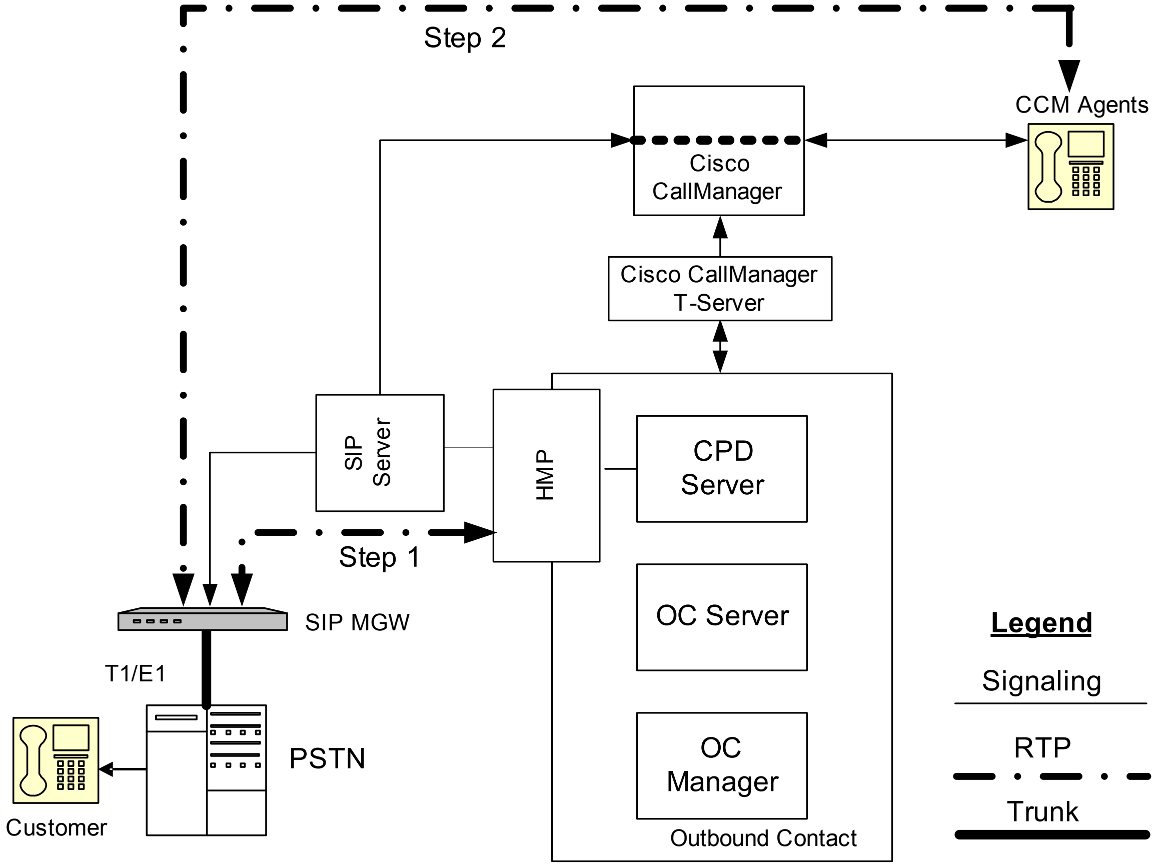

Figure 16: Transfer Mode—Cisco CallManager

Step 1

1. CPD Server places an outbound call to SIP Server where a Trunk DN points to a Media Gateway or a SIP endpoint.
2. The Media Gateway (MGW) then initiates the actual call.
3. Using HMP resources, CPD Server performs call progress analysis on the outbound call and sends the call result (`AttributeCallState`) to OCS.

Step 2

4. If a positive voice detection occurs, CPD Server informs URS.
5. URS routes the call (per the routing strategy) to a Route Point DN of an agent in the `Ready` state on the Cisco CallManager switch.
6. After the call is routed to the agent, no SIP signaling or RTP streams go through HMP.

Note: In this scenario, a transfer of the outbound call occurs rather than a bridging of two calls, as no engage call is placed, The agent is found after the outbound call is placed rather than before.

Outbound Contact with Genesys Voice Platform

Figure 17 illustrates how to use Outbound Contact with Genesys Voice Platform.

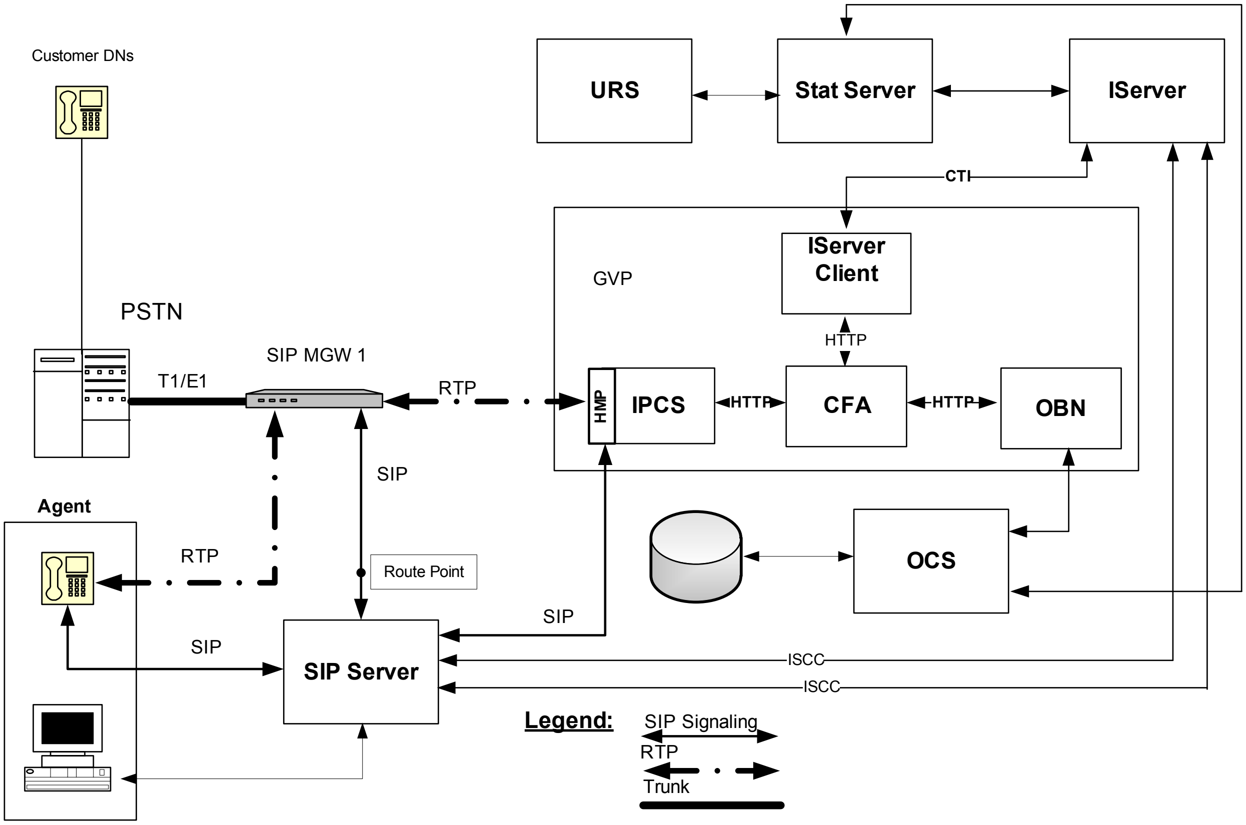

Figure 17: Overview of Outbound Contact with Genesys Voice Platform

In this scenario, the call flow proceeds as follows:

1. OCS uses an `Outbound Notification (OBN) Manager` application to make an outbound call using GVP.
2. When a `Power GVP` mode campaign is loaded in Outbound Contact Manager (OCM), OCS registers with OBN Manager using the `RequestOBNRegisterClient` and the `EventOBNAdapterConnected` messages. When the campaign is unloaded, then OBN Manager receives an `EventOBNAdapterDisconnected` message, which disconnects it from OCS.
3. OCS issues an outbound call request to GVP by sending a `RequestOBNRecordStartDial` message to OBN Manager. This message includes all dialing parameters and any `UserData` parameters. OBN Manager sends an `EventOBNRecordStartDialAck` message back to OCS when dialing has begun.

   Note: `UserData` is sent in this request only, and will not be sent in any other request.

4. OBN Manager forms a HTTP request using the `Start Dial` parameters and the `UserData` attributes to CFA in a `NEW_SESSION_REQ` message.
5. CFA processes the `NEW_SESSION_REQ` message from OBN Manager and requests IPCS/VCS to make an outbound call to the customer's DN using a `CREATE_LEG_REQ` message.
6. IPCS/VCS updates CFA with a call progress analysis (CPA) result in the `DIAL_AND_BRIDGE_DONE_REQ` message.

Notes: If the customer DN requests a call transfer to an agent, the call can be transferred by CTI or by the SIP `REFER` method on the platform.

- If you want to transfer using CTI, then perform the following steps:
  - Enable the CTI call transfer in the CFA section of EMPS. This enables a one-step call transfer from GVP to IServer.
- If you want to transfer a call using the SIP `REFER` method, then perform the following steps:
  - Enable the transfer type as `OneSignalChannel` containing the SIP `REFER` method as the Transfer Option in IVR Profile provisioning of EMPS.
  - Configure the Transfer Type in the CFA section of EMPS as `Transfer on Platform`. With this type of transfer, the SIP `REFER` request goes directly to SIP Server and the transfer occurs on platform.

7. CFA then updates OBN Manager with the following information:
   - The CPA result and the DN that it received from IPCS/VCS.
   - The UUID information that it received from the IServer client.

8. OBN Manager translates the CPA result from CFA into a call result that becomes the `GSW_CALL_RESULT` OCS key.
9. OBN Manager sends this information to OCS in the `EventOBNRecordProcessing` message. If the call result is `CPA_NORMAL`, IPCS/VCS instructs an `IVR` application to begin processing the call. The application information is provided by OCS to GVP using the `app_dbid` attribute in the `RequestOBNRecordStartDial` message.

Note: This `IVR` application can be Genesys Studio Voice XML Application or any another VXML application. The `IVR` application information is provisioned in the Element Management Provisioning System (EMPS) in GVP.

10. The application specific data is sent by the Genesys Studio Application to OBN Manger using the `HTTP Post` method. All the incremental data from the application is updated and saved locally by OBN Manager.
11. Once the application is successfully completed for the customer DN, IPCS will send an `END_SESSION_REQ` message back to CFA.
12. CFA ends the outbound call. It also translates the `END_SESSION_REQ` message into a call result and updates OBN Manager that the outbound call has ended.
13. OBN Manager informs OCS that the outbound call has ended using the `EventOBNRecordProcessed` message, which contains all updated `UserData` parameters. The `EventOBNRecordProcessed` message is the last message from OBN Manager to OCS.

Notes:

If Genesys Studio Application or any other VXML application wants to reschedule a record for a different phone number than the one originally dialed, the `RequestOBNAddRecord` message is used by OBN Manager. OCS acknowledges this request with an `EventOBNAddRecordAck` message after it has added a new record containing the new phone number into the calling list.

If OCS wants to cancel processing a record that has already been dialed, OCS sends a `RequestOBNRecordStopProc` message to OBN Manager. OBN Manager will respond with an `EventOBNAddRecordAck` message.

All error messages between OBN Manager and OCS use the `EventOBNError` message.

Any CTI communication between OCS and GVP uses the `iServer` client in GVP.

Index