Outbound Contact Reference Manual

Outbound Contact 8.1.5

12/29/2021

Table of Contents

Outbound Contact Reference Manual

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

SCXML Architecture

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.

> **Important**
>
> 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 7.6 Solution Guide*.

Event Overview

There are telephony and user events in Genesys.

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

Characteristics of Event Structures

There are two types of user events:

- Agent desktop request to OCS. All messages that travel from the desktop to OCS have the `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 the previous GSW_APPLICATION_ID for a while, 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), 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 the Error Event Attributes table should be contained in all error events.

Error Event Attributes

Key	Type	Comment
GSW_ERROR	String	Error name; see Error Names and Codes.
GSW_ERROR_NUMBER	Int	Mandatory

Error Names and Codes 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.

> **Important**
>
> 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 that the desktop application should not change the value of these key-value pairs (except `GSW_CALL_RESULT`).

The Default Record Information table contains a list of key-value pairs that OCS attaches to outbound call's user data by default.

Default Record Information

Key	Type	Description
GSW_ABANDONED	Int64 (Unix timestamp)	The `timeAbandoned` parameter from FTC timestamps.

Key	Type	Description
GSW_ACW_COMPLETED	Int64 (Unix timestamp)	The time when After Call Work is completed.
GSW_APPLICATION_ID	Integer	The OCS configuration application database ID. The unique identifier of the running OCS instance.
GSW_ATTEMPTS	Integer	The number of attempts for the record.
GSW_CALL_ATTEMPT_GUID	String	The global unique identifier of the call processing attempt used for historical reporting.
GSW_CALL_RESULT	Integer	Call Result saved from the previous call, or Call Result sent to change automatically detected call result; see Enumeration Table and Call Result Types.
GSW_CALLING_LIST	String	The name of the Calling List.
GSW_CAMPAIGN_GROUP_DESCRIPTION	String	The description of the Campaign Group.
GSW_CAMPAIGN_GROUP_GUID	String	The global unique identifier of the Campaign Group.
GSW_CAMPAIGN_GROUP_NAME	String	The name of the Campaign Group.
GSW_CAMPAIGN_NAME	String	The name of the Campaign.
GSW_CHAIN_ID	Integer	The unique Chain ID.
GSW_COMPLETE_PROCESSING	Int64 (Unix timestamp)	The time when processing of the call attempt was completed.
GSW_COUNTRY_CODE	String	The Client's country code.
GSW_D_AREA_CODE	String	The area code associated with the device.
GSW_HIST_SEQUENCE_NUM	Integer	Record submission sequential number 0 or 1. Value 0 means the first call report record submission for a given call attempt. Value 1 means the second call report record submission for a given finalized call.
GSW_OPTIMIZE_BY	Integer	The optimization method (enumeration). Available for predictive dialing modes only.
GSW_OPTIMIZE_GOAL	Integer	The optimization goal. Available for predictive dialing modes only.
GSW_PHONE	String	The customer’s phone number.
GSW_POSTAL_CODE	String	The Client's postal code.
GSW_RECORD_HANDLE	Integer	The unique Record Identifier.

Key	Type	Description
GSW_SCHEDULED_TIME	Int64 (Unix timestamp)	The reschedule time for a record. Available only with records of particular types.
GSW_START_PROCESSING	Int64 (Unix timestamp)	The time that OCS started processing the call attempt. If pre-dial validation is in place, this is the time that the pre-dial validation request was sent.
GSW_TENANT_NAME	String	The value is populated from the current Tenant (where the Campaign Group belongs). The `Tenant\Annex\devops\customer_name` option. Defaults to `undefined` if not set.
GSW_TZ_CODE	String	The client's time zone.
GSW_TZ_OFFSET	Integer	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. Values: -43200 ... 43200.

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.

Tip

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

The User Event Attributes table 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.

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: 1 = Predict 2 = Progress 3 = Preview 4 = ProgressAndSeize 5 = PredictAndSeize 8 = PushPreview 9 = ProgressiveGVP 11 = PowerGVP
GSW_APPLICATION_ID	Int	OCS application DBID Valid values begin at 101

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 Modifiable Mandatory Genesys Fields for modifiable mandatory Genesys fields.

Genesys Mandatory Fields

The Modifiable Mandatory Genesys Fields table contains the only Genesys mandatory fields that are modifiable by the `RecordProcessed` or `UpdateCallCompletionStats` events.

Modifiable Mandatory Genesys Fields

Genesys Mandatory Field Name	Recommended Key for send_attribute	Type	Description
call_result	GSW_CALL_RESULT	Int	Sent to change an automatically detected call result. See Enumeration Table and Call Result Types.
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 Enumeration Table.
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 Custom Data Formats.

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 Reserved Keys 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.

> **Important**
>
> 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`.

Reserved Keys

Key	Values	Type	Description
GSW_AGENT_ANSWERED		Int64 (Unix timestamp)	The time when an agent answers a call.
GSW_AGENT_RELEASED		Int64 (Unix timestamp)	The time when an agent

Key	Values	Type	Description
			releases a call.
GSW_AGENT_ID		String	The Login ID of the last agent who worked with the record.
GSW_AGENT_REQ_TYPE		String	The event identifier for events coming from desktops to OCS.
GSW_APPLICATION_ID	101…	Integer	The OCS configuration application database ID; unique identifier of the running OCS instance.
GSW_ASM_OVERDIAL	1	Integer	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 overdialled by OCS.
GSW_ASSURED_HANDLE	1…	Integer	The 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 …	Integer	The number of attempts for the record; used when a new record is added.
GSW_BLOCKING_RULE		String	The name of the rule that caused a negative result of pre-dial validation. OCS might receive it from a third-party validation server in a negative validation response. In such case, OCS passes it in a call attempt record submission unchanged.
GSW_BLOCKING_RULE_TYPE		Integer	The type of the compliance object that caused a negative pre-dial validation (Mandatory Suppression List, Custom Compliance Rule, etc). OCS may receive it from a third-party validation server in a negative

Key	Values	Type	Description
			validation response. In such case, OCS passes it in a call attempt record submission unchanged.
GSW_CALL_ATTEMPT_GUID		String	The 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		Integer	Call Result saved from the previous call, or Call Result sent to change an automatically detected call result; see Enumeration Table and Call Result Types.
GSW_CALL_RESULT_FEEDBACK	1	Integer	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...	Integer	The system time when a 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	The type of callback an agent wants to create,

Key	Values	Type	Description
			either Personal or Campaign.
GSW_CALLING_LIST		String	The name of the calling list.
GSW_CAMPAIGN_DESCRIPTION		String	The description of the Campaign. A 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 Enumeration Table. The values Engaged `Predictive` and Engaged `Progressive` correspond to the `Predictive with seizing` and `Progressive with seizing` dialing modes.
GSW_CONTACT_MEDIA_TYPE	any, email, voice	String	The media type for the calling record that corresponds to the value of the `contact_info_type` field in the calling list.
GSW_CAMPAIGN_GROUP_DBID		String	The DBID of the Campaign Group.
GSW_CAMPAIGN _GROUP_NAME		String	The name of the Campaign Group.
GSW_CAMPAIGN_GROUP_DESCRIPTION		String	The description of the Campaign Group. A value may be an empty string.
GSW_CAMPAIGN_NAME		String	The name of the Campaign.
GSW_CAMPAIGN_TEMPLATE_NAME		String	The name of the Campaign Template (the Script of type Outbound Campaign, referenced by the Campaign Group via the scriptDBID attribute).
GSW_CHAIN_ATTR	AllChain, RecordOnly	String	The flag determining whether to update a record chain or just a single record.
GSW_CHAIN_ID	0...	Integer	The unique chain ID.
GSW_CHAIN_N	0...	Integer	The unique identifier of a record in a chain.
GSW_CPD_COMPLETED		Int64 (Unix timestamp)	The time when CPD is completed.
GSW_CPN_DIGITS		String	CPNDigits as configured

Key	Values	Type	Description
			for a given record.
GSW_CPN_DIGITS_SET		String	The name of the Caller ID Set (the Script configuration object) that is included the CPNDigits value used in this call attempt. May not be present if no CPNDigits is configured or if CPNDigits is configured not via a Caller ID Set (via SCXML or the CPNDigits OCS option at any level).
GSW_CUSTOMER_ID		String	The customer ID that is used for requests.
GSW_DATE_TIME	A string represented in time in this format: MM/DD/YY(YYYY) HH:MM.	String	The date and time of a scheduled call, in the record's time zone.
GSW_DIALING		Int64 (Unix timestamp)	The time when call dialing is started.
GSW_D_STATE_CODE		String	The region code of a device.
GSW_D_TZ_JAVA_NAME		String	The time zone of a device.
GSW_DEVICE_MASK		String	The device mask.
GSW_ERROR	Error name	String	The error name. See Error Names and Codes.
GSW_ERROR_DESCRIPTION	Error description	String	The error description. See Error Names and Codes.
GSW_ERROR_NUMBER	Error Number	Integer	The error code. See Error Names and Codes.
GSW_FROM	0...	Integer	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	The DoNotCall message or a record cancellation message
GSW_PHONE		String	The customer's phone number.

Key	Values	Type	Description
GSW_PHONE_TYPE		Integer	The customer's phone type. See Enumeration Table.
GSW_RECORD_HANDLE	1...	Integer	The Unique Record Identifier.
GSW_RECORD_STATUS	See Enumeration Table.	Integer	The status of adding a record sent from a desktop.
GSW_RECORD_TYPE	See Enumeration Table.	Integer	The type of an added record sent from a desktop.
GSW_RELEASED		Int64 (Unix timestamp)	The time when a call is released.
GSW_SCRIPT_ID		Integer	The DBID of the `Script` configuration object.
GSW_STATE_CODE		String	The Client's region code.
GSW_SWITCH_DBID		Integer	The DBID of the `Switch` configuration object.
GSW_TENANT_CCID		Integer	The Contact Center ID. The value is populated from the current Tenant (where the Campaign Group belongs). The `Tenant\Annex\devops\customer_name` option. Defaults to `undefined` if not set.
GSW_TREATMENT	RecordTreat Personal, RecordTreat Campaign	String	The treatment that should be applied to a record chain for `RecordProcessed` event.
GSW_TZ_JAVA_NAME		String	The Client's time zone.
GSW_TZ_NAME		String	The Configuration Server time zone name (usually a standard three-letter abbreviation).
GSW_TZ_OFFSET	-43200 ... 43200	configuration	Offset (the time difference) in seconds between UTC and a particular time zone. It might contain different values throughout the year if Daylight Savings Time (DST) is used for the specified time zone.
GSW_UNTIL	0... > GSW_FROM	configuration	GSW_FROM -

Key	Values	Type	Description
			GSW_UNTIL: The time frame when a record can be called (in seconds from midnight).
GSW_USER_EVENT	Event Type, see All Desktop Protocol Events and Event Type Protocols.	String	The event identifier for events coming from OCS to desktops.
InteractionType	Outbound	String	The type of the interaction that is created by OCS. The value of this key is always set to `Outbound`.
InteractionSubtype	OutboundNew	String	The 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). Enumeration Table 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.

Enumeration Table

Genesys Mandatory Field in Calling List Table	Key	Enumeration Value	Data Type in User Event	Description
call_result	GSW_CALL_RESULT	See Call Result Types 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	Int	Customer phone type.

Genesys Mandatory Field in Calling List Table	Key	Enumeration Value	Data Type in User Event	Description
		7, Modem 8, Voice Mail 9, Pin Pager 10, E-mail Address 11, Instant Messaging		
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.

> **Important**
>
> 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.

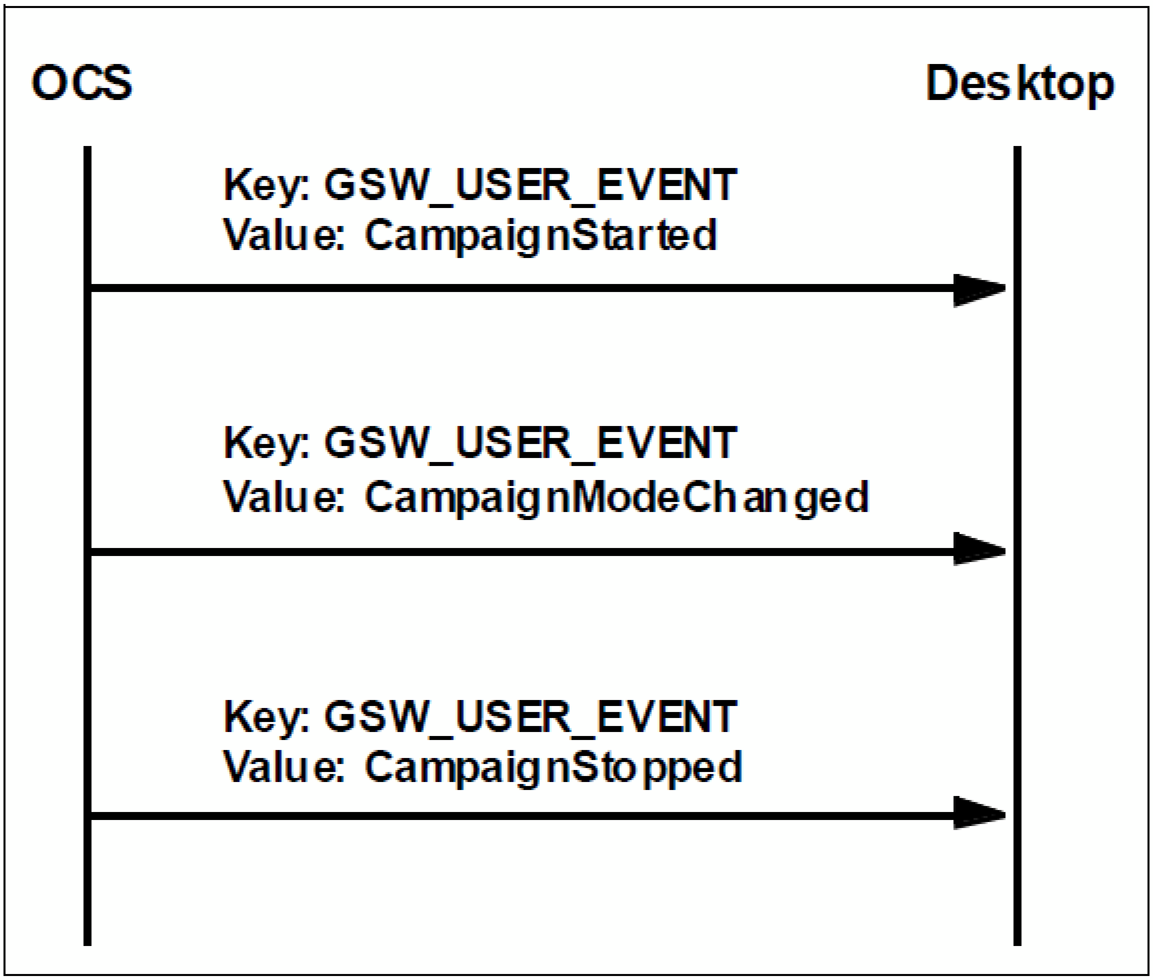
If the primary and backup OCS switch for any reason, a new primary server sends event `CampaignStarted/CampaignLoaded` to every agent in the Campaign to let the desktop know that the `GSW_APPLICATION_ID` attribute has changed. For more information, see Characteristics of Event Structures.

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.

Campaign Status from OCS to the Desktop shows the user events `CampaignStarted`, `CampaignStopped`, and `CampaignModeChanged`, which OCS sends to the desktop.



Campaign Status from OCS to the Desktop

CampaignStarted

OCS sends this event to the desktop when a dialing session for a Campaign is started. CampaignStarted contains more information.

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.

CampaignStarted User Event	
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.

The CampaignStarted Attached Data table lists the attached data for the `CampaignStarted` event.

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[a]	String	Yes	`Power GVP`, Progressive GVP.`Predictive`, `Predictive with Seizing`, `Preview`, `Progressive`, `Progressive with Seizing`, and `Push Preview`. **Note:** 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. CampaignLoaded contains more information.

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.

CampaignLoaded Attached Data lists the attached data for the `CampaignLoaded` event.

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. CampaignUnloaded contains more information.

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.

The CampaignUnloaded Attached Data table lists the attached data for the `CampaignUnloaded` event.

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. CampaignStopped contains more information.

CampaignStopped

CampignStopped 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.

The CampaignStopped Attached Data table lists the attached data for the `CampaignStopped` event.

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. CampaignModeChanged contains more information.

CampaignModeChanged

CampaignModeChanged User Event	
Description	Description of change sent to all logged-in agents when the

CampaignModeChanged User Event	
	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.

The CampaignModeChanged Attached Data table lists the attached data for the `CampaignModeChanged` event.

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 >	
GSW_CAMPAIGN_MODE	String	Yes	`Power GVP`, `Progressive GVP`, `Predictive`, `Predictive with Seizing`, `Preview`, `Progressive`, `Progressive with Seizing`, and `Push Preview`. **Note:** 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. CampaignGroupAssigned contains more information.

> **Important**
>
> This notification is a new part of Outbound Contact functionality. Refer to Agent Assignment in Multiple Campaigns in the *Outbound Contact Deployment Guide* for more information about this functionality.

CampaignGroupAssigned

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

The CampaignGroupAssigned Attached Data table lists the attached data for the `CampaignGroupChanged` event.

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

CampaignStatusRequest contains more information.

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.

The CampaignStatusRequest Attached Data table lists the attached data for the `CampaignStatusRequest` event.

CampaignStatusRequest Attached Data

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

> **Important**
>
> 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).

Campaign Group Status Notification from OCS to Web or Application Server

The notifications about the change in status of a Campaign Group can be sent from OCS to a Web or Application Server, such as Campaign Manager. The data is communicated to the Web or Application server using HTTP POST. The following states and information can be reported:

- Campaign Group loaded
- Campaign Group started
- Campaign Group stopped
- Campaign Group unloaded
- Campaign Group dial mode change

OCS can also periodically send data about active Campaign Groups, with a user-configurable time interval.

For information about the body of the HTTP POST message, see Fields in HTTP POST Requests to Web or Application Server in the *Outbound Contact Deployment Guide*.

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. PreviewDialingModeStart contains more information.

> **Important**
>
> 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.

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.

The PreviewDialingModeStart Attached Data table lists the attached data for the PreviewDialingModeStart request.

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[a]	String	No	Name of the Campaign Group.
GSW_CAMPAIGN_NAME	String	Yes	Name of the Campaign.
GSW_REFERENCE_ID[b]	Int	No	Reference identifier for the

Data Key	Type	Key Required	Description
			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. PreviewDialingModeStartAcknowledge contains more information.

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.

The Preview Dialing Mode Start Acknowledge Attached Data table lists the attached data for the `PreviewDialingModeStartAcknowledge` event.

Preview Dialing Mode Start Acknowledge Attached Data

Data Key The ! rowspan="1" colspan="1" | Type	Key Required	Description	
GSW_USER_EVENT	String	Yes	`PreviewDialingModeStartAcknowledge`
GSW_APPLICATION_ID	Int	Yes	Unique ID of OCS
GSW_CAMPAIGN_NAME	String	Yes	Name of the Campaign

PreviewDialingModeOver

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

> **Important**
>
> 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.

PreviewDialingModeOver

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

The PreviewDialingModeOver Attached Data table lists the attached data for the `PreviewDialingModeOver` request.

PreviewDialingModeOver Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	`PreviewDialingModeOver`
GSW_APPLICATION_ID	Int	Yes	Unique ID of OCS
GSW_CAMPAIGN_GROUP_NAME[a]	String	No	Name of the Campaign Group
GSW_CAMPAIGN_NAME	String	Yes	Name of the Campaign.
GSW_REFERENCE_ID[b]	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. PreviewDialingModeOverAcknowledge contains more information.

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.

The PreviewDialingModeOverAcknowledge Attached Data table lists the attached data for the PreviewDialingModeOverAcknowledge event.

PreviewDialingModeOverAcknowledge Attached Data

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

ReadyTime Request

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

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

Note:	- This request was added in the 7.6.1 release.

ReadyTime

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

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

The ReadyTime Attached Data table lists the attached data for the ReadyTime request.

ReadyTime Attached Data[a]

Data Key[b]	Type	Key Required	Value	Description
GSW_AGENT_REQ_TYPE	String	Yes	ReadyTime	Hard coded request name
GSW_READY_TIME	Int	Yes	<Expected time in seconds>	Expected time to go ready in seconds (N), where N >0
GSW_APPLICATION_ID	Int	Yes	<Unique ID of OCS>	Target OCS application DBID
GSW_RECORD_HANDLE[c]	Int	Yes	<Unique record handle>	Record handle for the record currently on the agent's desktop
GSW_REFERENCE_ID[d]	Int	No	<Reference ID>	Reference identifier for the request

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

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

OCS.

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

[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.

ReadyTimeAcknowledge

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

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.

The ReadyTimeAcknowledge Attributes table lists the attributes for the ReadyTimeAcknowledge event.

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 ReadyTime Error Codes is returned.

ReadyTime Error Codes

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

PreviewRecordRequest

PreviewRecordRequest	
Description	Request to send preview record.

PreviewRecordRequest	
OCS Action	Conditionally sends acknowledgment, depending on setting of the PreviewDialingModeStart event. See PreviewDialingModeStart Request.

The PreviewRecordRequest Attached Data table lists the attached data for PreviewRecordRequest.

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[a]	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. PreviewRecord contains more information. See Reserved Keys and Enumeration Table for predefined attribute values.

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).

The PreviewRecord Attached Data table lists the attached data for the PreviewRecord event.

PreviewRecord Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	`PreviewRecord`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_ATTEMPTS	Int	No	Number of attempts for the record. This key is used when a new record is added.
GSW_CALL_RESULT	Int	Yes	Call Result set by dialer or saved from previous call. (See the Enumeration Table.)
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[a]	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 Enumeration Table).
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.

Data Key	Type	Key Required	Description
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 in a few seconds. No Records Available contains more information.

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.

The No RecordsAvailable Attached Data table lists the attached data for the `No Records Available` event.

No RecordsAvailable Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	`PreviewRecordRequest`
GSW_ERROR_NUMBER	Int	Yes	Error code. See Error Names and Codes.
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. NoRunningPreviewCampaigns contains more information.

> Important
>
> In release 7.2, this scenario resulted in a `NoActivePreviewCampaign` message.

NoRunningPreviewCampaigns

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

The No Running Preview Campaign Error table lists the attached data for the `NoRunningPreviewCampaign` error.

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.
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 Call Result Types.)

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.

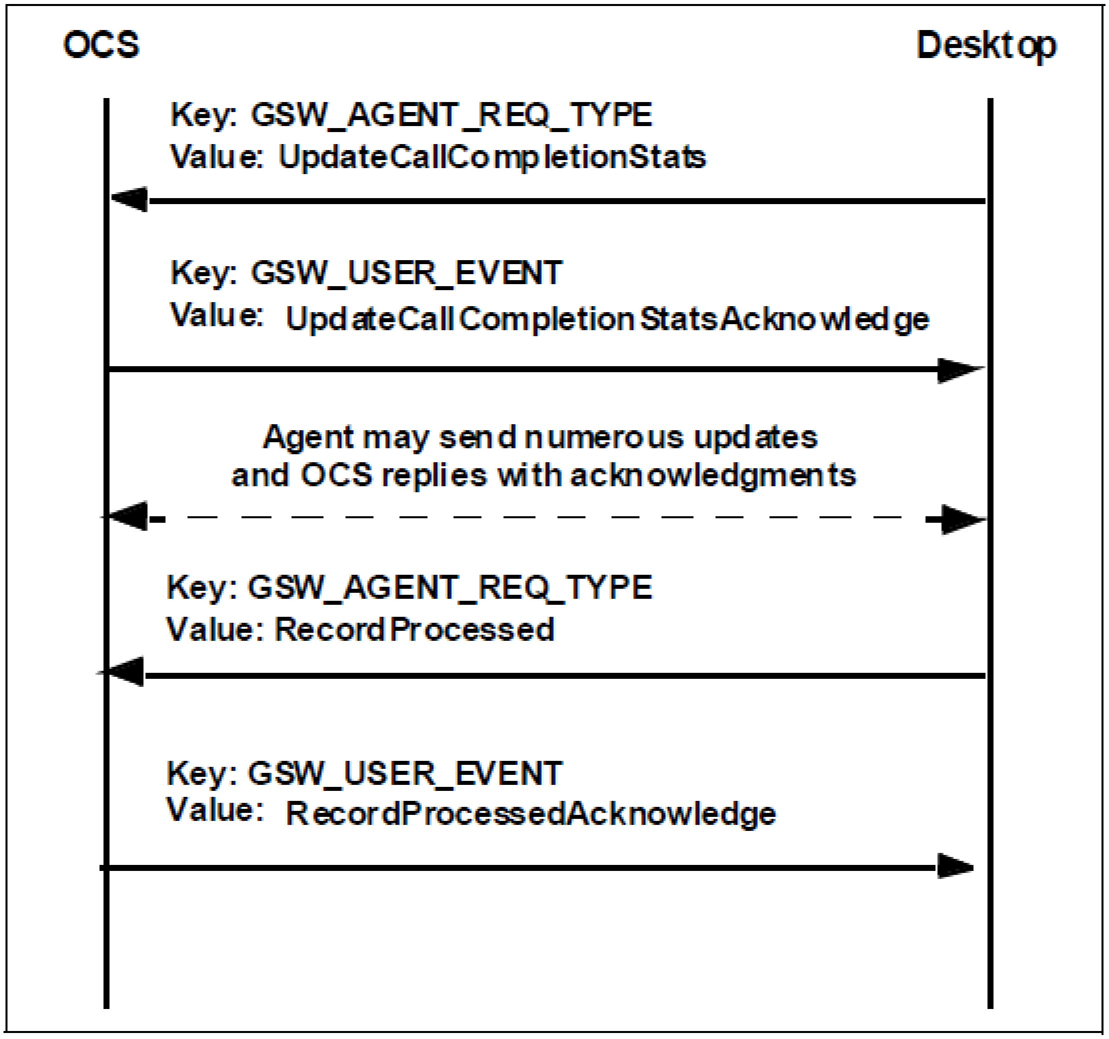
If you want OCS to apply a treatment to the call result entered in UpdateCallCompletionStats request, then the final RecordProcessed request for this record should contain an optional GSW_TREATMENT attribute, which has a possible value of RecordTreatPersonal or RecordTreatCampaign.

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.

Updating Call Results and Custom Fields Data Flow illustrates a typical data flow when updating Call Results and Custom Fields.


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

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.UpdateCallCompletionStats contains more information.

UpdateCallCompletionStats

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

The UpdateCallCompletionStats Attached Data table lists the attached data for the `UpdateCallCompletionStats` request.

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 Enumeration Table and Call Result Types.
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 Enumeration Table.
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[a]	String	No	Name of the Campaign Group

Data Key	Type	Key Required	Description
GSW_CAMPAIGN_NAME	String	Yes	Name of the Campaign.
GSW_REFERENCE_ID[b]	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. UpdateCallCompletionStatsAcknowledge contains more information.

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).

The UpdateCallCompletionAcknowledge Attached Data table lists the attached data for the `UpdateCallCompletionStatsAcknowledge` event.

UpdateCallCompletionAcknowledge Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	`UpdateCallCompletionStatsAcknowledge` or error.
GSW_APPLICATION_ID	Int	Yes	OCS configuration application

			database ID. Unique identifier of the running OCS instance.
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CAMPAIGN_NAME	String	Yes	Name of the Campaign.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier. Do not change this value.

RecordProcessed

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

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

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.

The RecordProcessed Attached Data table lists the attached data for the `RecordProcessed` request.

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.

Data Key	Type	Key Required	Description
GSW_CALL_RESULT	Int	No	Call Result sent to change automatically detected call result. See Enumeration Table and Call Result Types. **Note:** GSW_CALL_RESULT is an optional attribute in both UpdateCallCompletionStats and RecordProcessed requests.
GSW_CALLING_LIST	String	Yes	Name of the calling list.
GSW_CAMPAIGN_GROUP_NAME[a]	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 Enumeration Table.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.
GSW_REFERENCE_ID[b]	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. RecordProcessedAcknowledge contains more information.

RecordProcessedAcknowledge

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

The RecordProcessedAcknowledge Attached Data table lists the attached data for RecordProcessedAcknowledge event.

RecordProcessedAcknowledge Attached Data

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

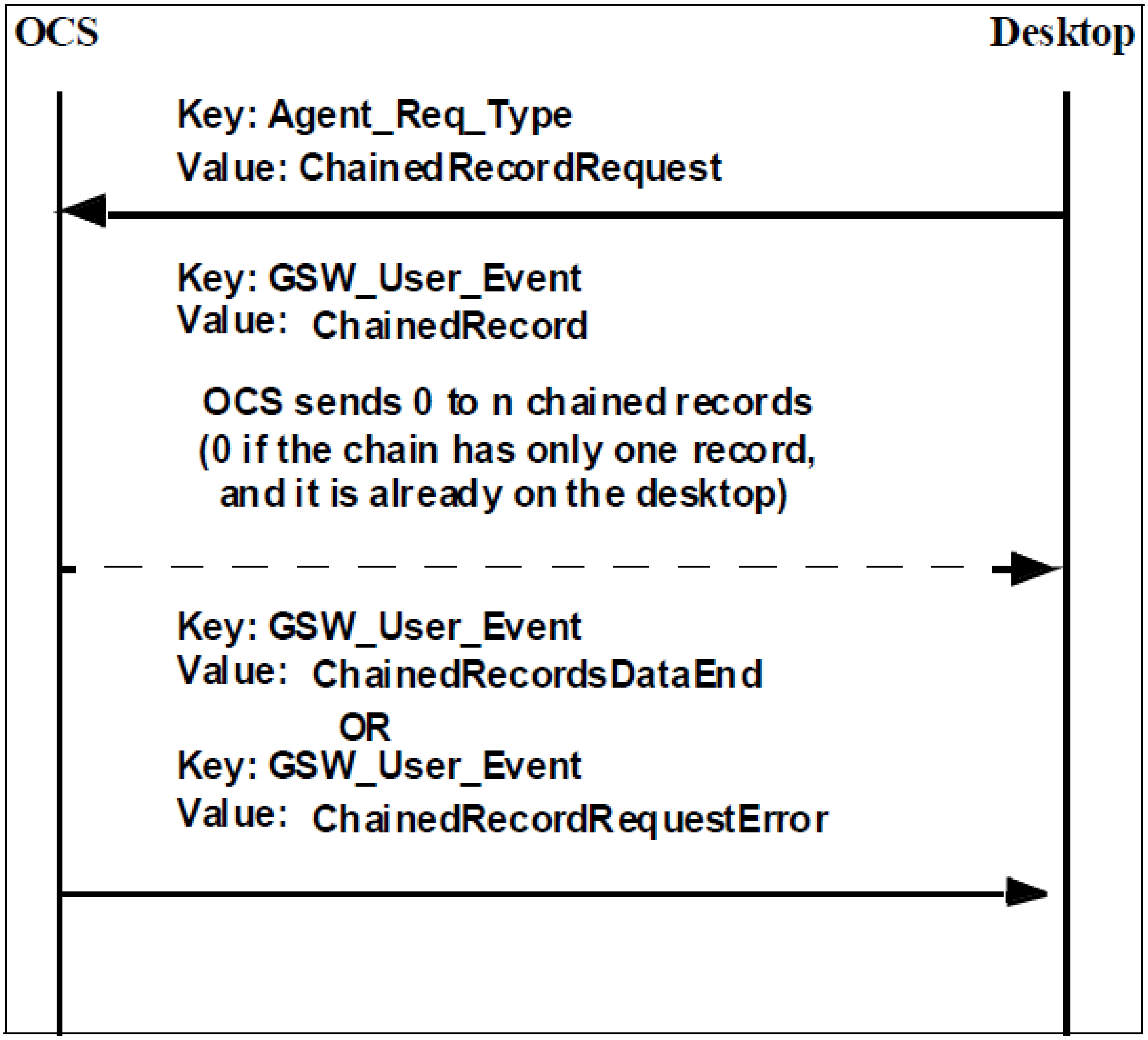
Chained Records

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

> ### Important
> 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, InPredictive 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. Chained Record Data Flow is an example of a typical chained record data flow.


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

Chained Record Data Flow

ChainedRecordRequest

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

ChainedRecordRequest

ChainedRecordRequest	
Description	Request to send all records from the chain defined by

ChainedRecordRequest	
	RecordHandle.
OCS Action	Send rest of a chain to the desktop.

The ChainedRecordRequest Attached Data table lists the attached data for `ChainedRecordRequest`.

ChainedRecordRequest Attached Data

Data Key	Type	Key Required	Description
GSW_AGENT_REQ_TYPE	String	Yes	`ChainedRecordRequest`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance.
GSW_CAMPAIGN_GROUP_NAME[a]	String	No	Name of the Campaign Group
GSW_CAMPAIGN_NAME	String	No	Name of the Campaign. **Note:** 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[b]	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. ChainedRecord contains more information.

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).

The ChainedRecord Attached Data table lists the attached data for the `ChainedRecord` event.

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 Enumeration Table.)
GSW_CAMPAIGN_NAME	String	No	Name of the Campaign. **Note:** 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:

Data Key	Type	Key Required	Description
			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 Enumeration Table).
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.
GSW_TZ_OFFSET	Int	No	Offset (time difference) in seconds between UTC and a particular time zone. It may contain different values throughout the year if Daylight Savings Time (DST) is used for the specified time zone.
GSW_UNTIL	Int	No	Time until, seconds since midnight.
GSW_CONTACT_MEDIA_TYPE	String	Yes	Describes the method of contact.
GSW_CAMPAIGN _GROUP_NAME	String	Yes	The name of the Campaign Group.
GSW_CAMPAIGN_GROUP__ DESCRIPTION	String	Yes	The description of the Campaign Group. Value may be an empty string.
Custom fields	Any	No	Custom Fields.

ChainedRecordsDataEnd

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

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).

The ChainedRecordsDataEnd Attached Data table lists the attached data for the `ChainedRecordsDataEnd` event.

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.

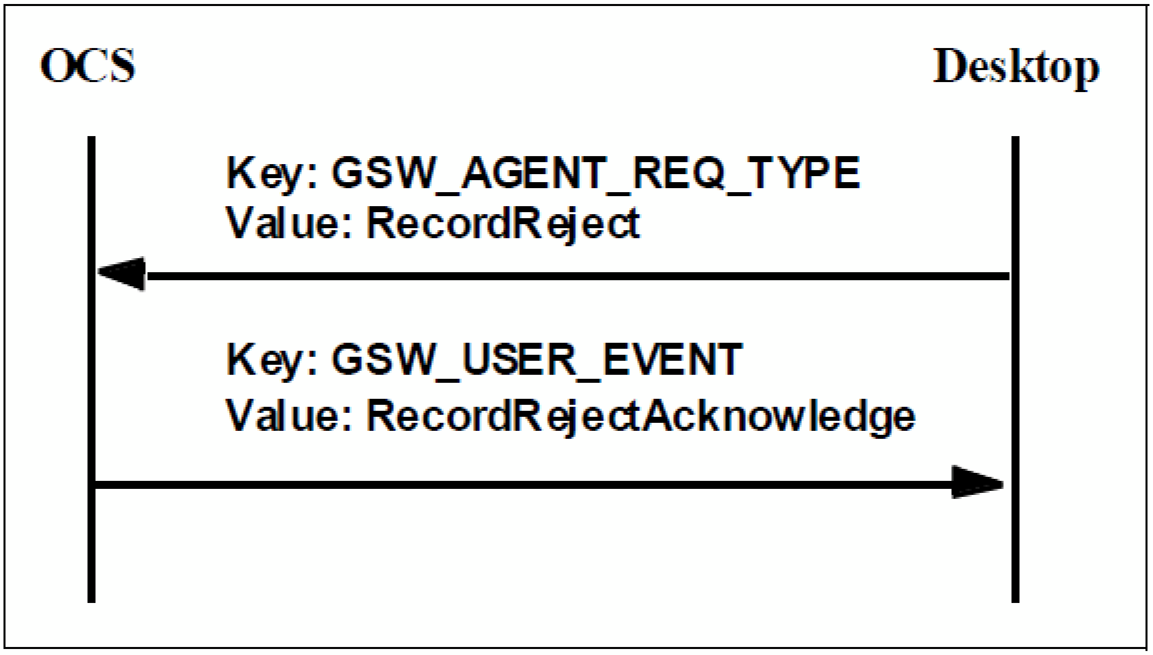
Rejecting Records

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.

The RejectRecord Data Flow diagram shows a typical `RejectRecord` data flow.



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. RecordReject contains more information.

RecordReject

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

The RecordReject Attached Data table lists the attached data for the `RejectRecord` event.

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[a]	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[b]	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. RecordRejectAcknowledge contains more information.

RecordRejectAcknowledge

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

The RecordRejectAcknowledge Attached Data table lists the attached data for the `RecordRejectAcknowledge` event.

RecordRejectAcknowledge Attached Data

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

Canceling Records

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

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

The following are three examples of record cancellations.

Example 1

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

Example 2

1. An agent is working in a *blended* environment (inbound and outbound) and has outbound agent desktop at his disposal.

2. The agent accepts an inbound call and sees that there is no longer a need for an outbound call to the customer.
3. This agent then sends `RequestRecordCancel` (with the phone number `GSW_PHONE`, but without `GSW_RECORD_HANDLE`) to OCS by means of Outbound-Desktop protocol.
4. OCS attempts to find a record that has the matching phone number in OCS memory and in calling lists assigned to loaded Campaigns/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.

> Tip
>
> 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.)
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.

> Tip
>
> 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 Communication DN API.

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 `Canceled` in the database. RequestRecordCancel contains more information.

RequestRecordCancel

RequestRecordCancel Request	
Description	Desktop sends a request to OCS to either: Cancel a record or a chain, in which case the preview record or scheduled call should not be dialed. or Cancel a record or chain for which the call has already been dialed, and the record displayed on the desktop. In both cases, the record should not be re-sent to another agent. It should be marked in the database as `canceled` .
OCS Action	Cancel record.

The RequestRecordCancel Attached Data table lists the attached data for the `RequestRecordCancel` request.

RequestRecordCancel Attached Data

Data Key	**Type**	**Key Required**	**Description**
GSW_AGENT_REQ_TYPE	String	Yes	`RequestRecordCancel`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance. Always required.
GSW_CALLING_LIST	String	No	Name of the calling list. Required only if `GSW_RECORD_HANDLE` is

Data Key	Type	Key Required	Description
			specified.
GSW_CAMPAIGN_GROUP_NAME[a]	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[b]	String	No	A user-defined field in the Calling List table that serves as a customer identifier.
GSW_PHONE[b]	String	No	Customer's phone number.
GSW_RECORD_HANDLE[b]	Int	No	Unique Record Identifier.
GSW_REFERENCE_ID[c]	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] 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).

[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.

Mandatory Fields

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

```
GSW_APPLICATION_ID
```

if (GSW_RECORD_HANDLE is specified)

```
        {
        GSW_CAMPAIGN_NAME
        GSW_CALLING_LIST
        }
    else
        {
        GSW_PHONE
        }
```

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

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

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

Under particular conditions, records with the same `chain_id` are not all cancelled by `RequestRecordCancel`. For more information, see the Filters that Break a Chain of Records section in the *Outbound Contact 8.1 Deployment Guide.*

RecordCancelAcknowledge

OCS sends this event to the desktop to acknowledge a `RequestRecordCancel` event. RecordCancelAcknowledge contains more information.

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.

The RecordCancelAcknowledge Attached Data table lists the attached data for the `RecordCancelAcknowledge` event.

RecordCancelAcknowledge Attached Data

Data Key	Type	Key Required	Description
GSW_USER_EVENT	String	Yes	`RecordCancelAcknowledge`
GSW_APPLICATION_ID	Int	Yes	OCS configuration application database ID. Unique identifier of the running OCS instance. Always required.
GSW_CALLING_LIST	String	No	Name of the calling list. Required only if `GSW_RECORD_HANDLE` is specified.
GSW_CAMPAIGN_NAME	String	No	Name of the Campaign. Required only if `GSW_RECORD_HANDLE` is specified.
GSW_CHAIN_ATTR	String	No	Flag determining whether to update the record chain or just the single record. Values are `AllChain` (default) or `RecordOnly` . See also Record Cancel for `AllChain` when Chain is Broken in the *Outbound Contact 8.1 Deployment Guide*.
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. **Note:** 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.

Important

The mandatory fields for theRequestRecordCancel 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. RecordCancel contains more information.

RecordCancel

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

The RecordCancel Attached Data table lists the attached data for the RecordCancel event.

RecordCancel Attached Data

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

Data Key	Type	Key Required	Description
GSW_CAMPAIGN_NAME	String	Yes	Name of the Campaign.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.
GSW_CHAIN_ATTR	String	No	Flag determining whether to update the record chain or just the single record. Values are AllChain or RecordOnly. (Default = AllChain)
GSW_PHONE	String	Yes	Customer’s phone number.
GSW_CUSTOMER_ID	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.

> **Tip**
>
> 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 Communication DN API.)
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.

Tip

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 Communication DN API. 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.

Important

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. DoNotCall contains more information.

DoNotCall

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

The DoNotCall Attached Data table lists the attached data for the `DoNotCall` request.

DoNotCall Attached Data

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

[a] If `GSW_RECORD_HANDLE` is specified, then `GSW_CALLING_LIST` and `GSW_CAMPAIGN_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_CAMPAIGN_NAME` attribute.

[c] If GSW_RECORD_HANDLE is not specified, then either GSW_PHONE or GSW_CUSTOMER_ID must be present. See Mandatory Fields.

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. DoNotCallAcknowledge contains more information.

DoNotCallAcknowledge

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

The DoNotCallAcknowledge Attached Data table lists the attached data for the `DoNotCallAcknowledge` event.

DoNotCallAcknowledge Attached Data

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

Data Key	Type	Key Required	Description
GSW_RECORD_HANDLE	Int	No[a]	Unique Record Identifier.
GSW_DURATION	Int	No	DNC duration, in seconds [c] Related Documentation: Do Not Call Duration

[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.

[c] OCS adds this value to the current time and stores the result as the expiration date in the Do Not Call table.

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. RecordReschedule contains more information.

> **Important**
>
> A callback is not scheduled at the time the 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.

RecordReschedule

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

The RecordReschedule Attached Data table lists the attached data for a `RecordReschedule` event.

RecordReschedule Attached Data

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

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.
GSW_REFERENCE_ID[b]	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.RecordRescheduleAcknowledge contains more information.

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).

The RecordRescheduleAcknowledge Attached Data table lists the attached data for a `RecordRescheduleAcknowledge` event.

RecordRescheduleAcknowledge Attached Data

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

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.

The ScheduledRecordReschedule Attached Data table lists the attached data for a `ScheduledRecordReschedule` event.

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[a]	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.
GSW_PHONE	String	No	Customer's phone number.
GSW_RECORD_HANDLE	Int	Yes	Unique Record Identifier.
GSW_REFERENCE_ID[b]	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. ScheduledRecordRescheduleAcknowledge contains more information.

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).

The ScheduledRecordRescheduleAcknowledge Attached Data table lists the attached data for a `ScheduledRecordRescheduleAcknowledge` event.

ScheduledRecordRescheduleAcknowledge Attached Data

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

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 `send_attribute` option configured.

The ScheduledCall Attached Data table lists the attached data for a `ScheduledCall` event.

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 Enumeration Table.)
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

Data Key	Type	Key Required	Description
			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 Enumeration Table).
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 AddRecord Attached Data). 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.

> ### Important
>
> OCS processes an `AddRecord` request with the attribute `chain_n = 0` as a request to add a new record to a chain, which is also how OCS processes the request if the `chain_n` attribute is not specified. If an agent specifies the `chain_n` attribute as not equal to 0 in the request, OCS interprets this as a request to add a specific record with the `chain_id` and `chain_n` attributes as defined in the request. This type of request will fail if the record with the `(chain_id, chain_n)` pair is already present in the calling list.

AddRecord contains more information.

AddRecord

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

The AddRecord Attached Data table lists the attached data for an `AddRecord` request.

AddRecord Attached Data

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

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 Enumeration Table). (Default = 2, DirectBusinessPhone)
GSW_RECORD_HANDLE[b]	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 Enumeration Table). Optional. (Default = 1, ready)
GSW_RECORD_TYPE	Int	No	Type of added record sent from a desktop. See the “Genesys Enumeration Table" on Enumeration Table. Optional. (Default = 2, general)
GSW_REFERENCE_ID[c]	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.

Data Key	Type	Key Required	Description
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. AddRecordAcknowledge contains more information.

AddRecordAcknowledge

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

The AddRecordAcknowledge Attached Data table lists the attached data for an `AddRecordAcknowledge` event.

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), agents are assigned to the Inbound activity.

Agent Logout Protocol

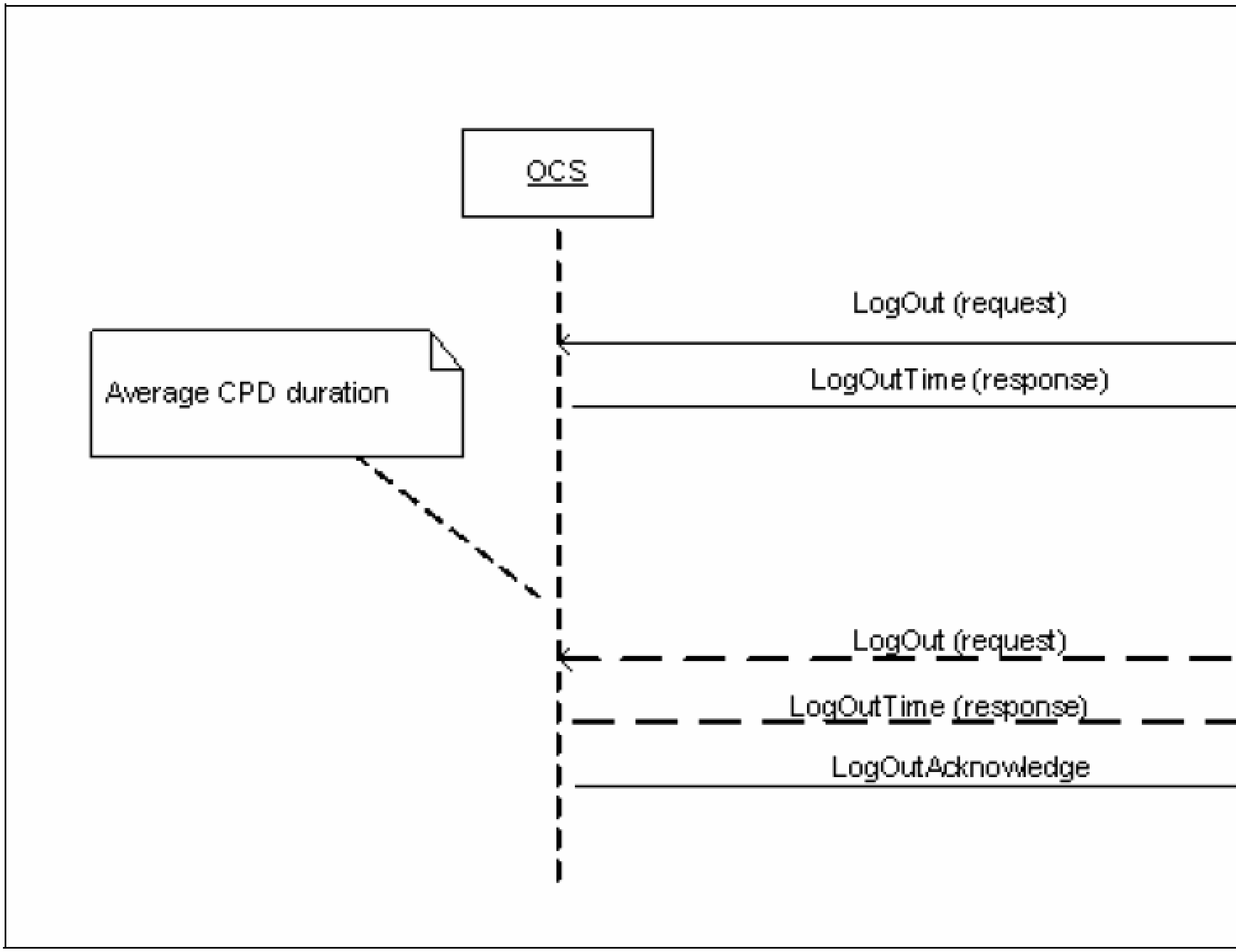
The extension of the Desktop Protocol (LogOut) addresses the issue of abandoned or dropped calls as a result of this combination of circumstances:

- 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. LogOutTime. 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.

The Logout Negotiation between Agent Desktop and OCS diagram shows the Desktop-OCS user events (request and responses) for an agent logout.



Logout Negotiation between Agent Desktop and OCS

LogOut

LogOut provides information on this event.

LogOut

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

The LogOut Attached Data table lists the attached data for the `LogOut` event.

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[a]	String	No	Name of the Campaign Group.
GSW_CAMPAIGN_NAME	String	Yes	Name of the Campaign.
GSW_REFERENCE_ID[b]	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. LogOutTime contains more information.

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.

The LogOutTime Attached Data table lists the attached data for the LogOutTime event.

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

LogOutAcknowledge provides information on this event.

LogOutAcknowledge

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

The LogOutAcknowledge Attached Data table lists the attached data for the LogOutAcknowledge event.

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

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

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.

The Media Type Business Attribute table provides information about identifying the media types that correspond to the `Media Type` business attribute. This defines how to contact the customer.

Media Type Business Attribute

Data Key	**Type**	**contact_info_type Field**	**Description**
GSW_CONTACT_MEDIA_TYPE	string	0	any (NoContactType)
		1	voice (HomePhone)

Data Key	Type	contact_info_type Field	Description
		2	voice (DirectBusinessPhone)
		3	voice (BusinessWithExt)
		4	voice (Mobile)
		5	voice (VacationPhone)
		8	voice (VoiceMail)
		10	email (E-mail)
		11	instant message

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.

> **Important**
> 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

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. Outbound Contact requires that agents be in a Ready state to receive outbound calls.

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 agent state notifications (`Logged In`, `Logged Out`, `Ready`, `Not Ready`) for Places associated with IVR Groups that only contain DNs of type Voice Treatment Port. (To determine if Voice Treatment Port type DNs are supported on your switch, refer to the documentation for the T-Server being used for outbound dialing.)
- 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.

> **Important**
>
> 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

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

Outbound Contact Library

The following section describes:

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

Error Names and Codes

The Error Names and Codes table displays error names and their corresponding codes for error conditions that occur while using communication protocols.

Error Names and Codes

GSW_ERROR	GSW_ ERROR_ NUMBER	Description
Invalid Request	101	Received request has the wrong request type.[a]
Attribute Not Found	102	Mandatory attribute cannot be found.

GSW_ERROR	GSW_ ERROR_ NUMBER	Description
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.

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

The All Desktop Protocol Events and Event Type Protocols table represents all Genesys event and event type protocols.

> **Important**
>
> 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.

All Desktop Protocol Events and Event Type Protocols

Messages	From > To	Descriptions and Actions
1. Notifications		
CampaignStarted	O > D	Should be sent when Campaign dialing is started or resumed, or as a response to event agent login if 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.

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.
2. Preview		
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.
3. Common		
UpdateCallCompletionStats	D > O	Desktop sends this event to update record details. Intermediate update.
UpdateCallCompletionStatsAcknowledge	O > D	OCS sends this event to confirm operation.
ReadyTime	D > O	Desktop sends this request to OCS, providing the number of seconds before the agent will go `Ready.`
ReadyTimeAcknowledge	O > D	OCS sends this event to the desktop to acknowledge the `ReadyTime` request.
RecordProcessed	D > O	Desktop sends this event to indicate that record is processed. OCS should update record if it is provided.

Messages	From > To	Descriptions and Actions
RecordProcessedAcknowledge	O > D	OCS confirms that record has been executed.
RecordReject	D > O	Desktop sends this request to indicate that the preview record or scheduled call will not be dialed by this agent. This record should be re-sent to another agent.
RecordRejectAcknowledge	O > D	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.
4. Chained Records		
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.
5. DoNotCall		
DoNotCall	D > O	Agent requests the number or customer ID in a record not to be called again.

Messages	From > To	Descriptions and Actions
DoNotCallAcknowledge	O > D	Confirmation that DoNotCall was accepted.
6. Record Cancel from OCS to Desktop		
RecordCancel	O > D	OCS sends this to the desktop to indicate that this record should not be dialed. Applicable for preview records and scheduled calls.
7. Request add record from Desktop		
AddRecord	D > O	Request to add a new record to the database.
AddRecordAcknowledge	O > D	Phone number can be used to relate request and response.
8. Request LogOut		
LogOut	D > O	Agent’s request to log out
LogOutTime	O > D	OCS response to LogOut request
LogOutAcknowledge	O > D	Automatic logout acknowledgement

Communication DN API

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

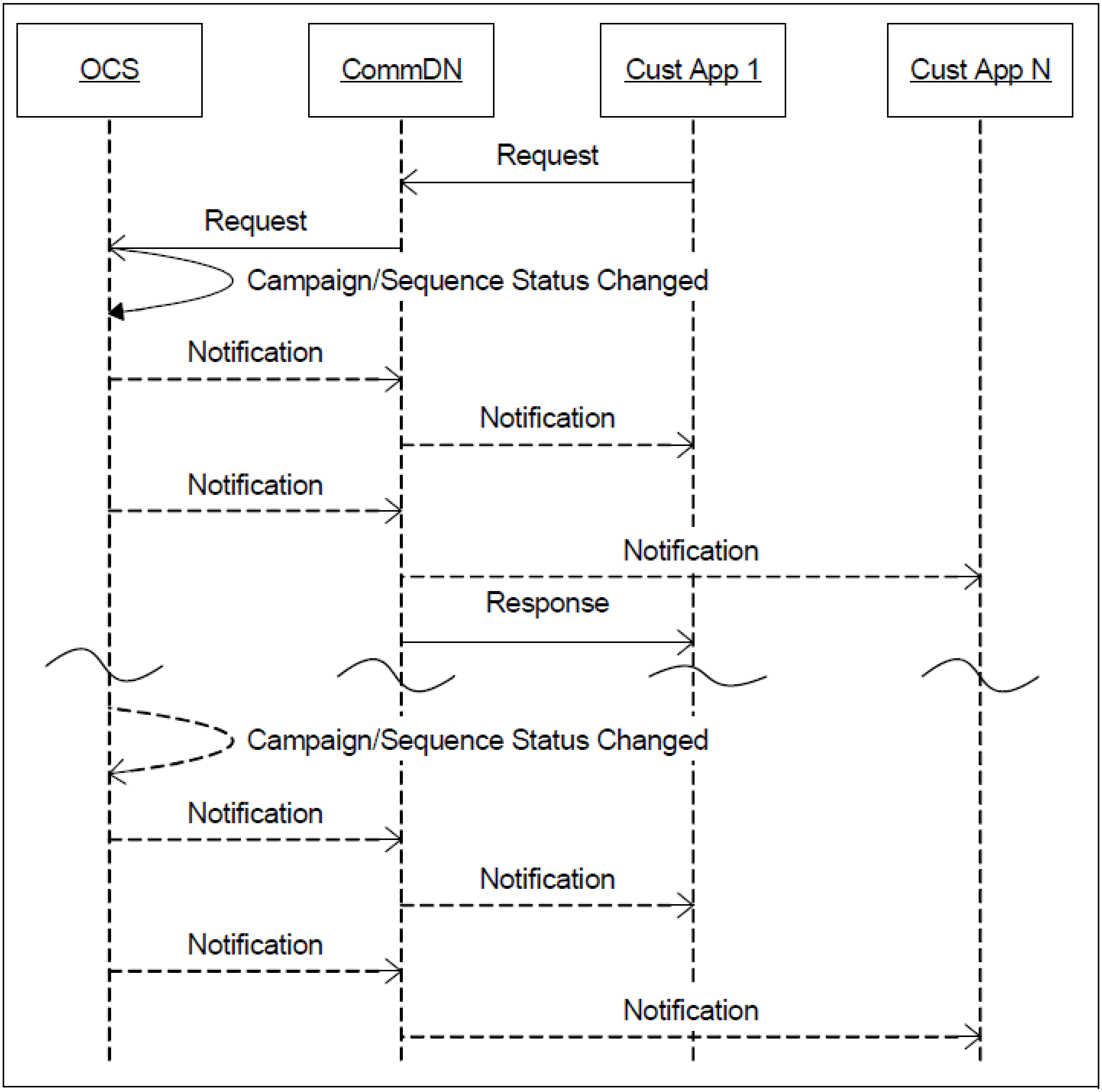
OCS and third-party applications communicate through the Communication DN API by means of `UserEvents` (with attached user data) that are sent and received on a CommDN. The attached user data is encoded as a key-value pairs list (TKVList). Values can be either string or integer. These values are described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

Protocol Sequencing for the Communication DN API shows the messaging sequence of the Communication DN API protocol.



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 Communication Protocols.

- TargetAppDBID (the DBID of the receiver)

Communication Structure 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.

Communication Structure

Request	Response/Notification	Mandatory Attributes
CM_ReqLoadCampaign	CM_EvCampaignLoaded	CampaignDBID or Properties OCS checks CampaignDBID . If the value is 0 , the request or event must have the proper schedule in the Properties attribute. GroupDBID DialMode OptimizeBy OptimizeGoa l
CM_ReqUnloadCampaign	CM_EvCampaignUnloaded	CampaignDBID GroupDBID
CM_ReqGetCampaignStatus	CM_EvCampaignStatus	Request CampaignDBID or Properties OCS checks CampaignDBID. If the value is 0, the request or event must have the proper schedule in theProperties attribute. If CM_ReqGetCampaignStatus CampaignDBID equals0 , OCS responds with the sequence status. GroupDBID Response or Notification CampaignDBID or Properties GroupDBID DialMode OptimizeBy OptimizeGoal GroupCampStatus. If a campaign belongs to a Sequence, then the attribute GroupCampStatus status represents the part of each scheduleItem (<n> in the Sequence. See User Event Structure for the Communication DN API.

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

Special OCS Option

Usually OCS works with all existing CommDNs in the Configuration Database. You can reduce the number of CommDNs OCS uses by assigning the outbound_contact_server option to these DNs. Set this option's value totrue 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.

- 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.

- 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.

- 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`

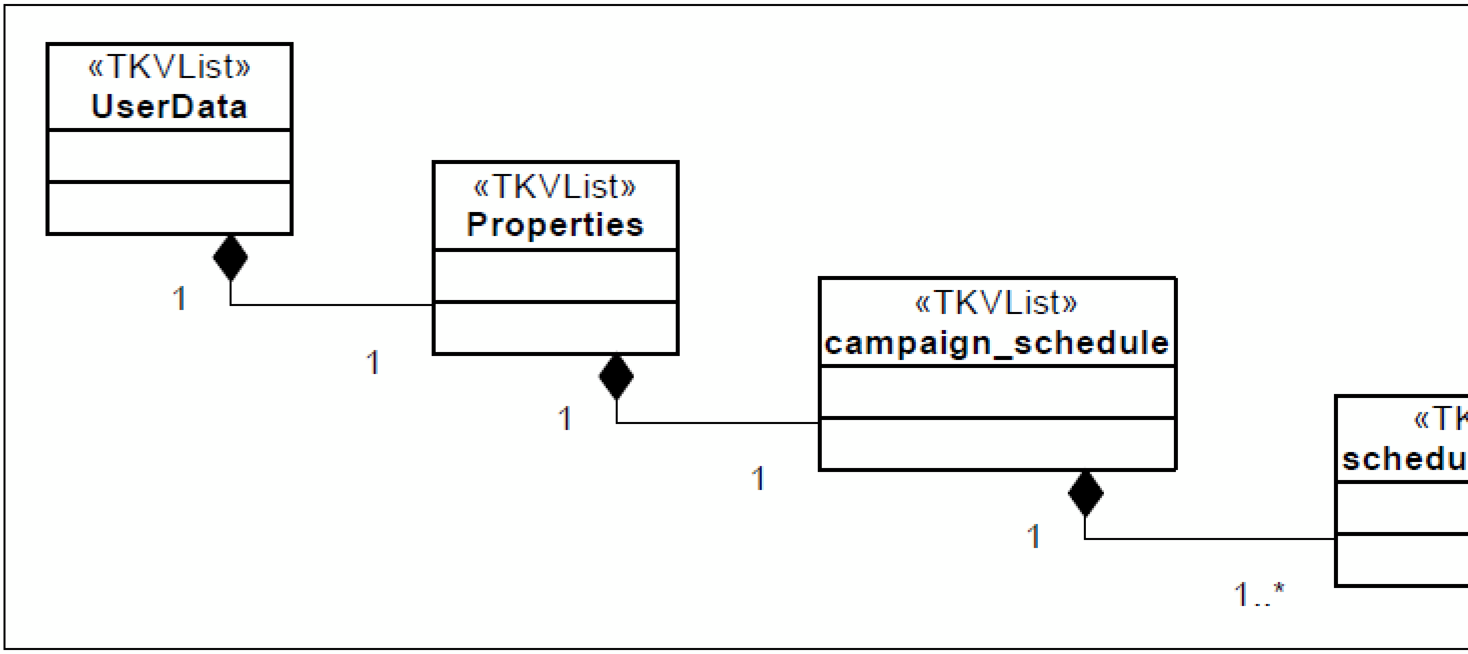
CommDN_2: `outbound_contact_server = undefined`

CommDN_3: `outbound_contact_server = undefined`

In this configuration, OCS uses all CommDNs.

User Event Structure

User Event Structure for the Communication DN API shows the user event structure for communication between third-party applications and the Communication DN API.



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 . . . n Items. For more information about the user event, campaign_schedule, see User Event Attributes for campaign_schedule (TKVList).

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). User Event Attributes for User Data (TKVList) lists the Genesys user event attributes sent with user data through the Communication DN API.

User Event Attributes for User Data (TKVList)

Key	Type	Description
GSW_CM_MessageType	Int	See the Enum values for GSW_CM_MessageType on Data Enumeration Values for GSW_CM_MessageType.
GSW_CM_AttrDialMode	Int	See the Enum values for GSW_CM_AttrDialMode on Enumeration Values for GSW_CM_AttrDialMode.
GSW_CM_AttrOptimizeBy	Int	See GSW_CM_AttrOptimizeBy Enum values on Enumeration Values for GSW_CM_AttrOptimizeBy.
GSW_CM_AttrOptimizeGoal	Int	Values from `0 - 100` percent, or from `0` to <as required> seconds, represent target values for the Optimization parameter.
GSW_CM_AttrGroupCampStatus	Int	See the Enum values for GSW_CM_AttrGroupCampStatus on page Enumeration Values for GSW_CM_AttrGroupCampStatus.
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 Enumeration Values for GSW_CM_AttrError.
GSW_CM_AttrErrorMessage	String	String describing the error that occurred.
GSW_CM_AttrOriginAppID	Int	Application DBID.
GSW_CM_AttrTargetAppID	Int	Application DBID.
GSW_CM_AttrProperties	TKVList	Attribute’s properties.
GSW_AUTO_COMPLETION	Int	This attribute is used with `CampaignLoad`, `SetDialingMode`, and `StartDialing` events. A value of 1 indicates that OCS

Key	Type	Description
		should stop the Campaign Group automatically, via `Stop` and `Unload` actions, after all records in a calling list are depleted. If the value is **0**, the functionality is disabled. See Campaign Group Auto Completion for more information.

User Event Attribute for GSW_CM_AttrProperties (TKVList) shows `UserEvent` attributes for `GSW_CM_AttrProperties` (TKVList).

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

User Event Attributes for campaign_schedule (TKVList) shows `UserEvent` attributes for `campaign_schedule` (TKVList).

User Event Attributes for campaign_schedule (TKVList)

Key	Type	Description
Description	String	Description of Campaign Sequence
startTime	Int	Time to start the Sequence (UTC)

Key	Type	Description
nItems	Int	Number of items in the Sequence
scheduleItem-<n>	TKVList	Properties of the Sequence item <n>

User Event Attributes for dialing_priority (TKVList) shows `UserEvent` attributes for `dialing_priority` (TKVList).

User Event Attributes for dialing_priority (TKVList)

Key	Type	Description
General	TKVList	Contains a list of the following key-value pairs: • `priority =<value>` • `n_records=<value>`
CampaignRescheduled	TKVList	Contains a list of the following key-value pairs: • `priority =<value>` • `n_records=<value>`
CampaignCallBack	TKVList	Contains a list of the following key-value pairs: • `priority =<value>` • `n_records=<value>`

User Event Attributes for scheduleItem-<*n*> shows the `UserEvent` attributes for scheduleItem-<*n*>.

User Event Attributes for scheduleItem-<*n*>

Key	Type	Description
stopAtTime	Int	Stop the campaign at a specified time.

Key	Type	Description
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 Enumeration Values for GSW_CM_AttrDialMode.
optMethod	Int	Optimization method for the campaign. See the Enum values for GSW_CM_AttrOptimizeBy on Enumeration Values for GSW_CM_AttrOptimizeBy.
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 Enumeration Values for GSW_CM_AttrGroupCampStatus.

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

Data Enumeration Values for GSW_CM_MessageType 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.

Data Enumeration Values for GSW_CM_MessageType

Data	Value	Comment
Messages That OCM/Genesys Administrator Uses to Communicate with OCS		
MSGCFG_NONE	0	not applicable
MSGCFG_UNKNOWN	1	not applicable
MSGCFG_ERROR	2	not applicable
MSGCFG_CLIENTREGISTER	3	not applicable
MSGCFG_DISCONNECTED	4	not applicable
CM_UnknownMessage	5	not applicable
Requests		
CM_ReqRegisterClient	6	not applicable
CM_ReqLoadCampaign	7	Request to load a campaign.
CM_ReqUnloadCampaign	8	Request to unload a campaign.
CM_ReqStartDialing	9	Request to start dialing a campaign.
CM_ReqStopDialing	10	Request to stop dialing a campaign.
CM_ReqSetDialingMode	11	Request to change dialing parameters for a campaign.
CM_ReqGetCampaignStatus	12	Request for campaign status.
CM_ReqCampaignRegistered	13	not applicable
CM_ReqCampaignUnregistered	14	not applicable

Data	Value	Comment
CM_ReqForceUnloadCampaign	29	Request to force campaign unloading.
CM_ReqCancelRecord	30	Request to cancel record from third-party application
CM_ReqDoNotCall	32	Request to add phone number of customer ID to Do-Not-Call List.
Responses		
CM_EvServerConnected	15	not applicable
CM_EvServerDisconnected	16	not applicable
CM_EvClientDisconnected	17	not applicable
CM_EvClientRegistered	18	not applicable
CM_EvCampaignLoaded	19	Acknowledge for request CM_ReqLoadCampaign.
CM_EvCampaignUnloaded	20	Acknowledge for request CM_ReqUnloadCampaign.
CM_EvDialingStarted	21	Acknowledge for request CM_ReqStartDialing.
CM_EvDialingStopped	22	Acknowledge for request CM_ReqStopDialing.
CM_EvDialingModeChanged	23	Acknowledge for request CM_ReqSetDialingMode
CM_EvCampaignStatus	24	Response or Notification when campaign mode is changed.
CM_EvCampaignRegistered	25	not applicable
CM_EvCampaignUnregistered	26	not applicable
CM_EvError	27	Wrong event error received.

Data	Value	Comment
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

Enumeration Values for GSW_CM_AttrError displays the Enumeration values for the user data GSW_CM_AttrError.

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.

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.
CM_ERROR_LICENSE_VIOLATION	15	Dial mode is not supported by current license.

Enumeration Values for GSW_CM_AttrDialMode shows the Enumeration values for the user data GSW_CM_AttrDialMode.

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

Enumeration	Value	Comment
CFGDMProgressGVP	9	Progressive GVP
CFGDMPowerGVP	11	Power GVP

Enumeration Values for GSW_CM_AttrOptimizeBy shows the Enumeration values for the user data GSW_CM_AttrOptimizeBy.

Enumeration Values for GSW_CM_AttrOptimizeBy

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

Enumeration Values for GSW_CM_AttrGroupCampStatus shows the Enumeration values for the user data GSW_CM_AttrGroupCampStatus.

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

Enumeration	Value	Comment
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_EvRecordCanceled`

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

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

UserEvent Structure

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

```
UserEvent
    I UserData

    I "GSW_CM_MessageType" 30

    ["GSW_CM_AttrError" 0]

    "GSW_CM_AttrOriginAppID" <value>
```

"GSW_CM_AttrTargetAppID" <value>

"GSW_CM_AttrProperties"
|

"cancel_record"
|

"GSW_PHONE" <value>

["GSW_CAMPAIGN_NAME" <value>]

["GSW_CHAIN_ATTR" <value>]

["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 User Event Attributes for User Data (TKVList) and Reserved Keys.

UserEvent Attributes

The `UserEvent` attributes in UserData (TKVList) pertain to the Record Cancel feature. GSW_CM_AttrProperties (TKVList) and cancel_record (TKVList) provide information on `GSW_CM_AttrProperties` and `cancel_record`. Also see User Event Attributes for User Data (TKVList).

UserData (TKVList)

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

Key	Type	Description
GSW_CM_AttrProperties	TKVList	See GSW_CM_MessageType Enum below.

GSW_CM_AttrProperties (TKVList)

Key	Type	Description
cancel_record	TKVList	Contains additional request attributes

cancel_record (TKVList)

Key	Type	Description
GSW_PHONE	String	Phone Number
GSW_CAMPAIGN_NAME	String	Campaign Name If specified, only records in this campaign will be canceled.
GSW_CHAIN_ATTR	String	AllChain, RecordOnly Specifies the scope of the request. `AllChain` is the default value
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. **Note:** 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. Data Enumerations and GSW_CM_AttrError Errors provide information on data enumerations and `GSW_CMAttrError` respectively. In addition, see Data Enumeration Values for GSW_CM_MessageType.

Data Enumerations

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

GSW_CM_AttrError 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 requestCM_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
     |

     UserData
          |

          "GSW_CM_MessageType" 32

          ["GSW_CM_AttrError" 0]

          "GSW_CM_AttrOriginAppID" <value=sender’s ID>

          "GSW_CM_AttrTargetAppID" <value=receiver’s ID>

          "GSW_CM_AttrProperties"
               |

               "do_not_call"
                    |

                    "GSW_PHONE" <value>

                    ["GSW_CUSTOMER_ID" <value>]

                    ["GSW_CHAIN_ATTR" <value>]
```

In this example, under `UserData`, the value of `GSW_CM_MessageType` is 32 for the request `CM_ReqDoNotCall`. The value would be33 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.

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).

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*.

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.

- `<parameters>`: The parameter portion of the URI defines the action for a resource; see the Supported Parameters section.

Supported Parameters

Query Parameter describes the supported parameters.

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 HTTP Request Actions and req Values for records Resource through HTTP Request Actions and req Values for the campaigngroups Resource 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.

Client Requests

OCS HTTP supports client requests using the POST method and, starting with version 8.1.524.04, using the GET method. 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. The GET Campaign Groups status is supported for all Campaign Groups and for an individual Campaign Group.

Client Request Headers

The POST request headers must be populated by the client in the POST request. POST Request Headers lists those headers. OCS HTTP ignores all other headers.

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 Note: 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

HTTP Request Actions and req Values for records Resource 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).

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.
DoNotCall	DoNotCall	Can be used for the entire chain or one record.
RecordReschedule	RecordReschedule	Non-finalizing for the Desktop Protocol, as

Action as Defined by the Outbound Contact Desktop Protocol	req Value	Note
		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.

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

HTTP Request Actions and req Values for the phones and customer_ids Resources 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 and Submitting DoNotCall Requests, respectively.)

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

Action as Defined by Outbound Contact Desktop Protocol	req Value	Note
		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.

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

HTTP Request Actions and req Values for the lists Resource 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

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. In addition, it must comply with the following:

- Be a full JSON representation (see See 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:).
- 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 Requests 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

HTTP Request Actions and req Values for the campaigngroups Resource 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.

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 BODY Attributes for campaigngroups Resource).
`GSW_CM_ReqSetDialingMode`	`SetDialingMode`	Must have additional parameters in the body of the HTTP request (see BODY Attributes for campaigngroups Resource).
`GSW_CM_ReqStopDialing`	`StopDialing`	
`GSW_CM_ReqUnloadCampaign`	`Unload`	
`GSW_CM_ReqForceUnloadCampaign`	`ForceUnload`	
`GSW_CM_ReqLaunchCampaign`	`Launch`	Must have additional parameters in the body of the HTTP request (see BODY Attributes for campaigngroups Resource).
`GSW_CM_ReqStageCampaign`	`Stage`	
`GSW_CM_ReqCompleteCampaign`	`Complete`	

BODY

The BODY must follow the format described in Client Request BODY. BODY Attributes for campaigngroups Resource also provides a list of attributes that can be included in the BODY of the request for the `campaigngroups` resource.

BODY Attributes for campaigngroups Resource

Attribute	Value	Description
`GSW_CM_AttrDialMode`	Integer	The dial mode enumeration value (see Enumeration Values for GSW_CM_AttrDialMode).
`GSW_CM_AttrOptimizedBy`	Integer	The optimization parameter type enumeration value (see Enumeration Values for GSW_CM_AttrOptimizeBy).
`GSW_CM_AttrOptimizeGoal`	Integer	The optimization parameter target value (see User Event Attributes for User Data (TKVList)).

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.

OCS HTTP Responses to Records, Phones, Customer IDs, Lists-Related Requests summarizes those responses for all resources but Campaign Groups.

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

OCS HTTP Responses to Campaign Groups-Related Requests summarizes the response for Campaign Group actions.

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	

OCS HTTP Responses to lists/AddRecord Requests with No Specified Record Handle summarizes the response for list/`AddRecord` requests.

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	

OCS HTTP Response Headers lists the headers in the OCS HTTP response that may be populated by OCS for all requests.

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
```

Successful request for individual Campaign Group status

1. The client sends the following request to OCS via the HTTP interface (where 115 is the DBID of the Campaign Group in question):

```
GET http://ocs.genesyslab.com:8080/campaigngroups/115 HTTP/1.1
Host: ocs.genesyslab.com
User-Agent: ORS
Accept: */*
```

2. If the specified Campaign Group is known to OCS, OCS HTTP replies with a similar message:

```
HTTP/1.1 200 OK
Server: OCS/8.1.524.03
Cache-control: private
Etag: 1569882947
Content-type: application/json
Content-length: 323

{
  "GSW_CM_MessageType": 24,
  "GSW_CM_AttrCampaignID": 139,
  "GSW_CM_AttrGroupID": 234,
  "GSW_CM_AttrDialMode": 1,
  "GSW_CM_AttrOptimizeBy": 1,
  "GSW_CM_AttrOptimizeGoal": 80,
  "GSW_CM_AttrGroupCampStatus": 5,
  "GSW_CM_AttrProperties": {
    "camp_group_data": {}
  },
  "GSW_SWITCH_DBID": 105,
  "GSW_CM_AttrOriginAppID": 398
}
```

Successful request for statuses of all Campaign Groups

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

```
GET http://ocs.genesyslab.com:8080/campaigngroups HTTP/1.1
Host: ocs.genesyslab.com
User-Agent: ORS
Accept: */*
```

2. OCS HTTP replies with a similar message containing information for all Campaign Groups known to OCS (that is active within OCS and configured, but not active).

```
HTTP/1.1 200 OK
Server: OCS/8.1.524.03
Cache-control: private
Etag: 1569882947
Content-type: application/json
Content-length: 763

{
  "GSW_CM_MessageType": 46,
  "CampaignGroups": {
    "Campaign_Group_1@Agent_Group_1": {
      "GSW_CM_AttrCampaignID": 146,
      "GSW_CM_AttrGroupID": 234,
      "GSW_CM_AttrGroupCampStatus": 0
    },
    "Campaign_Group_2@Agent_Group_2": {
      "GSW_CM_AttrReferenceID": 6,
      "GSW_CM_AttrCampaignID": 139,
      "GSW_CM_AttrGroupID": 234,
      "GSW_CM_AttrDialMode": 1,
      "GSW_CM_AttrOptimizeBy": 1,
      "GSW_CM_AttrOptimizeGoal": 80,
      "GSW_CM_AttrGroupCampStatus": 5,
      "GSW_CM_AttrProperties": {
        "camp_group_data": {}
      },
      "GSW_SWITCH_DBID": 105
    },
    "Campaign_Group_3@Agent_Group_3": {
      "GSW_CM_AttrCampaignID": 138,
      "GSW_CM_AttrGroupID": 162,
      "GSW_CM_AttrGroupCampStatus": 0
    }
  },
  "GSW_CM_AttrOriginAppID": 398
}
```

Defined Constants

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 Enumeration Table.

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

Outbound Contact Mandatory Fields in the Default Format provides a description of the Genesys mandatory fields in the Default Outbound Contact format.

Outbound Contact Mandatory Fields in the Default Format

Column Name	Data Type	Description
`record_id`	integer	Unique identification number of a calling record.
`contact_info`	varchar(128)	Customer's contact information, phone number in the voice campaign.
`contact_info_type`	integer	Type of contact information, phone type in the voice campaign. See Contact Information Types.
`record_type`	integer	Type of the record. See Record Types.
`record_status`	integer	Current status of the record. See Record Statuses.
`call_result`	integer	Final outcome of the record processing. See Call Result Types.
`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).

Column Name	Data Type	Description
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.
media_ref	integer	Empty, not used at this time.
switch_id	integer	DBID of the Switch where the agent who handled the record had logged in.
treatments	varchar(255)	Treatments application history. For more information, see the “Treatments" chapter in the *Outbound Contact 8.1 Deployment Guide.*

Call Results

The final outcome of a call is stored in the calling list from which the call is dialed. You can view call results for records in calling lists in 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.

Call Result Types shows the call result types.

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.
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.

Call Result	Enumeration Value	Description
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: • `call_wait_in_queue_timeout` • `call_wait_original_establish_timeout`
Dropped on No Answer	27	Call has been dropped or released from an established three-way call before being answered.
Fax Detected	17	Fax machine was detected at the dialed phone number.
Forwarded	23	Reserved for future use.

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.

Call Result	Enumeration Value	Description
Queue Full	18	Reserved for future use.
Redirected	22	Reserved for future use.
RemoteRelease	5	Call was released in response to an error on the switch or no contact with called party.
Silence	32	Call was dialed; however, there 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 `Stale` in the following scenario: 1. The following timer has expired: `stale_clean_timeout` 2. OCS marks the call result as `Stale` if an outbound call was transferred from:

Call Result	Enumeration Value	Description
		• a queue to a DN which is either not registered for OCS or does not have a logged in agent. • an agent in an outbound campaign to a DN that is not registered for OCS.
Switch Error	43	No dial tone received.
System Error	4	Dialing software error from the Dialogic driver or call progress detection (CPD) from the switch.
Transfer Error	45	Dialer has a problem transferring calls based on call action.
Transferred	1	Reserved for future use.
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 Call Result Types, 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 Call Result Types.) 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 Call Result Types, with the exception of Unknown and OK call results. The exception also applies to the Power mode (SMSs and E-Mails) for which OCS applies a treatment to the OK call result.

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.
Call Result Mappings Created by OCS shows examples of Call Result Mappings created by OCS.

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

Conflicting Call Result Mappings 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.

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.

Dialogic Result	T-Server Result	Possible Cause
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.
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 Data Type Values for data type values.

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 Contact Information Types.

Contact Information Types

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

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

Record Types

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

Record Types

Record Type	Description
Campaign CallBack	Used by the agent to reschedule a call and have the call delivered to any agent in the Campaign Group when the callback occurs. See also the `predictive_callback` option.
Campaign Rescheduled	Default value for a record that has been rescheduled by a call treatment. **Note:** 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 by applying a treatment that allows 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 Enumeration Table.

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 Record Statuses.

Record Statuses

Record Status	Description
Agent Error	An outbound call was distributed to a DN not monitored by OCS.

Record Status	Description
	OCS is not registered on that DN.
Canceled	A record has been canceled by Agent request or through the Communication DN Protocol.
Chain Ready	Reserved for future use
Chain Updated	Reserved for future use
Missed CallBack	Personal CallBack or Campaign CallBack has been missed and treatment action “Mark As Agent Error" was applied to record. Also see the `predictive_callback` option.
No Record Status	Record status is not set; that is, the call record is not ready to be dialed.
Ready	Default value; The record is ready to be dialed.
Retrieved	A record is retrieved from the database and is in the active dialing queue.
Stale	Outbound Contact has not received acknowledgment of any user event sent to the agent’s desktop application regarding this record. For all records that are considered stale (no longer useful) due to a desktop crash, and for all records that remain on the desktop without being updated in the database: OCS changes the `record_status` to `Stale` when a campaign is unloaded. The record status is marked as stale in the following scenarios: • 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`. • 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`. • 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`.

Record Status	Description
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 Enumeration Table.

Combining Record Statuses and Record Types

Record statuses and record types can be combined as shown in Record Status and Record Type Combinations.

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

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

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

Recommended DBMS Optimizations

This topic explains how to perform recommended DBMS optimizations.

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 dbspace where temporary tables reside. You may need to increase this size. The dbspace where temporary tables are stored is defined by the “`DBSPACETEMP" ONCONFIG` parameter.

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

DB2

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

If this table space is absent, an attempt to open a Calling List in 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 <`pagesize`> exists, and that the authorization ID <`user-name`> has USE privilege on this table space.

Oracle

Oracle 8.1 and below: On versions of Oracle before 8.1.7, Oracle does not use temporary tables, and OCM 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 Sever product documentation or your Database Administrator.

Sybase

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

```
UPDATE STATISTICS <calling_list_table_name>
```

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

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

Oracle

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

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

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

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

Informix

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

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

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

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

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.

Handling of the Backslash Escape Sequences

OCS stores the data in the DBMS as is, without processing of the backslash escape sequences. The DBMS must be configured to omit the processing of the backslash escape sequences; otherwise, the data might be altered in the DBMS or some unexpected errors might occur on the DBMS side while processing the queries containing any backslash escape sequences—for example,"`\0`".

Supported Functionality with IP Telephony

This topic describes the IP telephony functionality that Outbound Contact supports.

Overview

This topic 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.*

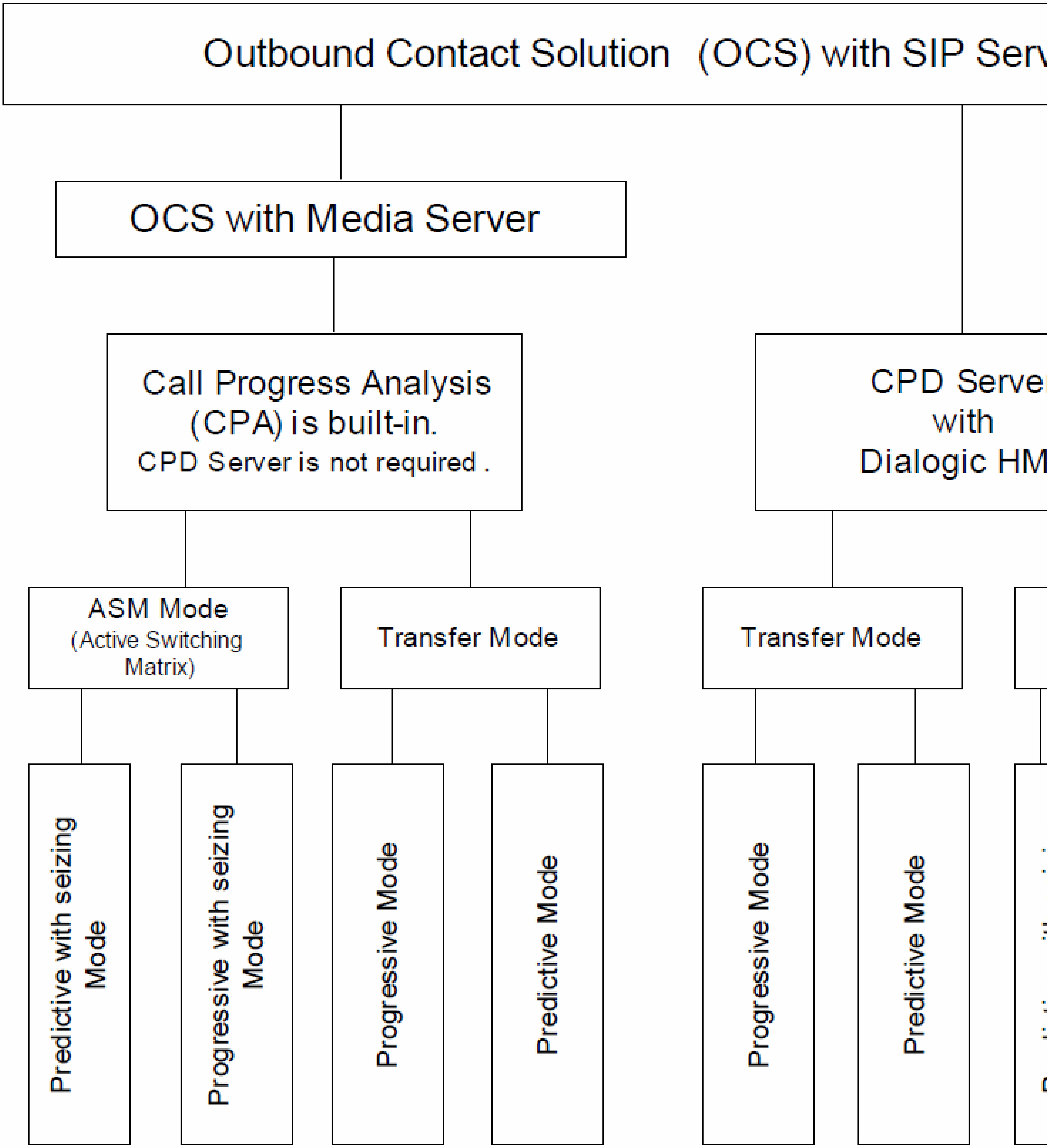
`Power GVP` dialing mode supports the Proactive Contact Solution. For more information on the Proactive Contact Solution, see the *Proactive Contact 8.0 Solution Guide*, which is located on the Genesys Technical Support web site and the Genesys Documentation Library DVD.

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

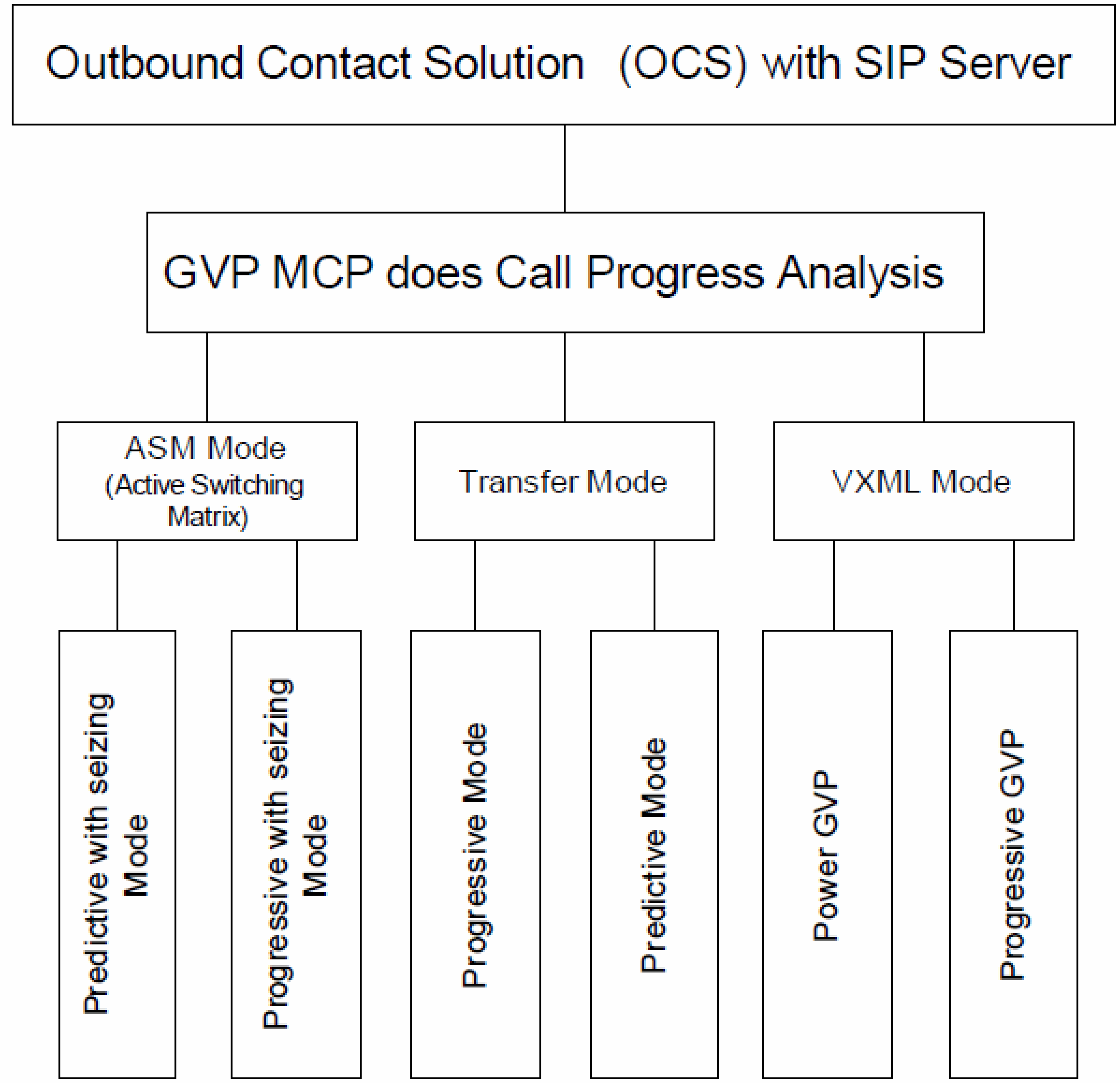
- Using CPD Server/Dialogic board as a dialer in a `Transfer` mode.
- Using CPD Server/Dialogic board as a dialer in the `ASM` mode.
- Using CPD Server/Dialogic's Host Media processing (HMP) software as a dialer.

- OCS and T-Server for Cisco Unified Communications Manager 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.

OCS with SIP Server Overview with Media Server or CPD Server and OCS with SIP Server Overview with GVP MCP provide an overview of each of these scenarios.


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

OCS with SIP Server Overview with Media Server or CPD Server


Outbound Contact Solution (OCS) with SIP Server
GVP MCP does Call Progress Analysis
ASM Mode (Active Switching Matrix)
Transfer Mode
VXML Mode
Predictive with seizing Mode
Progressive with seizing Mode
Progressive Mode
Predictive Mode
Power GVP
Progressive GVP

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.

Note:	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)
- Transfer Mode (MGW without CPA)
- Transfer Mode (MCP as the CPA Provider)
- ASM Mode (MCP as the CPA Provider)
- ASM Mode (MGW with CPA)
- ASM Mode (MGW without CPA)

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.

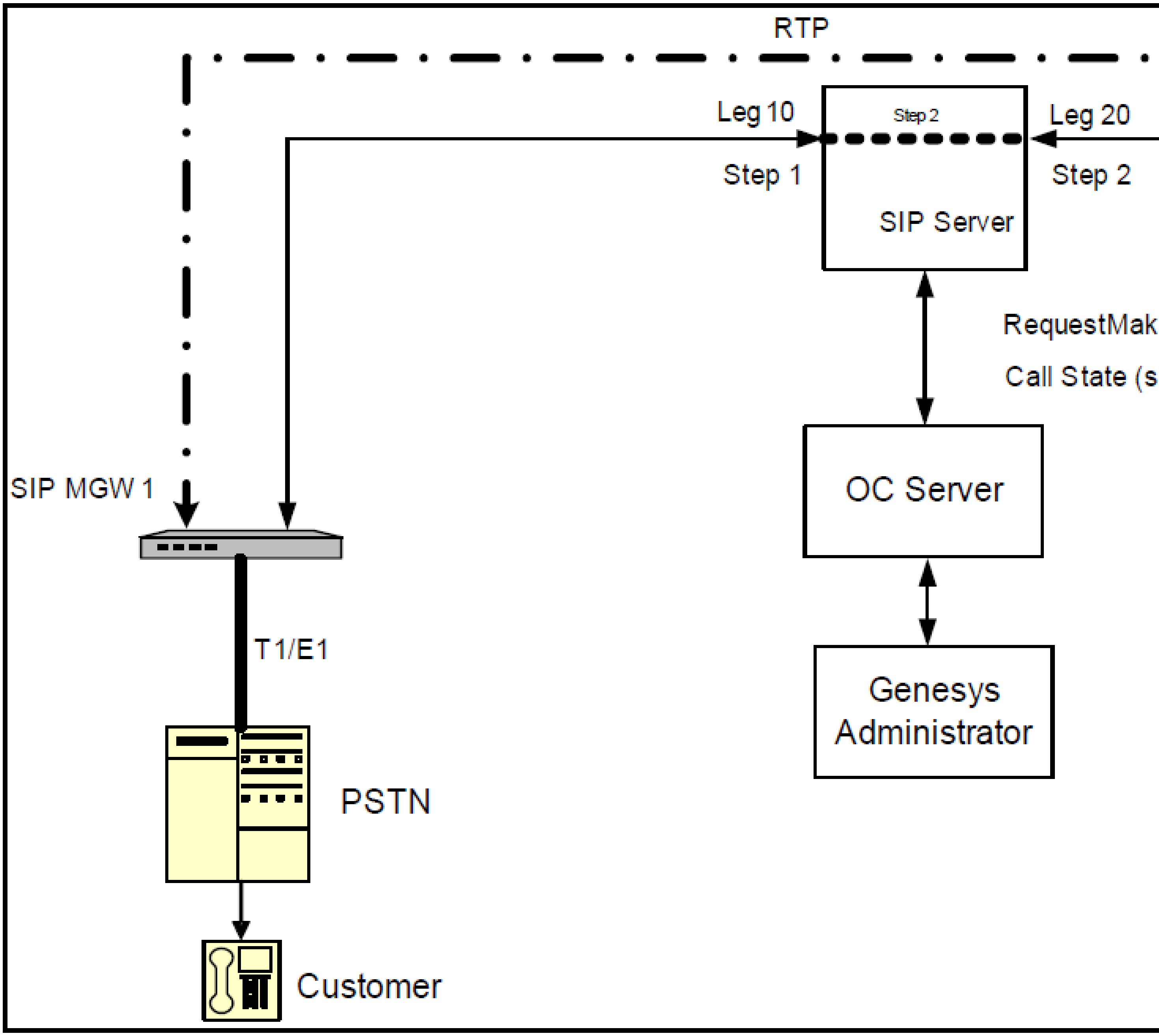
Transfer Mode (MGW with CPA)

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.

Transfer-Mode Call Flow (MGW with CPA) with a SIP Agent Endpoint illustrates a `Transfer`-mode call flow in VoIP environment that involves an MGW with CPA and a SIP agent endpoint.



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.

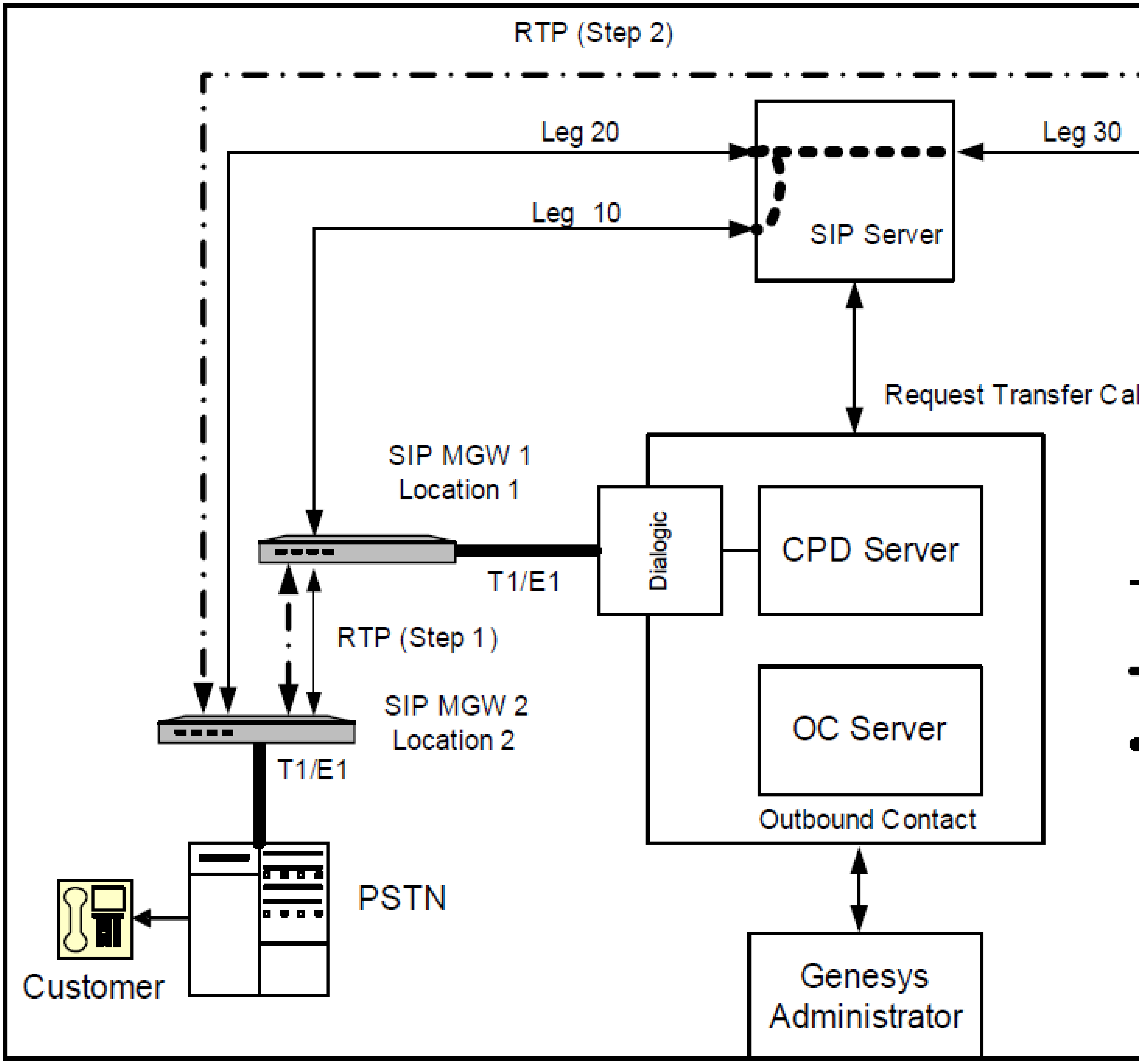
2. SIP Server creates call leg 10 with MGW 1 and establishes a call with the customer DN.
3. MGW 1 performs CPA and sends the call results to SIP Server.

Step 2

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

Transfer Mode (MGW without CPA)

The following scenario describes a media flow that involves an MGW without CPA abilities. Transfer-Mode Call Flow (MGW without CPA) in Two Locations with a SIP Agent Endpoint illustrates a `Transfer` -mode call flow in two locations in VoIP environment that involves an MGW without CPA and a SIP agent endpoint.



Transfer-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 `Req_MakePredictiveCall` to CPD Server. This request contains `SAttr_DialTo`, which is the customer's phone number.
2. CPD Server selects a Dialogic channel DN and places it off hook.
3. CPD Server makes a call via Dialogic board through MGW 1 from a selected Dialogic channel DN to a customer's phone number.
4. MGW 1 creates leg 10 of a call by sending `INVITE` to SIP Server to establish a session with a destination endpoint defined by `SAttr_DialTo`.
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.

Note:	• 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.

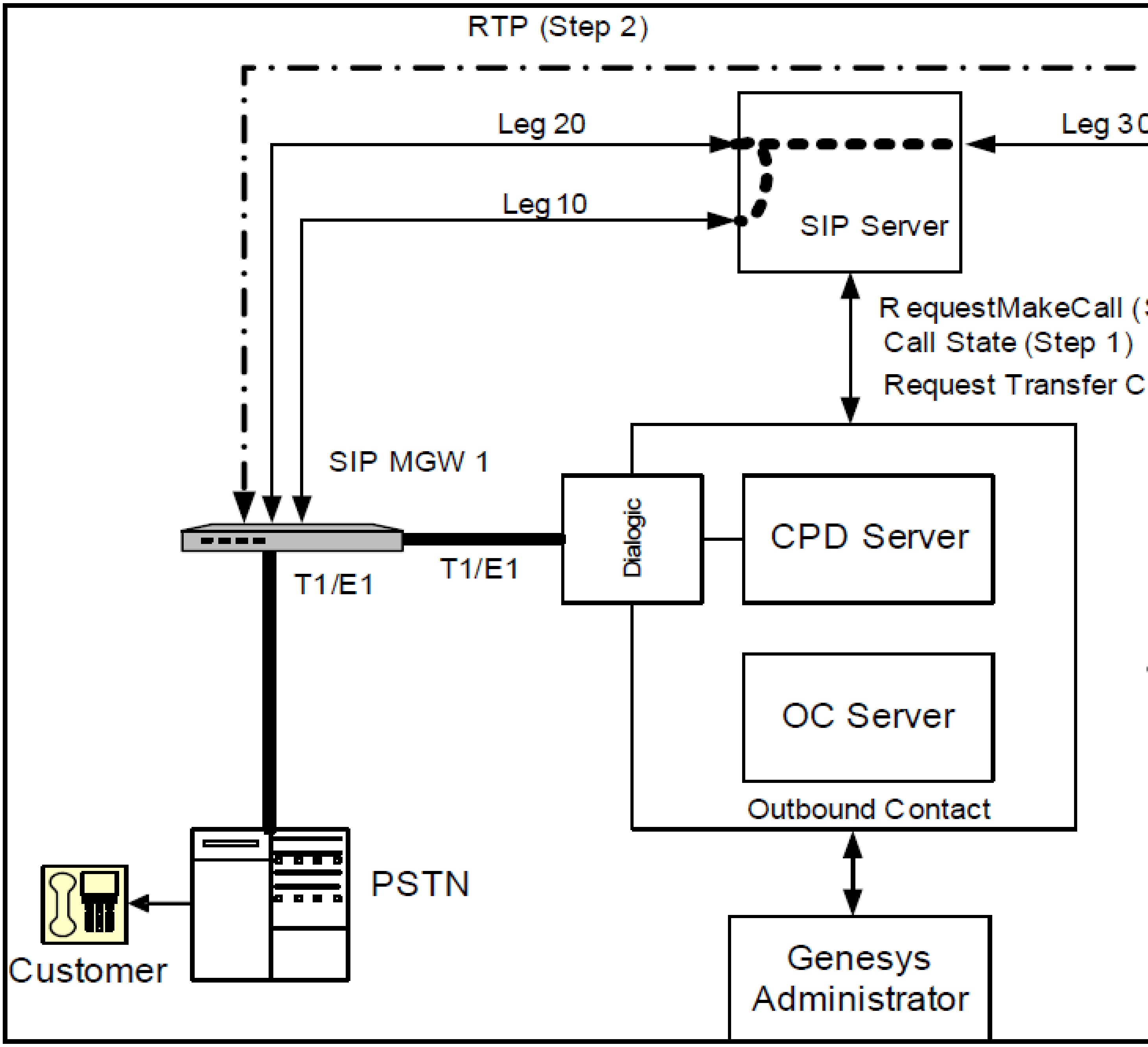
Step 2

1. CPD Server sends a request to SIP Server to initiate the transfer of the customer call to a SIP agent endpoint.
2. SIP Server creates and establishes call leg 30 with the SIP agent endpoint.
3. SIP Server joins call leg 20 and call leg 30. All media streams are now between the SIP agent endpoint and the customer.
4. 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 Call Flow (MGW without CPA) in One Location with a SIP Agent Endpoint illustrates a `Transfer`-mode call flow in one location in VoIP environment that involves a MGW without CPA and a SIP agent end point.

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



Transfer-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 Req_MakePredictiveCall to CPD Server. This request contains SAttr_DialTo, which is the customer's phone number.
2. CPD Server selects a Dialogic channel DN and places it off hook.
3. CPD Server makes a call via Dialogic board through MGW 1 from a selected Dialogic channel DN to a customer's phone number.
4. MGW creates leg 10 of a call by sending INVITE to SIP Server to establish a session with a destination endpoint defined by SAttr_DialTo.
5. SIP Server resolves the destination endpoint address and creates leg 20 of a call by sending INVITE back to MGW to establish a call with the customer's phone number. All media streams are now between the MGW 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.

Note:	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

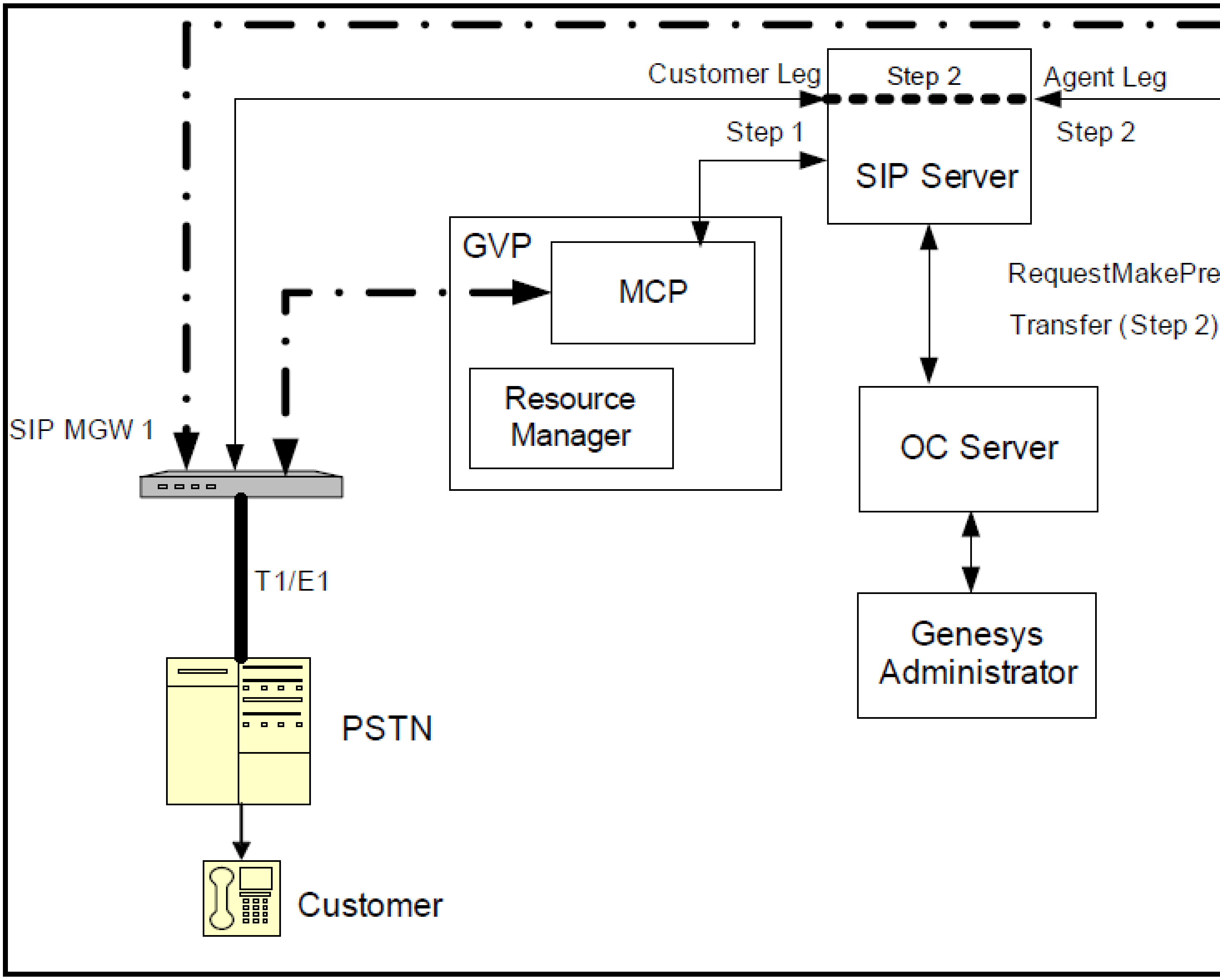
1. CPD Server sends a request to SIP Server to initiate the transfer of the customer call to a SIP agent endpoint.
2. SIP Server creates and establishes call leg 30 with the SIP agent endpoint.
3. SIP Server joins call leg 20 and call leg 30. All media streams are now between the SIP agent endpoint and the customer.
4. 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.

Transfer-Mode Call Flow (MCP as the CPA Provider) with a SIP Agent Endpoint illustrates a Transfer mode call flow in VoIP environment that involves MCP as the CPA provider and a SIP agent endpoint.



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.

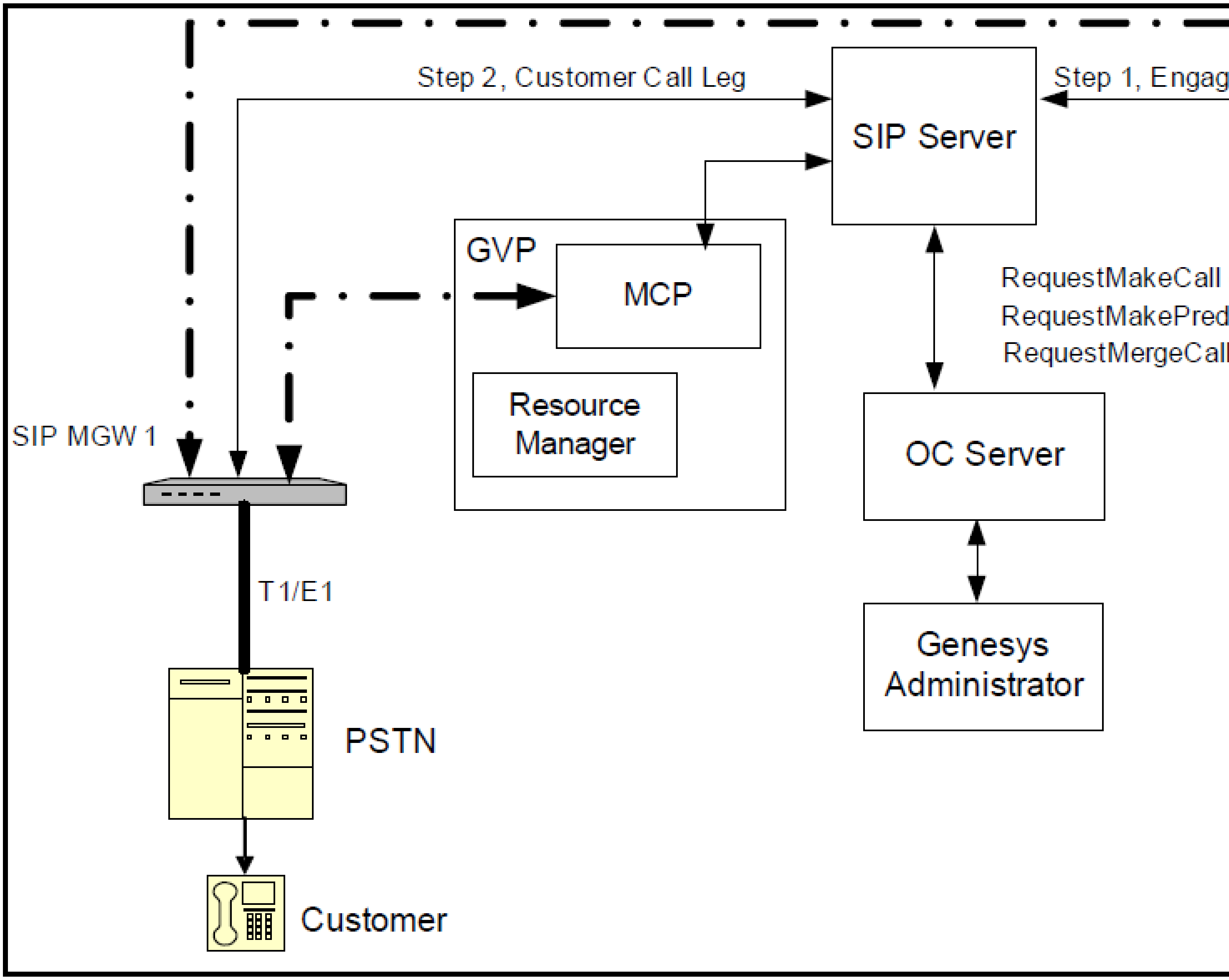
Step 2

1. 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.
2. SIP Server generates `EventQueued` and/or `RouteRequest` messages and establishes the call leg with a SIP agent endpoint.
3. 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.

ASM-Mode Call Flow (MCP as the CPA Provider) with a SIP Agent Endpoint illustrates a `Transfer` mode call flow in VoIP environment that involves MCP as the CPA provider and a SIP agent endpoint.



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

1. 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.
2. SIP Server invites MCP/Genesys Media Server to handle CPA.
3. SIP Server creates 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 3

1. 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.
2. 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.*

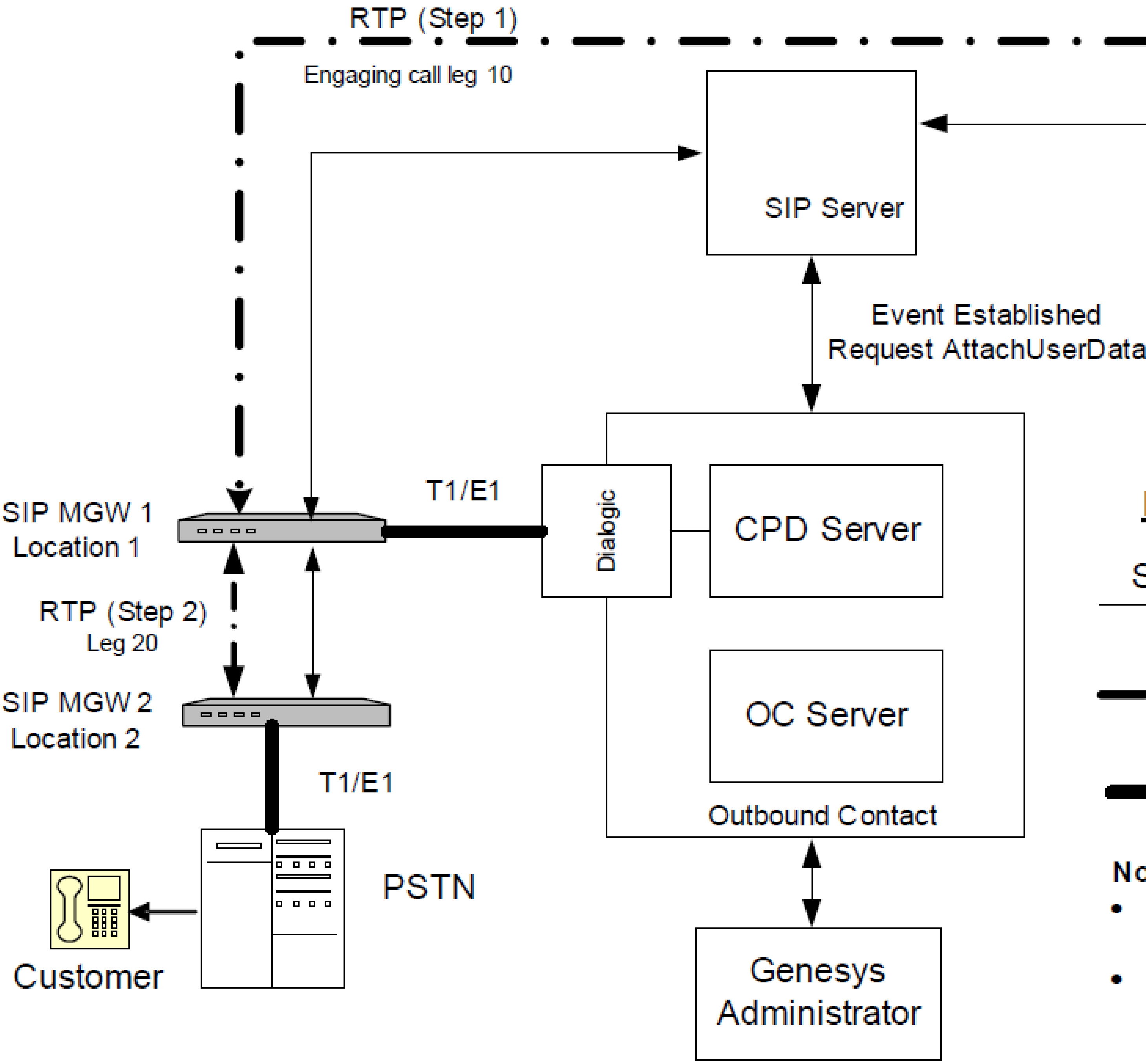
ASM Mode (MGW with CPA)

Supported scenarios are the same as in the previous section ASM Mode (MCP as the CPA Provider).

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.

ASM-Mode Call Flow (MGW without CPA) in Two Locations with a SIP Agent Endpoint illustrates an ASM-mode call flow in two locations in VoIP environment that involves a MGW without CPA and a SIP agent endpoint:



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

In this scenario, the call flow proceeds as follows:

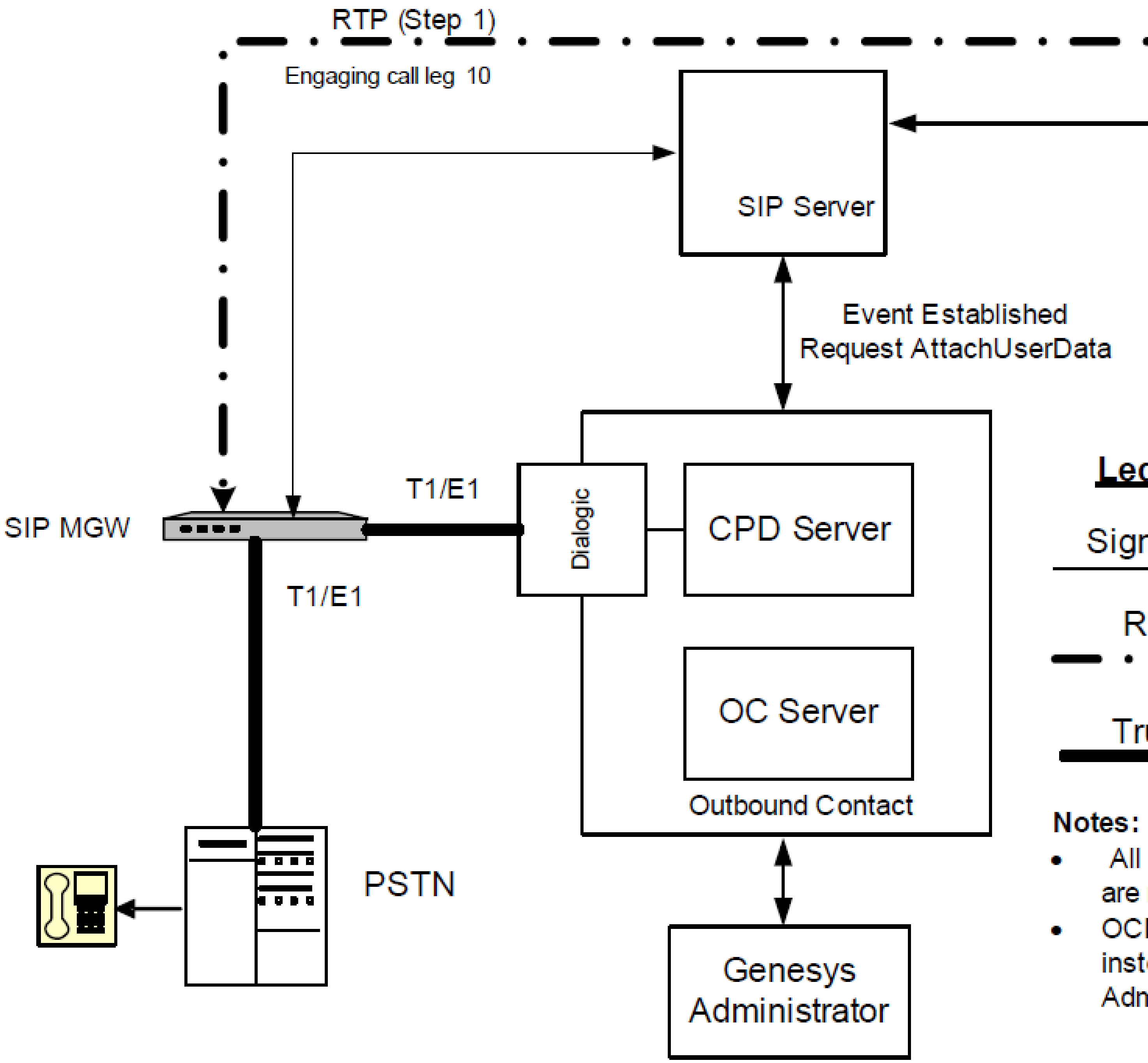
Step 1

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

1. OCS sends `Req_MakePredictiveCall` to CPD Server. This request contains `SAttr_DialTo`, which is the customer phone number.
2. CPD Server instructs the Dialogic board to place a call to the customer number that OCS provided.
3. If CPA has determined that there is a live voice, CPD Server attaches any customer data to the engage call (leg 20).
4. SIP Server delivers this data to the engaged agent’s desktop as a screen pop.
5. CPD Server connects the call’s internal and external leg. The call is established between the agent and the customer.
6. CPD Server informs OCS of the call result. The call is now handled as a normal outbound call.

ASM-Mode Call Flow (MGW without CPA) in One Location with a SIP Agent Endpoint illustrates an ASM-mode call flow in one location in VoIP environment that involves an MGW without CPA and a SIP agent endpoint.



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

In this scenario, the call flow proceeds as follows:

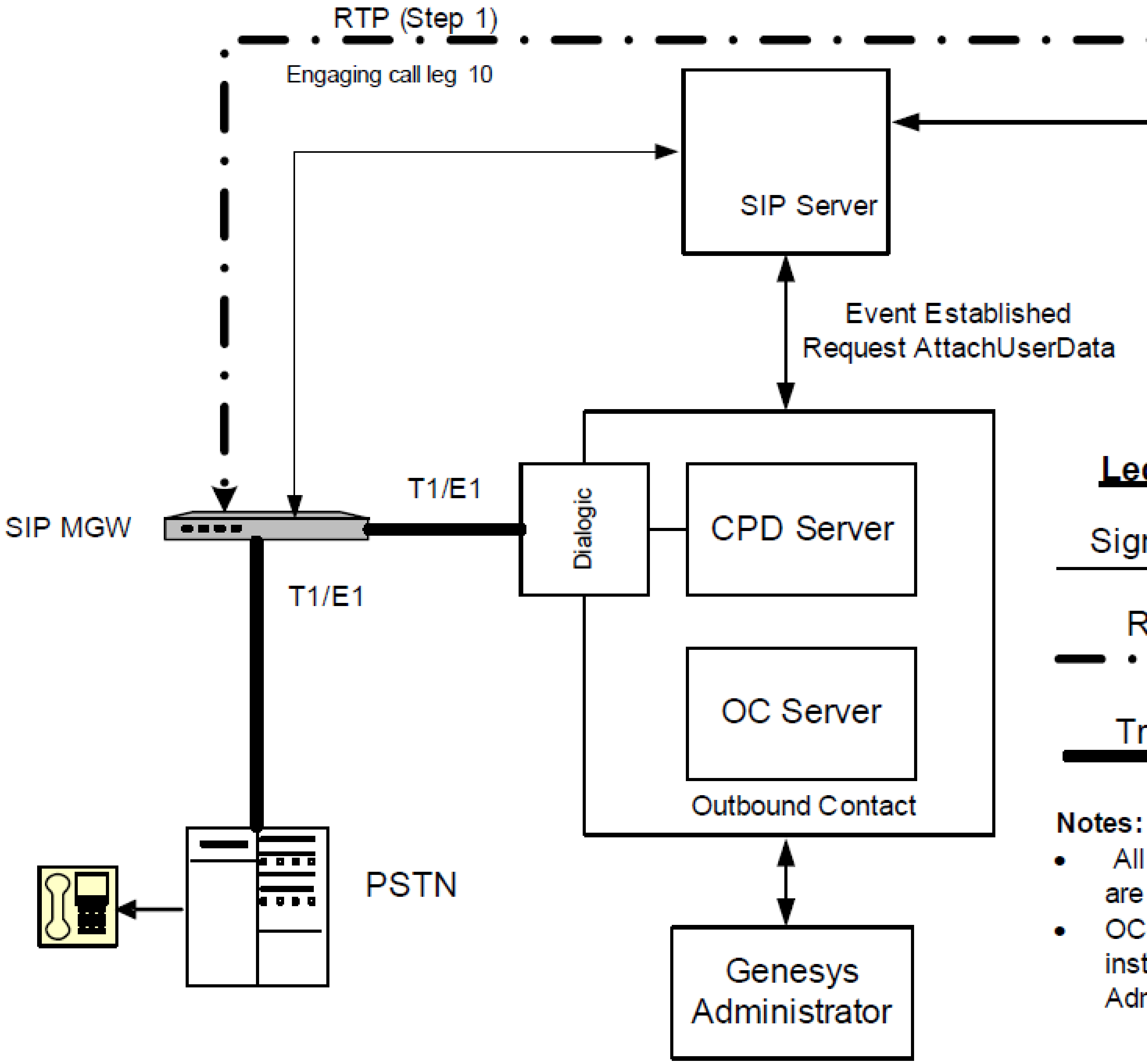
Step 1

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

1. OCS sends `Req_MakePredictiveCall` to CPD Server. This request contains `SAttr_DialTo`, which is the customer phone number.
2. CPD Server instructs the Dialogic board to place a call to the customer number that OCS provided.
3. If CPA has determined that there is a live voice, CPD Server attaches any customer data to the engage call (leg 20).
4. SIP Server delivers this data to the engaged agent's desktop as a screen pop.
5. CPD Server connects the call's internal and external leg. The call is established between the agent and the customer.
6. CPD Server informs OCS of the call result. The call is now handled as a normal outbound call.

ASM-Mode Call Flow with a SIP Agent Endpoint illustrates an `ASM`-mode call flow in VoIP environment that involves a SIP agent endpoint. HMP software is used for CPA.



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

1. OCS sends `Req_MakePredictiveCall` to CPD Server. This request contains `SAttr_DialTo`, which is the customer phone number.
2. CPD Server instructs HMP to place a call through SIP Server to the customer number that OCS provided.
3. If CPA has determined that there is a live voice, CPD Server attaches any customer data to the engage call (leg 10).
4. SIP Server delivers this data to the engaged agent's desktop as a screen pop.
5. 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.

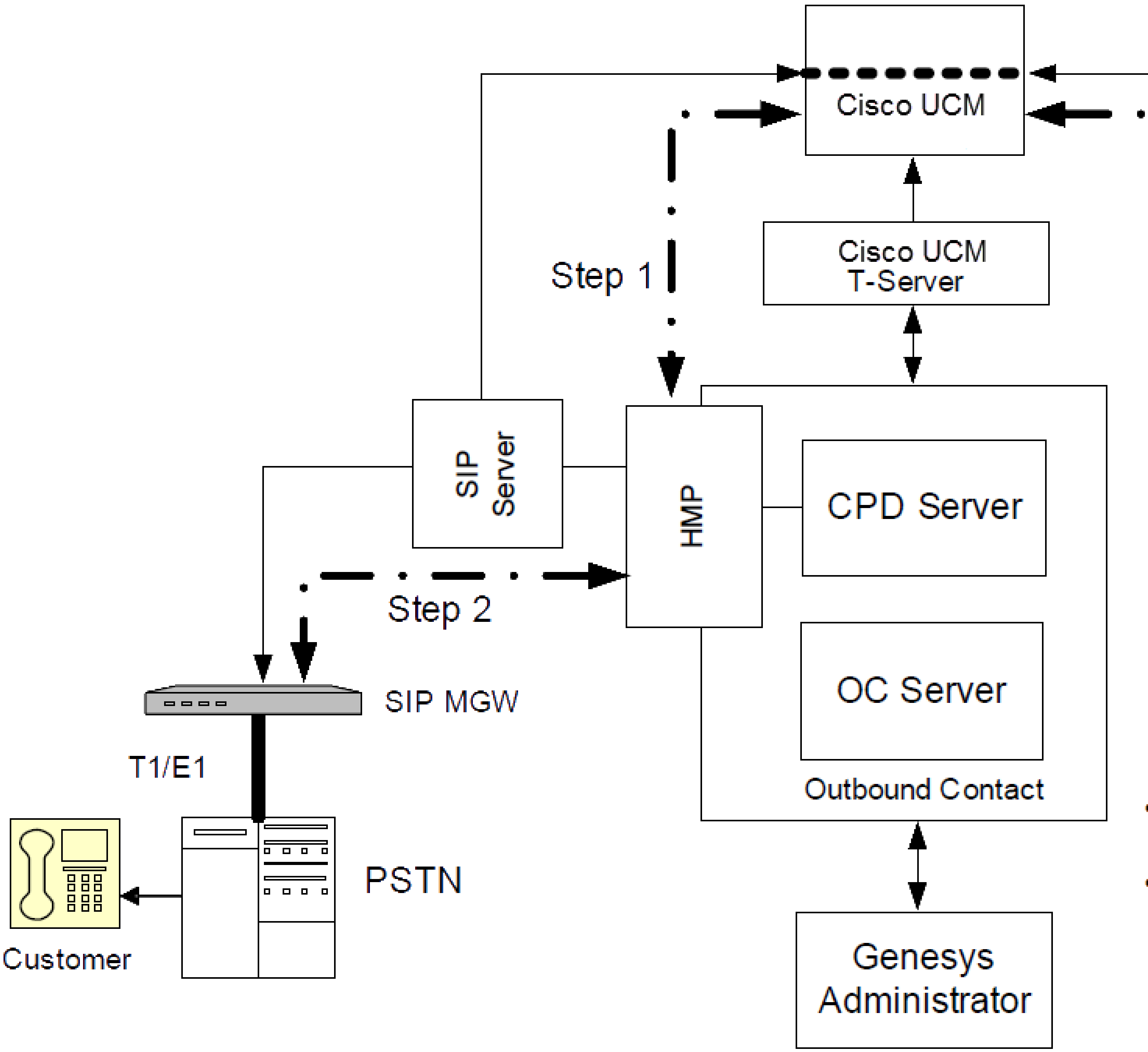
Outbound Contact with Cisco Unified Communications Manager

This section describes an `ASM` mode scenario and a `Transfer` mode scenario using T-Server for Cisco Unified Communications Manager (UCM) and agents.

ASM Mode

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

ASM-Mode Call Flow—Cisco Unified Communications Manager illustrates the architecture/call flow.



ASM-Mode Call Flow—Cisco Unified Communications Manager

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 UCM 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

OCS sends Req_MakePredictiveCall to CPD Server. This request contains SAttr_DialTo, which is the customer phone number.

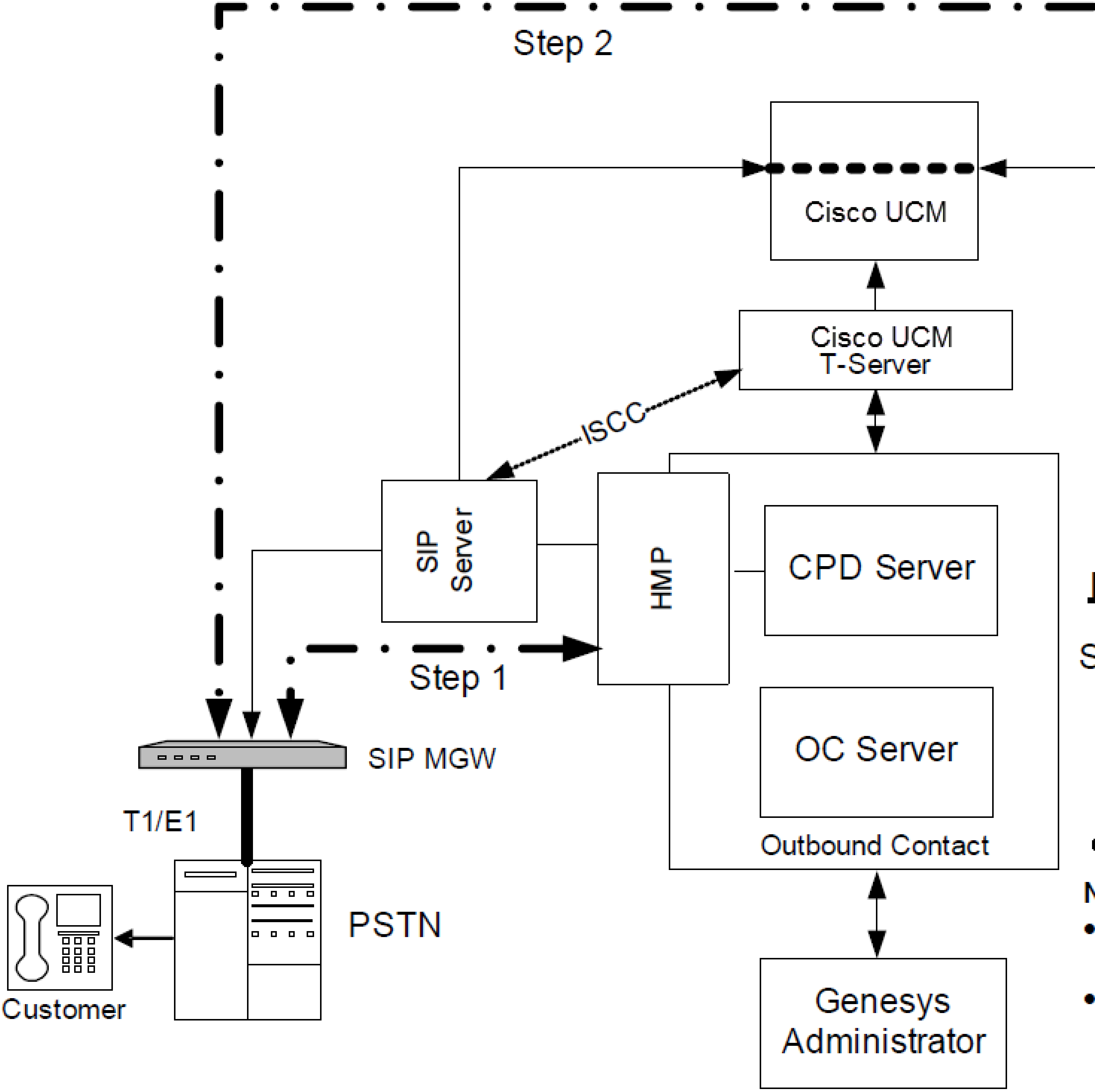
1. CPD Server initiates an outbound call to SIP Server, where another Trunk points to either a Media Gateway or an IP SIP client endpoint.
2. The SIP call in initiated call.
3. Using HMP resources, CPD Server performs call progress analysis.
4. If a positive voice detection occurs, CPD Server bridges the internal leg (the engaged call) and the external leg (the outbound call).
   The call is established between the agent and the customer.
5. CPD Server informs OCS of the call result.
6. 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 Cisco UCM switch is used to control outbound agents. So, when a call is established with the customer, the outbound call is transferred to the Cisco UCM agent using ISCC (data).

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

Transfer Mode—Cisco UCM illustrates the architecture/call flow.



Transfer Mode—Cisco UCM

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

1. 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 UCM.
2. Cisco UCM T-Server, which is monitoring the Cisco Unified Call Manager switch, informs the URS that the call was routed to the Routing Point.
3. URS routes the call (per the routing strategy, also stored at the central location) sends to Inter Server Call Control (ISCC).
4. ISCC sends the call to the Cisco UCM switch that is being monitored by the T-Server for Cisco UCM.
5. The Cisco UCM switch relays the call to Route Point for a group of agents.

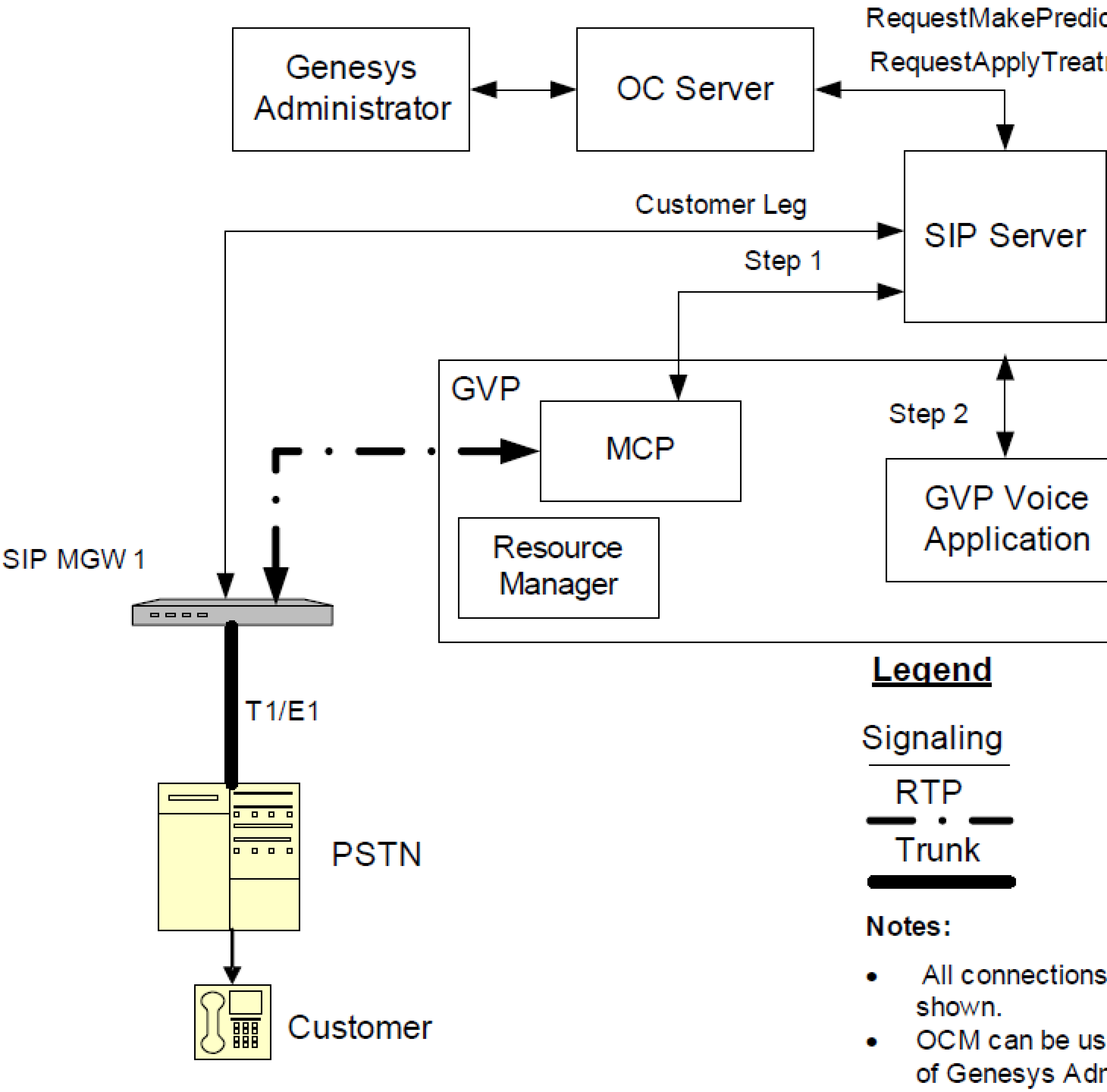
After the call is routed to an agent, no SIP signaling or RTP streams go through HMP.

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 and 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.

Outbound Contact, GVP 8.1—MCP as the CPA Provider illustrates how to use Outbound Contact with GVP 8.1.



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

1. For a positive call result (`EventEstablished` received), OCS sends the `RequestApplyTreatment` message to SIP Server to trigger a GVP Voice Application.
2. 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. TMakeCall Attached Data lists the attached data for `TMakeCall`.

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

Data Key	Type	Key Required	Value	Description
				Group DN
GSW_SESSION_DBID	Int	Yes	DIBD	Identifies the DBID of the Campaign Group (Dialing Session) for the initiated call.

TMakeCall Extensions lists the extensions for `TMakeCall`.

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. TMakePredictiveCall Attached Data lists the attached data for `TMakePredictiveCall`

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.

TMakePredictiveCall Extensions lists the extensions for `TMakePreictiveCall`.

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.
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 `call_answer_type_recognition` option.
cpd-on-connect	on or off	Indicates when CPA begins, according to the `cpd-on-connect` option. For Outbound-VoIP ASM modes, set this option to yes to specify that CPA begins after the call is connected. **Note:** Setting this option to yes accounts for the use of color ring back tones. If `cpd-on-connect` does not appear in the extensions, CPA starts as soon as the media stream is available. *Color ring back tones* (CRBT) 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 `call_timeguard_timeout` option, which specifies the maximum time allowed for CPA after the call is connected.

TMergeCall Extensions

TMergeCall Extensions lists the extensions for `TMergeCall`.

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 *Outbound Contact 8.1 Deployment Guide.*

TApplyTreatment Extensions

TApplyTreatment Extensions lists the extensions for `TApplyTreatment.`

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.

SCXML Architecture

In order to use SCXML-based treatments, Outbound Contact makes use of three components/applications:

- OCS (see Outbound Contact Server)
- Application Server
- SCXML engine/state machine built into OCS

Application Server

SCXML-based treatments are typically hosted/stored on an Application Server (along with other Genesys SCXML scripts, such as SCXML-based strategies) and provided to OCS based on parameters contained in HTTP requests. The `treatment-uri` option identifies the location of these scripts.

Application Server is a web server used to store and retrieve SCXML scripts. You deploy SCXML-based treatments to your production environment by publishing them to an Application Server. Upon an HTTP request, the Application Server is responsible for providing the treatment logic to OCS in the form of a document.

Document Retrieval from Web or Application Server

OCS connects to and retrieves SCXML documents (flexible treatments descriptions) from the Web or Application Server in the following ways:

- OCS uses a connection pool and HTTP/1.1 persistent connections. The size of the pool and connection timeout settings are controlled by the `http-connection-pool-size` and `http-response-timeout` options. Previously, OCS used HTTP 1.0 non-persistent connection to the Web or Application Server.
- OCS opens only a single connection to the given Web or Application Server to retrieve all SCXML documents from that server. Previously, OCS opened as many connections as there were simultaneous requests for SCXML documents from the loaded chains.
- OCS supports secure HTTP (HTTPS) with TLS protocol as a sub layer under regular HTTP for SCXML document retrieval from Web or Application Server. For more details about how to configure and use HTTPS, see Pre-Dial Validation Over Secure Connection.

Supported Application Servers

Genesys supports the following types of Application Server software:

- Microsoft Internet Information Services (IIS), formerly called Microsoft Internet Information Server). Genesys supports IIS 6..19.198.1.
- JBoss Application Server (or JBoss AS). This free software/open source Java EE-based Application Server is usable on any operating system that Java supports. Genesys supports version JBoss 8.1.

- IBM's WebSphere Application Server (WAS). This software Application Server, which is built using open standards (such as Java EE, XML, and Web Services) works with a number of web servers. Genesys supports IBM Websphere Application Server 5.0—8.1.
- Apache/Tomcat

Note:	While Web or Application Server is a more common and a more convenient way to store SCXML documents, you can simply place a file with the SCXML script in your file system. In this case, the absolute path to this file is configured as follows (example): `treatment-uri=file:///C:/GCTI/OCS/Scripts/sample01.scxml`

SCXML Engine/State Machine

The SCXML engine is a subcomponent of OCS that leverages the Genesys SCXML library built into OCS, and which parses the SCXML treatment script. When each record or chain of records associated with the campaign is accessed for processing, the engine creates an instance of the state machine and requests the SCXML treatment from the Application Server. When the record is processed (for example, the call is answered and then completed), the associated state machine instance is stopped.

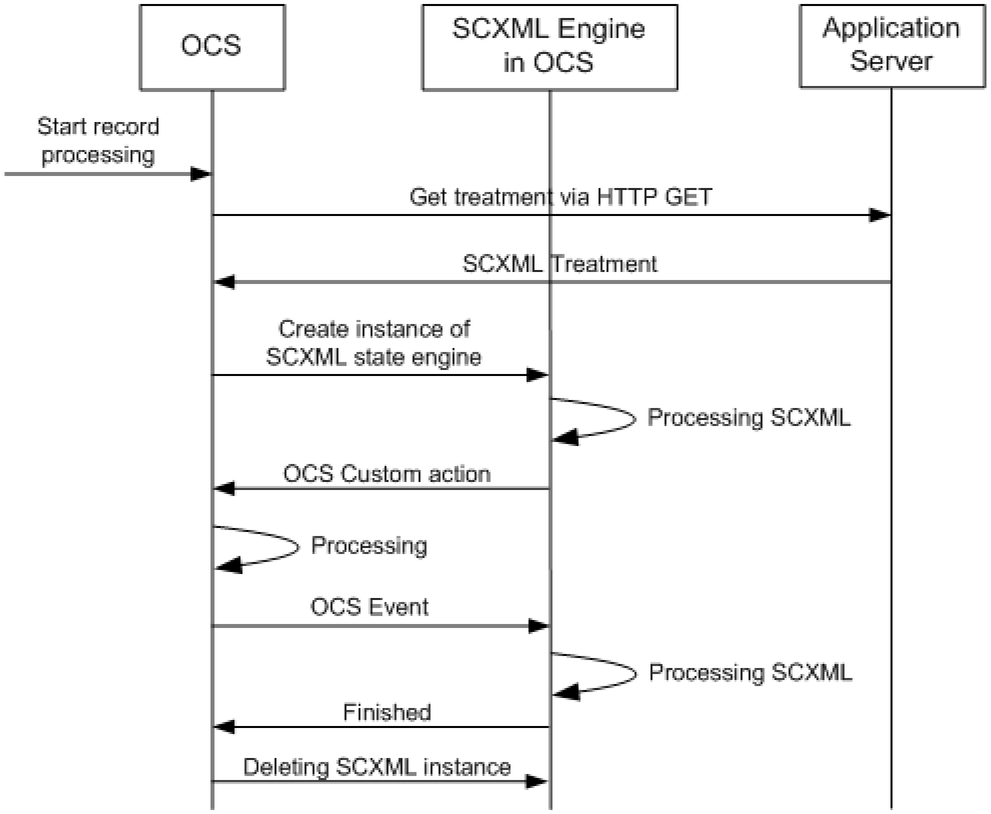
Note:	You specify the SCXML document that describes the treatment used by OCS in the `treatment-uri` option in the Campaign Group or Calling List object. For more information on configuring treatments, see Design and Configuration Task Summary.

A state machine is an SCXML treatment script launched by the SCXML engine, that includes states, actions, events, and transitions, designed to inform OCS on how to contact customers during a campaign.

- *States* provide information on the current status. A state machine operates in one state at a time, until an OCS event provides information that requires a state change, according to the design of the treatment script.
- *Actions* are activities that the state machine performs, according to the treatment script.
- *Events* are messages that OCS sends to the state machine in response to an action.
- *Transitions* occur when the state changes in response to an event.

The following figure illustrates the scenario in which the SCXML engine is used to apply an SCXML treatment.

SCXML Engine/State Machine Treatment Processing describes the diagram.



SCXML Treatment Processing

SCXML Engine/State Machine Treatment Processing

1. OCS receives a request to start processing a record of a Calling List associated with a campaign.
2. As configured in the Campaign Group or Calling List object, OCS requests the associated treatment from Application Server.
3. Application Server provides OCS with the SCXML treatment.
4. OCS requests the Genesys SCXML engine to create and start an instance of the SCXML state machine for the record.
5. The SCXML engine creates and starts the instance and requests that OCS execute customer actions for the record, according to the SCXML treatment script.
6. OCS initiates the requested action (for example, make a call, go to the next record, delay, and so on); for more information on possible actions, see Outbound Contact Custom Actions and Events).
7. After completing each action, OCS sends an event communicating the action result of the action to the instance of the state machine that SCXML engine is running.
8. OCS continues processing the actions, as defined in the state machine.
9. When the processing of the chain of records is completed and chain is finalized, OCS deletes the instance of that state machine.

Note:	A state machine may not be able to complete the

	script for a chain of records if: • The campaign is unloaded. • OCS shuts down.

After a particular instance of the state machine is restored, OCS sends the `ocs.daytime_change` event to the instance so it can adjust its behavior according to the current daytime interval.

Outbound Contact Custom Actions and Events

The following tables describe the custom actions and events supported in Outbound Contact for SCXML treatments.

Custom Actions

Action	Action Details
`timeout`	Response event: `ocs.timeout`
	Element attribute: `delay`--Timeout in seconds
	Child elements: none
	Description: Waits for the specified time.
	Examples • A five minute delay: `<ocs:timeout delay="300"/>` • A delay to the nearest holiday: `<ocs:timeout delay="HDAY"/>`
`next_record`	Reponse event: `ocs.next_record`
	Element attribute: `repeat_chain`-- (Optional) Controls whether the chain should be cycled again starting with the record at the beginning of the chain.
	Child elements: `contact_info_type`0 or more elements can be applied.
	Description: Requests the next record in the chain. The next record is selected according to the criteria defined by the `contact_info_type` child elements (for example, `type_id` or `type_name`). **Note:** An empty `record_id` indicates that there is no next record.
`contact_info_type`	Element attributes: • `type_id`--ID of the contact information type, as specified in the `contact_info_type` field in the Calling List

	• type_name--Mnemonic name of the contact information type. The list identifies each possible IDs and its corresponding type name: • 0--NO_TYPE • 1--HOME • 2--DIRECT_BUSINESS • 3--BUSINESS_WITH_EXTENSION • 4--MOBILE • 5--VACATION • 6--PAGER • 7--MODEM • 8--VOICE_MAIL • 9--PIN_PAGER • 10--E-MAIL • exclude--The value that determines whether records that match these criteria are excluded from the result. A value of no (the default) includes the records. A value of yes excludes the records.
	Child elements: none
	Description: Identifies the attribute for contact_type_info. At least one attribute (type_id or type_name) must be specified. If both are specified, type_id has priority. **Note:** If both are specified, confirm that the type_id and type_name values match each other. Otherwise, a warning appears in the log.

	Examples
	Example 1, Getting Next Record with the Home Phone
	`<ocs:next_record>`
	`<ocs:contact_info_type type_id=1 />`
	`</ocs:next_record>`
	
	Example 2, Getting the Next Record with Phone Other than Home and Mobile
	`<ocs:next_record>`
	`<ocs:contact_info_type type_name="'HOME'" exclude="'yes'" />`
	`<ocs:contact_info_type type_name="'MOBILE'" exclude="'yes'" />`
	`</ocs:next_record>`
	
	Example 3, Getting the Next Record According to the Database Order
	`<ocs:next_record/>`

	Example 4, Cycling Through the Chain and Getting the Next Record from the Beginning of the Chain <ocs:next_record repeat_chain="'yes'"/>
get_daytime	Request event: daytime_change
	Element attributes: none
	Child elements: none
	Description: Requests information about the current daytime interval.
get_info	Request event: ocs.info
	Element attributes: - what--(Mandatory) A string with the type of information being requested. The supported value is TIMERANGE. - param--(Optional) A string that is the parameter of the request. For the TIMERANGE type of information, valid values include: - BBH: Before business hours on a weekday - BH: Business hours on a weekday - ABH: After business hours on a weekday - WEND: Weekend - HDAY: Holiday
	Child events: none
	Description: Requests the execution of content-related information from OCS. TIMERANGE, param values = <BBH, BH, ABH, WEND, or HDAY> requests the time period (in seconds) when the specified param attribute timerange begins for the current record.
	Example<ocs:get_info what= "'TIMERANGE'" param = "'HDAY'"/>
make_call	Response event: ocs.callresult
	Element attribute: record_id--ID of the record for which the call will be made.
	Child elements: none
	Description: As a result of this action, OCS initiates a request to make a call using the contact information specified in this record.
	Example:<ocs:make_call record_id="record_id"/>
set_flex_attr	Response event: None if successfulError.Attribute if an invalid value is configured

	Element attributes: `record_id`--(Mandatory) Defines the ID of the record (record handle) for which the flexible attribute(s) (options, user data, or Attribute Extensions) are configured. This element can contain a special value of `0`, which is interpreted as `for all records in chain`. For example, it specifies that the same flexible attribute(s) (options) must be set for all the records in the chain that are processed.
	Child elements: `flex_attr1` or more elements can be defined.
	Child element attributes: • `attr`-- (Mandatory) Specifies the type of flexible attribute being configured. Allowed value: `OPTIONS`, `USERDATA`, `EXTENSIONS` • `type`--(Mandatory) Specifies the type of value that is used for the flexible attribute. Allowed values: `string` or `int` • key--(Mandatory) Specifies the mnemonic name of the flexible attribute. (option name)* `value`--(Mandatory) Specifies the value that is assigned to the attribute. (option value)Element description: Defines a single option, including its name, type, and value.
	Descriptions: `OPTIONS`--This custom action can be used to set one or more options for the specified record in the chain or for all records in the chain. The actual options and values are defined by the `flex_attr` child element(s). `USERDATA`--Used to update any mandatory or user-defined field that belongs to the current record or chain of records. `EXTENSIONS`--Used to pass an arbitrary key-value pair in the `AttributeExtensions` field of the Dialing request that OCS sends to the dialer (for example, to T-Server). • key - specifies the key in the pair • value - specifies the value for the key

	Examples:
	1. Set options for the current record:
	`<ocs:set_flex_attr record_id="_event.data.record_id">`
	`<ocs:flex_attr attr="'OPTIONS'" type="'string'" key="'CPNDigits'" value="'8884554040'"/>`
	`<ocs:flex_attr attr="'OPTIONS'" type="'string'" key="'pre-dial-validation'" value="'false'"/>`
	`</ocs:set_flex_attr>`
	2. Set option for all records in the chain:
	`<ocs:set_flex_attr record_id="'0'">`
	`<ocs:flex_attr attr="'OPTIONS'" type="'string'" key="'pre-dial-validation'" value="'true'"/>`
	`<ocs:flex_attr attr="'USERDATA'" type="'string'" key="'UserField'" value="'UserValue'"/>`
	`<ocs:flex_attr attr="'EXTENSIONS'" type="'string'"`

	`key="'DialAttr'" value="'AttrValue'"/>`
	`</ocs:set_flex_attr>`

`set_exec_flag`	Response event: None if successful`Error.Attribute` if an invalid value is configured
	Element attributes: • `flag`--(Mandatory) Defines the type of execution flag. Allowed Value: `PostProcessing` • `value`--(Mandatory) Defines the value of the execution flag. For `PostProcessing` flag type, valid values are: `Finalize` and `Switch`.
	Child elements: None
	Description: This custom action allows the flag, that controls SCXML treatment execution, to be set. • `PostProcessing` flag--Determines how the chain of records is processed when the SCXML treatment execution is completed. • `Finalize`--(default) Processing of the chain is done, chain is finalized (marked `completed`) in the database. • `Switch`--Processing of the chain continues after the SCXML treatment execution ceases, and traditional treatments (if configured) take over execution.

	Examples:
	1. Switch to traditional treatments:
	`<:set_exec_flag flag="'PostProcessing'" value="'Switch'"/>`
	2. Finalize chain processing:
	`<:set_exec_flag flag="'PostProcessing'" value="'Finalize'"/>`
	Note: This custom action is available in OCS release 8.0.001 and beyond.

Events from OCS

Event	Event Details
`ocs.callresult`	Response for `make_call`
	Event properties: `callresult`--The result of the call. Values include: • `Ok` • `GeneralError`

	• `SystemError` • `RemoteRelease` • `Busy` • `NoAnswer` • `SIT_Detected` • `AnswMachine` • `All_Trunks_Busy` • `SIT_Invalid_Num` • `SIT_Vacant` • `SIT_Oper_Intercept` • `SIT_Unknown` • `SIT_No_Circuit` • `SIT_Reorder` • `Fax` • `Abandoned` • `Dropped` • `DroppedNoAnswer` • `Unknown` • `Silence` • `Answer` • `NuTone` • `NoDialTone` • `NoProgress` • `NoRingBack` • `NoEstablished` • `PagerDetected` • `WrongParty` • `DialError` • `CallDropError` • `SwitchError` • `NoFreePortError` • `TransferError` • `Stale`

	• `AgentCallBackError` • `GroupCallBackError` • `DoNotCall` • `CancelRecord` • `WrongNumber`
	Description: OCS notifies the state machine instance of the call result.
`ocs.daytime_change`	Response for `get_daytime` or an unsolicited request
	Event properties: `daytime_code`--Code that defines the current daytime interval. Values include: • `BBH`: Before business hours on a weekday • `BH`: Business hours on a weekday • `ABH`: After business hours on a weekday • `WEND`: Weekend • `HDAY`: Holiday
	Description: Notifies instances of the SCXML scripts/state machine when the daytime interval changes. If the script is designed to recognize the change, it can result in a different number being dialed (for example, if the daytime interval changes from BH to ABH, a script could be configured to dial the home phone instead of a business phone).
`ocs.info`	Response for `get_info`.
	Event properties: `result`--A string that contains the information requested from OCS.
	Description: Returns information, based on what was requested from OCS in the context of the current record.
`ocs.next_record`	Response for `next_record`
	Event properties: • `record_id`--The OCS-assigned ID of the record in the Calling List. If it is equal to NULL, no next record is available and no other event properties are assigned • `contact_info`--The value of `contact_info` field for the record in the Calling List. • <pseudo-fields>--Helper fields defined by OCS including:

	• `gsw_chain_attempts:` (integer) Stores the number of attempts for each chain, which equals the sum of the attempts field values for all records that compose this chain. • `gsw_preferred_flag:` (string) Indicates whether the chain was prioritized by a custom field value, as defined by the `treatment-preferred-contact-field` option. Valid values for this event are yes and `no.` • All other fields defined in the Calling List.
	Description: Returns the next record according to the criteria set in the `next_record` custom action.
`ocs.timeout`	Response for `timeout`
	Event properties: none
	Description: Indicates that the `delay` specified in the `timeout` custom action has expired.
`Error.Attribute`	Response for any custom action.
	Event properties: `Error_code`--An error code (integer) that specifies the error that occurred.
	Description: Indicates that an incorrect attribute value is included in the custom action which triggered this event--for example, if the timeout custom action includes a negative delay value or the make_call custom action includes a record_id value that does not exist.
`Error.Configuration`	Response for any custom action
	Event properties: `error_code`--An error code (integer) that specifies the error that occurred.
	Description: Indicates a configuration error--for example, if the timeout custom action includes a delay value of `WEND` but the Statistical Table for weekdays is not configured.

Other Usages of SCXML Treatments

One important feature of SCXML treatments is the ability to define variables that are used later in the script to specify different actions. You define these values in the script. You can also specify their values. Whether or not you decide to specify the variable values, you can also specify the values in the `treatment-uri` option to enable you to reuse the same treatment for different Calling Lists/ Campaigns and their individual needs.

You can design the treatment script with variables for data such as the following:

- First contact type--Enables you to specify the contact type that OCS first attempts when contacting a customer. One example of a variable definition might be `start_contact_type`.
- Number of dial attempts--Enables you to specify the number of dial attempts made for a customer or contact type
- Call result--Enables you to specify an action in response to different call results.

Note:	For script examples, see `sample02.scxml`, `sample04.scxml` and `sample05.scxml`, which are sample treatment scripts that are installed when you install OCS. These scripts are located in the `scxml` folder.