



**Outbound Contact 8.1**

Reference Manual

**The information contained herein is proprietary and confidential and cannot be disclosed or duplicated without the prior written consent of Genesys Telecommunications Laboratories, Inc.**

Copyright © 2004–2015 Genesys Telecommunications Laboratories, Inc. All rights reserved.

## **About Genesys**

Genesys is the world's leading provider of customer service and contact center software—with more than 4,000 customers in 80 countries. Drawing on its more than 20 years of customer service innovation and experience, Genesys is uniquely positioned to help companies bring their people, insights and customer channels together to effectively drive today's customer conversation. Genesys software directs more than 100 million interactions every day, maximizing the value of customer engagement and differentiating the experience by driving personalization and multi-channel customer service—and extending customer service across the enterprise to optimize processes and the performance of customer-facing employees. Go to [www.genesys.com](http://www.genesys.com) for more information.

Each product has its own documentation for online viewing at the Genesys Documentation 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 and the Genesys logo are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other company names and logos may be trademarks or registered trademarks of their respective holders.

The Crystal monospace font is used by permission of Software Renovation Corporation, [www.SoftwareRenovation.com](http://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.

## **Customer Care from Genesys**

If you have purchased support directly from Genesys, please contact [Genesys Customer Care](#). Before contacting Customer Care, please refer to the [Genesys Care Program 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 Licensing Guide](#).

## **Released by**

Genesys Telecommunications Laboratories, Inc. [www.genesys.com](http://www.genesys.com)

**Document Version:** 81ou\_ref\_01-2015\_v8.1.501.00



# Table of Contents

Preface	7
About Outbound Contact	7
Intended Audience	8
Making Comments on This Document	8
Contacting Genesys Customer Care	8
Documentation Change History	8
Version 8.1.501.00	9
Version 8.1.301.00	9
Version 8.1.201.00	9
Version 8.1.101.00	9
Chapter 1	
<b>Communication Protocols</b>	<b>11</b>
Event Overview	12
Characteristics of Event Structures	12
Attaching Record Information to Desktop and OCS User Events	14
User Event Attributes	16
Attaching Scripts to OCS User Events and Telephony Events	25
Desktop Requests and OCS Responses	26
Campaign Status Notification from OCS to Desktop	27
PreviewDialingModeStart Request	36
ReadyTime Request	39
Preview Record Request and Acknowledgment	42
Updating Call Results and Custom Fields	46
Chained Records	52
Rejecting Records	57
Canceling Records	59
Record Cancel Requests	61
Canceled and DoNotCall Chained Records	65
Submitting DoNotCall Requests	66
Scheduling and Rescheduling Records	71
Adding Records to a Calling List	78
AddRecord Request	78
Unsolicited Notifications	82

	Agent Logout .....	82
	Agent Logout Protocol .....	82
	LogOutAcknowledge.....	85
	Proactive Interaction Support .....	86
	Caller ID Support .....	87
	Virtual Agent Support for Notifications.....	88
	Personalized Ring Tone Support .....	90
	Outbound Contact Library .....	90
	Error Names and Codes .....	91
	All Genesys Events and Event Type Protocols.....	92
Chapter 2	<b>Communication DN API .....</b>	<b>97</b>
	Overview.....	97
	Data Formats .....	98
	Protocol Sequencing .....	98
	User Event Structure .....	102
	User Data Enumeration Values .....	103
	Record Cancellation from a Third-Party Application.....	110
	DoNotCall Requests from a Third-Party Application .....	114
	DNC Messages.....	114
	Mandatory Attributes.....	115
	UserEvent Structure.....	115
Chapter 3	<b>OCS Support for the HTTP/HTTPS Protocol.....</b>	<b>117</b>
	URI Supported by OCS .....	118
	Format .....	118
	Supported Parameters.....	118
	URI Delivery to the Client .....	118
	OCS Resource Types Accessible via URI.....	119
	Client Requests .....	119
	Client Request Headers.....	119
	Client Request BODY .....	120
	HTTP Requests for the records Resource.....	121
	HTTP Requests for the phones and customer_ids Resources.....	122
	HTTP Requests for the lists Resource.....	123
	HTTP Request for the campaigngroups Resource .....	124
	HTTP Responses .....	125
	Guidelines for Client Connections .....	127
	Examples.....	127
	A Successful RecordProcessed Request .....	128
	An Unsuccessful RecordProcessed Request .....	128
	A Successful DoNotCall Request for a Specific Phone Number .....	129

	Successful AddRecord Request of the General Type .....	129
	Successful AddRecord Request of the Campaign Rescheduled Type .....	130
	Successful Load of a Campaign Group .....	130
	Setting Dialing Parameters for a Campaign Group.....	131
	Unsuccessful Load of a Campaign Group .....	131
Chapter 4	<b>Defined Constants .....</b>	<b>133</b>
	Field Definitions and Enumeration Values .....	133
	Call Results .....	135
	Call Result Mapping by OCS .....	139
	Data Types .....	141
	Contact Information Types.....	141
	Record Types .....	142
	Record Statuses .....	143
	Combining Record Statuses and Record Types.....	146
Chapter 5	<b>Recommended DBMS Optimizations .....</b>	<b>149</b>
	Optimizing Calling List Tables for Weight Rules .....	149
	Temporary Tables Considerations .....	150
	Maintaining Indexes for Large Calling Lists .....	151
	Configuring IBM DB2 for Use of the LANGUAGE SQL Option .....	153
Chapter 6	<b>Supported Functionality with IP Telephony.....</b>	<b>155</b>
	Overview.....	155
	Outbound Contact with SIP Server.....	159
	Transfer Mode (MGW with CPA) .....	159
	Transfer Mode (MGW without CPA) .....	161
	Transfer Mode (MCP as the CPA Provider).....	164
	ASM Mode (MCP as the CPA Provider) .....	166
	ASM Mode (MGW with CPA).....	167
	ASM Mode (MGW without CPA) .....	167
	Outbound Contact with Cisco CallManager.....	172
	ASM Mode .....	172
	Transfer Mode.....	174
	Outbound Contact with GVP 8.1 (Proactive Contact Solution).....	176
	T-Library Functions in an Outbound-VoIP Environment .....	178

Table of Contents

Supplements	<b>Related Documentation Resources .....</b>	<b>183</b>
	<b>Document Conventions .....</b>	<b>186</b>
Index	<b>.....</b>	<b>189</b>



## Preface

Welcome to the *Outbound Contact 8.1 Reference Manual*. This document provides reference information for performing configuration and installation procedures for Outbound Contact.

This document is valid for all 8.1 release(s) of this product.

---

Note: For versions of this document created for other releases of this product, visit the Genesys Documentation website, or request the Documentation Library DVD, which you can order by e-mail from Genesys Order Management at [orderman@genesys.com](mailto:orderman@genesys.com).

---

This preface contains the following sections:

- [About Outbound Contact, page 7](#)
- [Intended Audience, page 8](#)
- [Making Comments on This Document, page 8](#)
- [Contacting Genesys Customer Care, page 8](#)
- [Documentation Change History, page 8](#)

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on [page 183](#).

---

## About Outbound Contact

Outbound Contact 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 document is primarily intended for system engineers and other members of an implementation team who set and maintain Outbound Contact. It has been written with the assumption 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.

---

## Making Comments on This Document

If you especially like or dislike anything about this document, feel free to e-mail your comments to [Techpubs.webadmin@genesys.com](mailto:Techpubs.webadmin@genesys.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 scope of this document only and to the way in which the information is presented. Contact your Genesys Account Representative or Genesys Customer Care if you have suggestions about the product itself.

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

---

## Contacting Genesys Customer Care

If you have purchased support directly from Genesys, please contact [Genesys Customer Care](#).

Before contacting Customer Care, please refer to the [Genesys Care Program Guide](#) for complete contact information and procedures.

---

## Documentation Change History

This section lists topics that are new or have changed significantly since the first release of this document.



## Version 8.1.501.00

The following changes have been made in this document:

- Added GSW\_ASM\_OVERDIAL and GSW\_CPN\_DIGITS to the list of Reserved Keys in Table 6 on [page 18](#).
- Added GSW\_CPN\_DIGITS to the list of Preview Record Attached Data in Table 38 on [page 43](#).

## Version 8.1.301.00

The following changes have been made in this document:

- Information about Smart Logout was added to “Agent Logout” on [page 82](#).

## Version 8.1.201.00

The following changes have been made in this document:

- The call\_attempt field has been marked as a modifiable field in Table 4 on [page 17](#).
- Added cases for RequestRecordCancel request on [page 61](#).
- Clarified descriptions of Abandoned and Dial Error call result types in Table 123 on [page 135](#).
- Added information about Outbound Contact in VoIP environments on [page 159](#) and “ASM Mode (MGW with CPA)” on [page 167](#).

## Version 8.1.101.00

The following Reserved Keys are new in this release. Refer to Table 6 on [page 18](#) for more information.

- GSW\_CALL\_RESULT\_FEEDBACK
- GSW\_ASSURED\_HANDLE





Chapter

# 1

## Communication Protocols

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 either SIP Server or T-Server, to the agent desktop and back again until a call transaction is complete.

---

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

---

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

- [Event Overview, page 12](#)
- [Desktop Requests and OCS Responses, page 26](#)
- [Campaign Status Notification from OCS to Desktop, page 27](#)
- [PreviewDialingModeStart Request, page 36](#)
- [ReadyTime Request, page 39](#)
- [Preview Record Request and Acknowledgment, page 42](#)
- [Updating Call Results and Custom Fields, page 46](#)
- [Chained Records, page 52](#)
- [Rejecting Records, page 57](#)
- [Canceling Records, page 59](#)
- [Submitting DoNotCall Requests, page 66](#)
- [Scheduling and Rescheduling Records, page 71](#)
- [Adding Records to a Calling List, page 78](#)
- [Unsolicited Notifications, page 82](#)
- [Agent Logout, page 82](#)
- [Proactive Interaction Support, page 86](#)

- [Caller ID Support, page 87](#)
- [Virtual Agent Support for Notifications, page 88](#)
- [Personalized Ring Tone Support, page 90](#)
- [Outbound Contact Library, page 90](#)

---

## 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 `GSW_AGENT_REQ_TYPE` key.
- 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 `GSW_USER_EVENT` key.

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 dialing sessions for 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 1](#) should be contained in all error events.

**Table 1: Error Event Attributes**

Key	Type	Description
GSW_ERROR	String	Error name; see “Error Names and Codes” on <a href="#">page 91</a> .
GSW_ERROR_NUMBER	Int	Mandatory

Table 93 on [page 91](#) 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.

---

Note: Genesys recommends that you avoid naming user-defined fields in the calling list table the same as the name of the calling list table.

---

### 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 2](#) shows list of key-value pairs that OCS attaches to outbound call's user data by default.

**Table 2: 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

**Table 2: Default Record Information (Continued)**

Key	Type	Description
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 `send_attribute` option 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 “Outbound Contact Configuration Options” chapter of the *Outbound Contact 8.1 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 3 on [page 16](#) 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 3: 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	The sum of ready and retrieved chains for the campaign
GSW_LIST_COMPLETE	Int	Number of Ready chains for the given calling list
GSW_ERROR_DESCRIPTION	String	Error description
GSW_DIAL_MODE	Int	Dial mode. Valid values are as follows: <ul style="list-style-type: none"> <li>• 1 = Predict</li> <li>• 2 = Progress</li> <li>• 3 = Preview</li> <li>• 4 = ProgressAndSeize</li> <li>• 5 = PredictAndSeize</li> <li>• 8 = PushPreview</li> <li>• 9 = ProgressiveGVP</li> <li>• 11 = PowerGVP</li> </ul>
GSW_APPLICATION_ID	Int	OCS application DBID Valid values begin at 101



**Table 3: User Event Attributes (Continued)**

Key	Type	Description
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 `RecordProcessed` or `UpdateCallCompletionStats` request to modify the values in Genesys mandatory fields and custom fields. See [Table 4](#) for modifiable mandatory Genesys fields.

### Genesys Mandatory Fields

[Table 4](#) contains the only Genesys mandatory fields that are modifiable by the `RecordProcessed` or `UpdateCallCompletionStats` events.

**Table 4: Modifiable Mandatory Genesys Fields**

Genesys Mandatory Field Name	Recommended Key for send_attribute	Type	Description
call_attempt	GSW_ATTEMPTS	Int	Number of attempts made to call this record.
call_result	GSW_CALL_RESULT	Int	Sent to change an automatically detected call result. See <a href="#">Table 7</a> on <a href="#">page 24</a> and <a href="#">Table 123</a> on <a href="#">page 135</a> .
daily_from	GSW_FROM	Int	GSW_FROM to GSW_UNTIL: Time frame when a record can be called, in seconds from midnight (system or local time).
contact_info	GSW_PHONE	String	Customer's phone number.
contact_info_type	GSW_PHONE_TYPE	Int	Customer phone type. See <a href="#">Table 7</a> on <a href="#">page 24</a> .
daily_till	GSW_UNTIL	Int	GSW_FROM to GSW_UNTIL: Time until a record can be called, in seconds from 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 5 on [page 18](#).

**Table 5: 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 6](#) 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 `GSW_` prefix. Do not use this prefix for custom key names that you define using `send_attribute`.

**Table 6: 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.

**Table 6: Reserved Keys (Continued)**

Key	Values	Type	Description
GSW_ASM_OVERDIAL	1	Int	The presence of this key indicates that the outbound call was not merged with an engaging call in ASM mode (VoIP environment) and the call is considered overdialed by OCS.
GSW_ASSURED_HANDLE	1...	Int	Handle of a record which is dialed with guaranteed connection to an agent; added to an engaging call used to seize an agent for guaranteed connection.
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 7 on page 24 and Table 123 on page 135.

**Table 6: Reserved Keys (Continued)**

Key	Values	Type	Description
GSW_CALL_RESULT_FEEDBACK	1	Int	The presence of this key indicates that the given outbound call is selected as a test call for Answering Machine Detection (AMD) false positives and AMD false negatives. This call will be delivered to an agent, even if Answering Machine has been detected by the CPD engine and treatments on AM require otherwise. The call result for this call requires manual verification by an agent. Agent Desktop is expected to provide the actual call result to OCS in the GSW_CALL_RESULT attribute of the UpdateCallCompletionStats or RecordProcessed desktop request.
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.
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 7 on <a href="#">page 24</a> . The values Engaged Predictive and Engaged Progressive correspond to the Predictive with seizing and Progressive with seizing dialing modes.

**Table 6: Reserved Keys (Continued)**

Key	Values	Type	Description
GSW_CONTACT_MEDIA_TYPE	any, email, voice	String	Media type for the calling record that corresponds to the value of the <code>contact_info_type</code> 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_CPN_DIGITS		String	CPN Digits as configured for the given record.
GSW_CUSTOMER_ID		String	Customer ID that is used for requests.
GSW_DATE_TIME	A string represented in time in this format: MM/DD/YY(Y YYY) HH:MM.	String	Date and time of scheduled call, in the record's Time Zone.
GSW_ERROR	Error name	String	Error name. See Table 93 on <a href="#">page 91</a> .
GSW_ERROR_DESCRIPTION	Error description	String	Error name. See Table 93 on <a href="#">page 91</a> .
GSW_ERROR_NUMBER	Error Number	Int	Error code. See Table 93 on <a href="#">page 91</a> .

**Table 6: Reserved Keys (Continued)**

Key	Values	Type	Description
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 or a record cancellation message
GSW_PHONE		String	Customer's phone number.
GSW_PHONE_TYPE		Int	Customer phone type. See Table 7, "Enumeration Table," on page 24.
GSW_RECORD_HANDLE	1...	Int	Unique Record Identifier.
GSW_RECORD_STATUS	See Table 7, "Enumeration Table," on page 24.	Int	Status of adding record sent from a desktop.
GSW_RECORD_TYPE	See Table 7, "Enumeration Table," on page 24.	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).

**Table 6: Reserved Keys (Continued)**

Key	Values	Type	Description
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 94 on <a href="#">page 92</a> .	String	Event identifier for events coming from OCS to desktops.
InteractionType	Outbound	String	Type of the interaction that is 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 7](#) 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 Genesys Administrator). The desktop application should translate the enumeration value to the appropriate description when required for display.

**Table 7: 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 123 on <a href="#">page 135</a> 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 11, Instant Messaging	Int	Customer phone type.



**Table 7: 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 Scripts 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. Calling List (highest priority)
2. Campaign
3. Campaign Group (lowest priority)

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/Campaign group 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 12](#).

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

- One Campaign/Campaign group is running in auto dialing mode.
- Several Campaigns/Campaign groups 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 dialing session for 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.

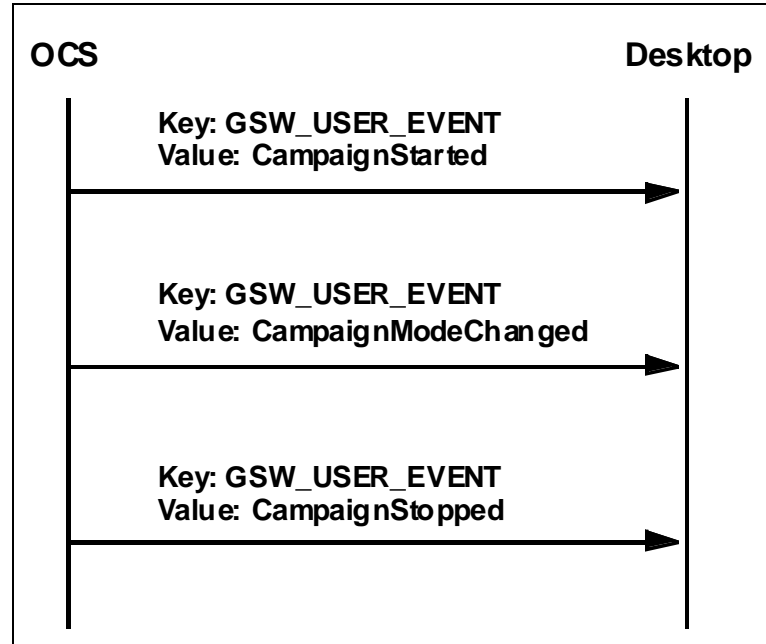


Figure 1: Campaign Status from OCS to the Desktop

## CampaignStarted

OCS sends this event to the desktop when a dialing session for a Campaign is started. [Table 8](#) contains more information.

Table 8: 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 dialing session for 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 9 on [page 29](#) lists the attached data for the CampaignStarted event.

**Table 9: 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 <sup>a</sup>	String	Yes	Power GVP, Progressive GVP, Predictive, Predictive with Seizing, Preview, Progressive, Progressive with Seizing, and Push Preview.  <b>Note:</b> In the OCS log, Predictive with Seizing and Progressive with Seizing appear as Engaged Predictive and Engaged Progressive respectively.	Mode in which the dialing session of a Campaign started.
GSW_CAMPAIGN_NAME	String	Yes	<Campaign name>	
GSW_CAMPAIGN_GROUP_NAME	String	Yes	<Campaign Group name>	

- a. In the OCS log, the Predictive with Seizing and Progressive with Seizing modes are referred to as Engaged Predictive and Engaged Progressive, respectively.

## CampaignLoaded

OCS sends this event to the desktop when a Campaign/Campaign group is loaded. Table 10 on [page 30](#) contains more information.

**Table 10: CampaignLoaded**

CampaignLoaded User Event	
Description	OCS sends this event to all logged in agents when a Dialing Session for a Campaign Group is loaded, or, as a response to an agent login when a Dialing Session for a Campaign Group 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 11 lists the attached data for the CampaignLoaded event.

**Table 11: 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
GSW_CAMPAIGN_NAME	String	Yes	<Campaign name>	
GSW_CAMPAIGN_GROUP_DESCRIPTION	String	Yes	<description> or an empty string	Description of Campaign Group
GSW_CAMPAIGN_GROUP_NAME	String	Yes	<Campaign Group name>	

## CampaignUnloaded

OCS sends this event to the desktop when a Dialing Session for a Campaign is unloaded. Table 12 on [page 31](#) contains more information.

**Table 12: CampaignUnloaded**

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

Table 13 lists the attached data for the CampaignUnloaded event.

**Table 13: 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
GSW_CAMPAIGN_GROUP_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 Dialing Session for a Campaign is stopped. Table 14 contains more information.

**Table 14: CampaignStopped**

CampaignStopped User Event	
Description	OCS sends this event to all logged in agents when a Dialing Session for a Campaign stops.
Recommended Desktop Action	The desktop should stop sending requests to the Campaign.

Table 15 on page 32 lists the attached data for the CampaignStopped event.

**Table 15: 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>	
GSW_CAMPAIGN_NAME	String	Yes	<Campaign name>	
GSW_CAMPAIGN_DESCRIPTION	String	Yes	<description> or an empty string	The description of the Campaign
GSW_CAMPAIGN_GROUP_DESCRIPTION	String	Yes	<description> or an empty string	Description of Campaign Group
GSW_CAMPAIGN_GROUP_NAME	String	Yes	<Campaign Group name>	

## CampaignModeChanged

OCS sends this event to the desktop when the dialing mode for a Campaign has changed. [Table 16](#) contains more information.

**Table 16: 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 17](#) lists the attached data for the CampaignModeChanged event.

**Table 17: 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 17: CampaignModeChanged Attached Data (Continued)**

Data Key	Type	Key Required	Value	Description
GSW_CAMPAIGN_MODE	String	Yes	Power GVP, Progressive GVP, Predictive, Predictive with Seizing, Preview, Progressive, Progressive with Seizing, and Push Preview.  <b>Note:</b> In the OCS log, Predictive with Seizing and Progressive with Seizing appear as Engaged Predictive and Engaged Progressive respectively.	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
GSW_CAMPAIGN_GROUP_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 Group. Table 18 on [page 34](#) contains more information.

---

Note: This notification is a new part of Outbound Contact functionality: agent assignment in multiple Campaigns. Refer to the *Outbound Contact 8.1 Deployment Guide* for more information about this functionality.

---

**Table 18: CampaignGroupAssigned**

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

Table 19 lists the attached data for the CampaignGroupChanged event.

**Table 19: 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
GSW_CAMPAIGN_GROUP_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 dialing session for a Campaign group. Possible status notification messages in a response to this request include:

- CampaignLoaded
- CampaignStarted
- CampaignGroupAssigned

Table 20 on [page 35](#) contains more information.

**Table 20: 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 21 lists the attached data for the CampaignStatusRequest event.

**Table 21: CampaignStatusRequest Attached Data**

Data Key	Type	Key Required	Value	Description
GSW_AGENT_REQ_TYPE	String	Yes	CampaignStatusRequest	Hard coded request name
GSW_APPLICATION_ID <sup>a</sup>	Int	No	OCS application DBID	Target OCS application DBID
GSW_REFERENCE_ID <sup>b</sup>	Int	No	Request ID	Reference identifier for the request

- a. GSW\_APPLICATION\_ID is an optional 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.
- b. GSW\_REFERENCE\_ID is an optional attribute in the message. When present, OCS guarantees to return this attribute (same key and same value) in the response to the desktop request, in both a positive response or an error.

---

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 dialing sessions for 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 Genesys Administrator. 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 22](#) 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 8.1 Deployment Guide*.

**Table 22: 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 23 on [page 37](#) lists the attached data for the PreviewDialingModeStart request.

**Table 23: 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_GROUP_NAME <sup>a</sup>	String	No	Name of the Campaign Group.
GSW_CAMPAIGN_NAME	String	Yes	Name of the Campaign.
GSW_REFERENCE_ID <sup>b</sup>	Int	No	Reference identifier for the request.

- a. Adding this attribute to the request enables the identification of the Campaign Group for environments where several groups are configured, active, and running for the same Campaign. This attribute has a higher priority than the GSW\_CAMPAIGN\_NAME attribute.
- b. GSW\_REFERENCE\_ID is an optional attribute in the message. When present, OCS guarantees to return this attribute (same key and same value) in the response to the desktop request, in both a positive response or an error.

## PreviewDialingModeStartAcknowledge

OCS sends this event to the desktop to acknowledge the start of Preview dialing mode. [Table 24](#) contains more information.

**Table 24: 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 25](#) lists the attached data for the PreviewDialingModeStartAcknowledge event.

**Table 25: Preview Dialing Mode Start Acknowledge Attached Data**

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	PreviewDialingModeStartAcknowledge

**Table 25: Preview Dialing Mode Start Acknowledge Attached Data (Continued)**

Data Key	Type	Key Required	Description
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 26](#) 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 8.1 Deployment Guide*.

**Table 26: 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 27](#) lists the attached data for the `PreviewDialingModeOver` request.

**Table 27: PreviewDialingModeOver Attached Data**

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	<code>PreviewDialingModeOver</code>
GSW_APPLICATION_ID	Int	Yes	Unique ID of OCS
GSW_CAMPAIGN_GROUP_NAME <sup>a</sup>	String	No	Name of the Campaign Group
GSW_CAMPAIGN_NAME	String	Yes	Name of the Campaign.
GSW_REFERENCE_ID <sup>b</sup>	Int	No	Reference identifier for the request.

- a. Adding this attribute to the request enables the identification of the Campaign Group for environments where several groups are configured, active, and running for the same Campaign. This attribute has a higher priority than the `GSW_CAMPAIGN_NAME` attribute.

- b. GSW\_REFERENCE\_ID is an optional attribute in the message. When present, OCS guarantees to return this attribute (same key and same value) in the response to the desktop request, in both a positive response or an error.

## PreviewDialingModeOverAcknowledge

OCS sends this event to the desktop to acknowledge the end of Preview dialing mode. [Table 28](#) contains more information.

**Table 28: 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 29](#) lists the attached data for the PreviewDialingModeOverAcknowledge event.

**Table 29: 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.

- 
- Notes:
- This request was added in the 7.6.1 release.
  - For more information about the `predictive_algorithm` option, see the *Outbound Contact 8.1 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 30 contains more information.

**Table 30: 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 31 lists the attached data for the ReadyTime request.

**Table 31: ReadyTime Attached Data<sup>a</sup>**

Data Key <sup>b</sup>	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 <sup>c</sup>	Int	Yes	<Unique record handle>	Record handle for the record currently on the agent's desktop
GSW_REFERENCE_ID <sup>d</sup>	Int	No	<Reference ID>	Reference identifier for the request

- ReadyTime can only be applied to a record currently being processed by the agent.
- ReadyTime supports only those key-value pairs listed in this table. Any other pairs will be ignored by OCS.
- The record handle is a mandatory attribute because it identifies the record currently being process by the agent for OCS.
- GSW\_REFERENCE\_ID is an optional attribute in the message. When present, OCS guarantees to return this attribute (same key and same value) in the response to the desktop request, in both a positive response or an error.



## ReadyTimeAcknowledge

OCS sends this event to the desktop to acknowledge the ReadyTime request, or sends an error (see “ReadyTime Error”). Table 32 contains more information.

**Table 32: 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 33 lists the attributes for the ReadyTimeAcknowledge event.

**Table 33: 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 34 is returned.

**Table 34: 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

**Table 34: ReadyTime Error Codes (Continued)**

Error Code	Error Description	Returned When:
104	Agent not found	This agent is unknown to OCS
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 CampaignStarted event 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 35](#) contains more information.

**Table 35: PreviewRecordRequest**

PreviewRecordRequest	
Description	Request to send preview record.
OCS Action	Conditionally sends acknowledgment, depending on setting of the PreviewDialingModeStart event. See “PreviewDialingModeStart Request” on <a href="#">page 36</a> .

Table 36 on [page 43](#) lists the attached data for PreviewRecordRequest.

**Table 36: 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.
GSW_REFERENCE_ID <sup>a</sup>	Int	No	Reference identifier for the request.

- a. GSW\_REFERENCE\_ID is an optional attribute in the message. When present, OCS guarantees to return this attribute (same key and same value) in the response to the desktop request, in both a positive response or an error.

## PreviewRecord

The following event is sent by OCS to the desktop in response to a request for records in the Preview dialing mode. [Table 37](#) contains more information. See “Reserved Keys” on [Page 18](#) and “Genesys Enumeration Table” on [page 24](#) for predefined attribute values.

**Table 37: 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 38](#) lists the attached data for the PreviewRecord event.

**Table 38: 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.

**Table 38: PreviewRecord Attached Data (Continued)**

Data Key	Type	Key Required	Description
GSW_CALL_RESULT	Int	Yes	Call Result set by dialer or saved from previous call. (See the “Genesys Enumeration Table” on <a href="#">page 24</a> .)
GSW_CALL_TIME	String	No	System time when record was called, in seconds from 1/1/70 (GMT).
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CAMPAIGN_GROUP_NAME <sup>a</sup>	String	No	Name of the Campaign Group.
GSW_CAMPAIGN_NAME	String	Yes	Name of the Campaign.
GSW_CHAIN_ID	Int	Yes	Unique chain ID
GSW_CPN_DIGITS	String	Yes	CPN Digits as configured for the given record.
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 <a href="#">page 24</a> ).
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.

- a. Adding this attribute to the request enables the identification of the Campaign Group for environments where several groups are configured, active, and running for the same Campaign. This attribute has a higher priority than the GSW\_CAMPAIGN\_NAME attribute.

## 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 42) in a few seconds. Table 39 contains more information.

**Table 39: 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 40 lists the attached data for the No Records Available event.

**Table 40: No Records Available 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 91.
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 41 on page 46 contains more information.

---

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

---

**Table 41: NoRunningPreviewCampaigns**

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

Table 42 lists the attached data for the NoRunningPreviewCampaign error.

**Table 42: 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 <a href="#">page 91</a> .
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 123 on [page 135](#).)

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.

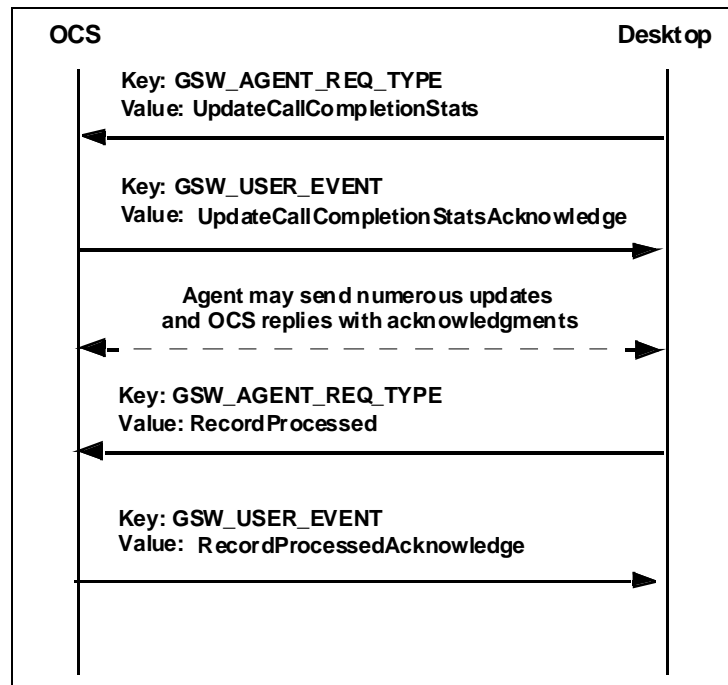


Figure 2: Updating Call Results and Custom Fields Data Flow

## UpdateCallCompletionStats

The desktop sends this request to OCS to update a record on completion of a call. Table 43 on [page 48](#) contains more information.

**Table 43: UpdateCallCompletionStats**

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

Table 44 lists the attached data for the UpdateCallCompletionStats request.

**Table 44: 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 7 on <a href="#">page 24</a> and Table 123 on <a href="#">page 135</a> .
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 7 on <a href="#">page 24</a> .
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_GROUP_NAME <sup>a</sup>	String	No	Name of the Campaign Group
GSW_CAMPAIGN_NAME	String	Yes	Name of the Campaign.
GSW_REFERENCE_ID <sup>b</sup>	Int	No	Reference identifier for the request.
Custom Fields	Any	No	Custom Fields.

- a. Adding this attribute to the request enables the identification of the Campaign Group for environments where several groups are configured, active, and running for the same campaign. This attribute has a higher priority than the GSW\_CAMPAIGN\_NAME attribute.



- b. GSW\_REFERENCE\_ID is an optional attribute in the message. When present, OCS guarantees to return this attribute (same key and same value) in the response to the desktop request, in both a positive response or an error.

## UpdateCallCompletionStatsAcknowledge

OCS sends this event to the desktop to acknowledge a call completion notification. [Table 45](#) contains more information.

**Table 45: 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 46](#) lists the attached data for the UpdateCallCompletionStatsAcknowledge event.

**Table 46: UpdateCallCompletionStatsAcknowledge 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 47](#) contains more information.

**Table 47: 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 48 lists the attached data for the RecordProcessed request.

**Table 48: 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 7 on <a href="#">page 24</a> and Table 123 on <a href="#">page 135</a> . <b>Note:</b> GSW_CALL_RESULT is an optional attribute in both UpdateCallCompletionStats and RecordProcessed requests. If omitted, in Preview dialing mode the default call result is Answer.
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CAMPAIGN_GROUP_NAME <sup>a</sup>	String	No	Name of the Campaign Group.
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 7 on <a href="#">page 24</a> .

**Table 48: RecordProcessed Attached Data (Continued)**

Data Key	Type	Key Required	Description
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.
GSW_REFERENCE_ID <sup>b</sup>	Int	No	Reference identifier for the request.
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.

- a. Adding this attribute to the request enables the identification of the Campaign Group for environments where several groups are configured, active, and running for the same Campaign. This attribute has a higher priority than the GSW\_CAMPAIGN\_NAME attribute.
- b. GSW\_REFERENCE\_ID is an optional attribute in the message. When present, OCS guarantees to return this attribute (same key and same value) in the response to the desktop request, in both a positive response or an error.

## RecordProcessedAcknowledge

OCS sends this event to the desktop to acknowledge a RecordProcessed notification. [Table 49](#) contains more information.

**Table 49: 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 50](#) lists the attached data for RecordProcessedAcknowledge event.

**Table 50: RecordProcessedAcknowledge Attached Data**

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	RecordProcessedAcknowledge

**Table 50: RecordProcessedAcknowledge Attached Data (Continued)**

Data Key	Type	Key Required	Description
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 Record Not Found response error 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 on [page 53](#) is an example of a typical chained record data flow.

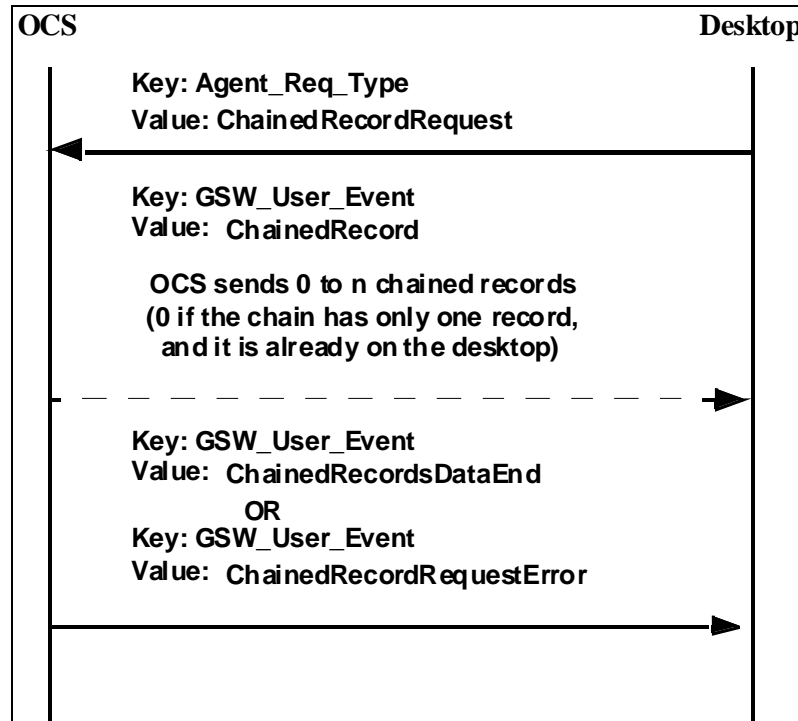


Figure 3: Chained Record Data Flow

## ChainedRecordRequest

The desktop sends this request to OCS to request a record chain. [Table 51](#) contains more information.

**Table 51: 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 52](#) lists the attached data for ChainedRecordRequest.

**Table 52: ChainedRecordRequest Attached Data**

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	ChainedRecordRequest

**Table 52: ChainedRecordRequest Attached Data (Continued)**

Data Key	Type	Key Required	Description
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CAMPAIGN_GROUP_NAME <sup>a</sup>	String	No	Name of the Campaign Group
GSW_CAMPAIGN_NAME	String	No	Name of the Campaign. <b>Note:</b> This value is optional in Outbound Contact releases 7.5 and higher. It was mandatory in previous releases.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.
GSW_REFERENCE_ID <sup>b</sup>	Int	No	Reference identifier for the request.

- a. Adding this attribute to the request enables the identification of the Campaign Group for environments where several groups are configured, active, and running for the same Campaign. This attribute has a higher priority than the GSW\_CAMPAIGN\_NAME attribute.
- b. GSW\_REFERENCE\_ID is an optional attribute in the message. When present, OCS guarantees to return this attribute (same key and same value) in the response to the desktop request, in both a positive response or an error.

## 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 53](#) contains more information.

**Table 53: 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 54 on [page 55](#) lists the attached data for the ChainedRecord event.

**Table 54: 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).
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 <a href="#">page 24</a> .)
GSW_CAMPAIGN_NAME	String	No	Name of the Campaign. <b>Note:</b> This value is optional in Outbound Contact releases 7.5 and higher. It was mandatory in previous releases.
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 <a href="#">page 24</a> ).
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.

**Table 54: ChainedRecord Attached Data (Continued)**

Data Key	Type	Key Required	Description
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 55](#) contains more information.

**Table 55: 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 56](#) lists the attached data for the ChainedRecordsDataEnd event.

**Table 56: 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.

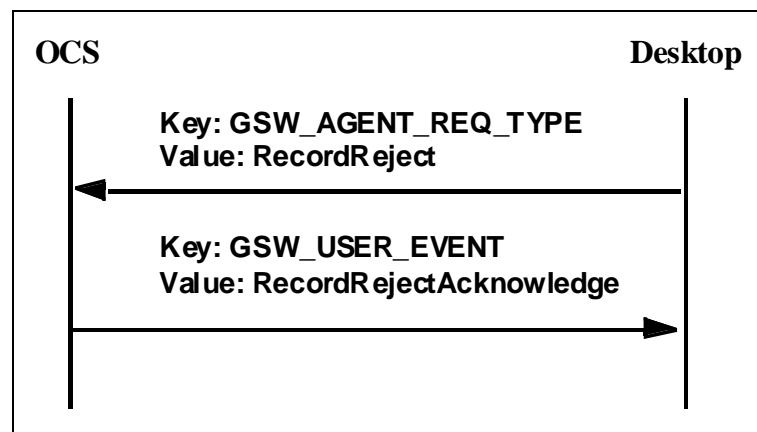


Figure 4: `RejectRecord` Data Flow

### 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 57 contains more information.

Table 57: `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 <code>RecordProcessed</code> after this request.

**Table 57: RecordReject (Continued)**

RecordReject Request	
OCS Action	OCS marks this record, and the rest of the chain, as general and ready.

Table 58 lists the attached data for the RejectRecord event.

**Table 58: 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.
GSW_CAMPAIGN_GROUP_NAME <sup>a</sup>	String	No	Name of the Campaign Group
GSW_CAMPAIGN_NAME	String	Yes	Name of the Campaign.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.
GSW_REFERENCE_ID <sup>b</sup>	Int	No	Reference identifier for the request.

- a. Adding this attribute to the request enables the identification of the Campaign Group for environments where several groups are configured, active, and running for the same Campaign. This attribute has a higher priority than the GSW\_CAMPAIGN\_NAME attribute.
- b. GSW\_REFERENCE\_ID is an optional attribute in the message. When present, OCS guarantees to return this attribute (same key and same value) in the response to the desktop request, in both a positive response or an error.

## RecordRejectAcknowledge

The following event is sent to the desktop by OCS to acknowledge a rejected record. Table 59 contains more information.

**Table 59: RecordRejectAcknowledge**

RecordRejectAcknowledge User Event	
Description	OCS accepts RejectRecord request.

**Table 59: RecordRejectAcknowledge (Continued)**

RecordRejectAcknowledge User Event	
Recommended Desktop Action	Kill the record and the chain.

Table 60 lists the attached data for the RecordRejectAcknowledge event.

**Table 60: 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 canceled. 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/Campaign groups.
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 110](#).)
3. OCS attempts to find a record that has the matching phone number in OCS memory and in calling lists assigned to loaded Campaigns/Campaign groups.
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 97](#).

## Record Cancel Requests

This section describes record cancel requests and acknowledgments.

### RequestRecordCancel

The desktop sends this request to OCS for one of the following reasons:

- To cancel a record or a chain.
- To cancel a record for which the call has already been dialed, and its record displayed on the desktop.

In both cases, the record is marked as `Cancelled` in the database. [Table 61](#) contains more information.

**Table 61: RequestRecordCancel**

RequestRecordCancel Request	
Description	<p>Desktop sends a request to OCS to either:</p> <ul style="list-style-type: none"> <li>• Cancel a record or a chain, in which case the preview record or scheduled call should not be dialed.</li> </ul> <p>or</p> <ul style="list-style-type: none"> <li>• Cancel a record or chain for which the call has already been dialed, and the record displayed on the desktop.</li> </ul> <p>In both cases, the record should not be re-sent to another agent. It should be marked in the database as <code>Cancelled</code>.</p>
OCS Action	Cancel record.

[Table 62](#) lists the attached data for the `RequestRecordCancel` request.

**Table 62: RequestRecordCancel Attached Data**

Data Key	Type	Key Required	Description
<code>GSW_AGENT_REQ_TYPE</code>	String	Yes	<code>RequestRecordCancel</code>

**Table 62: RequestRecordCancel Attached Data (Continued)**

Data Key	Type	Key Required	Description
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_GROUP_NAME <sup>a</sup>	String	No	Name of the Campaign Group
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_CUSTOMER_ID <sup>b</sup>	String	No	A user-defined field in the Calling List table that serves as a customer identifier.
GSW_PHONE <sup>b</sup>	String	No	Customer's phone number.
GSW_RECORD_HANDLE <sup>b</sup>	Int	No	Unique Record Identifier.
GSW_REFERENCE_ID <sup>c</sup>	Int	No	Reference identifier for the request.

- Adding this attribute to the request enables the identification of the Campaign Group for environments where several groups are configured, active, and running for the same Campaign. This attribute has a higher priority than the GSW\_CAMPAIGN\_NAME attribute.
- Record cancellation requests can identify sets of records by the record handle, phone number, and customer ID. If more than one record identifier is included in the same request, the identifiers are prioritized as follows: record handle (highest), phone (middle), and customer ID (lowest).
- GSW\_REFERENCE\_ID is an optional attribute in the message. When present, OCS guarantees to return this attribute (same key and same value) in the response to the desktop request, in both a positive response or an error.

### 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 `Cancelled`; if the field has the `RecordOnly` value, the record with the requested `GSW_PHONE` is marked as `Cancelled`, 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 8.1 Deployment Guide*.

## RecordCancelAcknowledge

OCS sends this event to the desktop to acknowledge a `RequestRecordCancel` event. [Table 63](#) contains more information.

**Table 63: 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 64](#) lists the attached data for the `RecordCancelAcknowledge` event.

**Table 64: RecordCancelAcknowledge Attached Data**

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

**Table 64: 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 <i>Outbound Contact 8.1 Deployment Guide</i> .
GSW_MESSAGE	String	No	OCS message (“Incomplete processing: record(s) on desktop”) notifying the RequestRecordCancel requester (agent desktop or third party) about OCS’s inability to completely handle the cancellation request, because calls records are still active on an agent’s desktop.  <b>Note:</b> This only affects cancellation by phone and customer ID. It does not affect RequestRecordCancel requests made by the record handle or DoNotCall requests.
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) or the same customer ID (GSW\_CUSTOMER



ID) on another desktop. When OCS sends a `RecordCancel` notification to the desktop, the agent should remove the record from the desktop. [Table 65](#) contains more information.

**Table 65: 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 <code>GSW_CHAIN_ATTR= RecordOnly</code> . Delete the chain if the <code>RecordCancel</code> contains <code>GSW_CHAIN_ATTR=AllChain</code> .

[Table 66](#) lists the attached data for the `RecordCancel` event.

**Table 66: RecordCancel Attached Data**

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

## 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/Campaign groups.
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 Group, 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 97](#).)
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/Campaign groups.
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 Group, 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 97](#).

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 dialing session for 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 either be restarted to acknowledge the changes or, alternatively, OCS will pick up these updates upon next reread of the Do Not Call list, if OCS is configured for such rereads. For a description of Do Not Call reread functionality, see the “Rereading of the Do Not Call List” section in the Outbound Contact 8.1 Deployment Guide.

Most administrators choose to synchronize OCS with the updated DoNotCall table (`gsw_donotcall_list`) during off-hour periods, so that restarting the server or rereading the DoNotCall table 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 67](#) contains more information.

**Table 67: DoNotCall**

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

[Table 68](#) lists the attached data for the `DoNotCall` request.

**Table 68: DoNotCall Attached Data**

Data Key	Type	Key Required	Description
<code>GSW_AGENT_REQ_TYPE</code>	String	Yes	<code>DoNotCall</code>
<code>GSW_APPLICATION_ID</code>	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
<code>GSW_CALLING_LIST</code>	String	No <sup>a</sup>	Name of the calling list.
<code>GSW_Campaign_GROUP_NAME<sup>b</sup></code>	String	No	Name of the Campaign Group
<code>GSW_CAMPAIGN_NAME</code>	String	No <sup>a</sup>	Name of the Campaign.

**Table 68: DoNotCall Attached Data (Continued)**

Data Key	Type	Key Required	Description
GSW_CHAIN_ATTR	String	No <sup>c</sup>	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 <sup>b</sup>	Customer's phone number.
GSW_CUSTOMER_ID	String	No <sup>c</sup>	A user-defined field in the Calling List table that serves as a customer identifier for DoNotCall requests.
GSW_RECORD_HANDLE	Int	No <sup>a</sup>	Unique Record Identifier.
GSW_REFERENCE_ID <sup>d</sup>	Int	No	Reference identifier for the request.

- a. If GSW\_RECORD\_HANDLE is specified, then GSW\_CALLING\_LIST and GSW\_CAMPAIN\_NAME are required.
- b. Adding this attribute to the request enables the identification of the Campaign Group for environments where several groups are configured, active, and running for the same Campaign. This attribute has a higher priority than the GSW\_CAMPAIN\_NAME attribute.
- c. If GSW\_RECORD\_HANDLE is not specified, then either GSW\_PHONE or GSW\_CUSTOMER\_ID must be present. See "Mandatory Fields" on [page 69](#).

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.

- d. GSW\_REFERENCE\_ID is an optional attribute in the message. When present, OCS guarantees to return this attribute (same key and same value) in the response to the desktop request, in both a positive response or an error.

### 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 69](#) contains more information.

**Table 69: DoNotCallAcknowledge**

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

[Table 70](#) lists the attached data for the `DoNotCallAcknowledge` event.

**Table 70: DoNotCallAcknowledge Attached Data**

Data Key	Type	Key Required	Description
<code>GSW_USER_EVENT</code>	String	Yes	<code>DoNotCallAcknowledge</code>
<code>GSW_APPLICATION_ID</code>	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
<code>GSW_CALLING_LIST</code>	String	No <sup>a</sup>	Name of the calling list.
<code>GSW_CAMPAIGN_NAME</code>	String	No <sup>a</sup>	Name of the Campaign.
<code>GSW_CHAIN_ATTR</code>	String	No	Flag determining whether to update the record chain or just the single record. Values are <code>AllChain</code> or <code>RecordOnly</code> . (Default = <code>RecordOnly</code> )
<code>GSW_PHONE</code>	String	No <sup>b</sup>	Customer's phone number.

**Table 70: DoNotCallAcknowledge Attached Data (Continued)**

Data Key	Type	Key Required	Description
GSW_CUSTOMER_ID	String	No <sup>b</sup>	A user-defined field in the Calling List table that serves as a customer identifier for DoNotCall requests.
GSW_RECORD_HANDLE	Int	No <sup>a</sup>	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 69](#).

---

## 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 `predictive_callback` option in the *Outbound Contact 8.1 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 71](#) 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 71: 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 72](#) lists the attached data for a `RecordReschedule` event.

**Table 72: RecordReschedule Attached Data**

Data Key	Type	Key Required	Description
<code>GSW_AGENT_REQ_TYPE</code>	String	Yes	<code>RecordReschedule</code>
<code>GSW_APPLICATION_ID</code>	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
<code>GSW_CALLBACK_TYPE</code>	String	Yes	Type of callback an agent wants to create, either <code>Personal</code> or <code>Campaign</code> .
<code>GSW_CAMPAIGN_GROUP_NAME<sup>a</sup></code>	String	No	Name of the Campaign Group.
<code>GSW_CAMPAIGN_NAME</code>	String	Yes	Name of the Campaign.



**Table 72: RecordReschedule Attached Data (Continued)**

Data Key	Type	Key Required	Description
GSW_DATE_TIME	String	Yes	Date and time of scheduled call, in the record's Time Zone.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.
GSW_REFERENCE_ID <sup>b</sup>	Int	No	Reference identifier for the request.

- a. Adding this attribute to the request enables the identification of the Campaign Group for environments where several groups are configured, active, and running for the same campaign. This attribute has a higher priority than the GSW\_CAMPAIGN\_NAME attribute.
- b. GSW\_REFERENCE\_ID is an optional attribute in the message. When present, OCS guarantees to return this attribute (same key and same value) in the response to the desktop request, in both a positive response or an error.

## RecordRescheduleAcknowledge

OCS sends this event to the desktop to acknowledge a rescheduled record. [Table 73](#) contains more information.

**Table 73: 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 74](#) lists the attached data for a RecordRescheduleAcknowledge event.

**Table 74: 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.

**Table 74: RecordRescheduleAcknowledge 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, in the record's Time Zone.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.

## ScheduledRecordReschedule

The desktop sends this event to OCS to reschedule a previously rescheduled record. [Table 75](#) contains more information.

**Table 75: ScheduleRecordReschedule**

ScheduleRecordReschedule User Event	
Description	Request a reschedule of Preview Record, Predictive Call, or Scheduled Call when a rescheduled call cannot be completed and must be set for another time.
OCS Action	Update a record chain and reschedule the record.

[Table 76](#) lists the attached data for a ScheduledRecordReschedule event.

**Table 76: 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.
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_GROUP_NAME <sup>a</sup>	String	No	Name of the Campaign Group.
GSW_CAMPAIGN_NAME	String	Yes	Name of the Campaign.
GSW_DATE_TIME	String	Yes	Date and time of scheduled call, in the record's Time Zone.

**Table 76: ScheduledRecordReschedule Attached Data (Continued)**

Data Key	Type	Key Required	Description
GSW_PHONE	String	No	Customer's phone number.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.
GSW_REFERENCE_ID <sup>b</sup>	Int	No	Reference identifier for the request.

- a. Adding this attribute to the request enables the identification of the Campaign Group for environments where several groups are configured, active, and running for the same Campaign. This attribute has a higher priority than the GSW\_CAMPAIGN\_NAME attribute.
- b. GSW\_REFERENCE\_ID is an optional attribute in the message. When present, OCS guarantees to return this attribute (same key and same value) in the response to the desktop request, in both a positive response or an error.

## ScheduledRecordRescheduleAcknowledge

OCS sends this event to the desktop to acknowledge the rescheduling of a scheduled record. [Table 77](#) contains more information.

**Table 77: 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 78](#) lists the attached data for a ScheduledRecordRescheduleAcknowledge event.

**Table 78: 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.

**Table 78: ScheduledRecordRescheduleAcknowledge 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_DATE_TIME	String	Yes	Date and time of scheduled call, in the record's Time Zone.
GSW_PHONE	String	No	Customer's phone number.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.

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 79](#) contains more information.

**Table 79: 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 that have the <code>send_attribute</code> option configured.

Table 80 lists the attached data for a ScheduledCall event.

**Table 80: 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 <a href="#">page 24</a> .)
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, in the record’s Time Zone.
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 <a href="#">page 24</a> ).
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/Campaign group 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. When adding new record that is not a part of the existing chain to the calling list, you can set up this new record as one of the following types, provided the fields of the record are set up correctly for the type of the record: General, Campaign Callback, Personal Callback, Campaign Rescheduled or Personal Rescheduled.

## 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/Campaign group. 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 82](#)). 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.

When a record is added to the existing chain, it assumes the type of the chain, regardless of the type assigned to this record. Therefore, if a chain of records is on an agent desktop and new record of type `Callback` needs to be added, the agent should perform the following steps:

1. Add a record of type `General` to the existing chain.
2. Issue a `RequestChainedRecords` desktop request, to have OCS deliver the newly added record to the desktop.
3. Issue a `RecordReschedule` request using the handle of the newly added 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 81](#) contains more information.

**Table 81: 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 82](#) lists the attached data for an AddRecord request.

**Table 82: AddRecord Attached Data**

Data Key	Type	Key Required	Description
<code>GSW_AGENT_REQ_TYPE</code>	String	Yes	AddRecord
<code>GSW_AGENT_ID</code>	String	No	Login ID of last agent who worked with the record. Optional. (Default = 0)
<code>GSW_APPLICATION_ID</code>	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
<code>GSW_ATTEMPTS</code>	Int	No	Number of attempts for the record. This key is used when a new record is added. Optional. (Default = 0)
<code>GSW_CALL_RESULT</code>	Int	No	Call Result sent to change automatically detected call result. (See the “Genesys Enumeration Table” on <a href="#">page 24</a> .)
<code>GSW_CALL_TIME</code>	String	No	System time when record was called, in seconds from 1/1/70 (GMT). Optional. (Default = 0)
<code>GSW_CALLING_LIST</code>	String	Yes	Name of the calling list.
<code>GSW_CAMPAIGN_GROUP_NAME<sup>a</sup></code>	String	No	Name of the Campaign Group
<code>GSW_CAMPAIGN_NAME<sup>b</sup></code>	String	Yes	Name of the Campaign.

**Table 82: AddRecord Attached Data (Continued)**

Data Key	Type	Key Required	Description
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, in the record's Time Zone. (Default = 0)
GSW_FROM	Int	No	GSW_FROM - GSW_UNTIL: Time frame when a record can be called, seconds from midnight. (Default = 28800, which represents 8 AM)
GSW_PHONE	String	Yes	Customer's phone number.
GSW_PHONE_TYPE	Int	No	Customer phone type (See the "Genesys Enumeration Table" on <a href="#">page 24</a> ). (Default = 2, DirectBusinessPhone)
GSW_RECORD_HANDLE <sup>b</sup>	Int	No	Unique Record Handle value for the record.
GSW_RECORD_STATUS	Int	No	Status of adding record sent from a desktop (See the "Genesys Enumeration Table" on <a href="#">page 24</a> ). Optional. (Default = 1, ready)
GSW_RECORD_TYPE	Int	No	Type of added record sent from a desktop. See the "Genesys Enumeration Table" on <a href="#">page 24</a> . Optional. (Default = 2, general)
GSW_REFERENCE_ID <sup>c</sup>	Int	No	Reference identifier for the request.
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.



- a. Adding this attribute to the request enables the identification of the Campaign Group for environments where several groups are configured, active, and running for the same Campaign. This attribute has a higher priority than the GSW\_CAMPAIGN\_NAME attribute.
- b. To handle a scenario in which several dialing sessions are active or running for the same Campaign and ensure that a new record is added to an existing chain for the appropriate group, OCS places a higher priority on processing the GSW\_RECORD\_HANDLE attribute if present in the request over the GSW\_CAMPAIGN\_NAME attribute. The GSW\_RECORD\_HANDLE attribute provides information to identify the Campaign Group, and with GSW\_CHAIN\_ID, enables a new record to be added correctly. In addition, if the GSW\_CHAIN\_ID does not match the ID of the chain, OCS returns error code 103.
- c. GSW\_REFERENCE\_ID is an optional attribute in the message. When present, OCS guarantees to return this attribute (same key and same value) in the response to the desktop request, in both a positive response or an error.

## AddRecordAcknowledge

OCS sends this event to the desktop to acknowledge an added record. [Table 83](#) contains more information.

**Table 83: AddRecordAcknowledge**

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

[Table 84](#) lists the attached data for an AddRecordAcknowledge event.

**Table 84: 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.
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.

### Smart Logout

The Smart Logout feature ensures that an agent is not logged out of a campaign until all outbound calls that OCS dialed based on the agent's availability are returned to the calling list or are completed by the agent (refer to “[Agent Logout Protocol](#)”). After the completion of Smart Logout (after OCS sends the LogoutAcknowledge message—refer to “[LogoutAcknowledge](#)” on [page 85](#)), agents are assigned to the Inbound activity.

## Agent Logout Protocol

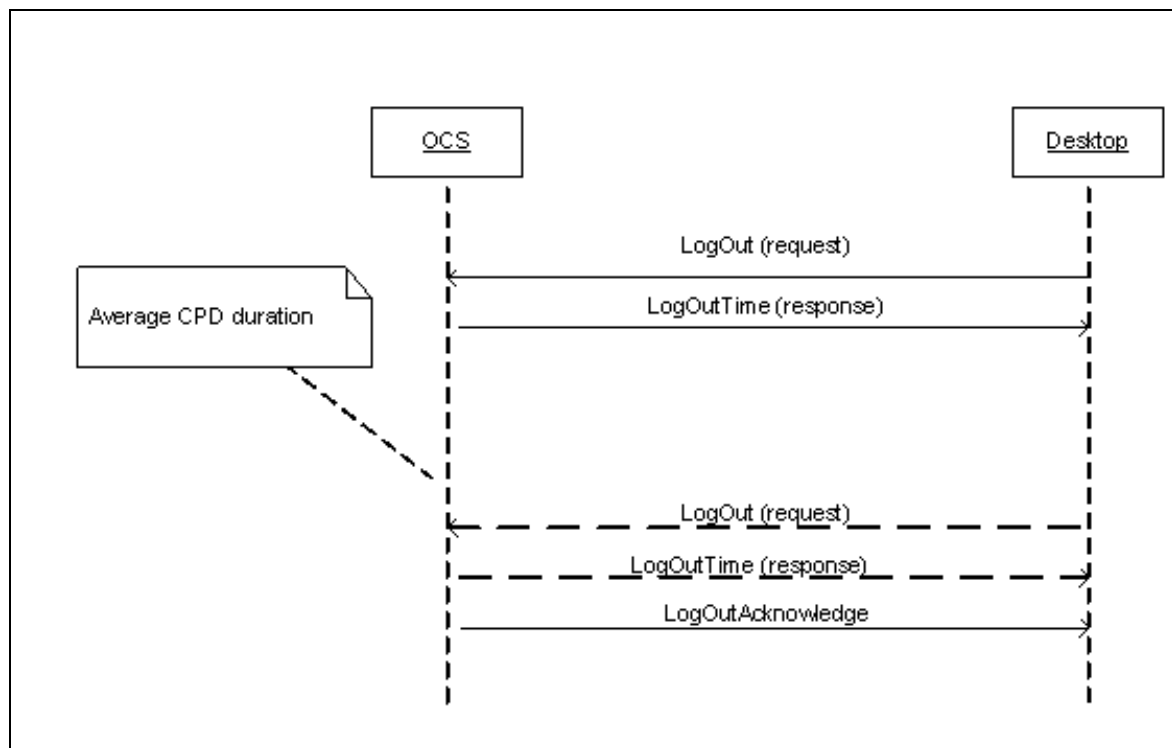
The extension of the Desktop Protocol (Logout) addresses the issue of abandoned or dropped calls as a result of this combination of circumstances:

- A significant number of Agents in a dialing session for 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 85](#). 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 85](#) provides information on this event.

**Table 85: 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 <code>call_wait_connected_timeout</code> 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 86 lists the attached data for the LogOut event.

**Table 86: 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_GROUP_NAME <sup>a</sup>	String	No	Name of the Campaign Group.
GSW_CAMPAIGN_NAME	String	Yes	Name of the Campaign.
GSW_REFERENCE_ID <sup>b</sup>	Int	No	Reference identifier for the request.

- a. Adding this attribute to the request enables the identification of the Campaign Group for environments where several groups are configured, active, and running for the same Campaign. This attribute has a higher priority than the `GSW_CAMPAIGN_NAME` attribute.
- b. `GSW_REFERENCE_ID` is an optional attribute in the message. When present, OCS guarantees to return this attribute (same key and same value) in the response to the desktop request, in both a positive response or an error.

## LogOutTime

OCS sends this response to the desktop for the agent's LogOut request. [Table 87](#) contains more information.

**Table 87: 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 88](#) lists the attached data for the LogOutTime event.

**Table 88: 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 89](#) provides information on this event.

**Table 89: LogOutAcknowledge**

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

[Table 90](#) lists the attached data for the LogOutAcknowledge event.

**Table 90: 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 91](#) 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 91: 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 92](#) provides information about identifying the media types that correspond to the Media Type business attribute. This defines how to contact the customer.

**Table 92: 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.

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.

---

Note: When using CPD Server in an ASM mode, set the `line-type` option to `isdn`, `isdn-dm3`, `cas-dm3`, or `sip-hmp-asm`.

---

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 8.1 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.
- OCS treats an outbound call as being transferred to an Unknown DN if, while a campaign is running in IVR mode, after being diverted to an IVR port, the outbound call is transferred to an agent who is not participating in the IVR campaign.
- 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/Campaign group. After this, OCS does not take dynamic changes of the option until unloading the Campaign/Campaign group.
- 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 about these options, see the “Outbound Contact Configuration Options” chapter in the *Outbound Contact 8.1 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 8.1 Deployment Guide*.

---

## Outbound Contact Library

The following section describes:

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

## Error Names and Codes

Table 93 displays error names and their corresponding codes for error conditions that occur while using communication protocols.

**Table 93: Error Names and Codes**

<b>GSW_ERROR</b>	<b>GSW_ERROR_NUMBER</b>	<b>Description</b>
Invalid Request	101	Received request has the wrong request type. <sup>a</sup>
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.
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.

**Table 93: Error Names and Codes (Continued)**

GSW_ERROR	GSW_ERROR_NUMBER	Description
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 94](#) 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 94: All Desktop Protocol Events and Event Type Protocols**

Messages	From > To	Descriptions and Actions
<b>1. Notifications</b>		
CampaignStarted	O > D	Should be sent when Campaign dialing is started or resumed, or as a response to event agent login if a dialing session for the Campaign is started.
CampaignStopped	O > D	Should be sent when dialing for Campaign is stopped or paused. All lists in Campaign deactivated.

**Table 94: All Desktop Protocol Events and Event Type Protocols (Continued)**

Messages	From > To	Descriptions and Actions
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 dialing session/Campaign Group(s) statuses.
<b>2. Preview</b>		
PreviewRecordRequest	D > O	Request to send preview record.
PreviewDialingModeStart	D > O	Request to activate preview session for the agent. Needed if the agent_preview_mode_start option is set to true.
PreviewRecord	O > D	Preview record to dial.
NoRecordsAvailable	O > D	No more records available.
<b>3. Common</b>		
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.

**Table 94: All Desktop Protocol Events and Event Type Protocols (Continued)**

Messages	From > To	Descriptions and Actions
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	OCS accepts RejectRecord request.
RecordCancelAcknowledge	O > D	Desktop sends a request to OCS to cancel a record or a chain.
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.
ScheduledCall	O > D	OCS sends this event to an agent to indicate that scheduled call should be executed.
ScheduledRecordReschedule	D > O	Request a reschedule of Preview Record, Predictive Call, or Scheduled Call when a rescheduled call cannot be completed and must be set for another time.
<b>4. Chained Records</b>		
ChainedRecordRequest	D > O	Request to send all records from the chain defined by Record Handle (Unique Record Identifier).
ChainedRecord	O > D	Request to send all records from the chain defined by RecordHandle.
ChainedRecordsDataEnd	O > D	All chain has been delivered.
<b>5. DoNotCall</b>		
DoNotCall	D > O	Agent requests the number or customer ID in a record not to be called again.

**Table 94: All Desktop Protocol Events and Event Type Protocols (Continued)**

Messages	From > To	Descriptions and Actions
DoNotCallAcknowledge	O > D	Confirmation that DoNotCall was accepted.
<b>6. Record Cancel from OCS to Desktop</b>		
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.
<b>7. Request add record from Desktop</b>		
AddRecord	D > O	Request to add a new record to the database.
AddRecordAcknowledge	O > D	Phone number can be used to relate request and response.
<b>8. Request LogOut</b>		
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, page 97](#)
- [Protocol Sequencing, page 98](#)
- [User Event Structure, page 102](#)
- [User Data Enumeration Values, page 103](#)
- [Record Cancellation from a Third-Party Application, page 110](#)
- [DoNotCall Requests from a Third-Party Application, page 114](#)

---

### Overview

Outbound Contact provides a Communication DN (CommDN) API that enables 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 open APIs.

The Communication DN API also enables users to control campaigns and campaign sequences through third-party applications or scripts instead of OCM or Genesys Administrator. 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 Genesys Administrator.
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 Genesys Administrator.
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 disregarded.

## 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 103](#). The communication is based on two types of messages: Request-Response and unsolicited notification.

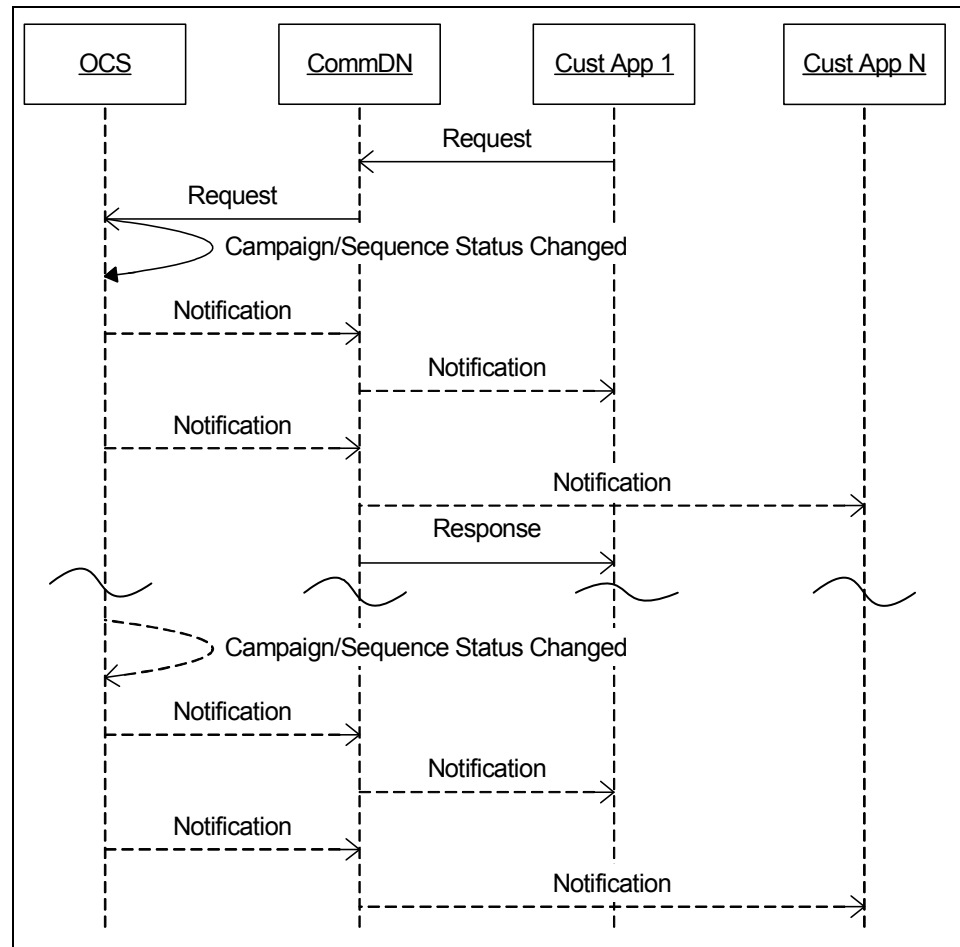
---

## 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 11](#).

- `TargetAppDBID` (the DBID of the receiver)

Table 95 on [page 100](#) 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 95: Communication Structure**

Request	Response/Notification	Mandatory Attributes
CM_ReqLoadCampaign	CM_EvCampaignLoaded	<ul style="list-style-type: none"> <li>• CampaignDBID or Properties</li> </ul> <p>OCS checks CampaignDBID. If the value is 0, the request or event must have the proper schedule in the Properties attribute.</p> <ul style="list-style-type: none"> <li>• GroupDBID</li> <li>• DialMode</li> <li>• OptimizeBy</li> <li>• OptimizeGoal</li> </ul>
CM_ReqUnloadCampaign	CM_EvCampaignUnloaded	<ul style="list-style-type: none"> <li>• CampaignDBID</li> <li>• GroupDBID</li> </ul>
CM_ReqGetCampaignStatus	CM_EvCampaignStatus	<p>Request</p> <ul style="list-style-type: none"> <li>• CampaignDBID or Properties</li> </ul> <p>OCS checks CampaignDBID. If the value is 0, the request or event must have the proper schedule in the Properties attribute.</p> <p>If CM_ReqGetCampaignStatus CampaignDBID equals 0, OCS responds with the sequence status.</p> <ul style="list-style-type: none"> <li>• GroupDBID</li> </ul> <p>Response or Notification</p> <ul style="list-style-type: none"> <li>• CampaignDBID or Properties</li> <li>• GroupDBID</li> <li>• DialMode</li> <li>• OptimizeBy</li> <li>• OptimizeGoal</li> <li>• GroupCampStatus. If a campaign belongs to a Sequence, then the attribute GroupCampStatus status represents the part of each scheduleItem (&lt;n&gt; in the Sequence. See Figure 7 on <a href="#">page 102</a>.</li> </ul>

**Table 95: Communication Structure (Continued)**

Request	Response/Notification	Mandatory Attributes
CM_ReqSetDialingMode	CM_EvDialingModeChanged	<ul style="list-style-type: none"> <li>• CampaignDBID or Properties</li> <li>• GroupDBID</li> <li>• DialMode</li> <li>• OptimizeBy</li> <li>• OptimizeGoal</li> </ul>
CM_ReqStartDialing	CM_EvDialingStarted	<ul style="list-style-type: none"> <li>• CampaignDBID or Properties</li> <li>• GroupDBID</li> <li>• DialMode</li> <li>• OptimizeBy</li> <li>• OptimizeGoal</li> </ul>
CM_ReqStopDialing	CM_EvDialingStopped	<ul style="list-style-type: none"> <li>• CampaignDBID or Properties</li> <li>• GroupDBID</li> </ul>
CM_ReqDoNotCall	CM_EvDoNotCallProcessed	<ul style="list-style-type: none"> <li>• Phone</li> <li>• CustomerID</li> </ul>
CM_ReqCancelRecord	CM_EvRecordCanceled or CMEvError	<ul style="list-style-type: none"> <li>• OriginAppDBID (the DBID of the sender)</li> <li>• TargetAppDBID (the DBID of the receiver)</li> <li>• Phone</li> </ul> <p>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.</p>

## 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 `outbound_contact_server` option to these DN. 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 8.1 Deployment Guide* for more information.

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

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.

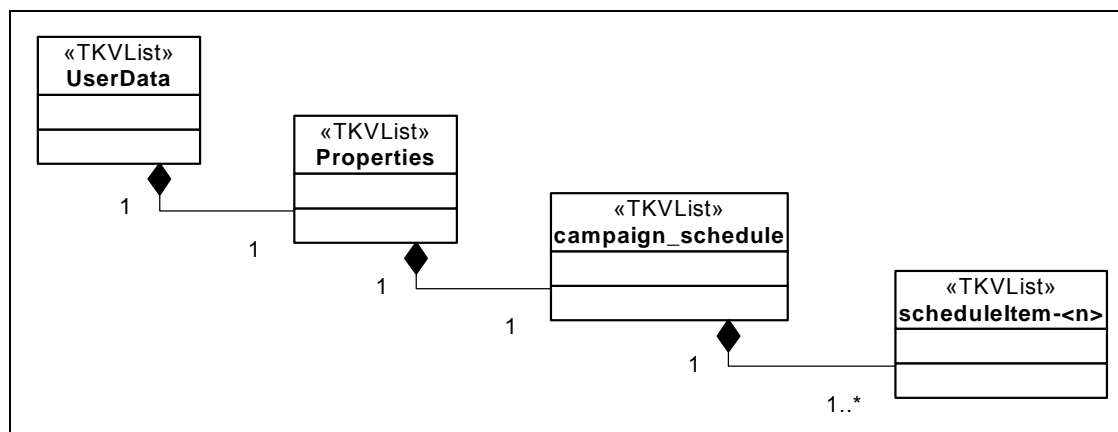


Figure 7: User Event Structure for the Communication DN API

---

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 98 on [page 104](#).

---

## 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 96](#) lists the Genesys user event attributes sent with user data through the Communication DN API.

**Table 96: User Event Attributes for User Data (TKVList)**

Key	Type	Description
GSW_CM_MessageType	Int	See the Enum values for GSW_CM_MessageType on <a href="#">page 106</a> .
GSW_CM_AttrDialMode	Int	See the Enum values for GSW_CM_AttrDialMode on <a href="#">page 109</a> .
GSW_CM_AttrOptimizeBy	Int	See GSW_CM_AttrOptimizeBy Enum values on <a href="#">page 110</a> .
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 <a href="#">page 110</a> .
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 <a href="#">page 108</a> .
GSW_CM_AttrErrorMessage	String	String describing the error that occurred.
GSW_CM_AttrOriginAppID	Int	Application DBID.

**Table 96: 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 97](#) shows UserEvent attributes for GSW\_CM\_AttrProperties (TKVList).

**Table 97: 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
dialing_priority	TKVList	Contains information about the dialing priority of specific record types.
do_not_call	TKVList	Contains additional request attributes

[Table 98](#) shows UserEvent attributes for campaign\_schedule (TKVList).

**Table 98: 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 99](#) shows UserEvent attributes for dialing\_priority (TKVList).

**Table 99: User Event Attributes for dialing\_priority (TKVList)**

Key	Type	Description
General	TKVList	Contains a list of the following key-value pairs: <ul style="list-style-type: none"> <li>· priority =&lt;value&gt;</li> <li>· n_records=&lt;value&gt;</li> </ul>



**Table 99: User Event Attributes for dialing\_priority (TKVList) (Continued)**

Key	Type	Description
CampaignRescheduled	TKVList	Contains a list of the following key-value pairs: <ul style="list-style-type: none"> <li>· priority =&lt;value&gt;</li> <li>· n_records=&lt;value&gt;</li> </ul>
CampaignCallBack	TKVList	Contains a list of the following key-value pairs: <ul style="list-style-type: none"> <li>· priority =&lt;value&gt;</li> <li>· n_records=&lt;value&gt;</li> </ul>

[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.
sleepBeforeNextStart	Int	The wait time, in minutes, before the start of this campaign/campaign group.
campaignDBID	Int	DBID of the campaign.
dialMode	Int	Dial mode for the campaign. See the Enum values for GSW_CM_AttrDialMode on <a href="#">page 109</a> .
optMethod	Int	Optimization method for the campaign. See the Enum values for GSW_CM_AttrOptimizeBy on <a href="#">page 110</a> .

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

Key	Type	Description
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 <a href="#">page 110</a> .

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
<b>Messages That OCM/Genesys Administrator Uses to Communicate with OCS</b>		
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
CM_UnknownMessage	5	not applicable
<b>Requests</b>		
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.

**Table 101: Data Enumeration Values for GSW\_CM\_MessageType (Continued)**

<b>Data</b>	<b>Value</b>	<b>Comment</b>
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.
<b>Responses</b>		
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.
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

**Table 101: Data Enumeration Values for GSW\_CM\_MessageType (Continued)**

Data	Value	Comment
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.
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/campaign group.
CM_ERROR_GROUP_NOT_FOUND	8	Requested group not found in configuration.

**Table 102: Enumeration Values for GSW\_CM\_AttrError (Continued)**

Error	Value	Comment
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
CFGDMProgressAndSeize	4	Progressive with engage mode
CFGDMPredictAndSeize	5	Predictive with engage mode
CFGDMPushPreview	8	Push Preview
CFGDMProgressGVP	9	Progressive GVP
CFGDMPowerGVP	11	Power GVP

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_NotLoaded	0	Status not loaded
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 (and optionally, by customer ID) 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\_EvRecordCancelled

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
  I
  UserData
    I
    "GSW_CM_MessageType" 30
    ["GSW_CM_AttrError" 0]
    "GSW_CM_AttrOriginAppID" <value>
    "GSW_CM_AttrTargetAppID" <value>
    "GSW_CM_AttrProperties"
      I
      "cancel_record"
        I
        "GSW_PHONE" <value>
        ["GSW_CAMPAIGN_NAME" <value>]
        ["GSW_CHAIN_ATTR" <value>]
        ["GSW_MESSAGE" Incomplete processing: record(s) on desktop]
```

---

Note: The user event might also include GSW\_CUSTOMER\_ID (an optional attribute) that you can add to a third-party cancellation request.

---

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

See Table 96, “User Event Attributes for User Data (TKVList),” on [page 103](#) and Table 6, “Reserved Keys,” on [page 18](#).

## 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 96](#) on [page 103](#).

**Table 106: UserData (TKVList)**

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

**Table 107: GSW\_CM\_AttrProperties (TKVList)**

Key	Type	Description
<code>cancel_record</code>	TKVList	Contains additional request attributes

**Table 108: cancel\_record (TKVList)**

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



**Table 108: cancel\_record (TKVList) (Continued)**

Key	Type	Description
GSW_CUSTOMER_ID	String	(Optional) A user-defined field in the Calling List table that serves as a customer identifier.
GSW_MESSAGE	String	OCS message (“Incomplete processing: record(s) on desktop”) notifying the RequestRecordCancel requester (agent desktop or third-party) about OCS’s inability to completely handle the cancellation request, because calls records are still active on an agent’s desktop. <b>Note:</b> This only affects cancellation by phone and customer ID. It does not affect RequestRecordCancel request made by the record handle or DoNotCall requests.

## 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. In addition, see [Table 101](#) on [page 106](#).

**Table 109: Data Enumerations**

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

**Table 110: GSW\_CM\_AttrError 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"
      I
      "do_not_call"
        I
        "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.

---



# 3

## OCS Support for the HTTP/HTTPS Protocol

This chapter describes a subset of the HTTP Protocol that is supported by OCS. OCS can receive requests compliant with the HTTP Protocol, process them, and return responses to clients. When doing so, OCS acts as the HTTP Server. The client that is seeking to contact OCS using the HTTP Protocol acts as an HTTP client. OCS can also process requests and generate responses over secure HTTP (HTTPS).

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

- [URI Supported by OCS, page 118](#)
- [OCS Resource Types Accessible via URI, page 119](#)
- [Client Requests, page 119](#)
- [HTTP Responses, page 125](#)
- [Guidelines for Client Connections, page 127](#)
- [Examples, page 127](#)

---

Note: OCS supports the HTTP Protocol only if the OCS Application object has an HTTP port configured. For more information, see the *Outbound Contact 8.1 Deployment Guide*.

---

### HTTPS Support

OCS supports communication over HTTPS, or strictly speaking, HTTP over Transport Layer Security (TLS) connection, using a Genesys TLS implementation.

For a detailed description of a Genesys TLS implementation, see the *Genesys Security Deployment Guide*, Part 3, “Server Integrity - Transport Layer Security”.

# URI Supported by OCS

This section describes the URI format, supported parameters, and how OCS delivers the URI to the client.

## Format

OCS supports the following format for the URI:

```
http://<OCS HTTP listener host name>:<listener port>/<path to OCS resource>?<parameters>
```

Where:

- <OCS HTTP listener host name>: The host name of the OCS HTTP listener.
- <listener port>: The listener port for the OCS HTTP listener.
- <path to OCS resource>: The path to the Outbound Contact resource in the following format:
  - /<resource type>/<resource identifier>

For example, /campaigngroups/<DBID of the Campaign Group>

Supported resource types include records, phones, customer IDs, Calling Lists, and Campaign Groups. For more information, see “OCS Resource Types Accessible via URI” on [page 119](#).

- <parameters>: The parameter portion of the URI defines the action for a resource; see the “[Supported Parameters](#)” section.

## Supported Parameters

[Table 111](#) describes the supported parameters.

**Table 111: Query Parameter**

Query Parameter Name	Query Parameter Value
req	The name of the action required for the resource, as defined by the Outbound Contact Desktop and Third Party Protocols. See <a href="#">Tables 113 through 116</a> for a list of supported query request/ operations. The format for this resource is as follows: req=<action>

## URI Delivery to the Client

If your OCS application has a configured HTTP port, then when a record is dialed in any dialing mode, OCS provides the client with the URI for that record in the GSW\_RECORD\_URI key. For example, for a record with a record handle of 15, the URI would be:

```
GSW_RECORD_URI = http://ocs.us.int.genesyslab.com:8080/records/15
```

OCS uses the IP address property of the host on which it is running to construct the host section of the URI or, if the host has no IP address configured, OCS uses the name of the host.

---

Note: In IPv6 deployments, you cannot set the IP address of the host—only IPv4 addresses can presently be set for the host. Therefore, *do not* try to specify the IP address of the host. OCS will use the host name in the URI.

---

---

## OCS Resource Types Accessible via URI

As mentioned in the “[Format](#)” section, the OCS resource types that can be accessed using the URI include the following:

- Records—You can request actions based on the record handle of the record, using the `records` resource.
- Phones—You can request actions based on the phone number, using the `phones` resource.
- Customer IDs—You can request actions based on the customer ID, using the `customer_ids` resource.
- Campaign Groups—You can request actions for a Campaign Group based on its configuration DBID, using the `campaigngroups` resource.
- Lists—You can request actions for a Calling List based on the name of that list, using the `lists` resource.

For details about how to specify each resource type in the URI, see “[Format](#)” on [page 118](#).

---

## Client Requests

OCS HTTP supports client requests using the `POST` method only. In the `POST` request, the client specifies the resource and action in the URI. Other information required for processing the request is specified in the `BODY` of the HTTP request.

### Client Request Headers

The `POST` request headers must be populated by the client in the `POST` request. Table 112 on [page 120](#) lists those headers. OCS HTTP ignores all other headers.

**Table 112: POST Request Headers**

Header Name	Value	Example
Host	OCS HTTP listener host/port	ocs.us.int.genesyslab.com:8080
User Agent	Name of the client that is connecting	GVP/8.1 Bank self-service #02
Content-type	MIME type	application/json <b>Note:</b> application/json is the only MIME type supported by OCS in HTTP requests.
Content-length	Payload (content) message length	77

## Client Request BODY

The BODY is the payload or content of the message, in the JavaScript Object Notation (JSON) format (MIME application/json type). All key-value pairs included in the BODY are packaged in this format. In addition, all pairs must conform to the Desktop or Third Party Protocol described in the following HTTP request sections.

---

Note: For more information about the JSON format, see <http://json.org>.

---

The message can contain a full or partial representation of the resource object.

- A full representation is needed when OCS has no prior information on the resource. For example, the full representation of an AddRecord request for a record resource is as follows:

```
{
  "GSW_PHONE": "567567567545656",
  "GSW_TZ_NAME": "PST",
  "GSW_CALL_RESULT": 28,
  "STATUS_CODE": "New",
  "CUSTOMER_STATUS": 5
}
```

- A partial representation is needed when an update is requested for an existing record and only updated fields can be delivered. For example, a partial request is as follows:

```
{
  "GSW_CALL_RESULT": 33,
  "STATUS_CODE": "Accepted",
  "CUSTOMER_STATUS": 7
}
```



In addition to the object representation, the BODY may include key-value pairs that control how a request is processed, per the Desktop Protocol. Some of these pairs include:

- GSW\_CHAIN\_ATTR = AllChain or RecordOnly
- GSW\_TREATMENT = RecordTreatPersonal or RecordTreatCampaign
- GSW\_TENANT\_DBID = <DBID>
- GSW\_CAMPAIGN\_NAME = <Campaign name>
- GSW\_TENANT\_NAME = <Tenant name>

---

Notes:

- The BODY can contain national alphabet symbols in the string values.
- For the HTTP interface, when GSW\_RECORD\_HANDLE, GSW\_PHONE, GSW\_CUSTOMER\_ID, and GSW\_CALLING\_LIST are used with the request for the same resource name, they have a lower priority than the parameters specified in the URI and should not be used in the JSON body.

---

## HTTP Requests for the records Resource

This section describes the actions and req values for the records resource and the content of the BODY.

### Actions and req Values

Table 113 provides a list of request actions available and their associated req values for the records resource. These actions are all inherited from the Desktop Protocol (see “Desktop Requests and OCS Responses” on [page 26](#)).

**Table 113: HTTP Request Actions and req Values for records Resource**

Action as Defined by the Outbound Contact Desktop Protocol	req Value	Note
UpdateCallCompletionStats	UpdateCallCompletionStats	
RecordProcessed	RecordProcessed	A Treatment is applied if GSW_TREATMENT= <Campaign or Personal> is specified in the body of the request.
RecordReject	RecordReject	
RequestRecordCancel	RequestRecordCancel	Can be used for the entire chain or one record.

**Table 113: HTTP Request Actions and req Values for records Resource (Continued)**

Action as Defined by the Outbound Contact Desktop Protocol	req Value	Note
DoNotCall	DoNotCall	Can be used for the entire chain or one record.
RecordReschedule	RecordReschedule	Non-finalizing for the Desktop Protocol, as this protocol requires RecordProcessed; finalizing in the HTTP Protocol.
AddRecord	AddRecord	Contains the handle of the record being processed, which caused an AddRecord action. Used to determine the target Campaign Group and Calling List. The original record must still be finalized.

## BODY

The BODY must follow the format described in “Client Request BODY” on [page 120](#).

## HTTP Requests for the phones and customer\_ids Resources

This section describes the request actions and req values for the phones and customer\_ids resources and the content of the BODY.

If the client wants to update the resource addressed by phone number or by Customer ID, it should explicitly specify the Tenant for which the operation is intended, by providing OCS with either the DBID or name of the Tenant in the message body, according to the Desktop Protocol. If both the DBID and name of the Tenant are provided the DBID takes precedence over the name.

### Actions and req Values

Table 114 on [page 123](#) provides a list of request actions and their associated req values available for the phones and customer\_ids resource. These actions are all inherited from the Desktop Protocol. (For information on this protocol associated with phones and customer\_ids for canceling records and placing Do Not Call requests, see “Canceling Records” on [page 59](#) and “Submitting DoNotCall Requests” on [page 66](#), respectively.)

**Table 114: HTTP Request Actions and req Values for the phones and customer\_ids Resources**

Action as Defined by Outbound Contact Desktop Protocol	req Value	Note
DoNotCall	DoNotCall	You must also specify either GSW_TENANT_NAME GSW_TENANT_DBID in the BODY of the request.
RequestRecordCancel	RequestRecordCancel	You must also specify either GSW_TENANT_NAME GSW_TENANT_DBID in the BODY of the request.  You can also specify whether the request is applicable for a specific Campaign only or all Campaigns.

## BODY

The BODY must follow the format described in “Client Request BODY” on [page 120](#)

## HTTP Requests for the lists Resource

This section describes the actions and req values for the lists resource and the content of the BODY.

### Actions and req Values

[Table 115](#) provides a list of request actions and their associated req values available for the lists resource. These actions are all inherited from the Desktop Protocol

**Table 115: HTTP Request Actions and req Values for the lists Resource**

Action as Defined by Outbound Contact Desktop Protocol	req Value	Note
AddRecord	AddRecord	No record handle is specified. Instead, the Calling List name and Campaign name are specified.

## BODY

The BODY must follow the format described in “Client Request BODY” on [page 120](#). In addition, it must comply with the following:

- Be a full JSON representation (see [page 120](#)).
- Contain the properties of the record being added (mandatory and user-defined), including the Campaign name (in the GSW\_CAMPAIGN attribute), phone number (in the GSW\_PHONE attribute) and time zone (in the GSW\_TZ\_NAME attribute).

## HTTP Request for the campaigngroups Resource

This section describes the request types and req values for the campaigngroups resource and the content of the BODY.

### Request and req Values

Table 116 on [page 124](#) provides a list of request types and their associated req values available for the campaigngroups resource. These actions are all inherited from the Communication DN Protocol.

**Table 116: HTTP Request Actions and req Values for the campaigngroups Resource**

Request Type (Per Communication DN Protocol)	req Value	Notes
GSW_CM_ReqLoadCampaign	Load	
GSW_CM_ReqStartDialing	StartDialing	Must have additional parameters in the body of the HTTP request (see <a href="#">Table 117</a> ).
GSW_CM_ReqSetDialingMode	SetDialingMode	Must have additional parameters in the body of the HTTP request (see <a href="#">Table 117</a> ).
GSW_CM_ReqStopDialing	StopDialing	
GSW_CM_ReqUnloadCampaign	Unload	
GSW_CM_ReqForceUnloadCampaign	ForceUnload	

## BODY

The BODY must follow the format described in “Client Request BODY” on [page 120](#). Table 117 on [page 125](#) also provides a list of attributes that can be included in the BODY of the request for the campaigngroups resource.

**Table 117: BODY Attributes for campaigngroups Resource**

Attribute	Value	Description
GSW_CM_AttrDialMode	Integer	The dial mode enumeration value (see Table 103 on <a href="#">page 109</a> ).
GSW_CM_AttrOptimizedBy	Integer	The optimization parameter type enumeration value (see Table 104 on <a href="#">page 110</a> ).
GSW_CM_AttrOptimizeGoal	Integer	The optimization parameter target value (see Table 96 on <a href="#">page 103</a> ).

**BODY Content Example**

```
{
  "GSW_CM_AttrDialMode": 1,
  "GSW_CM_AttrOptimizeBy": 1,
  "GSW_CM_AttrOptimizeGoal": 80
}
```

---

## HTTP Responses

OCS HTTP responses to client requests can be either positive or negative. Positive HTTP responses delivered by OCS to the client do not contain any message body. Negative responses may include a plain text or HTML error message in the response body.

Note: For all client requests to all resources but campaign groups, the error message contains the GSW\_ERROR attribute of the Desktop Protocol. For Campaign Group requests, the error message contains the GSW\_CM\_AttrError attribute.

[Table 118](#) summarizes those responses for all resources but Campaign Groups.

**Table 118: OCS HTTP Responses to Records, Phones, Customer IDs, Lists-Related Requests**

Response Type	Sent When
200 OK	Request executed successfully
400 Bad Request	Incorrect/unknown request
410 Gone	Record was not found
500 Internal server error	OCS internal processing error

[Table 119](#) summarizes the response for Campaign Group actions.

**Table 119: OCS HTTP Responses to Campaign Groups-Related Requests**

Response Type	Sent When	Notes
200 OK	Request executed successfully	
400 Bad Request	Incorrect/unknown request	Incorrect request or wrong resource for the request; for example, an attempt to load a record handle
404 Not Found	Campaign Group was not found	Unknown Campaign Group
409 Conflict	Request is incompatible with the resource state	An example of this is an attempt to load a Campaign Group that is already loaded
410 Gone	The Campaign Group was not found	The Campaign Group exists in the configuration but is not active.
500 Internal server error	OCS internal processing error	

[Table 120](#) summarizes the response for list/AddRecord requests.

**Table 120: OCS HTTP Responses to lists/AddRecord Requests with No Specified Record Handle**

Response Type	Sent When	Notes
200 OK	Request executed successfully	
400 Bad Request	Incorrect/unknown request	Incorrect request or wrong resource for the request; for example, an attempt to load a record handle
404 Not Found	Specified Campaign or Calling List was not found	Unknown Campaign Group
409 Conflict	Record could not be added	An example of this is when DB Server is disconnected or DBMS has returned an error in response to the INSERT SQL request.
500 Internal server error	OCS internal processing error	

Table 121 on [page 127](#) lists the headers in the OCS HTTP response that may be populated by OCS for all requests.

**Table 121: OCS HTTP Response Headers**

Header Name	Value	Example	Notes
Server	Name and version information for the server	OCS_Alpha/8.1.000.07	For all response types
Warning	Error ID per the Desktop Protocol	112	For negative response types
Content-type	MIME type	text/plain or text/html	For negative response types
Content-length	Text error message length	14	For negative response types

---

## Guidelines for Client Connections

OCS HTTP supports both direct connections from clients and connections through a proxy (for example, Squid proxy). If a proxy is involved, it is assumed that the proxy is transparent to both the client and the server. The connection between OCS HTTP and the client is also persistent.

The client must open a TCP connection to the HTTP Server when it receives a URI for the resource and a connection is not opened yet. HTTP Server will not close the connection to the client unless the connection closes by itself (network disruption) or the server is switched to a backup mode.

HTTP Server ignores the `Connection: close` header of the client request. For performance purposes, the client should not close the connection to the HTTP Server at will, especially upon processing each resource-related request.

---

Note: OCS HTTP does not support pipelining requests (that is, a client sending multiple requests without waiting for each response before sending the next request).

---



---

## Examples

This section contains examples of successful and unsuccessful requests via HTTP.

## A Successful RecordProcessed Request

For a successful RecordProcessed request from the client, the processing flow is as follows:

1. The client sends the following request to OCS via the HTTP interface:

```
POST http://ocs.us.int.genesyslab.com:8080/records/15?req=RecordProcessed HTTP/1.1
Host: ocs.us.int.genesyslab.com:8080
User-Agent: GVP/8.1 Banking self-service #2
Content-type: application/json
Content-length: 54
```

```
{
"CUSTOMER_CODE": 22, "GSW_CALL_RESULT": 33,
"DATE_LAST_SERVED": "10/30/2008"
}
```

2. OCS HTTP replies with the following message:

```
HTTP/1.1 200 OK
Server: OCS/8.1.000.07
Content-length: 0
```

## An Unsuccessful RecordProcessed Request

For an unsuccessful RecordProcessed request from the client, the processing flow is as follows:

1. The client sends the following request to OCS via the HTTP interface:

```
POST http://ocs.us.int.genesyslab.com:8080/records/15?req=RecordProcessed HTTP/1.1
Host: ocs.us.int.genesyslab.com:8080
User-Agent: GVP/8.1 Banking self-service #2
Content-type: application/json
Content-length: 54
```

```
{
"CUSTOMER_CODE": 22, "GSW_CALL_RESULT": 33,
"DATE_LAST_SERVED": "10/30/2008"
}
```

2. The identified record was already deleted due to a stale timeout.
3. OCS HTTP replies with the following message:

```
HTTP/1.1 410 Gone
Server: OCS/8.1.000.07
Warning: 112
Content-type: text/html
Content-length: 35
```

```
<HTML><body>410 Gone : 0 No call found for the record
handle</body></HTML>
```



## A Successful DoNotCall Request for a Specific Phone Number

For a successful DoNotCall request for a specific phone number from the client, the processing flow is as follows:

1. The client sends the following request to OCS via the HTTP interface:

```
POST http://ocs.us.int.genesyslab.com:8080/phones/4155555555?req=DoNotCall HTTP/1.1
Host: ocs.us.int.genesyslab.com:8080
User-Agent: ORS Server strategy #2
Content-type: application/json
Content-length: 35
```

```
{
  "GSW_TENANT_NAME": "Alpha Tenant"
}
```

2. OCS replies with the following message:

```
HTTP/1.1 200 OK
Server: OCS/8.1.000.07
Content-length: 0
```

## Successful AddRecord Request of the General Type

For a successful AddRecord request (General record type) from the client, the processing flow is as follows:

1. The client sends the following request to OCS via the HTTP interface:

```
POST http://ocs.genesyslab.com:8080/lists/Alpha%20List?req=AddRecord HTTP/1.1
Host: ocs.genesyslab.com
User-Agent: Genesys Orchestration/8.0.000.15 ORS Strategy #1
Content-type: application/json
Content-length: 84
```

```
{
  "GSW_CAMPAIGN_NAME": "Alpha Campaign",
  "GSW_PHONE": "4155670000",
  "GSW_TZ_NAME": "PST"
}
```

2. If the specified Campaign is active or running and DBMS returned a positive response to the INSERT SQL statement, OCS HTTP replies with the following message:

```
HTTP/1.1 200 OK
Server: OCS/8.1.000.19
Content-length: 0
```

## Successful AddRecord Request of the Campaign Rescheduled Type

For a successful AddRecord request (Campaign Rescheduled record type) from the client, the processing flow is as follows:

1. The client sends the following request to OCS via the HTTP interface:

```
POST http://ocs.genesyslab.com:8080/lists/Alpha%20List?req=AddRecord HTTP/1.1
Host: ocs.genesyslab.com
User-Agent: Genesys Orchestration/8.0.000.15 ORS Strategy #2
Content-type: application/json
Content-length: 208
```

```
{
  "GSW_CAMPAIGN_NAME": "Alpha Campaign",
  "GSW_PHONE": "4155670000",
  "GSW_PHONE_TYPE": 4,
  "GSW_TZ_NAME": "PST",
  "GSW_RECORD_TYPE": 6,
  "GSW_DATE_TIME": "12/12/2009 13:00",
  "CUSTOMER_CODE": 22,
  "DATE_LAST_SERVED": "10/30/2008"
}
```

2. If the specified Campaign is active or running and DBMS returned a positive response to the INSERT SQL statement, OCS HTTP replies with the following message:

```
HTTP/1.1 200 OK
Server: OCS/8.1.000.19
Content-length: 0
```

## Successful Load of a Campaign Group

For a successful Load request from the client, the processing flow is as follows:

1. The client sends the following request to OCS using the HTTP interface:

```
POST http://ocs.us.int.genesyslab.com:8080/campaigngroups/106?req=Load HTTP/1.1
Host: ocs.us.int.genesyslab.com:8080
User-Agent: Genesys Orchestration/8.0.000.15 Banking administration #3
Content-length: 0
```

2. If the specified Campaign Group is present in configuration and successfully loaded by OCS, OCS HTTP replies with the following message:

```
HTTP/1.1 200 OK
```

```
Server: OCS/8.1.000.12
Content-Length: 0
```

## Setting Dialing Parameters for a Campaign Group

For a request to set dialing parameters for a Campaign Group from the client, the processing flow is as follows:

1. The client delivers the following request to OCS via the HTTP interface:  
 POST http://ocs.genesyslab.com:8080/campaigngroups/106?req=SetDialingMode HTTP/1.1  
 Host: ocs.genesyslab.com:8080  
 User-Agent: Genesys Orchestration/8.0.000.15 Banking administration #3  
 Content-type: application/json  
 Content-length: 83

```
{
  "GSW_CM_AttrDialMode": 1,
  "GSW_CM_AttrOptimizeBy": 2,
  "GSW_CM_AttrOptimizeGoal": 5
}
```

2. If the specified Campaign Group is running within OCS, the new dial mode transition is allowed, and the specified optimization parameters are valid, OCS HTTP replies with the following message:

```
HTTP/1.1 200 OK
Server: OCS/8.1.000.12
Content-Length: 0
```

## Unsuccessful Load of a Campaign Group

For a request to load from the client, the processing flow is as follows:

1. The client sends the following request to OCS via the HTTP interface:

```
POST http://ocs.us.int.genesyslab.com:8080/campaigngroups/106?req=Load HTTP/1.1
Host: ocs.us.int.genesyslab.com:8080
User-Agent: Genesys Orchestration/8.0.000.15 Banking administration #3
Content-length: 0
```

2. If Campaign@Alpha Agent Group Campaign Group is already loaded in OCS, OCS HTTP replies with the following negative response:

```
HTTP/1.1 409 Conflict
Server: OCS/8.1.000.12
Warning: 5
Content-length: 0
```



# 4 Defined Constants

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

- [Field Definitions and Enumeration Values, page 133](#)
- [Call Results, page 135](#)
- [Data Types, page 141](#)
- [Contact Information Types, page 141](#)
- [Record Types, page 142](#)
- [Record Statuses, page 143](#)
- [Combining Record Statuses and Record Types, page 146](#)

---

## Field Definitions and Enumeration Values

Some Genesys mandatory fields in a Calling List table are represented as predefined integer constants, called enumeration values. The actual enumeration values are provided in the `cfg_locale` table in the Configuration Database.

This section provides field definitions. For enumeration values, see Table 7, “Enumeration Table,” on [page 24](#).

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

[Table 122](#) provides a description of the Genesys mandatory fields in the Default Outbound Contact format.

**Table 122: Outbound Contact Mandatory Fields in the Default Format**

Column Name	Data Type	Description
<code>record_id</code>	integer	Unique identification number of a calling record.

**Table 122: Outbound Contact Mandatory Fields in the Default Format (Continued)**

Column Name	Data Type	Description
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 127, "Contact Information Types," on <a href="#">page 141</a> .
record_type	integer	Type of the record. See Table 128, "Record Types," on <a href="#">page 142</a> .
record_status	integer	Current status of the record. See Table 129, "Record Statuses," on <a href="#">page 144</a> .
call_result	integer	Final outcome of the record processing. See Table 123, "Call Result Types," on <a href="#">page 135</a> .
attempt	integer	Number of attempts made to reach the customer.
dial_sched_time	integer	Date and time (in the record's Time Zone) 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 to which the record belongs.
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.
group_id	integer	Empty, not used at this time.

**Table 122: Outbound Contact Mandatory Fields in the Default Format (Continued)**

Column Name	Data Type	Description
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 <i>Outbound Contact 8.1 Deployment Guide</i> .

---

## 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 Genesys Administrator or Outbound Contact Manager. Call results can also be used to select treatments. For example, if a busy signal is reached, then the record can be marked for redialing at a later time. Call results for selecting treatments are chosen in the Outbound Contact Wizard or from the Call Result drop-down list for the Treatment object in Genesys Administrator or Outbound Contact Manager.

---

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

---

[Table 123](#) shows the call result types.

**Table 123: Call Result Types**

Call Result	Enumeration Value	Description
Abandoned	21	Call dropped by the customer, while waiting in queue or on routing point; or calls that were dropped by the customer while the call was ringing on the agent’s DN.
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.
Answer	33	Customer was reached at the dialed phone number.

**Table 123: Call Result Types (Continued)**

Call Result	Enumeration Value	Description
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. This call result might have also been assigned to a dial attempt in scenarios where a pre-connect CPA finished with a contradictory result such as “no ringback” or “positive voice”, or CPD Server was not able to recognize certain elements in ISDN or SIP messages received when the call disconnected.
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: <ul style="list-style-type: none"> <li>• <code>call_wait_in_queue_timeout</code></li> <li>• <code>call_wait_original_establish_timeout</code></li> </ul>
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.



**Table 123: Call Result Types (Continued)**

Call Result	Enumeration Value	Description
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.
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 in which 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.

**Table 123: Call Result Types (Continued)**

Call Result	Enumeration Value	Description
Silence	32	Call was dialed; however, there was no call progress indication.
SIT Detected	8	Any type of network tone.
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 <code>Stale</code> in the following scenario: <ol style="list-style-type: none"> <li>The following timer has expired: <code>stale_clean_timeout</code></li> <li>OCS marks the call result as <code>Stale</code> if an outbound call was transferred from: <ul style="list-style-type: none"> <li>a queue to a DN which is either not registered for OCS or does not have a logged in agent.</li> <li>an agent in an outbound campaign to a DN that is not registered for OCS.</li> </ul> </li> </ol>
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.

**Table 123: Call Result Types (Continued)**

Call Result	Enumeration Value	Description
Transfer Error	45	Dialer has a problem transferring calls based on call action.
Transferred	1	Reserved for future use.
Unknown Call Result	28	Default. All records should be set to this call result before starting a campaign/campaign group.
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 123](#), 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 123](#).) 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 123](#).

## 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 124 on [page 140](#) shows examples of Call Result Mappings created by OCS.

**Table 124: 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 125](#) 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 8.1 Deployment Guide* for more information about these options.

**Table 125: 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 125: 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 126](#) for data type values.

**Table 126: 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 127](#).

**Table 127: Contact Information Types**

Phone Type	Description
Business With Extension	Office number with extension

**Table 127: Contact Information Types (Continued)**

Phone Type	Description
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 7 on [page 24](#).

---



---

## Record Types

Record types show if a record is to be dialed, not dialed, or scheduled. See [Table 128](#).

**Table 128: 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 <code>predictive_callback</code> option.

**Table 128: Record Types (Continued)**

Record Type	Description
Campaign Rescheduled	Default value for a record that has been rescheduled by a call treatment. <b>Note:</b> If a record originally of the Callback type receives a busy or no answer result and has a treatment automatically applied, it retains the Callback type following the treatment. However, if such a Callback is explicitly rescheduled by an agent after connecting to the destination, the record takes the new type of Rescheduled.
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 that agent when the callback occurs.
Personal Rescheduled	Used by the agent to reschedule a call treatment and allow only that agent 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 7 on [page 24](#).

---



---

## 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 or Genesys Administrator. See Table 129 on [page 144](#).

**Table 129: 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 <code>predictive_callback</code> 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 129: Record Statuses (Continued)**

Record Status	Description
Stale	<p>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.</p> <p>The record status is marked as stale in the following scenarios:</p> <ol style="list-style-type: none"> <li>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.</li> <li>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.</li> <li>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.</li> </ol>
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 7 on [page 24](#).

---

# Combining Record Statuses and Record Types

Record statuses and record types can be combined as shown in [Table 130](#).

**Table 130: 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 130: Record Status and Record Type Combinations (Continued)**

<b>Record Status</b>	<b>Record Type</b>
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 7 on [page 24](#).

---



# 5

## Recommended DBMS Optimizations

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

- [Optimizing Calling List Tables for Weight Rules, page 149](#)
- [Temporary Tables Considerations, page 150](#)
- [Maintaining Indexes for Large Calling Lists, page 151](#)
- [Configuring IBM DB2 for Use of the LANGUAGE SQL Option, page 153](#)

---

### 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/campaign groups 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 (three to four thousand or more records in the calling list table), OCM or Genesys Administrator 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 or Genesys Administrator's GUI. To resolve this problem and to enable GUI 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 `dbospace` where temporary tables reside. You may need to increase this size. The `dbospace` 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 Genesys Administrator or 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 `<page_size>` 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 or Genesys Administrator 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 Genesys Administrator or OCM to disconnect if the application 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 two to three thousand).

---

Note: If you are using OCM, you can also 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 DBMS 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/campaign group, 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 Server 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.



---

## Configuring IBM DB2 for Use of the LANGUAGE SQL Option

IBM DB2 must be properly configured to provide the ability to create and execute Stored Procedures with the LANGUAGE SQL option in the CREATE PROCEDURE statement. Refer to the IBM Support Website or DB2 product documentation for additional details.



# 6

## Supported Functionality with IP Telephony

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

- [Overview, page 155](#)
- [Outbound Contact with SIP Server, page 159](#)
- [Outbound Contact with Cisco CallManager, page 172](#)
- [Outbound Contact with GVP 8.1 \(Proactive Contact Solution\), page 176](#)

---

### 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 SIP Server as a dialer and a VDN as the DN on behalf of which OCS originates calls with Genesys Media Server (a GVP component), which is able to perform Call Progress Analysis (CPA).
  - Using SIP Server as a dialer and Trunk Group DN as the DN on behalf of which OCS originates calls with Genesys Media Server to perform CPA in Transfer and Active Switching Matrix (ASM) modes.
- OCS, SIP Server, and GVP 8.1 together provide support for self-service campaigns that may or may not involve an agent. SIP Server performs dialing, Genesys Media Server provides CPA, and the interaction outcome is supplied to OCS by the VXML application using HTTP protocol requests. Power GVP and Progressive GVP modes are supported in this scenario.

---

Note: For an architectural description and configuration information about using Outbound Contact in a VoIP environment, see the *Outbound Contact 8.1 Deployment Guide*.

---

Power GVP dialing mode supports the Proactive Contact Solution. For more information on the Proactive Contact Solution, see the *Proactive Contact 8.0 Solution Guide*, which is located on the Genesys Technical Support web site and the Genesys Documentation Library DVD.

For backwards compatibility, Outbound Contact also continues to support use of CPD Server and a Dialogic board.

- Using CPD Server/Dialogic board as a dialer in a Transfer mode.
- Using CPD Server/Dialogic board as a dialer in the ASM mode.
- Using CPD Server/Dialogic's Host Media processing (HMP) software as a dialer.
- 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 CPD Server/Dialogic board in the ASM mode only.
  - Using CPD Server/HMP in the ASM or Transfer modes.

Figure 8 on [page 157](#) and Figure 9 on [page 158](#) provide an overview of each of these scenarios.

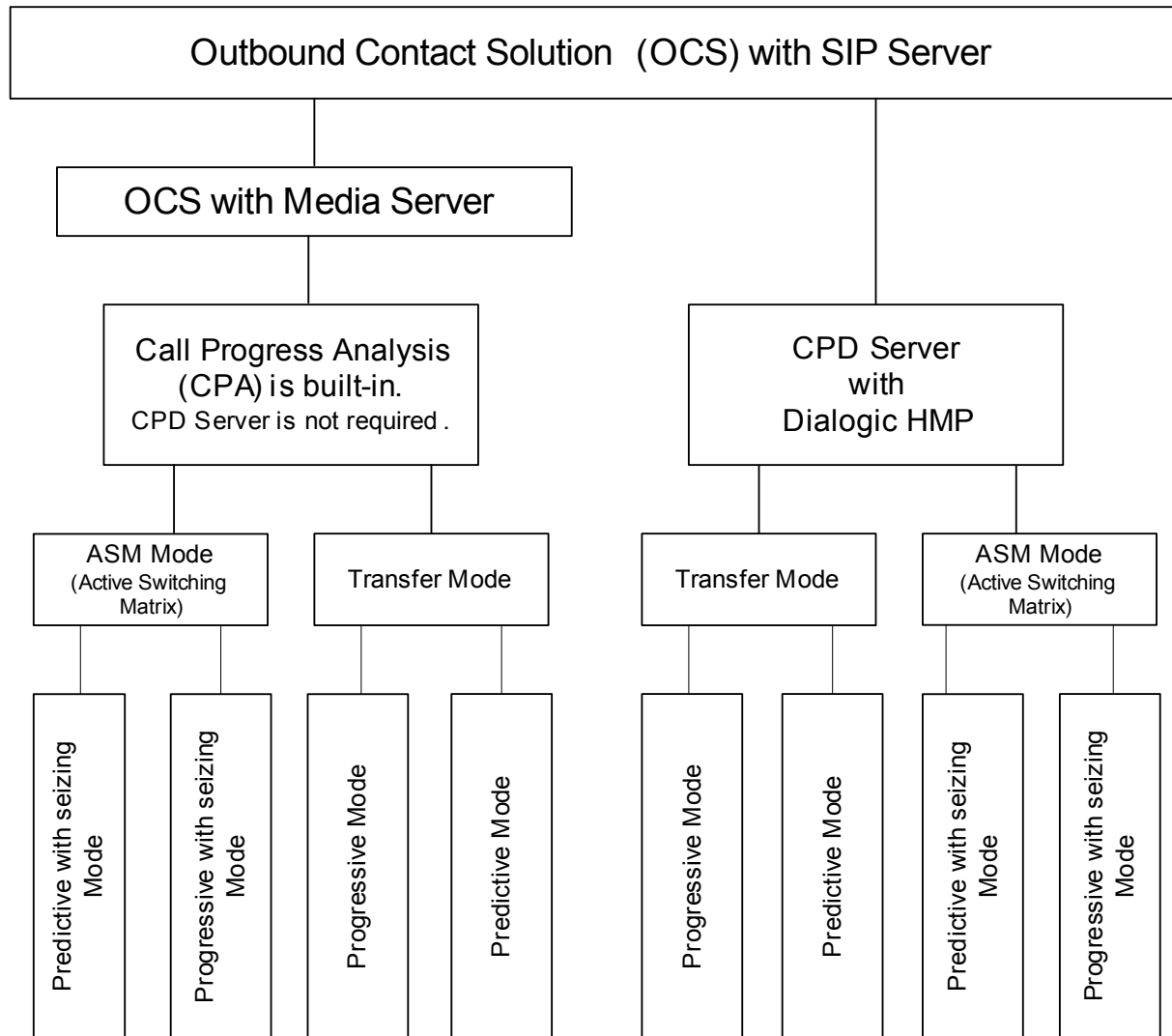
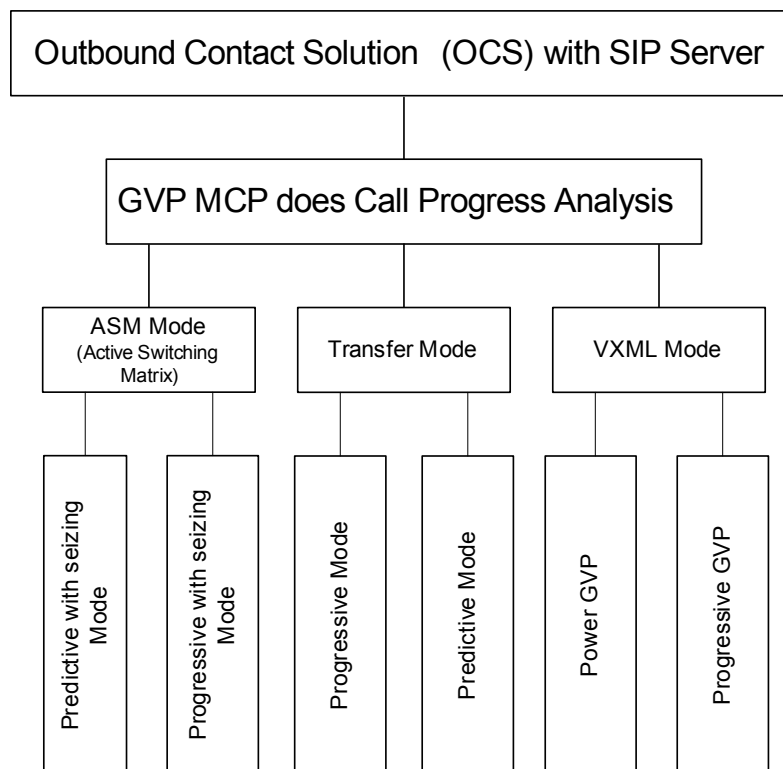


Figure 8: OCS with SIP Server Overview with Media Server or CPD Server



**Figure 9: OCS with SIP Server Overview with GVP MCP**

The following sections describe predictive and progressive dialing mode scenarios for a VoIP environment. 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.

---

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

Please consult with your media gateway 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.

---

---

# Outbound Contact with SIP Server

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

- “Transfer Mode (MGW with CPA)” on [page 159](#)
- “Transfer Mode (MGW without CPA)” on [page 161](#)
- “Transfer Mode (MCP as the CPA Provider)” on [page 164](#)
- “ASM Mode (MCP as the CPA Provider)” on [page 166](#)
- “ASM Mode (MGW with CPA)” on [page 167](#)
- “ASM Mode (MGW without CPA)” on [page 167](#)

---

Note: This section provide detailed descriptions of supported calling scenarios involving Outbound Contact in a VoIP environment. In general, there are no differences in supported scenarios for OC dialing in both ASM and Transfer modes with CPA, regardless of where CPA is done, either on MCP or on MGW.

---

## 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

---

Note: In this scenario, you can also use GVP 8.1 Media Control Platform (MCP) to handle CPA in place of MGW.

---

Figure 10 on [page 160](#) illustrates a Transfer-mode call flow in VoIP environment that involves an MGW with CPA and a SIP agent endpoint.

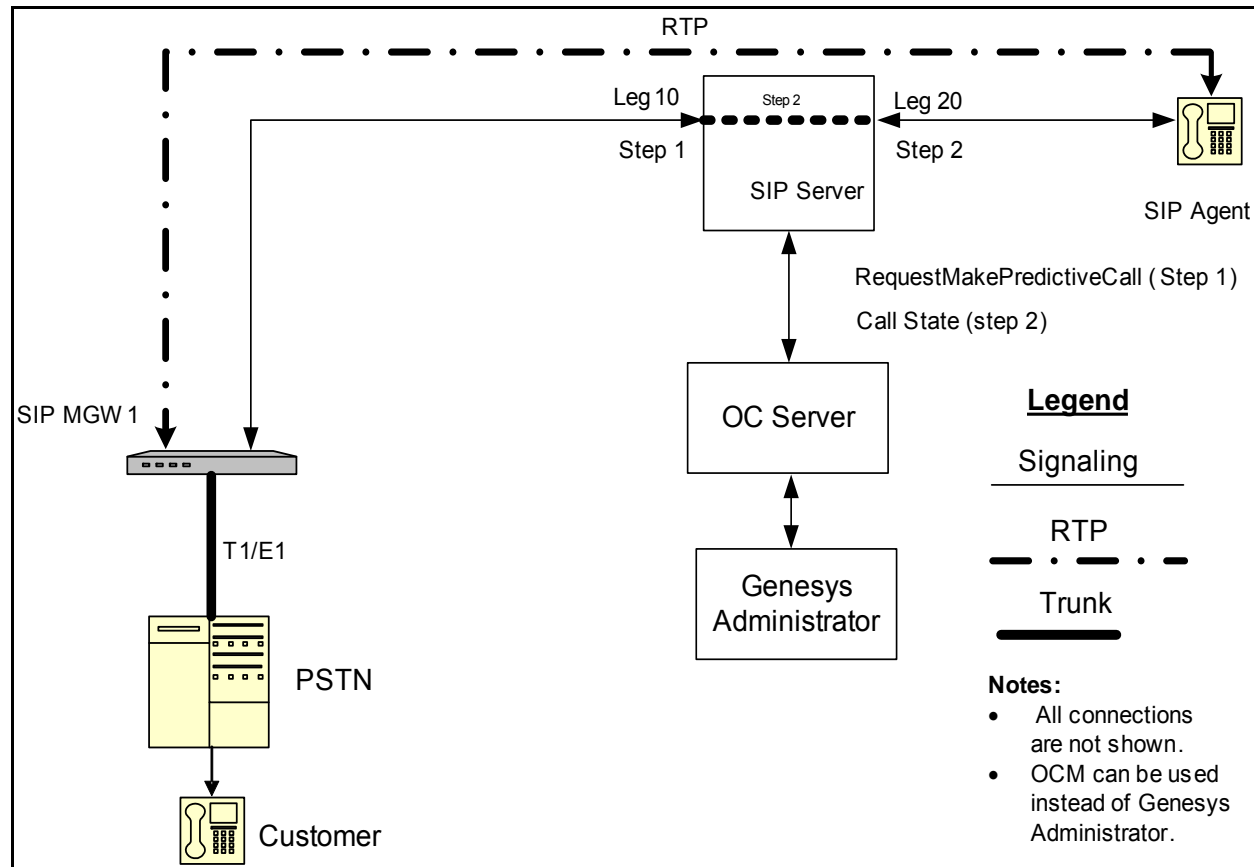


Figure 10: Transfer-Mode Call Flow (MGW with CPA) with a SIP Agent Endpoint

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.





5. SIP Server resolves the destination endpoint address and, assuming that this address matches the trunk address on MGW 2, creates leg 20 of a call by sending INVITE to MGW 2 to establish a call with the customer's phone number. All media streams are now between the MGW1 endpoint and the customer.
6. SIP Server conferences call leg 10 and call leg 20.
7. CPD Server performs CPA and reports the call result to OCS.

---

Notes:

- When using CPD Server with Dialogic board connected to MGW the CPD Server `tsCall` option must be set to no.
- MGW must be configured to match a channel number from which it originates an outbound call with a DN assigned to the Dialogic channel selected by CPD Server.

---

## Step 2

8. CPD Server sends a request to SIP Server to initiate the transfer of the customer call to a SIP agent endpoint.
9. SIP Server creates and establishes call leg 30 with the SIP agent endpoint.
10. SIP Server joins call leg 20 and call leg 30. All media streams are now between the SIP agent endpoint and the customer.
11. 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.

Figure 12 on [page 163](#) illustrates a Transfer-mode call flow in one location in VoIP environment that involves a MGW without CPA and a SIP agent endpoint.

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



### 7. CPD Server performs CPA and reports the call result to OCS.

- 
- Notes:
- When using CPD Server with Dialogic board connected to MGW the CPD Server `tscal l` option must be set to `no`.
  - MGW must be configured to match a channel number from which it originates an outbound call with a DN assigned to the Dialogic channel selected by CPD Server.
- 

### Step 2

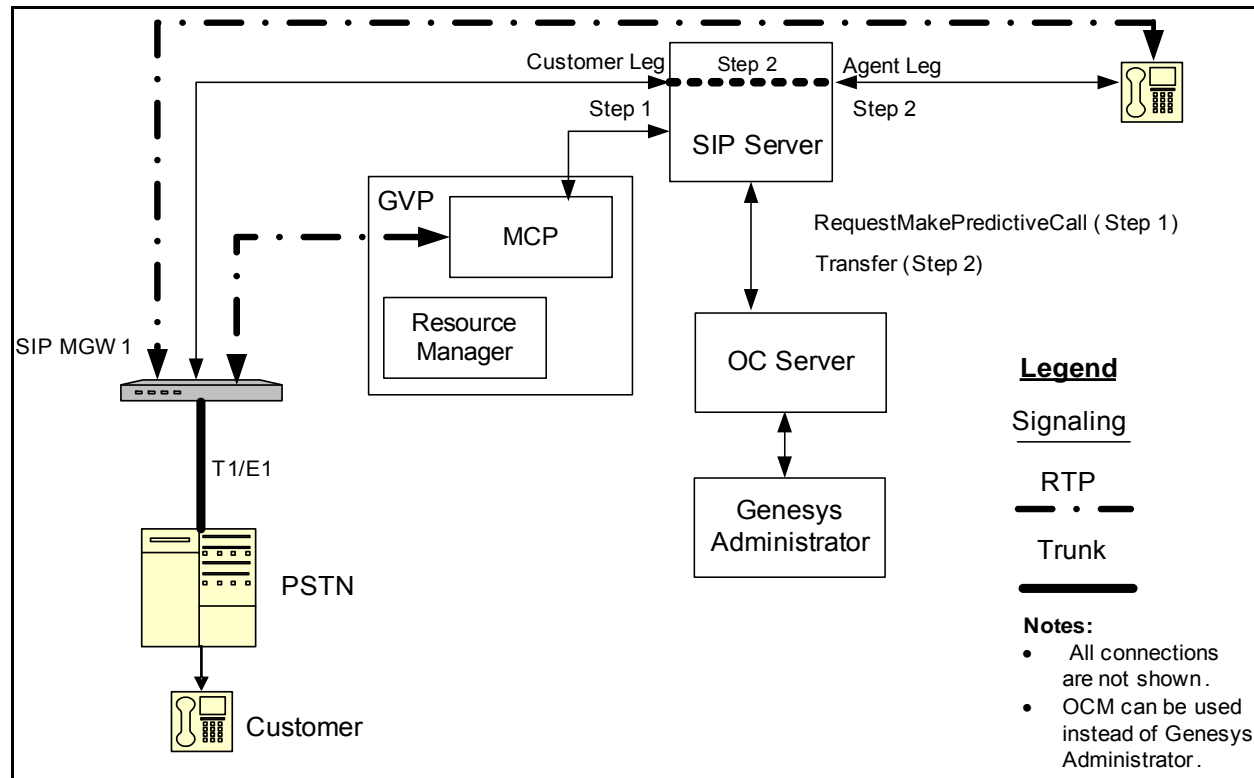
8. CPD Server sends a request to SIP Server to initiate the transfer of the customer call to a SIP agent endpoint.
9. SIP Server creates and establishes call leg 30 with the SIP agent endpoint.
10. SIP Server joins call leg 20 and call leg 30. All media streams are now between the SIP agent endpoint and the customer.
11. 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.

## Transfer Mode (MCP as the CPA Provider)

The following scenario describes a media flow that involves an GVP 8.1 MCP providing CPA abilities, and using SIP Server as the dialer for the Trunk Group DN.

Figure 13 on [page 165](#) illustrates a Transfer mode call flow in VoIP environment that involves MCP as the CPA provider and a SIP agent endpoint.



**Figure 13: Transfer-Mode Call Flow (MCP as the CPA Provider) 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 invites MCP/Genesys Media Server to handle CPA.
3. SIP Server creates the call leg with MCP and establishes a call with the customer DN.
4. MCP performs CPA and sends the call result to SIP Server.
5. SIP Server reports the call state to OCS by either `EventReleased` (for negative call results) or `EventEstablished` (for positive ones) on the Trunk Group DN.

#### Step 2

6. For a positive call result (`EventEstablished` received), OCS requests SIP Server to transfer the outbound call from the Trunk Group DN to Voice Transfer Destination DN.
7. SIP Server generates `EventQueued` and/or `RouteRequest` messages and establishes the call leg with a SIP agent endpoint.

8. When SIP Server joins the customer call leg to the agent call leg, all media streams are between the SIP agent endpoint and the customer.

## ASM Mode (MCP as the CPA Provider)

The following scenario describes a media flow that involves an GVP 8.1 MCP providing CPA abilities, and using SIP Server as the dialer for the Trunk Group DN.

Figure 14 illustrates a Transfer mode call flow in VoIP environment that involves MCP as the CPA provider and a SIP agent endpoint.

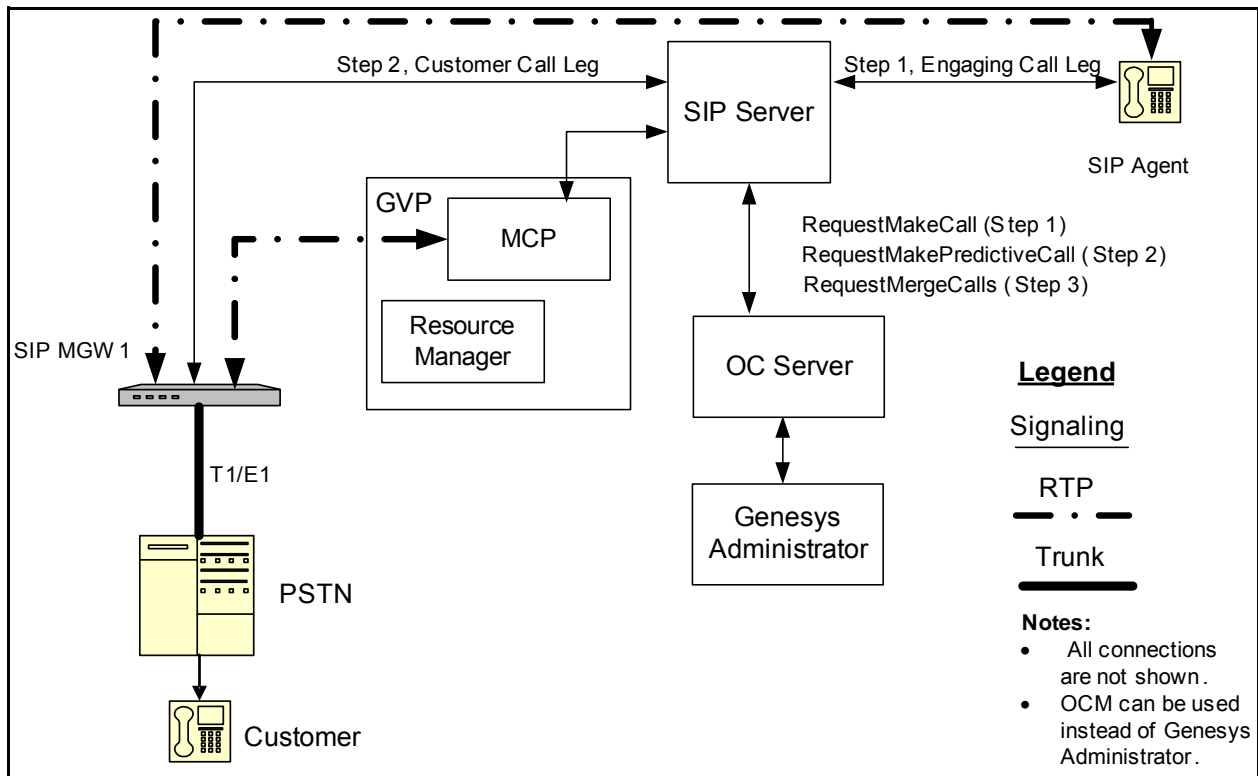


Figure 14: ASM-Mode Call Flow (MCP as the CPA Provider) with a SIP Agent Endpoint

In this scenario, the call flow proceeds as follows:

### Step 1

1. OCS sends a `RequestMakeCall` to SIP Server. This request contains the `'GSW_CALL_TYPE': 'ENGAGING'` key-value pair in its `AttributeUserData`, that identifies an engage call and the `AttributeOtherDN`, that is the Voice Transfer Destination DN.
2. SIP Server invites MCP/Genesys Media Server to handle the bridging of the engaging and customer legs and perform CPA on the customer leg.
3. SIP Server initiates an engaging call leg from the Trunk Group DN to the Voice Transfer Destination DN.

4. SIP Server generates an `EventQueued` and/or `RouteRequest` message and establishes the engaging call leg with a SIP agent endpoint.
5. The agent answers the engaging call, which generates an `EventEstablished` message (that includes the Genesys Media Server ID). The agent now waits for OCS and SIP Server to generate a second call leg to a calling list number.

### Step 2

6. OCS sends a `RequestMakePredictiveCall` message to SIP Server. This request contains `'GSW_CALL_TYPE': 'REGULAR'` key-value pair in its `AttributeUserData`, that identifies the customer call leg and `AttributeOtherDN`, that is the customer's DN.
7. SIP Server invites MCP/Genesys Media Server to handle CPA.
8. SIP Server creates call leg with MCP and establishes a call with the customer DN.
9. MCP performs CPA and sends the call result to SIP Server.
10. SIP Server reports the call state to OCS by either `EventReleased` (for negative call results) or `EventEstablished` (for positive ones) on the Trunk Group DN.

### Step 3

11. For a positive call result (`EventEstablished` received), OCS sends a `RequestMergeCalls` to SIP Server. This request contains both the engage and customer calls Connection IDs.
12. MCP connects the internal engaging and external customer call legs. The call is established between the agent and the customer.

---

Note: MCP bridges these legs by either merging them or transferring the agent to the customer call, depending on the configuration of the `merge-method` option. For more information on this option or on bridging calls, see the *Outbound Contact 8.1 Deployment Guide*.

---

## ASM Mode (MGW with CPA)

Supported scenarios are the same as in the previous section “ASM Mode (MCP as the CPA Provider)” on [page 166](#).

## ASM Mode (MGW without CPA)

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

Figure 15 illustrates an ASM-mode call flow in two locations in VoIP environment that involves a MGW without CPA and a SIP agent endpoint:

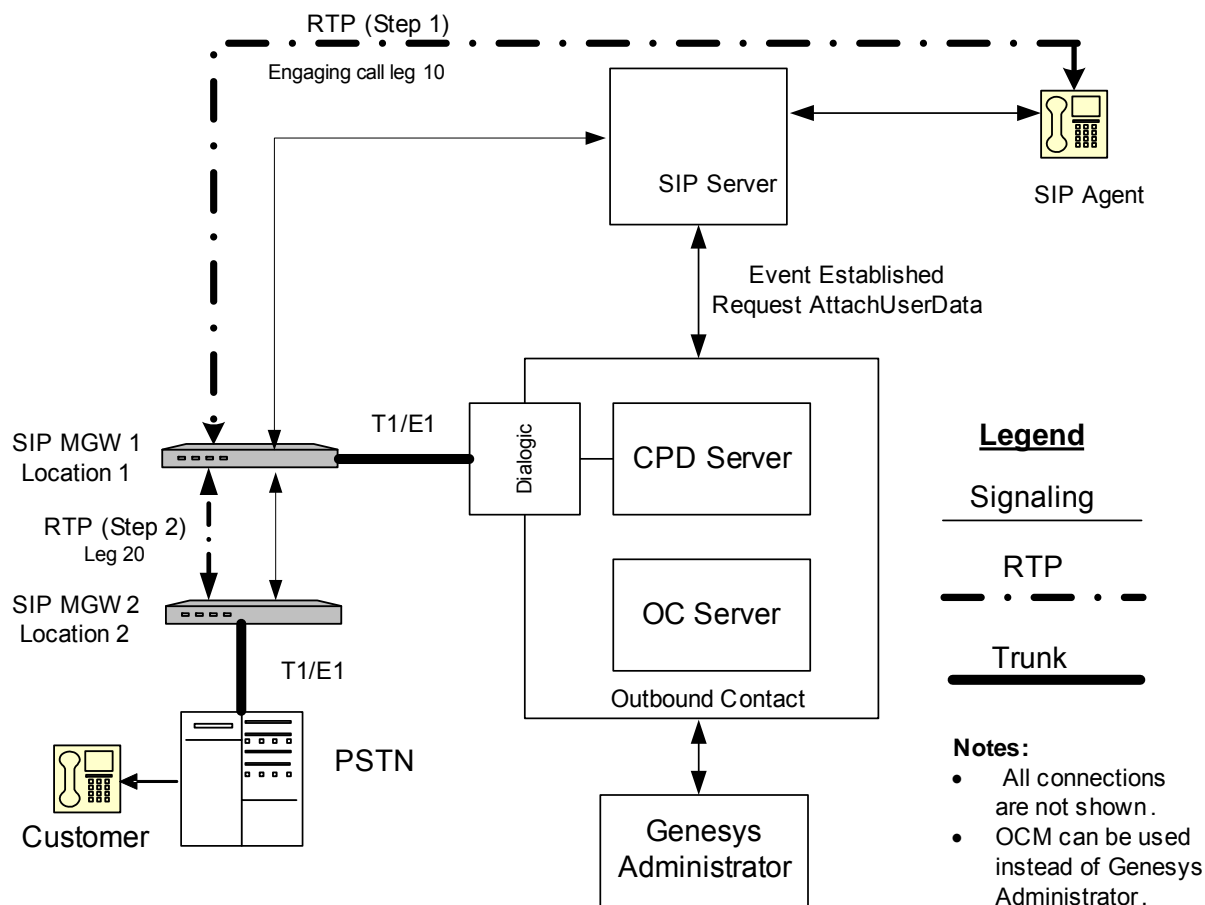


Figure 15: ASM-Mode Call Flow (MGW without CPA) in Two Locations with a SIP Agent Endpoint

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. OCS sends Req\_MakePredictiveCall to CPD Server. This request contains SAttr\_DialTo, which is the customer phone number.
7. CPD Server instructs the Dialogic board to place a call to the customer number that OCS provided.
8. If CPA has determined that there is a live voice, CPD Server attaches any customer data to the engage call (leg 20).
9. SIP Server delivers this data to the engaged agent's desktop as a screen pop.
10. CPD Server connects the call's internal and external leg. The call is established between the agent and the customer.
11. CPD Server informs OCS of the call result. The call is now handled as a normal outbound call.

Figure 16 illustrates an ASM-mode call flow in one location in VoIP environment that involves an MGW without CPA and a SIP agent endpoint.

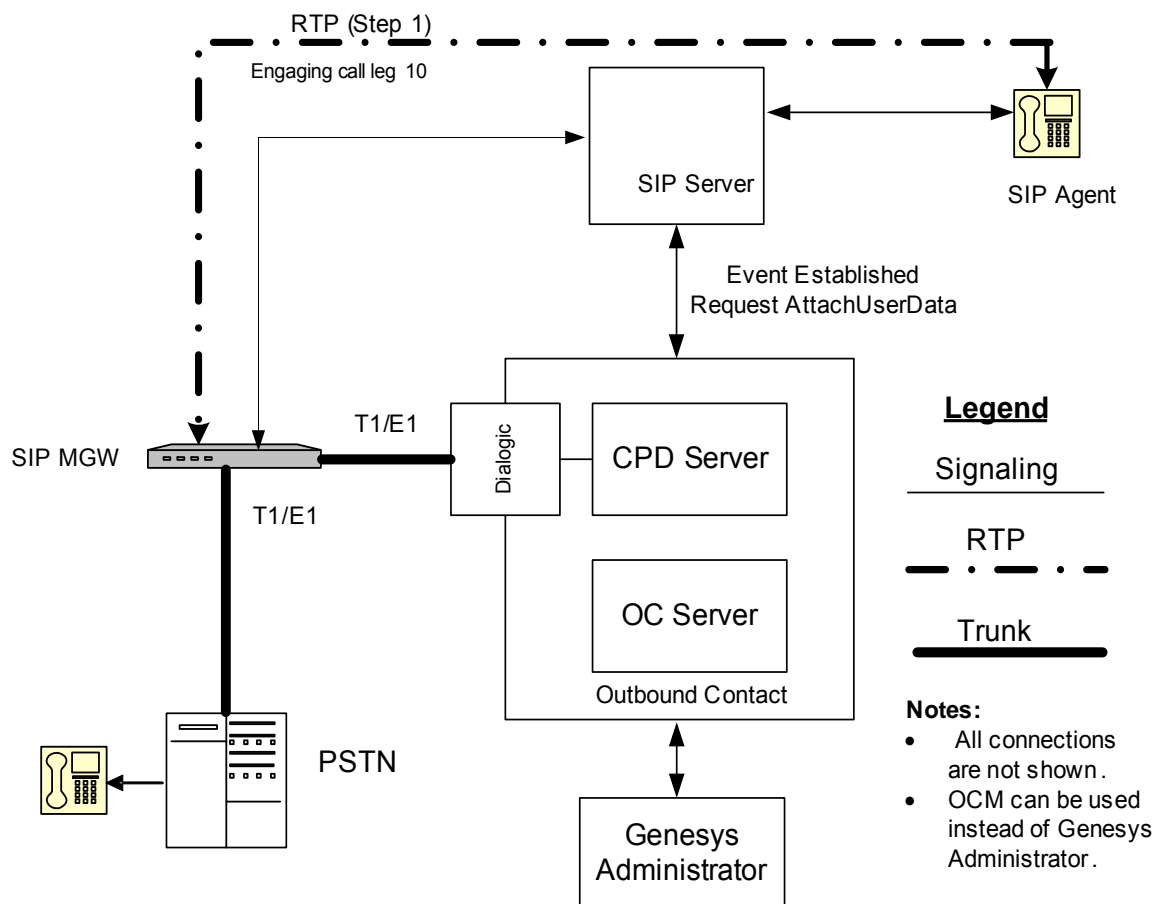


Figure 16: ASM-Mode Call Flow (MGW without CPA) in One Location with a SIP Agent Endpoint

In this scenario, the call flow proceeds as follows:

#### Step 1

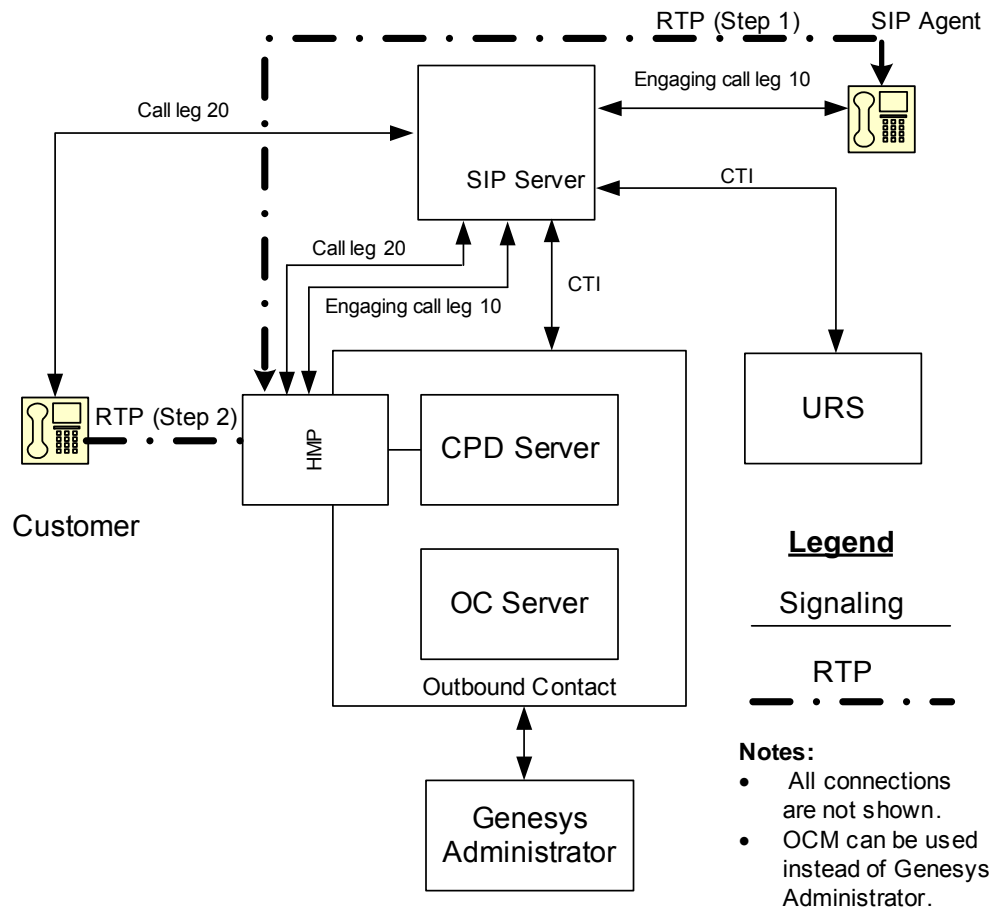
1. OCS sends an engage agent request to CPD Server.
2. CPD Server instructs the Dialogic board to create an engage call (leg 10) with an available agent's queue.
3. The engage call is queued, which generates an `EventQueued` message.
4. The agent's queue diverts the engage call to an agent's desktop.
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. OCS sends `Req_MakePredictiveCall` to CPD Server. This request contains `SAttr_DialTo`, which is the customer phone number.
7. CPD Server instructs the Dialogic board to place a call to the customer number that OCS provided.
8. If CPA has determined that there is a live voice, CPD Server attaches any customer data to the engage call (leg 20).
9. SIP Server delivers this data to the engaged agent's desktop as a screen pop.
10. 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 17 on [page 171](#) illustrates an ASM-mode call flow in VoIP environment that involves a SIP agent endpoint. HMP software is used for CPA.



**Figure 17: ASM-Mode Call Flow 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. OCS sends Req\_MakePredictiveCall to CPD Server. This request contains SAttr\_DialTo, which is the customer phone number.

6. CPD Server instructs HMP to place a call through SIP Server to the customer number that OCS provided.
7. If CPA has determined that there is a live voice, CPD Server attaches any customer data to the engage call (leg 10).
8. SIP Server delivers this data to the engaged agent's desktop as a screen pop.
9. 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.

---

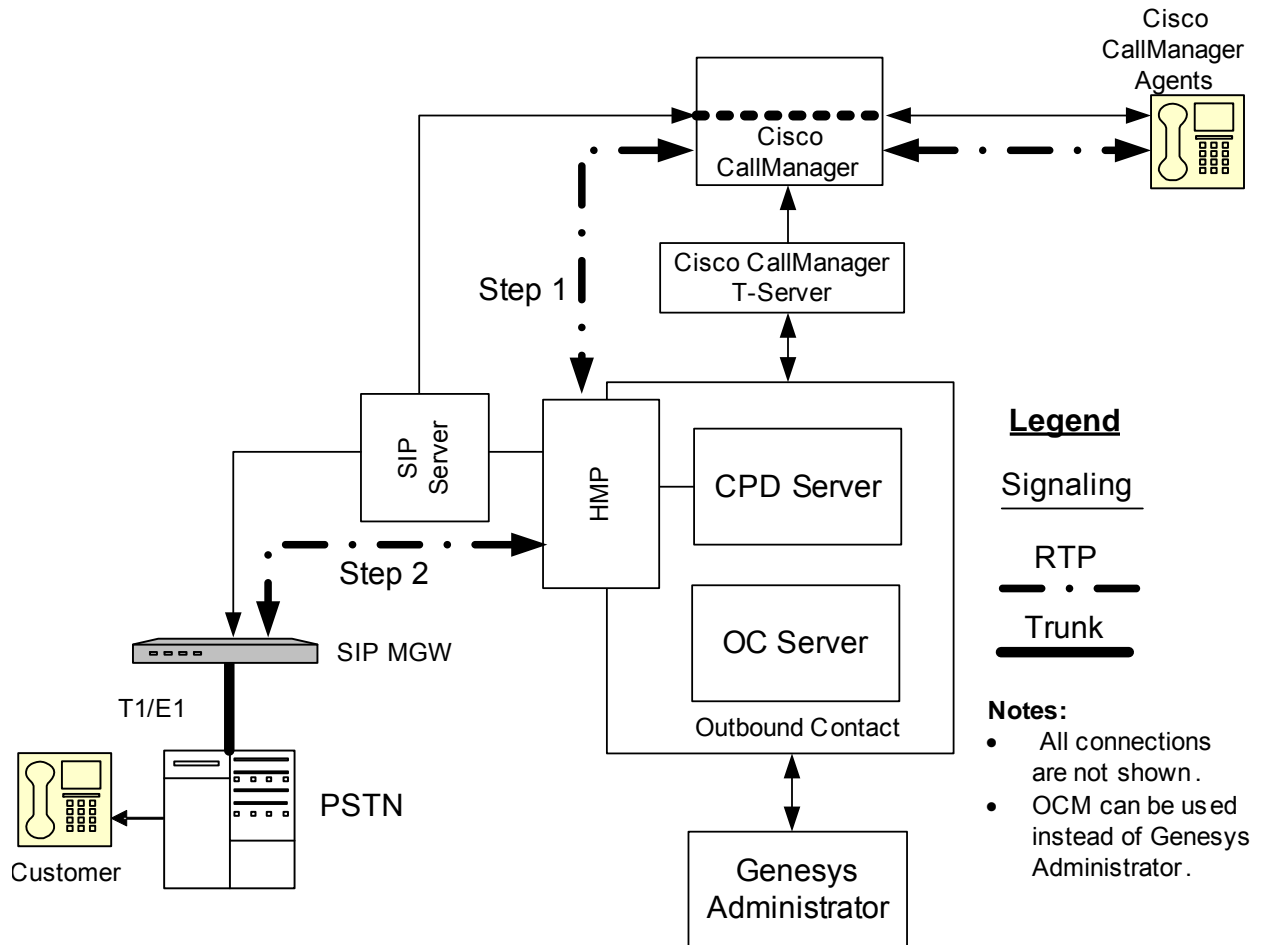
## Outbound Contact with Cisco CallManager

This section describes an ASM mode scenario and a Transfer mode 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 18 on [page 173](#) illustrates the architecture/call flow.



**Figure 18: ASM-Mode Call Flow—Cisco CallManager**

In this scenario, the call flow proceeds as follows:

#### Step 1

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 Group DN configuration, SIP Server redirects the engage call to Cisco CallManager Route Point DN.
4. As a result of a IRD strategy, URS 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

6. OCS sends Req\_MakePredictiveCall to CPD Server. This request contains SAttr\_DialTo, which is the customer phone number.

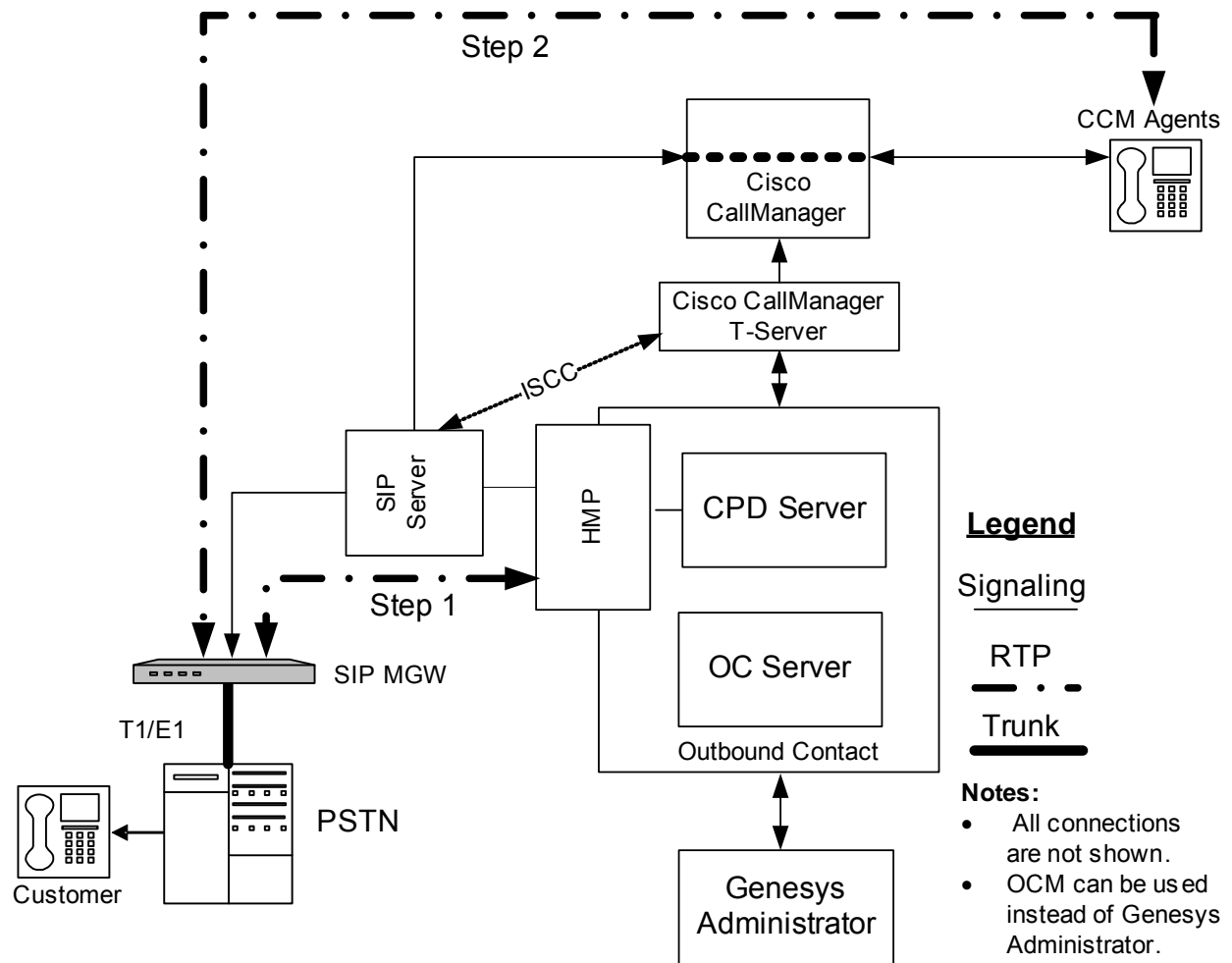
7. CPD Server initiates an outbound call to SIP Server, where another Trunk points to either a Media Gateway or an IP SIP client endpoint.
8. The SIP call is initiated.
9. Using HMP resources, CPD Server performs call progress analysis.
10. If a positive voice detection occurs, CPD Server bridges the internal leg (the engaged call) and the external leg (the outbound call).  
The call is established between the agent and the customer.
11. CPD Server informs OCS of the call result.
12. After the calls are bridged between the customer and the agent, SIP signalling occurs and RTP streams go through HMP.

## Transfer Mode

For Transfer mode, Outbound Contact is configured with a centralized CPD Server. In this configuration, SIP Server is used as a switch for dialing to customers and CCM switch is used to control outbound agents. So, when call is established with the customer, the outbound call is transferred to the Cisco CallManager agent using ISCC (data).

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

Figure 19 on [page 175](#) illustrates the architecture/call flow.



**Figure 19: Transfer Mode—Cisco CallManager**

### Step 1

1. The OCS sends a dial request to the CPD Server. (Both the OCS and CPD Server are at the central location.)
2. CPD Server sends a dial request to HMP.
3. HMP dials the customer's number.
4. SIP Server directs the call to the customer through the MGW.
5. HMP performs call progress detection and sends the call result to CPD Server/OCS.

### Step 2

6. After receiving an Answer call result, CPD Server transfers the call to a Routing Point. (HMP sends REFER to SIP Server and SIP Server sends INVITE to Cisco CallManager.)

7. Cisco CallManager T-Server, which is monitoring the Cisco CallManager switch, informs the URS that the call was routed to the Routing Point.
8. URS routes the call (per the routing strategy, also stored at the central location) sends to Inter Server Call Control (ISCC).
9. ISCC sends the call to the Cisco CallManager switch that is being monitored by the Cisco CallManager T-Server.
10. The Cisco CallManager switch relays the call to Route Point for a group of agents.

After the call is routed to an 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.

---

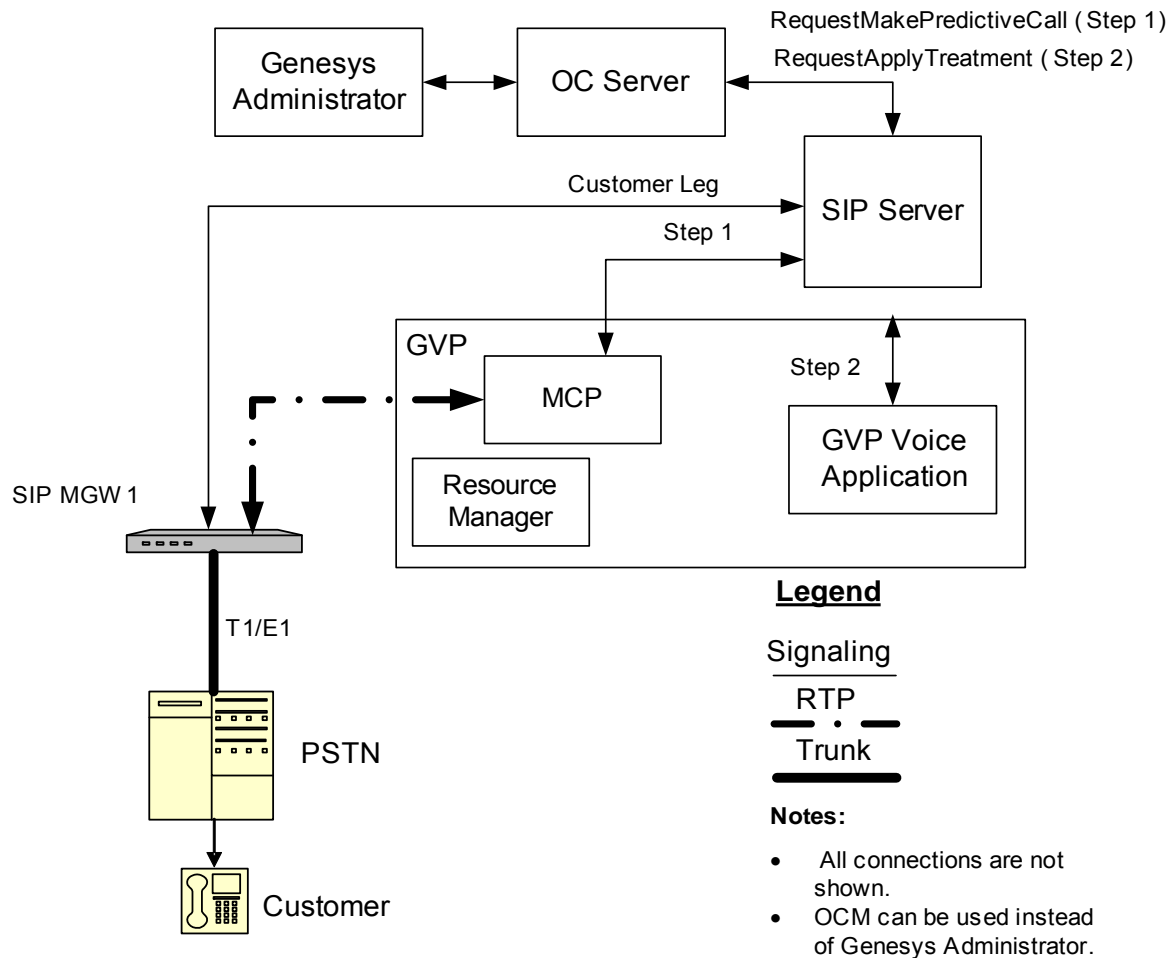
---

## Outbound Contact with GVP 8.1 (Proactive Contact Solution)

In this scenario, OCS, SIP Server, and GVP 8.1 provide support for self-service campaigns that may or may not involve an agent. SIP Server performs the dialing and MCP provides CPA. The VXML application using HTTP protocol supplies OCS with the interaction outcome. The Power GVP and Progressive GVP modes are supported.

Figure 20 on [page 177](#) illustrates how to use Outbound Contact with GVP 8.1.





**Figure 20: Outbound Contact, GVP 8.1—MCP as the CPA Provider**

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 invites MCP/Genesys Media Server to handle CPA.
3. SIP Server creates the call leg with MCP and establishes a call with the customer DN.
4. MCP performs CPA and sends the call result to SIP Server.
5. SIP Server reports the call state to OCS by either `EventReleased` (for negative call results) or `EventEstablished` (for positive ones) on the Trunk Group DN.

**Step 2**

6. For a positive call result (EventEstablished received), OCS sends the RequestApplyTreatment message to SIP Server to trigger a GVP Voice Application.
7. SIP Server generates an EventTreatmentApplied message.  
All media streams are between the GVP Voice Application and the customer.

## T-Library Functions in an Outbound-VoIP Environment

This section provides attached data and extensions information for T-Library functions when they are used in a VoIP environment.

---

Note: For information on the options identified in this section, see the *Outbound Contact 8.1 Deployment Guide*.

---

### TMakeCall Attached Data and Extensions

OCS uses TMakeCall to initiate an engaging call. [Table 131](#) lists the attached data for TMakeCall.

**Table 131: TMakeCall Attached Data**

Data Key	Type	Key Required	Value	Description
GSW_CALL_TYPE	String	Yes	ENGAGING	Identifies the call as an engaging call.
GSW_QUEUE_DBID	Int	Yes	DBID	Identifies the DBID of the Voice Transfer Group DN
GSW_SESSION_DBID	Int	Yes	DIBD	Identifies the DBID of the Campaign Group (Dialing Session) for the initiated call.

Table 132 on [page 179](#) lists the extensions for TMakeCall.

**Table 132: TMakeCall Extensions**

Data Key	Value	Description
beep	on or off	Enables the playing of a beep tone on an engaging call right before its bridged to a customer call.  This key is associated with beep-on-merge option.

### TMakePredictiveCall Attached Data and Extensions

OCS uses TMakePredictiveCall to initiate a customer call. [Table 133](#) lists the attached data for TMakePredictiveCall

**Table 133: TMakePredictiveCall Attached Data**

Data Key	Type	Key Required	Value	Description
GSW_CALL_TYPE	String	Yes	REGULAR	Identifies the call as customer call.
GSW_QUEUE_DBID	Int	Yes	DBID	Identifies the DBID of the Voice Transfer Group DN
GSW_SESSION_DBID	Int	Yes	DIBD	Identifies the DBID of the Campaign Group (Dialing Session) for the initiated call.

[Table 134](#) lists the extensions for TMakePreictiveCall.

**Table 134: TMakePredictiveCall Extensions**

Data Key	Value	Description
cpd-record	on or off	Enables or disables the recording of the call progress detection phase of the call, as set in the cpd-recording option.

**Table 134: TMakePredictiveCall Extensions (Continued)**

Data Key	Value	Description
call_answer_type_recognition	String	Identifies the call progress analysis (CPA) for both the pre-connect (via SIT tones) and the post-connect (either a fax or an answering machine) phases of the call, as set in the <code>call_answer_type_recognition</code> option.
cpd-on-connect	on or off	Indicates when CPA begins, according to the <code>cpd-on-connect</code> option. For Outbound-VoIP ASM modes, set this option to <code>yes</code> to specify that CPA begins after the call is connected.  <b>Note:</b> Setting this option to <code>yes</code> accounts for the use of color ring back tones. If <code>cpd-on-connect</code> does not appear in the extensions, CPA starts as soon as the media stream is available.  <i>Color ring back tones (CRBT)</i> refers to the ability to play other audio sounds (music, voice, and so on) for a busy signal for example, instead of a standard ring back tone.
call_timeguard_timeout	Time in milliseconds	Reflects the setting of the <code>call_timeguard_timeout</code> option, which specifies the maximum time allowed for CPA after the call is connected.

## TMergeCall Extensions

Table 135 lists the extensions for TMergeCall.

**Table 135: TMergeCall Extensions**

Data Key	Value	Description
method	bridging or transfer	Specifies the method of connection to be used. For more information on these methods, see the “Outbound-VoIP Deployment” chapter of the <i>Outbound Contact 8.1 Deployment Guide</i> .

## TApplyTreatment Extensions

Table 136 lists the extensions for TApplyTreatment.

**Table 136: TApplyTreatment Extensions**

Data Key	Value	Description
am-beep-detection	on or off	Specifies whether GVP is forced to detect an answering machine beep tone before playing music or starting the VXML application in Outbound VoIP dialing modes, as set in the am-beep-detection option.





Supplements

## Related Documentation Resources

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

### Outbound Contact

- The *Outbound Contact 8.1 Deployment Guide*, which describes application features, architectural information, instructions on how to install and configure Outbound Contact 8.1 components, configuration option descriptions, and other related information.
- *Outbound Contact Manager 7.6 Help*, which describes how to use Outbound Contact Manager.
- Release Notes and Product Advisories for this product, which are available on the [Genesys Customer Care website](#).

### Genesys Administrator

- *Framework Genesys Administrator Help*, which describes how to use Genesys Administrator, including topics covering Outbound Contact specific tasks.

### Proactive Contact/Routing

- The *Genesys 8.0 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 7.6 Proactive Routing Solution Guide*, which provides instructions on how to integrate Outbound Contact with the Customer Interaction Management (CIM) Platform.

## Management Framework

- The *Framework 8.1 Deployment Guide*, which will help you configure, install, start, and stop Framework components.
- The *Framework 8.1 Configuration Options Reference Manual*, which will provide you with descriptions of configuration options for other Framework components.
- The *Framework 8.1 Configuration Manager Help*, which will help you use Configuration Manager.

## Genesys

- The *Genesys Events and Models Reference Manual*, which contains the T-Library API, information on TEvents, and an extensive collection of call models.
- *Genesys Glossary*, which provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms used in this document.
- *Genesys Migration Guide*, which provides documented migration strategies for Genesys product releases. Contact Genesys Customer Care for more information.

Information about supported operating systems and third-party software is available on the Genesys Documentation website in the following documents:

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

Consult the following additional resources as necessary:

- *Genesys Hardware Sizing Guide*, which provides information about Genesys hardware sizing guidelines for the Genesys 8.x releases.
- *Genesys Interoperability Guide*, which provides information on the compatibility of Genesys products with various Configuration Layer Environments; Interoperability of Reporting Templates and Solutions; and Gplus Adapters Interoperability.
- *Genesys Licensing Guide*, which introduces you to the concepts, terminology, and procedures that are relevant to the Genesys licensing system.
- *Genesys DB Sizing Worksheets*, which provides a range of expected database sizes for various Genesys products.

For additional system-wide planning tools and information, see the release-specific listings of [System-Level Documents](#) on the [Genesys Documentation website](#).

Genesys product documentation is available on the:

- [Genesys Customer Care website](#).



## Related Documentation Resources

- [Genesys Documentation website](#).
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at [orderman@genesys.com](mailto:orderman@genesys.com).

# Document Conventions

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

## Document Version Number

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

```
80fr_ref_06-2008_v8.1.001.00
```

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

## Screen Captures Used in This Document

Screen captures from the product graphical user interface (GUI), as used in this document, may sometimes contain minor spelling, capitalization, or grammatical errors. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

## Type Styles

[Table 137](#) describes and illustrates the type conventions that are used in this document.

**Table 137: Type Styles**

Type Style	Used For	Examples
Italic	<ul style="list-style-type: none"> <li>Document titles</li> <li>Emphasis</li> <li>Definitions of (or first references to) unfamiliar terms</li> <li>Mathematical variables</li> </ul> <p>Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on <a href="#">page 187</a>).</p>	<p>Please consult the <i>Genesys Migration Guide</i> for more information.</p> <p>Do <i>not</i> use this value for this option.</p> <p>A <i>customary and usual</i> practice is one that is widely accepted and used within a particular industry or profession.</p> <p>The formula, <math>x + 1 = 7</math> where <math>x</math> stands for...</p>

**Table 137: Type Styles (Continued)**

Type Style	Used For	Examples
<p>Monospace font (Looks like teletype or typewriter text)</p>	<p>All programming identifiers and GUI elements. This convention includes:</p> <ul style="list-style-type: none"> <li>• The <i>names</i> 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.</li> <li>• The values of options.</li> <li>• Logical arguments and command syntax.</li> <li>• Code samples.</li> </ul> <p>Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.</p>	<p>Select the Show variables on screen check box.</p> <p>In the Operand text box, enter your formula.</p> <p>Click OK to exit the Properties dialog box.</p> <p>T-Server distributes the error messages in EventError events.</p> <p>If you select true for the inbound-bsns-calls option, all established inbound calls on a local agent are considered business calls.</p> <p>Enter exit on the command line.</p>
<p>Square brackets ([ ])</p>	<p>A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.</p>	<p>smcp_server -host [/flags]</p>
<p>Angle brackets (&lt; &gt;)</p>	<p>A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise.</p> <p><b>Note:</b> In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.</p>	<p>smcp_server -host &lt;confighost&gt;</p>





# Index

## A

Add record events	
AddRecord	79
AddRecordAcknowledge	81
AddRecord event	79, 95
AddRecordAcknowledge event	81, 95
Agent logout events	
LogOut	83
LogOutAcknowledge	85
LogOutTime	85
ASM Mode Call Flow (MGW without CPA)	167
attached data	14
Attaching Script Information to OCS User Events and Telephony Events	25
attributes	
send	15
user event	16
audiocodes	
media gateway	159

## C

Call Progress Analysis	159
call result	
completion, updating	47
Call Result Mapping by OCS	139
Call Result Mappings Made by CPD Server	140
call results	
updating	46
call_result field	24
call_wait_in_queue_timeout	136
call_wait_original_establish_timeout	136
CallerID Per Campaign	87
CallerID Per Campaign options	
for CPN (Calling Party Number)	88
CallerID Support	87
CallerID Support for Avaya CTI	88
CallerID Support for ISDN Connections	87
CallerID Support Per Campaign	87
Campaign status notification events	

CampaignGroupAssigned	34
CampaignGroupAssigned event	93
CampaignLoaded	29, 30
CampaignLoaded event	93
CampaignModeChanged	32
CampaignModeChanged event	93
CampaignStarted	28
CampaignStarted event	92
CampaignStatusRequest	34
CampaignStatusRequest event	93
CampaignStopped	31, 92
CampaignUnloaded	30, 93
campaigns	
mode change	32
notification	27
started	28
Canceling records events	
RecordCancel	65
RecordCancelAcknowledge	63
RequestRecordCancel	61
chained records	
request	52, 53
Chained records events	
ChainedRecord	54
ChainedRecordRequest	53
ChainedRecordsDataEnd	56
ChainedRecord event	54, 94
ChainedRecord Request event	53
ChainedRecordRequest event	94
ChainedRecordsDataEnd event	56, 94
char data type	141
CommDN with OCS and third-party apps	98
commenting on this document	8
Communication DN protocol	
definition	97
enumeration values and	103
mandatory attributes and	99
Contact Information Types	141
contact_info_type field	24
CPA	159
CPNDigits	88

CPNPlan	88	ReadyTimeAcknowledge	41
CPNPresentation	88	RecordProcessed	49
CPNScreening	88	RecordProcessedAcknowledge	51
Creating Indexes	151	RecordReschedule	72
custom data formats	18	RecordRescheduleAcknowledge	73
custom field		ScheduledCall	76
updating	46	ScheduledRecordRescheduleAcknowledge	75
		scheduleItem-	103
		ScheduleRecordReschedule	73
		UpdateCallCompletionStats	47
		UpdateCallCompletionStatsAcknowledge	49
<b>D</b>		<b>F</b>	
Data Enums	113	fields	
data formats		call_result field	24
custom	18	contact_info_type field	24
data type		custom, updating	46
char data type	141	mandatory	17
datetime data type	141	record_status	25
float data type	141	reserved names	18
int data type	141	updating	17
unknown data type	141	float data type	141
varchar data type	141		
datetime data type	141		
DBMS optimizations	149		
Default Record Information	14	<b>G</b>	
default record information	14	GSW_AGENT_REQ_TYPE	12
Defined Constants	133	GSW_APPLICATION_ID	13
dialing modes		GSW_CM_AttrError	114
Predictive	155	GSW_CM_MessageType	113
Progressive	155	GSW_RECORD_HANDLE	12
Do Not Call events		GVP	
DoNotCall	68	Outbound integration	176
DoNotCallAcknowledge	70	GVP 8.1	155
do not call request	66		
document			
audience	8		
conventions	186		
errors, commenting on	8		
version number	186		
DoNotCall event	68, 94		
DoNotCallAcknowledge event	70, 95		
		<b>H</b>	
<b>E</b>		HMP	
enum. See enumeration tables		asm-mode	170
enumeration tables	23	HTTP	
error names	91	client requests	119
Event Overview	12	OCS responses	125
Event Responses	13	Post header	119
events		req parameter	118
event structures	12	supported resource types	119
messages	13	URI format	118
No RecordsAvailable	45	HTTP Protocol	117
NoRunningPreviewCampaigns	45	HTTPS	
PreviewDialingModeStartAcknowledge	37	Transport Layer Security (TLS)	117
PreviewRecord	43		
ReadyTime	40		
		<b>I</b>	
		int data type	141
		IP Telephony	155

## L

LogOut event . . . . . 83, 95  
 LogOutAcknowledge event . . . . . 85, 95  
 LogOutTime event . . . . . 85, 95

## M

Maintaining Indexes for Large Calling Lists . 151  
 mandatory fields . . . . . 17  
 media gateway  
   audiocodes . . . . . 159  
   paraxip . . . . . 159  
 MGW . . . . . 159  
 Microsoft SQL Server . . . . . 150

## N

New Call Mappings . . . . . 140  
 No RecordsAvailable event . . . . . 45  
 NoRunningPreviewCampaigns event . . . . . 45  
 notification . . . . . 27  
   unsolicited . . . . . 82

## O

OCS DBID . . . . . 12  
 Optimizing Calling List Tables . . . . . 149  
 options  
   outbound\_contact\_server . . . . . 101  
 OriginAppDBID . . . . . 99, 101  
 Outbound  
   GVP integration . . . . . 176  
 outbound\_contact\_server . . . . . 101

## P

paraxip  
   media gateway . . . . . 159  
 Personalized Ring Tone Support . . . . . 90  
 POST header . . . . . 119  
 Predictive dialing mode . . . . . 155  
 Preview dialing mode events  
   PreviewDialingModeOver . . . . . 38  
   PreviewDialingModeOverAcknowledge . . . 39  
   PreviewDialingModeStart . . . . . 36  
 Preview Record  
   No Running Preview Campaigns . . . . . 45  
   PreviewDialingMode event . . . . . 93  
   PreviewRecord event . . . . . 43, 93  
   PreviewRecordRequest event . . . . . 42, 93  
 Proactive Interaction . . . . . 86  
   media type . . . . . 86  
 Progressive

dialing modes . . . . . 155

## R

ReadyTime agent request . . . . . 40  
 ReadyTime event . . . . . 93  
 ReadyTimeAcknowledge event . . . . . 93  
 Record Not Found . . . . . 12, 52  
 record processed . . . . . 49  
 record status  
   stale . . . . . 145  
 record\_status field . . . . . 25  
 record\_type field . . . . . 25  
 RecordCancel event . . . . . 65, 95  
 RecordCancelAcknowledge event . . . . . 63, 94  
 RecordProcessed event . . . . . 49, 93  
 RecordProcessedAcknowledge event . . . . . 51, 93  
 RecordReject event . . . . . 57, 94  
 RecordRejectAcknowledge event . . . . . 58, 94  
 RecordReschedule event . . . . . 72, 94  
 RecordRescheduleAcknowledge event . . . . . 73, 94  
 records  
   attached data . . . . . 14  
   canceling . . . . . 59  
   chained records request . . . . . 52  
   do not call . . . . . 66  
   Preview record request . . . . . 42  
   rejecting . . . . . 57  
   reschedule . . . . . 72  
   scheduling . . . . . 71  
 Rejecting records events  
   RecordReject . . . . . 57  
   RecordRejectAcknowledge . . . . . 58  
 req parameter . . . . . 118  
 RequestRecordCancel event . . . . . 61, 94  
 reschedule . . . . . 72  
 reserved names . . . . . 18

## S

ScheduledCall event . . . . . 76, 94  
 ScheduledRecordReschedule event . . . . . 73  
 ScheduledRecordRescheduledAcknowledge  
   event . . . . . 75  
 scheduleItem- event . . . . . 103  
 ScheduleRecordReschedule event . . . . . 94  
 scheduling . . . . . 71  
 send attributes . . . . . 15  
 send\_attribute  
   names . . . . . 17  
 stale record status . . . . . 145  
 started . . . . . 28  
 status notification . . . . . 27  
 Supported Functionality . . . . . 155

## T

TargetAppDBID . . . . .	99, 101
Temporary Tables Considerations . . . . .	150
Transfer Mode (MGW with CPA) . . . . .	159
Transfer Mode (MGW without CPA) . . . . .	161
Transport Layer Security (TLS) HTTPS over . . . . .	117

## U

unknown data type . . . . .	141
unsolicited notifications . . . . .	82
UpdateCallCompletionStats event . . . . .	47, 93
UpdateCallCompletionStatsAcknowledge event . . . . .	49, 93
Updating Indexes . . . . .	151
user event attributes . . . . .	16
UserEvent Attributes . . . . .	112

## V

varchar data type . . . . .	141
VoIP environment . . . . .	158
T-Library functions . . . . .	178