This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Outbound Contact Deployment Guide

Outbound Contact 8.1.5

11/6/2022

Table of Contents

Outbound Contact 8.1 Deployment Guide

Welcome to the *Outbound Contact 8.1 Deployment Guide*. These topics provide step-by-step instructions for the configuration and installation of the applications and components that are required to run Outbound Contact 8.1.

About Outbound Contact

Outbound Contact Deployment

Outbound VoIP Deployment

Configuration Options Reference

Dialing Regulations and Related Functionality

Call Handling and Treatments

Outbound Deployment with CPD Server

Database-related Functionality

Predictive Algorithm

Introduction

Outbound Contact is an automated system that is used to create, modify, and run outbound dialing campaigns/dialing sessions in which agents interact with customers. Running a dialing session, involves launching a dialing session that is associated with the campaign, monitoring it, and making any necessary adjustments. To perform these tasks, Outbound Contact users can:

- Create calling lists from customer contact information.
- Group calling lists into campaigns for outbound dialing.
- Share campaigns among multiple agent groups.
- Create campaign sequences with the Campaign Sequence object, which provides predefined thresholds and automatic agent assignments.
- Automatically load, start, unload, and perform other management functions for Campaign Groups using Outbound Schedules.
- Optimize usage of agent resources by using assignments. Agents can shift as needed between different dialing sessions and activities, including switching between inbound and outbound call handling.
- Chain records for a customer (multiple call records).
- Choose different dialing modes.
- Define treatments and scheduled calls for unsuccessful calls on each calling list.
- Monitor campaigns using Reporting tools.
- Apply Do Not Call restrictions by phone number or customer ID.

Outbound Contact can use traditional telephony or VoIP architecture. In both cases, it has a client/server architecture consisting of the one of the following:

- Outbound Contact consists of Outbound Contact Server (OCS) and optionally can include Call Progress Detection Server (CPD Server) and/or CPD Proxy Server.

In release 8.x, Genesys Administrator replaces Outbound Contact Manager (OCM). Genesys Administrator is a web-based user interface (UI) that provides provisioning, monitoring, deployment, and operations of Genesys solutions.

Note:	If you already use Outbound Contact Manager, you can continue to use OCM 7.6 with Outbound Contact 8.x.

Migration

Refer to the *Genesys Migration Guide* for detailed information about migrating Outbound Contact 7.x or 8.0 to Outbound Contact 8.1.

License Control

Outbound Contact 8.1 requires an 8.0 license. See the *Genesys Licensing Guide* for the supported versions of FlexLM, which is the application Genesys uses to authenticate licenses. If you want to run Outbound Contact in a mixed environment (in which some Outbound Contact components are release 8.1, and some others are release 7.5, 7.6, or 8.0), 8.0 Outbound Contact licenses must be present on License Server, along with licenses for earlier versions of Outbound Contact. Outbound Contact licenses are required for all dialing modes except Push Preview and Power GVP with OBN Manager.

Note:	If you are using Dialogic HMP software, additional licenses are necessary. See HMP Software and contact your Dialogic representative for more information.

Refer to the *Genesys Licensing Guide* for detailed information about how to license Outbound Contact 8.1.

Outbound Contact Terms and Concepts

These topics introduce the terminology and concepts that pertain to Outbound Contact:

- Formats
- Fields
- Records
- Calling Lists
- Filters
- Treatments
- Campaign Group
- Campaigns
- Display Name on a Per-campaign Basis
- Outbound Schedules

Formats

The layout of each calling list is determined by a format. The term *format* has two meanings in Outbound Contact:

The field layout in a table in a relational database.

A configuration object in Genesys Administrator.

Note:	You can also configure this in Configuration Manager.

A format consists of a number of Field objects.

You can create multiple calling lists of the same format, but each calling list can have only one corresponding format. The configuration database provides default formats; these consist of all of the mandatory fields that are required by Outbound Contact 7.x and 6.5. You can create new formats that consist of user-defined fields and the mandatory fields.

Each format within a tenant has a unique name, which you specify if you create a new format.

Fields

The term *field* has two definitions for Outbound Contact:

- The name of a column in a table in a relational database.
- A configuration object in Genesys Administrator.

Note:	You can also configure this in Configuration Manager.

Each Calling List field object contains both database-related information (such as data type) and business-related information (such as field type).

Records

A *record* is a row in a Calling List table. Among its mandatory fields, each record contains:

- A phone number.
- The fields that determine how the records should be processed (Record Type and Record Status).
- The fields that store time boundaries and time zone information; these enable runtime determination of whether the record can be dialed.

A record can contain as many custom fields as needed as defined in a Calling List Format object.

Chained Records

Chained records are multiple records for the same customer in a calling list. These records typically contain alternative phone numbers such as home, business, and cellular phone. In addition, each record in the same chain can have different time boundaries and time zone information, and it can also have different values stored in its business data fields.

Flexible Prioritization of Record Types

Flexible prioritization means that a user can assign his or her own order of priority to specific record types for dialing purposes.

A record type with a higher priority is dialed before a record with a lower priority. Genesys Administrator (Outbound) automatically passes the priority for the record type and its parameters to Outbound Contact Server with events.

The user can prioritize the following record types in the Predictive/Progressive dialing mode and in the Preview dialing mode.

- General records--All records are initially marked General in an Outbound Contact calling list. The status may change after the record has been dialed. Either an agent or a treatment in response to a call result can change the status of a record.
- Campaign Callback records--An agent reschedules a campaign callback record at the customer's chosen time. The callback may be dialed automatically or delivered to an agent's desktop softphone as a preview record depending on the value of the predictive_callback option. See Outbound Contact Configuration Options for more information about this option.

Note:	See the "Communication Protocols" chapter of the *Outbound Contact Reference Manual* for information on the events distributed when

	rescheduling records.

- Campaign Rescheduled records--Treatments reschedule these records based on the detected call results. A campaign rescheduled call is dialed automatically and handled by any agent who is working on that particular campaign.

Record types can be prioritized in Genesys Administrator (Outbound Contact section) or in a third-party application that manages the campaigns.

The following options define the default record priorities for Outbound Contact:

- `default_campaigncallback_priority`
- `default_campaignrescheduled_priority`
- `default_general_priority`

These options must be configured in the `Outbound` section of the `Application`, `Campaign` or `Campaign Group` objects. Wherever these options are configured, they apply to `Campaign Group` objects.

Priorities set at the Campaign Group level have the most precedence while those set at the Application level have the least. In addition:

- Values set at the Application level are applicable for all Campaign Groups.
- Values set at the Campaign level should be applicable for all Campaign Groups created for the given Campaign.

The priorities set by the options are displayed as a grid in Genesys Administrator at `Operations > Outbound Contact > Dialing Sessions > Priorities`. The same grid also appears for the associated Sequence on the `Sequences` tab, and for the associated Schedule at `Provisioning > Outbound Contact > Schedules`, if the associated Campaign Group is part of the Sequence or has an assigned Schedule. In all cases, the grid is read-only.

Note:	You can also configure SCXML treatments to select which record in a chain is dialed first and which is dialed next. For more information, see SCXML-Based Treatments.

If you are using a third-party application (instead of Genesys Administrator) to communicate with OCS, you must use the KeyValue List from the Genesys Common Library with the events to transfer the priority information to OCS.

You must define the priorities in the following user requests, that are sent through the Communication DN:

- `CM_ReqLoadCampaign`
- `CM_ReqStartDialing`
- `CM_ReqSetDialingMode`

For more information about these user requests, see "Communication DN API" in the *Outbound Contact Reference Manual*.

Note that flexible prioritization does not apply to Personal Callback records or Personal Rescheduled records. An agent schedules or reschedules a personal call back record at the customer's chosen time. OCS automatically delivers a Personal record to the agent's desktop at the specified time, and the same agent who initially handled the record processes the Personal Call Back or Personal Rescheduled record.

Calling Lists

The term *calling list* has two meanings in Outbound Contact:

- A subset of records from a table in a relational database that satisfies the conditions of a filter that might be associated with it.
- A configuration object in Genesys Administrator.

The two meanings converge in the following way: the Table Access Point configuration object in Genesys Administrator contains the name of the table in the database. Each table in the database requires a separate Table Access Point object. The Table Access Point object is a property of the Calling List object in Genesys Administrator. One or more Calling List objects can share the same Table Access Point object.

Each Calling List table contains a number of mandatory fields that, for example, identify customers and the status of each record.

The user can add a number of user-defined fields; these typically store business-related data.

Do Not Call

The Do Not Call (DNC) function prevents a particular telephone number or customer ID from being dialed. Record-blocking can occur in Outbound Contact either by customer request or by a decision on the part of a contact center manager.

A Do Not Call request can be handled during either an outbound call or an inbound call--for example:

- During a conversation initiated by an outbound call, a customer tells an agent that he or she does not want to be called anymore. In outbound mode, the agent uses a unique record identifier generated by OCS (GSW_RECORD_HANDLE) to refer to the record.
- A customer calls the contact center (inbound call) and explicitly asks not to be called. In inbound mode, the phone number or customer ID serves as a reference to the record.

In both cases, having received a Do Not Call request, OCS makes an entry in a special table (referenced in the gsw_donotcall_list Table Access) in the database. OCS reads all of the records from this table and populates two separate tables (buffers) in memory with unique values from the phone field and from the customer_id field. These tables can also be populated by Do Not Call requests to OCS from the following sources:

- An agent desktop--If an agent belongs to a Campaign Group, and a dialing session for that Campaign Group is loaded, Outbound Contact Server uses the OCS-Agent Desktop protocol to handle DNC requests.
- A third-party application, through the Communication DN API -- If an agent is not associated with a Campaign Group, OCS uses the Communication DN API.
- A routing strategy can insert records into the DoNotCall list.

- GVP can insert records into the DoNotCall list when OCS is running in Power GVP dialing mode.

A Do Not Call request (by phone number or customer ID) from the agent's desktop updates not only the records that are in the database, but also retrieves records that are currently in the memory buffers of the Outbound Contact Server.

Outbound Contact makes a provision for several customers with the same phone number in a calling list. The value of the customer_id option can specify a user-defined field in the Calling List table to identify DNC customers by customer ID instead of phone number. You define the field name in the Field object with the send attribute assigned to it. (For more information, see Field Object.) You then specify the field name as the value of the customer_ID option in the OCS Application object. The new field is part of the Format object. Through these configurations, you can apply the DNC restriction to a particular customer instead of placing the restriction on a particular telephone number.

To enable this enhanced Do Not Call functionality in Outbound Contact, see User-Defined Field for Do Not Call.

OCS stores records marked as NoCall in the gsw_donotcall_list (one per tenant) and monitors them in the following way:

When a dialing session associated with an outbound campaign is running, OCS retrieves records that are ready to dial and checks them against the gsw_donotcall_list.

If a record that is retrieved from the calling list matches any record from the gsw_donotcall_list, OCS does not dial this record. Instead, it returns the record to the calling list and changes its record_type to NoCall.

Rereading of the Do-Not-Call List

The DoNotCall list might be updated from outside Outbound Contact, for example by a third-party application that accesses the DoNotCall list database table directly. To avoid missing updates from external sources, you can have OCS periodically reread the already-loaded DoNotCall list.

Note:	The Do Not Call list table for the given Tenant is read when the first dialing session from this Tenant is activated (loaded). Rereading of the Do Not Call list for the specific Tenant halts dialing in all Dialing Sessions that belong to this Tenant until the list is fully read. Therefore, Genesys recommends that you configure Do Not Call rereads for periods of low activity in the contact center (such as night time).

You can choose to have OCS reread already-loaded Do Not Call lists periodically by setting the `dnc-reread` option. For details, see Outbound Contact Configuration Options.

If a manual update to the Do Not Call list is required, restart OCS in order to acknowledge the changes. If you choose to restart OCS, do so during off-hour periods, so that restarting the server does not disrupt calling activities. Alternatively, OCS will pick-up these updates upon next reread of the Do Not Call list, if such a reread is configured.

Primary and Backup Impact

When OCS switches from the backup to the primary, OCS rereads the Do Not Call records that were added after the Do Not Call list was initially read by OCS for all Tenants that have active/ running dialing sessions associated with campaigns whose Do Not Call list(s) were imported. This action synchronizes the Do Not Call list between the backup OCS and the primary OCS if the primary OCS updates the list after the backup OCS reads it, because of the addition of new records to the Do Not Call list. No call requests will be created by OCS until the Do Not Call list table is completely read.

Dealing with a Large Do Not Call List

Outbound Contact uses a database table to store Do Not Call contact numbers and other information related to DNC customers' requests. This table serves as a persistent storage of Do Not Call--related information. While loading a dialing session for a campaign, OCS retrieves the data that is stored in this table, puts it into memory, and checks each phone number from the calling list against this table to determine whether a phone number should be dialed.

Generally this Do Not Call database table is intended to hold information regarding only internal Do Not Call requests rather than external Do Not Call requests.

An internal DNC request is one that is specific to a particular contact center. For example, a Do Not Call request is internal if either of the following is true:

- A called party requests to be marked as Do Not Call when an agent contacts him or her through an outbound call.
- A called party contacts (through an inbound phone call, e-mail, or personal visit) the organization that makes outbound calls and requests to be marked as Do Not Call.

An external DNC request is one that is submitted to an authority that collects such requests and distributes them to contact centers.

The number of internal Do Not Call requests is relatively small compared to the number of external ones. A contact center may receive several thousand DNC requests from customers. The number of DNC requests nation-wide could number in the tens of millions. Because OCS stores all phone numbers from the Do Not Call table in memory (RAM), the following OCS performance factors are affected as the size of the table increases:

- The amount of time that OCS takes to load a dialing session for a campaign--or, more specifically, the amount of time that it spends reading the table data for a campaign and storing it in memory.
- The amount of memory that is allocated by OCS after it reads the whole table.

To improve the performance of OCS, minimize the amount of data stored in the Do Not Call database table (accessible to OCS through gsw_donotcall_list Table Access). Ideally, this table would contain only internal DNC records. Information about external Do Not Call requests would be kept in a separate, *custom* DNC database table.

See the "Recommended DBMS Optimizations" chapter in the *Outbound Contact Reference Manual* for more information about tuning your database(s).

To address potential performance issues, Genesys recommends two methods for handling large Do

Not Call tables (over 1 million records):

- Use a dialing filter to improve OCS handling of large database tables.
- Run an SQL query to completely avoid any performance problems.

Method 1: Use a Dialing Filter

Dialing filters are used to bypass records from the Calling List table that should not be dialed. Each calling list can have its own dialing filter. Suppose that a custom DNC database table contains the field dnc_phone for storing all phone numbers that should not be dialed. In this case, the dialing filter will be as follows:

- phone NOT IN (SELECT dnc_phone FROM <custom_do_not_call_table>)
- NOT EXISTS (SELECT dnc_phone FROM <custom_do_not_call_table> WHERE dnc_phone = phone)

Note that you must complete the following steps:

- Replace the <custom_do_not_call_table> placeholder with the actual name of the custom Do Not Call table.
- Replace the column name phone (in the Outbound 6.X format) with contact_info if you are using the Outbound 7.X format for the calling list.

The second dialing filter might be faster, but your Database Management System (DBMS) might not support an EXISTS SQL clause. Check your DBMS documentation or consult your Database Administrator if necessary.

Method 2: Run an SQL Query

You can run an SQL query directly on calling lists in your database, before loading a dialing session for a campaign. This query marks as NoCall any records in a calling list with phone numbers matching those in a custom Do Not Call table. Run the following SQL query on all Calling Lists that you plan to use for dialing:

UPDATE <*calling_list_table*> SET record_type = 7 WHERE phone IN (SELECT dnc_phone FROM <*custom_do_not_call_table*>)

You must complete the following steps:

- Replace the placeholder <custom_do_not_call_table> with the actual name of the custom Do Not Call table.
- Replace the placeholder <calling_list_table> with the actual name of the Calling List table.
- Change permissions, as needed, for modifying tables through the execution of the UPDATE SET statement.

If you choose to maintain Do Not Call support manually, you must run this SQL query every time after you modify the calling list (by changing record_type to General and record_status to Ready) while preparing a calling list for the next campaign.

Note:	For more information about working with Do Not

	Call lists, see Deploying Do Not Call Functionality.

Filters

Outbound Contact uses two kinds of filters: dialing filters and viewing filters.

Dialing Filter

A *dialing filter* customizes a calling list so that only certain phone numbers are dialed during a campaign. One use of dialing filters is to divide records that share the same calling list (database table) between two or more campaigns. As a configuration object in Genesys Administrator, each Calling List object requires its own filter in order to select records according to specific criteria. For example, if you intend to divide records in the same table between two campaigns, you must create two Calling List objects, each with its own filter.

You create dialing filters and apply them to calling lists in either Genesys Administrator or Configuration Manager.

OCS supports WHERE and ORDER BY clauses in dialing filters that are longer than 255 characters. For more information, see Filter Object Options Tab Fields.

Viewing Filter

A *viewing filter* selects records in a calling list for display in Genesys Administrator. One use of viewing filters is to display records in a certain order — for example, alphabetically by the name of the city. Another use of viewing filters is to select customers by specific criteria that you define—for example, all records that have a call_result value of NoAnswer.

It is possible to combine a dialing filter and viewing filter for browsing records in on the Operations tab of Genesys Administrator (Outbound Contact) when both filters are enabled. For more information, see *Framework Genesys Administrator Help* (Operating Your Call Center > Outbound Contact Operations).

When OCS receives notification from Configuration Server that a dialing filter has changed, OCS removes the retrieved records from the buffer (memory), returns them to the Calling List database, and then retrieves the records that meet the criteria of the newly applied filter. Records on which the agent is already working remain in OCS memory until they have been processed.

Treatments

Treatments define options for handling calls by describing sets of actions to be performed on each call result. For example, if the call result for an outbound call is NoAnswer, a treatment might be to redial the number at a later time or to dial a different phone number for the customer.

In addition to the standard treatments for busy signals, no answers, and so on, Outbound Contact gives the user the option to route, to an alternate DN, calls answered by fax machines or answering machines. For example, it can run a fax-on-demand campaign along with a campaign that plays messages to answering machines. This scenario requires a separate application that is responsible for playing prerecorded messages (for example, IVR or voice mail). You can create treatments and assign them to a calling list by using Genesys Administrator or Configuration Manager.

You can design customer-focused treatments by using SCXML scripts that allow you to customize contacting customers according to time of day, contact type/first contact, dial attempts, call results and so on.

For more information on treatments, see Call Handling and Treatments and SCXML-Based Treatments.

Campaign Group

A *Campaign Group* is the main configuration object in Outbound Contact. It can be found within the Campaign object in Genesys Administrator. A Campaign Group is defined as a Campaign (a set of calling lists) that is assigned to work resources such as an Agent Group or a Place Group.

A Campaign Group can be associated with Agent Groups or Place Groups. Agents are individuals who are set up as Person objects in Genesys Administrator in order to handle customer contacts. Places can be telephone sets. There are two ways to group these individual objects:

- An Agent Group is a logical grouping of agents. Agents can be grouped by skill (for example, language or marketing knowledge) or time (for example, shifts). If the Agent Group is configured as "virtual", Outbound Contact uses the information that is provided by the Stat Server to monitor the Agent or Place Group and the Agent Status. Outbound Contact monitors the virtual agent group through the Stat Server that is configured in the Campaign Group object in Genesys Administrator. If it is not specified, the group will not be processed, and Outbound Contact Server prints an error message in the log and stops processing the group. If Outbound Contact Server disconnects from the specified Stat Server, it will stop dialing for that group because of the unavailability of agent-state information. Dialing will resume when reconnection to the Stat Server is reestablished. For more information on creating virtual agent groups, see the *Framework Stat Server User's Guide.*
- A Place Group is a logical grouping of telephone sets. An example of a place group is a group of telephone sets (sets shared by agents) in a contact center. When a place group is configured for a Campaign Group, Outbound Contact Server does not validate who logs in to the group. For its purpose, all that matters is that someone has logged in to the place group to handle outbound calls. A place group is commonly configured when a contact center requires a generic group (in other words, a physical group of phone sets) for a performance report on a campaign and does not need to know who participated in the campaign. In Contact Center Pulse Plus (CCPulse+), the Genesys real-time reporting software where outbound statistics are reported, the place group report shows activities and statistics of the DNs that belong to places that are related to the Campaign Group.

The Agent Group or Place Group must be assigned to a campaign as a Campaign Group object. Stat Server provides status information about these configuration objects (real or virtual) to Outbound Contact Server. When loading a dialing session for a particular campaign group, OCS registers and monitors the following DNs:

- All DNs of type Extension, ACD Position, and Voice Treatment Port that are associated with Places and belong to a Switch object associated with a T-Server application to which OCS is connected.
- The DN specified as the Voice Transfer Destination (VTD) on the Advanced tab of the Campaign Group configuration object.
- For VoIP environments, a Trunk Group DN configured on the Advanced tab of the Campaign Group configuration object. For more information, see VoIP Environment Configuration.

If any other DNs of the ACD Queue, Routing Point, External Routing Point, Service Number, Virtual Queue types or any others are involved in an outbound call flow, they must be monitored by OCS. These DNs should be specified as an Origination DN in the Advanced tab of the Agent Group or Place Group object that is linked to the Campaign Group in order for OCS to register those DNs with T-Server and to receive call-related events on those DNs.

Notes:	• If a Voice Transfer Destination DN and a T-Server used as a Dialer belong to one switch in a multi-site environment, but the call is distributed by external routing to an agent who is on a different site, at least one queue from this site should be listed in the Origination DN list of the corresponding Agent or Place Group. This forces OCS to register regular DNs on the remote site and receive related events. • After OCS has registered a DN specified as VTD, Trunk Group DN, or Origination DN, OCS keeps monitoring it until OCS is restarted.

VoIP Environment Configuration

When running Outbound Contact in a VoIP environment, you must configure the Campaign Group so that it is VoIP-ready by doing the following:

- A Trunk Group DN is configured.
- The SIP Server, that controls the Trunk Group DN, is listed in the Configuration tab/Connections section of the OCS application object in Genesys Administrator.

Note:	If these conditions are not met, the dialing modes function in the traditional, non-IP mode for that dialing session/Campaign Group. For more information about the Outbound-VoIP environment, see Outbound VoIP Deployment.

Campaigns

A campaign is a flexible master plan that organizes calling lists for dialing calls and handling call results. An outbound campaign specifies what kind of work the agents perform (for example, collections, telemarketing, or fund-raising). A campaign is structured as follows:

- Generally, Outbound Contact Server (OCS) does not allow you to run more than one campaign for the same Agent or Place Group at the same time. For example, if you are currently running Campaign Group A (Campaign 1, Group 1), you cannot run Campaign Group B (Campaign 2, Group 1). However, there are certain exceptions to this rule based on the types of dialing modes being used by each Campaign Group, as shown in the following table:

	Dialing Modes						
	Progressive/ Progressive with Seizing	**Predictive/ Predictive with Seizing**	**Preview**	**Push Preview**	**Power GVP**	**Progressive GVP**	**Predictive GVP**
Progressive/ Progressive with Seizing	N	N	Y	Y	Y	Y	N
Predictive/ Predictive with Seizing	N	N	Y	Y	Y	Y	N
Preview	Y	Y	Y	Y	Y	Y	Y
Push Preview	Y	Y	Y	Y	Y	Y	Y
Power GVP	Y	Y	Y	Y	Y	Y	Y
Progressive GVP	Y	Y	Y	Y	Y	Y	Y
Predictive GVP	N	N	Y	Y	Y	Y	N

(**Y** = can be run simultaneously; **N** = cannot be run simultaneously)

See Campaign Group for more information.

- One campaign can use several calling lists, and one calling list can be shared among several campaigns. A special record-level locking mechanism prevents simultaneous dialing of the same dialing record for different campaigns.

These configuration elements can be modified at runtime without stopping the dialing session that is associated with a campaign.

> **Warning**
>
> Genesys recommends that you do not change the dialer for an outbound campaign while a campaign is active. Doing so can result in unexpected behavior, such as the campaign stopping or problems with historical reporting.

Interaction with Agents

Outbound Contact maintains a documented protocol of interactions between itself and an agent's desktop application. To simplify desktop application architecture and development, this protocol is based on the user events that T-Server distributes at the client application's requests. The supported types of interactions between Outbound Contact and the desktop application are described in the following subsections. For more information about these interactions, see Communication Protocols in the *Outbound Contact Reference Manual*.

Call Delivery Prioritization

In modern call centers, delivery of outbound calls to agents is handled by a routing strategy/ Designer application. Depending on the business needs in your environment, you can prioritize delivery of outbound calls over inbound calls. Outbound calls have distinct differentiators (call type, presence of the GSW_RECORD_HANDLE pair in UserData), which can be utilized by the strategy logic.

Agent Assignment in Multiple Campaigns

Outbound Contact supports agent assignment in multiple campaigns for both the Transfer Mode and ASM Modes. When using Universal Routing Server (URS) 8.0.1 and higher, you can enforce the routing of any call to an agent assigned to a specific Campaign Group by invoking the UseActivityType routing strategy function. For details, see the description of the UseActivityType function in the *Universal Routing Reference Manual*. Keep in mind, however, that use of the UseActivityType function may affect performance of predictive algorithm and cause abandonment of outbound calls if outbound-dedicated Agents are occupied by non-outbound calls.

Transfer Mode

Universal Routing Server (URS) must be used in order to support agent groups in multiple campaigns. OCS will attach CampGroupID data to every outbound call to represent the CampaignGroup DBID. For every agent assigned to more then one active CampaignGroup, Outbound Contact designates only one Campaign Group for this agent. OCS will change the Campaign Group designation when the configuration of campaign priorities and the staffing limits have been changed, when agents have been logged in/out, or when dialing sessions for some Campaign Groups have been started/stopped.

Stat Server uses the same information in the CurrentCampaignGroupDBID statistic. URS can then target agents by using the CampaignGroup target.

Note:	A Campaign Group routing target should be used only when routing calls to groups with an agent assignment. When URS receives agent-assignment statistics from Stat Server, it might not apply them immediately to the strategy. This can cause the outbound call to be abandoned during processing by the strategy. Set URS pulse_time option to a very low value (for example, 1 second) to prevent such calls from being abandoned. For more information on this URS option, see the *Universal Routing Reference Manual.*

ASM Modes

OCS also supports agent assignments in multiple campaigns in ASM modes. When you set the agent-assignment option to true, OCS includes the Campaign Group name and DBID in its request for an engaging call. This ensures that a customer call will be merged only with an engaging call that originated for the same Campaign Group.

When OCS is making assignment decisions in ASM modes, OCS considers an engaged agent eligible for reassignment to other Campaign Groups and can make a reassignment decision and release the engaging call.

ASM Modes & CPD Server

CPD Server attaches the Campaign Group identifiers to the user data of the engaging call as follows:

- For ISDN conventional configurations, CPD Server monitors the Voice Transfer Destination DN and requests an update of user data that has these identifiers when EventRouteRequest is signaled for the engaging call.
- When using CPD Server with HMP, the identifiers are attached during the call-setup phase, as SIP Server can attach data to new a call whenever this data is provided in the SIP header of the initial call-setup message. SIP Server must be properly configured to attach these identifiers to an engaging call. In the configuration section of the SIP Server Application object, INVITE section must contain the following options (example):
    - userdata-1 = GSW_CAMPAIGN_GROUP_NAME
    - userdata-2 = GSW_CAMPAIGN_GROUP_DBID

Note:	Because user data is passed via SIP headers, the name of the Campaign Group used for agent assignment can contain only English letters, numbers and underscores.

When CPD Server bridges an outbound call to an engaging call, CPD Server gives first priority to the Campaign Group DBID (GSW_CAMPAIGN_GROUP_DBID) in order to match calls associated with the same Campaign Group.

VoIP ASM Modes Without CPD Server

Because all VoIP ASM modes do not use CPD Server, OCS can attach Campaign Group identifiers to both engaging and outbound calls when they are created. As a result, these calls can be routed and bridged to the correct Campaign Group. To enable this, the routing strategy must be designed to route engaging calls to Campaign Group targets.

Routing Strategy Design and CampaignGroupID

For agent assignment, you should design the routing strategy to distribute only outbound calls and engaging calls (if running a dialing session/campaign in an ASM mode) that contain the correct CampGroupID attached data to agents who meet the following criteria:

- The agent belongs to the specified group that is configured for that CampaignGroup.
- The agent's current statistic in Stat Server contains the CurrentCampaignGroupDBID, which is equivalent to the CampGroupID.

Both requirements can be achieved by using routing targets of Campaign Group type. In addition, for engaging calls placed in conventional ISDN scenarios in which an engaging call arrives to the Route Point DN without attached data, you must include the SuspendForEvent function (EventAttachedDataChanged) in the routing strategy before the routing decision is made.

Note:	This SuspendForEvent function should be applied only to engaging calls, (that is, those calls that originated from a certain DN list.)

Campaign Prioritization

Campaigns are prioritized according to the following definitions:

- An understaffed campaign is one in which the number of assigned agents is less than what is required to maintain a minimum service-level agreement (SLA).
- A staffed campaign is one in which the number of assigned agents is sufficient to maintain a minimum SLA, but is less than the maximum SLA.
- An overstaffed campaign is one in which that the number of assigned agents is sufficient to maintain the maximum SLA.

Agents are assigned based on the following priorities:

- All active campaigns that are configured with an equal priority will have an equal number of agents assigned to each campaign if all of the campaigns are equally understaffed or equally staffed.

Note:	If agents are not configured equally for all Campaign Groups—for example, if they have different skills—the above behavior does not apply. In this situation, the distribution for campaigns could be uneven, and some agents could be moved to an Inbound session instead of being reassigned to an understaffed campaign.

- When some campaigns with equal priority are understaffed, all non-assigned agents are assigned to the understaffed campaigns only. If there are several understaffed campaigns, an equal number of agents are assigned to each campaign that has equal priority.

- When there are no understaffed campaigns, all non-assigned agents are assigned to staffed campaigns only. If there are several staffed campaigns, an equal number of agents are assigned to each campaign that has equal priority.
- Agents are not assigned to overstaffed campaigns.

Refer to the descriptions of the following configuration options for more information about how the following options help to configure agent assignment:

- agent-assignment
- agent-assignment-priority
- agent-assignment-min-num
- agent-assignment-max-num
- inbound-agent-assignment-priority
- inbound-agent-assignment-min-num

Refining How Agent Assignment Functions

The following options address specific situations, providing you with more flexibility to refine the effectiveness of agent reassignment. OCS can now reassign agents to other activities if the dialing session to which they are currently assigned cannot use them because there are no available calling records or there are no free dialing ports.

- Agent reassignment if there are no available dialing ports or if there are no dialing records:
  - agent-reassignment-if-waiting-ports
  - agent-reassignment-if-waiting-ports-timeout
  - agent-reassignment-if-waiting-records
  - agent-reassignment-if-waiting-records-timeout
- Preventing agent assignment to inbound or low-priority outbound activities:
  - inbound-agent-assignment-min-num

Claiming of Agents by a Routing Strategy

The ClaimAgentsFromOCS routing function in Universal Routing 8.1, enables a routing strategy to make an immediate reassignment of one or more agents from outbound call handling to handling of inbound interactions.

This functionality is enabled using the ocs-urs-interact configuration option.

This function is used when a strategy that is processing inbound calls cannot process a call within the service objective criteria, and the call is important enough to require the use of agents currently busy with outbound activity.

For example, outbound agents can be claimed if a call has been waiting longer than some specified value.

For additional information on this functionality, see the chapter on Universal Routing Functions in the *Universal Routing Reference Manual*.

Agent Selection Logic

To select an agent for reassignment to an inbound activity, OCS goes through a list of agents specified in the `claim_agent.GSW_PERSON_DBIDS` key of the CM_ReqClaimAgent message and checks their states and assigned sessions. While checking agent states, OCS takes in consideration only Ready or Not Ready states, without differentiating them on their type or cause (such as ACW, on call, reason code, and so on). OCS gives preference to an agent in the Ready state, who is currently assigned to the Campaign Group with the lowest priority and the largest number of assigned agents.

OCS selects the most appropriate agent from all outbound-assigned agents satisfying request conditions, and reassigns that agent to the inbound activity for the time period specified in the request. If there is only one active dialing session and all agents in that session are not in the Ready state, OCS selects the first agent in the `claim_agent.GSW_PERSON_DBIDS` list. If OCS determines that agent is already claimed, it selects the next agent in the list.

An agent may not be reassigned to the inbound activity immediately if there are some present conditions in the Campaign Group. For example, if there are no Ready agents suitable for reassignment, OCS waits until a selected agent finishes the outbound call before reassigning the agent.

After the time expires, OCS reassigns the agent back to the outbound activity from where that agent was claimed, or to the other running outbound activity this agent could be assigned to, giving preference to the Campaign Group with the highest priority and the smallest number of assigned agents.

OCS attempts to accommodate each Claim Agent request from URS by reassigning one agent per request to inbound activities. It is expected that URS does not claim more agents than necessary by sending Claim Agent requests repeatedly.

Personal and Campaign Callbacks

The OCS-Desktop protocol enables the agent to schedule Personal or Campaign callbacks for specific records. A *Personal callback* is delivered to the agent who scheduled it. A *Campaign callback* is sent to any agent working in the specified dialing session. The records scheduled as either Personal or Campaign callbacks are marked accordingly in the calling list, so that reports can be generated.

For Personal callbacks, OCS follows up on them if a dialing session is stopped but not unloaded.

For Campaign callbacks, processing depends on the settings specified in the predictive_callback option.

Note:	See Communication Protocols in the *Outbound Contact Reference Manual* for information on the events that are distributed when an agent reschedules records.

Campaign Sequencing

Outbound Contact supports dynamic manual sequencing of running campaigns (campaign sequencing). Only one campaign per dialing session can be running at a time, but multiple campaigns can be loaded into memory. This enables the user to stop a previous dialing session for a campaign and automatically start the next one without disrupting the agents' work. When the new dialing session starts, the agents who are working in the group are notified. If new agents log in to the group after a dialing session for an outbound campaign has started, they are informed of the name and status of the currently running dialing session. OCS sends the name and status of a dialing session for a campaign as user events by means of a communication protocol.

Campaign sequencing can be configured according to the following parameters:

- Stop at Contacts—This counter is increased when a dialed record is updated in the database with the Answer call result. As soon as the counter exceeds the campaign threshold, the dialing session for the campaign is stopped and unloaded.
- Stop at Dials—This counter is increased when a dialing request or a PreviewRecord user event is issued. When the counter exceeds the campaign threshold, the dialing session for a campaign is stopped and unloaded.
- Stop at time—This occurs when a dialing session for a campaign is stopped and unloaded after it had been running up to the specified time.

Display Name on a Per-campaign Basis

This functionality enables you to specify a Display Name (in addition to CPN Digits) when dialing calls in an outbound campaign. The Display Name can be set at the level of the OCS application, Campaign Group, or individual record or chain of records.

Starting in release 8.1.1, OCS can use the CPNDisplayName configuration option to specify the name to be displayed. When dialing with SIP Server, this value is passed to SIP Server as the DisplayName parameter in the AttributeExtensions. When dialing with CPD Server in HMP transfer mode, this option is supported only if the CPD Server option tscall is set to true/yes.

For an individual record, or for a chain of records, this option can be set using the set_flex_attr custom action of the SCXML treatment. See Setting Options for Individual Records or Chain of Records for more information about custom actions.

Note:	Display Name is only supported when CPNDigits is also defined. Therefore, you must specify both the CPNDigits and CPNDisplayName options.

Outbound Schedules

Outbound Schedules enable you to automate control of dialing sessions that run in the contact center more than once; that is, their execution recurs. For example, if specific dialing sessions run every weekday, by using Outbound Schedules you can configure OCS to automatically run these dialing sessions without the need for manual everyday operations to manage them.

In Outbound Schedule terminology a *dialing session* is a Campaign Group.

An Outbound Schedule is an object that:

- Contains the instructions for when its execution recurs;
- Unites one or more dialing sessions that are to be executed at each recurrence.

Each dialing session that is a part of the Schedule contains additional configuration parameters that specify which commands are automatically executed by OCS for this dialing session and when they should occur.

The following actions are available for automation:

- Load
- Start (with dialing parameters)
- Stop
- Set Dialing Mode and/or Set Dialing Parameters
- Unload
- Force Unload
- Complete (Stop if required, followed by Unload)

The Outbound Schedule determines when and at what intervals the included dialing sessions are executed. A Schedule typically unites a set of dialing sessions that should be activated and deactivated based on the same rules.

OCS supports an unlimited number of Schedules, each Schedule containing an unlimited number of dialing sessions. Additionally, the same dialing session can be included in more than one Schedule. Dialing sessions that are included in the same Schedule do not need to be associated with the same Campaign, Group, or have anything else in common.

Note:	Genesys Administrator is used for configuration of Outbound Schedules. For instructions on creating and configuring Outbound Schedules, see the Provisioning > Outbound Contact > Schedules section in the *Framework Genesys Administrator Help*.

The following table shows the types of recurrence that are supported for Outbound Schedules:

Recurrence Types for Outbound Schedules

Type of recurrence	Available Settings	Duration of Execution	Example
Daily	Repeat every N day(s) at specific time	From specified start date until specified end date or indefinitely	* Run every day from 8 am to 5 pm, all summer, from 1st of June, 2011 until 31st of August, 2011. • Run every three days from 8 am to 12 pm, from 1st of September, 2011.
Weekly	Repeat every N week(s) on specified days of week at specific time	From specified start date until specified end date or indefinitely	* Run every Saturday and Sunday, 8 am to 1 pm, from 1st of June, 2011 until 31st of August, 2011. • Run every second Monday, beginning at 8 am, from September 1st, 2011.
Monthly	Repeat every N month(s) on the specified days at specific time.	From specified start date until specified end date or indefinitely.	* Run every month, on 15th, 16th,19th and last day of the month, 8 am to 5 pm, from 1st of June, 2011 until 31st of August, 2011. • Run every other month, on all days except for 10th and 11th, beginning at 8 am, from September 1st, 2011.
Once	Execute once at specific time.	Execute on specified date.	* Run once, 3 pm to 4 pm, on the 1st of July, 2011.

A Schedule with any type of recurrence may also specify stop conditions, such as "stop at the specified time of day" and/or "stop after a specified time interval". When an Outbound Schedule is stopped, it sends the Complete command (that is Stop, if necessary, followed by Unload) to all dialing sessions that are configured for this Schedule and are still active or running.

Schedule Items

Each dialing session that is associated with an Outbound Schedule is an item in that Schedule.

Execution of each Schedule item is controlled by execution points. Each execution point comprises the following:

- An action to be done to the dialing session.
- A condition that specifies when the action should be performed.

Execution points are of two types: instant and sequential. Any action that occurs at an execution point can be either sequential or instant. However, OCS executes instant execution points differently from sequential execution points, as explained in the following sections.

Sequential Execution Points

This category or set contains execution points that should logically be executed one after another. For example a dialing session can be started only after it has been loaded, and therefore Load should generally be the first execution point and Start should be the second in the set of sequential execution points. OCS activates the first execution point of this set when the Schedule is activated and only switches to the next (N + 1) execution point when current (Nth) execution point has been executed. There must be at least one sequential execution point for each Schedule item. Sequential execution points cannot be configured as repeatable.

Note:	When defining sequential execution points for a Schedule, the At time condition specified for a Stop action must not be earlier than the At time condition specified for the corresponding Start action. OCS does not assume that the earlier Stop time is on the next calendar day; instead, it starts the action at the specified time and then immediately stops it. This is not an issue when defining Stop actions with the After condition; OCS supports execution on the following day in this case.

Instant Execution Points

This set of execution points is activated immediately upon Schedule activation. Instant execution points are not dependent on each other. For example, if a Schedule item should be completed when a specific condition is met, the Complete action should be added to the set of instant execution points. This set can be empty or contain one or more execution points. Instant execution points can be configured as repeatable, which means they are executed each time the specified condition for the given execution point is met.

Execution Points

Execution Points that are created for each Schedule item define what actions are executed for the Schedule item and when they are executed. An execution point consists of an action (for example, Load, Start, or Unload) and conditions. The following table summarizes the conditions that can be applied to each action.

Conditions Applicable to Schedule Actions

Type of Condition	Parameters	Description
No condition	Not applicable	Executes action immediately
At time	Time of day	Executes action at specified time of day, for example at 3 o'clock.
After	Time period	Executes action after specified time period has elapsed, for example after 30 minutes.
Time in state	Name of dialing sessionName of stateTime spent in state	Executes action when specified dialing session assumes the specified state and spends specified time interval in that state, for example when dialing session 1 has unloaded, or when dialing session 2 has been running for 1 hour.
Value of statistics	Name of statisticsName of objectName of Stat ServerTarget value of the statistics	Executes when specified statistics on specified object received from specified Stat Server assumes specified value, for example when there are no more dialing records in the calling lists.

Note:	Conditions cannot be logically combined and each action can have one, and only one, associated condition. OCS attempts to execute the action specified for each execution point and continues Schedule execution regardless of the success or failure of the action. For example, if you attempt to Stop and Unload a Schedule item that is already stopped, OCS will fail to execute the Stop action. Nevertheless, OCS continues to the next action configured for the Schedule item and will next try to execute the Unload action.

How Schedules are Executed

All Outbound Schedules are executed independently. When it starts up, OCS reads all Outbound Schedules from the configuration and calculates the nearest date and time for each Outbound Schedule to be activated. When that activation date and time comes, for each Schedule item, OCS activates the first execution point of the sequential set and (if an instant set is present in the Schedule) all execution points of the instant set.

Activation of an execution point means that it is ready to be executed whenever the associated condition is met. For example, if the condition is the value of a statistic, OCS begins monitoring the specified statistic for the target value. If the condition specified as time interval after which the action occurs, OCS starts the countdown for the given time interval at which the execution point is to be activated.

An Outbound Schedule is deactivated after all Schedule items that it contains are unloaded or when the stop conditions for the Schedule are met. When the Schedule is completed, OCS calculates the time for the next activation (unless the Schedule was configured to run only once). The Schedule then waits in a dormant state until the next activation. In addition to control by the configured sets of sequential and instant execution points, OCS enables you to manually control the Schedule items at any time during Schedule execution in the same way as for independent dialing sessions. You can manually stop, start, and unload any Scheduled item using any administrative OCS client (that is, Genesys Administrator or Outbound Contact Manager). You can also check the status of any Schedule item in Genesys Administrator or Outbound Contact Manager.

Manual Operations in Schedules

Although you can perform manual operations on schedule items, manually unloading a schedule item breaks its link to a schedule and the schedule item loses its connection to the schedule. Then, when the standalone session is activated again, the schedule does not resume control over it. Only unloading disassociates the item from the schedule. The stop, start, and set dialing modes do not affect this association.

Dynamic Disabling of Schedules

Whenever an active Schedule is marked in Genesys Administrator as disabled, OCS continues executing it until it is deactivated (after all Schedule items that it contains are unloaded), or when the stop conditions for the Schedule are met. If a deactivated Schedule (in a dormant state) is disabled, OCS does not activate it at the next activation time.

Starting with release 8.1.507.08, the on-schedule-disable option determines how OCS responds to the dynamic disabling of an Outbound Schedule. This option is configured at the Application or Outbound Schedule level (at the Outbound Schedule level, the option can be configured only in the `default` section of the **Annex** tab of an Outbound Schedule configuration object).

Regardless of the option value, OCS does not activate a disabled Schedule at the next activation time.

Important

OCS applies any other dynamic changes to schedule parameters at the next activation.

Primary / Backup Notifications

For each active Schedule, the primary OCS notifies its backup about the Schedule's current state, including all of the parameters for activating and deactivating each Schedule item, and provides updated information any time a change to the Schedule status occurs. Because it is fully informed of all Schedule configuration and status information, in case of a switchover, the backup OCS can resume execution of the Schedule from exactly the same position where primary OCS has left off.

Architecture

These topics explain the required architecture for outbound campaigns:

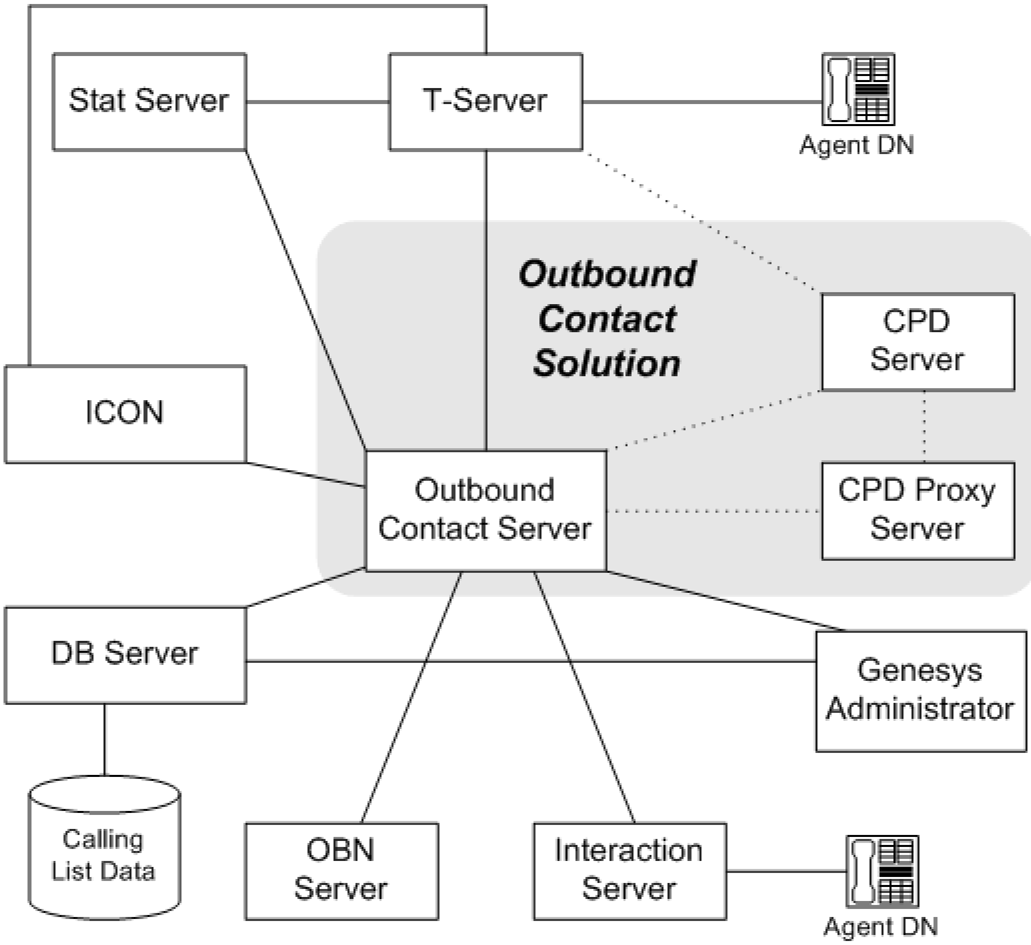
- Outbound Architecture Overview
- Dialing Overview
- Multiple Sites, Users, and Tenants
- Outbound Contact and HTTP Server
- Other Supporting Solutions

Note:	For architectural and related information about using Outbound Contact in an Outbound-VoIP environment, see Outbound VoIP Deployment.

Outbound Architecture Overview

Outbound Contact provides an integrated environment for creating, running, and reporting on outbound campaigns. The following figure illustrates how the Outbound Contact components integrate with other Genesys components in a standard T-Server environment.

The following subsections describes (respectively) the Outbound Contact components, Genesys Administrator, and how they fit into the architecture that is illustrated in the following figure.



Outbound Contact Architecture - T-Server Configuration

Outbound Contact Server

Outbound Contact Server (OCS), as the dialing engine, is the core of Outbound Contact. It retrieves information from Calling List(s) with the help of the DB Server and requests that T-Server, or a Dialogic board (through CPD Server), or another dialer (when using GVP 7.6 Outbound Notification (OBN) or Interaction Server) dials outbound calls or initiates outbound interactions. OCS monitors the call flow and reacts to call results and delivers responses to requests from agent desktops.

Desktop protocol for Outbound Contact enables OCS to communicate through T-Server with agent desktops. Third-party protocol for Outbound Contact enables OCS to communicate through a Communication DN with third-party applications. Outbound Contact is also capable of communicating with another OCS, Stat Server, and ICON.

Note:	Outbound Contact Server can also use the following: • Genesys Voice Platform (GVP) to dial outbound calls. For more information, see Genesys Voice Platform. • Interaction Server to deliver interactions to agent's desktops. See Genesys eServices (formerly, Multimedia) for more information. • Stat Server to provide Agent/Place status information solely. In any dialing mode, when monitoring an Agent/Place group is required, Stat Server should be properly configured; otherwise, Outbound Contact will not operate.

OCS also performs the following functions:

- Reads configuration data—including the options for configuration objects such as Agent Groups, Place Groups, and Campaign Groups — from the Configuration Database.
- Receives commands from Outbound Contact Manager, Genesys Administrator, or third-party applications for starting and stopping dialing sessions for Campaign Groups.
- Provides pacing algorithms for the dialing of outbound calls.
- Updates call results.
- Applies treatments to records based on call results.
- Processes Do Not Call (DNC) requests
- Checks the Do Not Call list for telephone numbers and customer IDs that should not be contacted.

Genesys Administrator

In release 8.x, Genesys Administrator replaces Outbound Contact Manager (OCM) as the UI for Outbound Contact. Genesys Administrator is a web-based UI that provides provisioning, monitoring, and deployment abilities for Genesys solutions. It supports the same functionality that was previously provided by OCM 7.6.

The Outbound-specific features of Genesys Administrator enable contact center managers to perform the following tasks:

- Manage calling lists:
  - Add, modify, and delete records in a list.
    Note: Genesys does not recommend changing `record_status` from Retrieved to Ready in a calling list while a campaign is loaded or running.
  - Create chained records in a list.
  - Move records among calling lists that have the same table format and that exist in the same type of DBMS.
  - Import/export calling lists.
- Browse Calling Lists, Campaign Groups, and their components:
  - Use dialing and viewing filters to select records by particular criteria.

Outbound Contact Manager or Genesys Administrator can *automatically* save a viewing filter when the user leaves the current calling list. When the user reopens the list, Genesys Administrator retrieves and displays the filter.

- View the relationships among campaigns, agent groups, and calling lists.
- Start and stop dialing sessions.

> If you start a dialing session with Genesys Administrator, and then Genesys Administrator stops, the dialing session continues to run. When Genesys Administrator restarts, OCS updates Genesys Administrator with the current status of the dialing session.

- Create and manage Schedules for Campaign Groups that enable the automatic control of the included Campaign Groups based on user-configurable criteria.
- Manage campaign sequences.
- Monitor the current status of dialing sessions for outbound campaigns.
- Specify or modify the optimization parameters for any campaigns and campaign sequences that are running.
- Update Do Not Call lists with data from external sources.
- Prioritize records for dialing by record type.

Note:	OCM 7.6 still can be used with Outbound Contact 8.x. If you are using OCM, Dialing Sessions are called Campaign Groups for starting and stopping operations.

Call Progress Detection Server

Call Progress Detection Server is an optional component for switches that do not have Call Progress Detection (CPD) capability. This server uses a Dialogic board to perform CPD for outbound calls and, if configured, to record voice files. As the dialer, CPD Server interfaces with OCS and the Dialogic board.

Dialing Methods

The way in which outbound dialing is performed depends on the dialing method and equipment that you use. There are several ways to perform outbound dialing:

- No Call Progress Detection (CPD) Server required if
  - A switch already has call progress detection capability, or
  - SIP Server is used with GVP applications.
- Transfer modes—The dialing mode in which calls successfully placed and identified with a positive CPD are transferred from the dialing resource to the VDN (Vector DN) for distribution to agents.
- ASM (Active Switching Matrix mode) modes:
  - For the Dialogic boards, the board connects to the switch through an ISDN carrier. In this mode, outbound calls are a combination of the following:
    - An internal call from the board to the agent desktop (an *engaging* call).
    - An outbound call from the board to the public switched telephone network (PSTN).

CPD Server connects these two calls on the Dialogic board when it is time to establish a call between an agent and a dialed number. Therefore, this method requires twice as many Dialogic ports as Transfer mode requires.

Note:	ASM mode is the only supported dialing mode when using T-Server for Cisco Call Manager or Unified Communications Manager and CPD Server with Dialogic boards. This limited support is due to T-Server's inability to monitor CPD or IVR ports.

For more information on these dialing modes, see Dialing Modes.

Engaging Calls

An *engaging* call usually is an internal call dialed from the call-processing port (dialing modes that used in conjunction when using CPD server) or from the Trunk Group DN (dialing modes that used in conjunction when using VoIP environment) to the Voice Transfer Destination DN and is delivered to an agent from this DN. An agent answers this call, which makes the agent engaged—that is ready to be connected (bridged) to an outbound call with a customer.

The ASM mode provides the following functionality to enhance performance:

- Eliminates unacceptable delays between the time when a customer picks up the phone and when the call is connected to an agent.
- Avoids lack of support for call transfers.
- Eliminates force-tone problems in which a customer can hear a signal intended only for an agent — for example, when a signal alerts the agent that a call is transferred.

In the Predictive with seizing dialing mode, a call is dialed automatically, based on a prediction of agent availability; the agent is then *seized (engaged)* before the called party answers. There is no delay after the customer is connected. OCS includes a campaign-pacing engine with a Predictive dialing algorithm that initiates the record dialing. In the *Progressive with seizing dialing mode*, a call is dialed automatically when an agent is already on the line and ready to take the call. *Progressive with seizing* means Progressive dialing in ASM mode.

In both modes, the agent should be on the line *(engaged)* before the customer answers. The differences between these two dialing modes are as follows:

- In *Progressive with seizing* mode, the agent is already reserved for the call when it is dialed.
- In Predictive with seizing mode, the agent is *predicted* to be available when the dialer places the call.

Note:	In release 7.6.101 and higher, the predictive algorithm accounts for the average engaging call dialing duration (defined as the time between an agent going ready and becoming ASM Engaged when the engaging call is delivered to the agent) for a given campaign and uses this value when calculating dialing pacing.

Centralized CPD Server

Outbound Contact supports a centralized CPD Server.

A contact center that has a multi-site environment can use a single CPD Server at its central location. With a centralized CPD Server, a routing strategy at the central site can send calls and events to Routing Points that are distributed across the network.

For example, in a multi-site environment where a group of agents are logged into a switch, you can use a single CPD Server that is connected to any T-Server at that same site. The agents do not have to be logged in to the same switch to which the CPD Server is connected. With a centralized CPD Server, a routing strategy at the central site can send calls and events to Routing Points that are distributed across the network. For more information, see also Multi-Sites.

Sample Call Flow with Centralized CPD Server

The following is a sample call flow using a centralized CPD Server. In this example, Steps 1 through 7 occur at the central location in a wide-area network (WAN).

The OCS sends a dial request to the CPD Server. (Both the OCS and CPD Server are at the central location.)

CPD Server sends a dial request to its Dialogic board.

The Dialogic board dials the customer's number. A switch conveys the call to the customer, while the Dialogic board performs call-progress detection.

After receiving an Answer call result, the CPD Server transfers the call to a Routing Point.

A T-Server, that is monitoring the switch to which it is linked (by a CTI link), informs the Router about the call at the Routing Point.

The routing strategy at the central location determines how to route the call. In this example, the routing strategy determines that the call should be routed to an agent at an outlying site.

The Router sends the call to Inter Server Call Control (ISCC), an external routing feature of T-Server.

ISCC sends the call to a second switch that is being monitored by a second T-Server. The second switch relays the call to an ACD Queue for a group of agents.

In this step, the second switch, the T-Server that monitors it, and the group of agents associated with the ACD Queue, are all at an outlying location in the network.

Voice File Recordings

CPD Server is able to record files in a plain VOX format (PCM/8000Hz/8bps). It creates two voice files for each outbound call:

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

Recording on Dialogic Ports

CPD Server records these voice files on the Dialogic ports used for outbound dialing. *Specific* ports are *not* dedicated for recording, but the *number* of ports to use for recording is configurable.

The option max-number-ports-to-record specifies the maximum number of agent ports on which CPD Server records at the same time. You configure this option in the CPD Server Application object in Genesys Administrator or Configuration Manager.

The cpd-recording option enables or disables the recording of File 1 only (call progress detection stage).

CPD Server counts the ports being recorded, and stops recording when the specified number of ports is engaged for this purpose. CPD Server will not record on another port until one is disengaged.

The number of times that the same call transfers from Agent DN to Agent DN does not increase the number of ports being used for the recording of that call. The call remains on the same port when it passes from agent to agent. This extended port usage for recording might necessitate the allocation of more Dialogic resources—that is, an increased number of ports to be used for outbound dialing in general.

For information about the max-number-ports-to-record and cpd-recording options, See CPD Server Options.

Voice File Names and Content

Voice file names have the format <Prefix>_<RecRef>_<PhoneNumber>_<Date>_<Time>_<AgentCustPortNames> where each of the variables in angle brackets is replaced by an alpha/numeric string. The following table lists and describes each of these variables.

Information in File Names

Variable in File Name Structure	Description
<Prefix>	Prefix for file type
<RecRef>	Recording reference number
<PhoneNumber>	Phone number
<Date>	Date of the recording
<Time>	Starting time of the recording
<Agent/CustPortNames>	Agent Dialogic port name

<Prefix>

The prefix in the file name is set by the following CPD Server configuration options:

- cpd-file-name-prefix for call result (the default is cpd_).
- conversation-file-name-prefix for conversation (the default is conv_).

For more information about these options, See CPD Server Options.

<RecRef>

CPD Server assigns a recording reference number to each voice file as part of its name.

<Date>

The date is presented in the following format:

<Month (two digits)> <day (two digits)> <year (two digits)>

For example, 060703 represents June 7, 2003.

<Time>

Time is presented in this format:

<Hour (two digits)> <Minutes (two digits)> <seconds (two digits)>

For example, 111545 represents 11 hours, 15 minutes, 45 seconds.

The hour is presented in 24-hour format as shown in these examples:

- 6:00 AM is 060000.
- 6:00 PM is 180000.

Examples of File Names

The following are sample file names:

- Sample file name for a call result voice file:

Call_Result_5551212_060703_060000_dtiB1T1.vox

- Sample file name for a conversation voice file:

Conv_5551212_060703_060005_dtiB1T1_dtiB1T2.vox

Note:	The call result file is created for call-progress detection only, and the file name contains only one CPD port. A conversation file, which records an actual conversation between an agent and a customer, is recorded in ASM mode only and the file name contains two CPD ports.

Answering Machine Final Beep Detection

Starting in release 8.1.1, CPD Server can be configured to detect the final beep after an answering machine greeting before processing the call any further. This functionality is especially useful when a message should be recorded on the customer's Answering Machine device as a result of the outbound campaign contact. Beep detection ensures that the message begins playing (and being recorded by the Answering Machine) only after AM greeting is fully over. This ensures that a message is recorded on the AM in full, and no starting part of the message is lost.

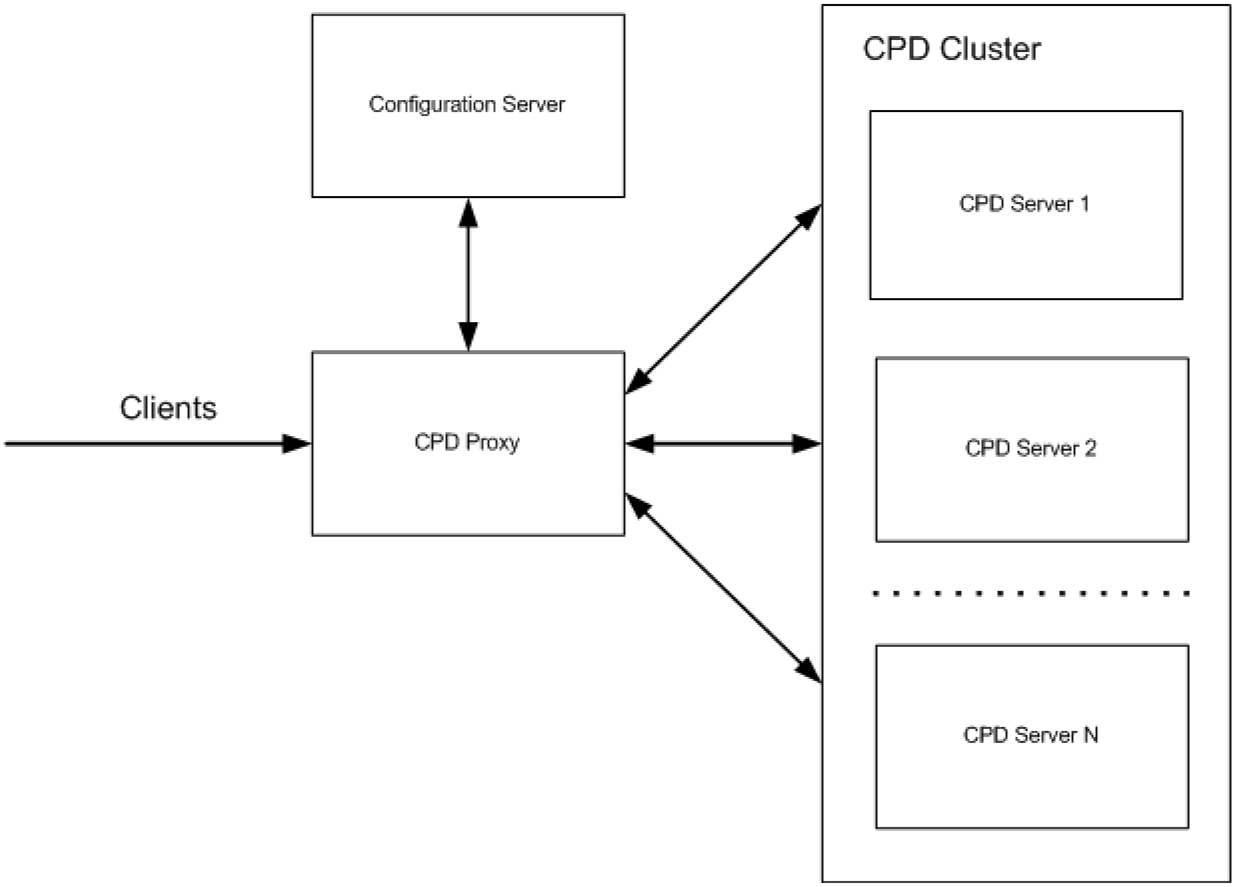
This functionality is available only when CPD Server is used with Dialogic DM3 boards or HMP software.

The following configuration options are used to provision this feature:

- am-beep-detect--Enables or disables this feature.
- am-beep-detect-timeout--Defines the time that CPD Server waits for the final beep.
- am-beep-tone-<n>--Defines AM final beep tones. See AM-beep-tones Section.

CPD Proxy Server

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



Configuration with CPD Proxy Server

Dialing Overview

A typical Genesys system consists of agent desktops, a T-Server or a SIP Server, one or more OCS applications, and a calling list database.

For an enterprise that uses T-Server, this central component, a computer-telephony integration (CTI) server provides a transport layer for all the system's telephony information. All information to and from the desktop and other servers passes through T-Server. The agent desktop and OCS are clients of T-Server. For an Outbound-IP environment, SIP Server functions as the transport layer.

Notes:	• The value of the T-Server and SIP Server `consult-user-data` option should NOT be set to `separate`. This value prevents OCS from correctly processing a call flow when an outbound call is transferred by a consultation call. • For information about a VoIP environment that uses SIP Server instead of T-Server, see Outbound VoIP Deployment.

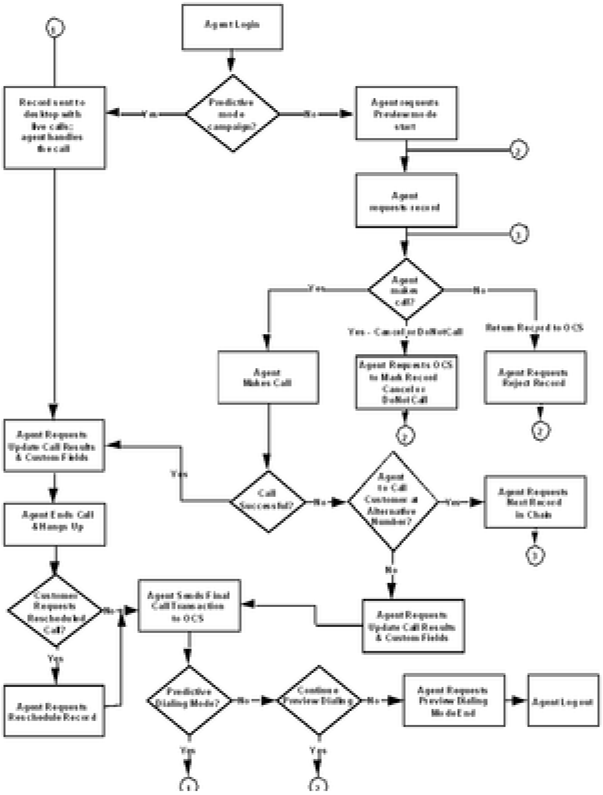
Smart Logout

The Smart Logout feature ensures that an agent is not logged out of a campaign until all calls that were assigned to the agent are returned to the calling list or are completed by the agent (refer to the "Agent Logout" section in the "Communication Protocols" chapter of the *Outbound Contact Reference Manual*). After the completion of Smart Logout (after OCS sends the `LogOutAcknowledge` message—refer to the "LogOutAcknowledge" section in the "Communication Protocols" chapter of the *Outbound Contact Reference Manual*), agents are assigned to the Inbound activity.

General Agent Desktop Interaction with OCS

The following figure illustrates the data flow for a typical customer record.

Note:	This figure is not specific to any particular sequence of events.

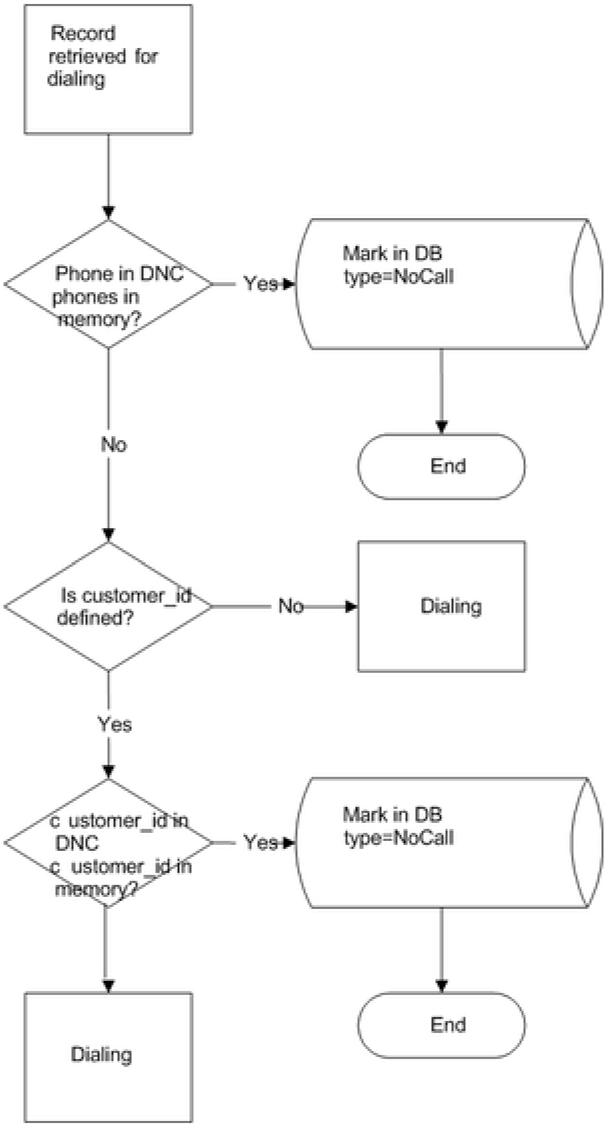

Agent Login
Predictive mode campaign?
Yes
No
Record sent to desktop with live calls; agent handles the call
Agent requests Preview mode start
Agent requests record
Agent makes call?
Yes
No
Yes - Cancel or DoNotCall
Return Record to OCS
Agent Makes Call
Agent Requests OCS to Mark Record Cancel or DoNotCall
Agent Requests Reject Record
Agent Requests Update Call Results & Custom Fields
Call Successful?
Yes
No
Agent to Call Customer at Alternative Number?
Yes
No
Agent Requests Next Record in Chain
Agent Ends Call & Hangs Up
Customer Requests Rescheduled Call?
No
Yes
Agent Sends Final Call Transaction to OCS
Agent Requests Update Call Results & Custom Fields
Agent Requests Reschedule Record
Predictive Dialing Mode?
No
Yes
Continue Preview Dialing
No
Yes
Agent Requests Preview Dialing Mode End
Agent Logout
1
2
3

Data Flow for Typical Customer Record

Check Against Do Not Call List

OCS reads all records from the table that is referenced in the gsw_donotcall_list Table Access and populates separate tables in memory with the unique values from the phone and customer_id fields. The tables in memory mirror the DNC List in the database. OCS checks these tables in memory during a Do Not Call predial check.

If OCS finds a phone number in the Do Not Call table in memory during a predial check, it applies the DNC restriction to the phone number and does not check the customer ID. The phone number has a higher priority than the customer ID if they are both in a dialing record. The following figure illustrates the predial check process.


Record retrieved for dialing
Phone in DNC phones in memory?
Yes
Mark in DB type=NoCall
End
No
Is customer_id defined?
No
Dialing
Yes
c ustomer_id in DNC c ustomer_id in memory?
Yes
Mark in DB type=NoCall
End
Dialing

Predial Check by Phone and Customer ID

Dialing Modes

Outbound Contact supports the following dialing modes:

- Predictive (a Transfer mode). For a standard environment, see Predictive Dialing Modes; for a VoIP environment, see Predictive GVP Dialing Mode.
- Predictive with seizing (an Active Switch Matrix [ASM]) mode). For a standard environment, see Predictive with Seizing Dialing Mode; for a VoIP environment, see Outbound-VoIP in ASM Dialing Modes.
- Progressive (a Transfer mode). For a standard environment, see Progressive Dialing Mode; for a VoIP environment, see Progressive GVP Dialing Mode.
- Progressive with seizing (an ASM mode). For a standard environment, see Progressive with Seizing Dialing Mode; for a VoIP environment, see Outbound-VoIP in ASM Dialing Modes.
- Preview. See Preview Dialing Mode.
- Push Preview. See Push Preview Dialing Mode.
- Direct Push Preview. See Direct Push Preview Dialing Mode.
- Power GVP. See Power GVP Dialing Mode.
- Last In First Out (LIFO). See Last In First Out Campaign Groups for Record Processing.

Note:	For more information about a VoIP deployment, see Outbound VoIP Deployment.

After a dialing session for a campaign has started, the user can change the dialing mode in the following ways:

- From Predictive to Progressive, and vice versa

Note:	If you change the mode from Progressive to Predictive while the Campaign is running, the Campaign switches to the new mode when it reaches the value set by the history_length option. If that value has already been reached at the time of the change, the mode changes immediately.

- From Predictive with seizing to Progressive with seizing, or vice versa

To change from any Predictive or Progressive dialing mode to Preview dialing mode, you must stop and restart the dialing session for the campaign group.

Outbound Contact supports call blending in any dialing mode. Among other considerations, OCS takes into account queued inbound calls. By using the progressive_blending_reserved_agents option, you can specify the number of reserved agents in Progressive dialing mode that should be available for inbound traffic. For more information about the progressive_blending_reserved_agents option, see Outbound Contact Configuration Options.

Dialing Modes and Third-Party Clients

Outbound Contact supports communication with Interaction Server, as a third party client, over a dedicated ESP port. For Campaign Groups in Push Preview and autodial modes the following third party client operations are supported if they meet the Request Attribute Requirements:

- Cancel records
- Mark records as Do Not Call
- Update records
- Process records
- Reschedule records
- Add records

Request Attribute Requirements

To support all operations of third-party clients:

- For requests to cancel records or mark records as Do Not Call, the request must include one of the following to identify the target record(s):
    - A valid record handle.
      If the record handle is specified in the request and OCS finds a record that has that record handle, the record or the entire chain is marked as canceled in the Calling List(s).
    - Contact Info.
      If Contact Info is specified in the request without the record handle, all records that have that Contact Info (and all chains, if any, that include these records) are marked as canceled in the Calling List(s).
    - Customer ID.
      If Customer ID is specified in the request without the record handle, all records that have that Customer ID (and all chains, if any, that include those records) are marked as cancelled in the Calling List(s).

Note:	Optionally, the Campaign name can also be specified to limit record cancellation to a Campaign that has the specified name. Interaction ID should be specified if URS is processing an interaction that was created by Interaction Server.

- For all other records, the request must include a valid record handle.

Routing Strategy Design

You can enhance your strategies which are used to route outbound call to the group of agents or group of places. You do so by using strategy blocks from the Outbound folder in Interaction Routing Designer (IRD). As with all the other blocks that you use in the strategy, you select the block you want to use (for example, Update record) and set the properties within this block, such as the name of Outbound Contact Server, the record handle (which you can extract from the user data of the outbound call), and user data values that you want to pass to OCS for update.

Predictive Dialing Modes

There are three predictive dialing modes:

- Predictive Dialing Mode
- Predictive with Seizing Dialing Mode
- Predictive GVP Dialing Mode

In the predictive dialing modes, Outbound Contact can dial more calls than the number of available agents in a group--making use of the predictive algorithm to determine the pacing.

Predictive Dialing Mode

The Predictive dialing mode predicts agent availability when it dials calls from a calling list. Predictive mode is recommended for high-volume, low-value campaigns.

Predictive with Seizing Dialing Mode

In the *Predictive with seizing* dialing mode (an ASM mode), a call is dialed automatically, based on a prediction of agent availability, and the agent is "seized" (engaged) before the called party answers. There is no delay time after the customer is connected. Predictive with seizing is an Active Switching Matrix (ASM) mode.

Predictive GVP Dialing Mode

In the Predictive GVP dialing mode, the non-zero waiting time is taken into account, implementing a call flow that uses a system queue for impatient customers. In this model, a waiting customer might give up (or abandon) the call after a period of time. Based on the assumption that these time periods are exponentially distributed random values with an average value that characterizes the customer's patience, the Predictive GVP algorithm calculates an outbound dialing rate to achieve the Target Value of the optimization method, by using the average customer patience time. The algorithm dynamically estimates the average patience time, based on the last call's statistics.

The following optimization methods can be used with Predictive GVP dialing mode:

- Agent Busy Factor
- Overdial Rate
- Average Waiting Time
- Average Distribution Time

Note:	OCS always uses Hot Start in the Predictive GVP dialing mode. If the options for the predefined statistical data (predictive_patience_time, predictive_hit_ratio,

	predictive_outbound_call_duration, predictive_inbound_rate, and predictive_inbound_call_duration) are not configured, OCS applies their default values, which may not be efficient.

Agent Feedback for the Predictive Dialing Modes

To improve dialing efficiency in the predictive modes (for example, to increase the Busy Factor for a given value of the Abandon Rate), OCS supports better optimization by accepting estimated call-completion time notifications from agents through the ReadyTime event. In this event, the agent desktop provides OCS with preliminary information on the time remaining for an agent to complete call processing.

As a result of these notifications, OCS evaluates whether an agent is trusted or distrusted. A *trusted agent* is one with a calculated trust factor that is less than the Overdial Rate optimization parameter for that campaign. In other words, the agent is available to handle another call when OCS expected the agent to be available.

A *distrusted agent* is one with a trust factor that is greater than the Overdial Rate for that same campaign.

The *trust factor* is the percentage of false notifications in comparison to all ReadyTime events received. A notification is considered false when an agent is not ready within the tolerance set in the time-to-ready-tolerance option after OCS receives the ReadyTime event. This tolerance is the allowed accuracy in the agent's estimation of when that agent will complete call processing and become Ready.

OCS and the predictive algorithm rely on the accuracy of this information to determine when the agent is Ready.

Note:	All agents are distrusted when a dialing session for a campaign first starts. The predictive_algorithm option must be set to advanced_small_group for the time estimate (when the agent will become Ready) that is delivered to OCS from the agent desktop to be considered by the predictive algorithm. Time to going Ready estimates are taken into account only if the Overdial Rate is used as the optimization parameter.

Note:	OCS does not support the Agent Feedback feature in the Predictive GVP dialing mode.

Predictive Algorithm

The predictive dialing algorithm performs real-time pacing calculations based on continuously sampled values for the following parameters:

- Statistics:

    - Average talk time (for both outbound and inbound calls)
    - Average not ready time
    - Average wrap-up time
    - Average contact rate per campaign (percentage of answered calls over total dialed calls)
    - Number of currently queued calls (both inbound and outbound)
    - Average inbound traffic volume based on the value of the history_length option
- Resources:
    - Number of available agents logged in the group
    - Number of available dialing ports
- Controllable parameters:
    - Optimized parameters, such as Agent Busy Factor, Overdial Rate, Average Waiting Time, Average Distribution Time (for Predictive GVP dialing mode only)
    - Goal value of the chosen optimized parameter

Outbound Contact can support an unlimited number of agent groups, each of which has its own independent dialing engine. In other words, each group can run its own outbound campaign with its own pacing based on real-time, campaign-specific statistics. By default, Outbound Contact treats all agent groups with which it works as blended groups by immediately adjusting pacing if both inbound and outbound calls are routed to the same group.

Optimization Parameters

In predictive dialing modes, Outbound Contact offers *optimization parameters*. These parameters directly affect the performance of dialing algorithms in the predictive dialing modes. These parameters include the following:

- Agent Busy Factor
- Overdial Rate
- Average Waiting Time
- Average Distribution Time
- Maximum Gain

You specify these parameters as part of the campaign group configuration process within Genesys Administrator > Provisioning tab > Outbound Contact > Campaign > Campaign Group.

Predictive Algorithm for Customer Abandoned Calls

This algorithm, used in the Predictive GVP dialing mode, implements the Average Distribution Time optimization method (in addition to the other four optimization methods). See Optimization Parameters. The Target Value for this optimization method is the average waiting time in queue for all calls, including abandoned calls. The waiting-time-in-queue begins the moment a call is queued to be transferred to an agent and ends when the call is connected to the agent or abandoned. The Predictive GVP algorithm calculates and logs the actual value of the average waiting-time-in-queue, which is the actual average distribution time.

The Target Value for the Average Distribution Time optimization method should be less than the average patience time. In accordance with the Erlang-A model for exponentially-distributed customer waiting times, the expected Abandon Rate is calculated as follows:

Abandon Rate = Average Distribution Time / Average Patience Time

The Predictive algorithm for the Predictive GVP dialing mode handles inbound traffic in the same way as all other Predictive algorithms, but also applies the following restrictions, and performs the following calculations:

Restrictions

- Restricts the outbound dialing pace so that the expected Abandon Rate does not exceed 30 percent.
- Restricts the outbound dialing rate so that the number of busy ports does not exceed the total number of ports available for Campaign Group. The number of busy ports equals the current number of calls in the dialing and queued states. The total number of available ports is the lesser of the total ports for the Trunk Group as reported by SIP Server in the EventResourceInfo parameter and the non-zero value of the number of CPD ports that are specified for the Campaign Group.
- Restricts the outbound dialing rate so that the number of requested outbound calls-per-second does not exceed the maximum number of dialing-requests-per-second, as specified for Campaign Group by using the dialing_rate_limit configuration option.

Calculations

- Calculates the Hit Ratio as the percentage of queued calls, relative to all dialed calls. Calls answered by a live persons and released without this queuing are considered a special type of No Contact call.
- Calculates the Overdial Rate as the percentage of abandoned queued calls, relative to all calls that are queued to be transferred to an agent. When the actual Abandon Rate (calculated from the time when the Campaign Group is started) exceeds the value that is specified by the predictive_max_overdial_rate configuration option, OCS switches the predictive algorithm to the Advanced Small Group algorithm, where the Target Value equals the predictive_max_overdial_rate configuration option or the pa-abandon-rate-limit configuration option, if the latter is smaller.

The Predictive algorithm for the Predictive GVP dialing mode supports the features that are currently implemented for the other Predictive algorithms, including Agent Assignment, the dynamic allocation of dialing rate and ports, and the self-diagnostic.

Configuration Options

The following options can be configured to optimize the Predictive GVP algorithm:

- predictive_patience_time--Use this option to specify the starting value of the average patience time.
- pa-selfcheck-adt-threshold--Use this option to specify the percentage of the target Average Distribution Time that the Predictive algorithm uses as the threshold to report the dialing performance degradation.

Note:	When you are using the Predictive GVP dialing mode, there are some important things to note. The pa-dial-expire and pa-queue-expire configuration options specify the timeouts for cleaning up stuck calls that are in dialed and queued states. The timeouts are used to distinguish normal

	and abnormal call flows. These options should be set to values that are higher than the maximum expected durations of each state. When you are using Predictive GVP mode, these durations might be significantly higher than for other dialing modes. The pa-hitratio-min configuration option specifies (as a percentage), the minimum value of the Hit Ratio, which is used to predict the dialing pace. The Hit Ratio is the percentage of queued calls, relative to all dialed calls. For Predictive GVP mode, calls are queued only when conversations with live agents are required. Therefore, the Hit Ratio can be significantly less than that of a regular Predictive mode. This option should be set to a value that is less than the minimum expected percentage of conversations with agents to all dialed calls.

Predictive Algorithm for Pacing Based on Gain/Loss Criteria

This algorithm uses the Maximum Gain optimization method for the Predictive dialing mode. This method dynamically predicts the optimal dialing pace to maximize the total gain of a campaign, based on a given ratio of benefit-per-established-call to loss-per-abandoned-call. This method has no target value in the strict sense, but Target Value is used to set the benefit-to-loss specific ratio (see Configuring Maximum Gain). Contact center managers can set this ratio in accordance with the real impact of established and abandoned calls on a campaign.

The Maximum Gain optimization method can be used in the Predictive, Predictive with Seizing, and Predictive GVP dialing modes. This method supports existing features for the other optimization methods, such as a preliminary Progressive-in-Predictive mode (using *Hot Start* with predefined statistical data), Agent Assignment, the dynamic allocation of the dialing rate and ports, and the self-diagnostic algorithm.

In addition, the Maximum Gain optimization method includes a special algorithm that maximizes the total gain for small groups of agents. This algorithm considers the benefit-to-loss specific ratio, the current readiness of agents to answer calls (including estimated call-completion time notifications from agents), the current numbers of dialed and queued calls in progress, and the estimated value of the Hit Ratio. It is configured in the same way as the Advanced Small Group algorithm for the OverDial Rate optimization method.

Example

To find the optimal Target Value for this optimization method, compare the beneficial effect of an increase in Agent Busy Factor to a negative effect of an increase in Overdial Rate. The Agent Busy Factor is determined by the number of established calls (answered by agents) and the Overdial Rate is determined by the number of abandoned calls.

Assuming that each established call provides a benefit (B) and each abandoned call incurs a loss (L), the total gain (TG) (the total B minus the total L) per average service time is calculated as follows:

TG = ABF*N*B - ODR*DP*L

Where:

- ABF = Agent Busy Factor
- N = Number of agents
- B = Benefit value per each established call

- ODR = Overdial Rate
- DP = Dialing pace (in Erlang units)
- L = Loss value per each abandoned call

Configuring Maximum Gain

Use Genesys Administrator to set the Maximum Gain optimization parameter for any type of Predictive dialing mode that uses this parameter. When this optimization parameter is selected, you will be prompted for the following two values:

- Established outbound call gain
- Abandoned outbound call loss

When Genesys Administrator communicates with OCS to store information in the Campaign Group configuration object, it uses the following formula to convert the two values to the following single value:

Target Value = (Established Gain/Abandoned Loss) * 100

Restricting the Abandon Rate

The predictive_max_overdial_rate configuration option is used to restrict the real Abandon Rate by switching the dialing mode from Predictive to Progressive if the current actual Abandon Rate exceeds the value of this option. However, switching to Progressive mode can significantly reduce the Busy Factor. Therefore, this option is best used to guarantee compliance with the legal requirements of Abandon Rates.

Genesys recommends that you use the pa-abandon-rate-limit configuration option (see pa-abandon-rate-limit) with the Predictive algorithm to restrict the expected Abandon Rate in all types of Predictive modes and for all the types of optimization methods.

Predictive Algorithm for Small Groups

Outbound Contact provides you with two predictive algorithms for dialing outbound calls during a campaign that are specifically designed to handle small groups of agents.

- Small group algorithm--This algorithm waits until all dialed calls are completed before dialing new calls.
- Advanced small group algorithm--This algorithm accounts for all inbound calls and also allows calls to be initiated, even if one or more calls remain in the dialing stage, instead of waiting until all dialed calls are completed.

You specify the small group algorithm by configuring the predictive_algorithm option with either the small_group or advanced_small_group value. You can choose to exclude current outbound calls with long dialing durations from pacing calculations made by the advanced small group algorithm. To do so, set the value for the pa-exclude-long-dialing configuration option to true.

Predictive Algorithm Self-Diagnostic

In release 7.6.101 and higher, the predictive algorithm has been enhanced with self-diagnostic capabilities to limit performance degradation, which could result in a deviation from the optimization parameters (Busy Factor, Overdial Rate, and Average Waiting Time) that are associated with the loss of a dialing pace.

Four options are associated with this self-diagnostic capability:

- pa-selfcheck-awt-threshold
- pa-selfcheck-bf-threshold
- pa-selfcheck-odr-threshold
- pa-selfcheck-interval

Diagnostic results are reported using log events that identify the expected cause of the degradation and addresses four areas:

- 12-50131: call-related factors, which include the following scenarios:
    - Unexpected call-related agents' activity, which can be caused by:
        - Agent state PA_StatAgentBusyRingOrDial that is not caused by Outbound, OutboundOther, or Inbound call types.
        - Agent goes to the Not Ready state from the Ready state but is not checked out.
        - Agents log out from the Ready state, but does not use Smart LogOut.
    - Low Hit Ratio.
    - Transferring outbound calls to another campaign.
    - Receiving outbound calls that are transferred from another campaign.
    - Engaging call duration influence.

A sample log event of this type might be:

12:35:49.844 Std 50131 Campaign Group 'Alpha Campaign@AlphaAgentGroup', Busy Factor defined 80, actual 50, call-related reason: Hit Ratio is low

- 12-50132: configuration factors, which include the following scenarios:
    - Long calls, which last more than outbound_agent_outlier_limit) on agents and thus agents become Not Available.
    - A dials-per-second limitation on the switch or Campaign Group level
    - Violation of Overdial Rate limitation (predictive_max_overdial_rate) with a transition to the Progressive-in-Predictive mode.
- 12-50133: malfunctions, which include the following scenarios:
    - Abandoned calls in the presence of Ready agents.
    - Cleaning or removing outbound calls (calls that are frozen in either a Dialed or Queued state and removed by the Garbage Collector).
    - Not Ready agents become Busy with new outbound calls (usually caused by an inadequate telephony event sequence).
    - The Outbound dialing rate is greater or less than predicted.

- 12-50134: external conditions, which include the following scenarios:
    - Lack of dialing ports.
    - Campaign Group--related waits:
        - Lack of dialing records.
        - Lost connections to important supporting servers (for example, T-Server or Stat Server).
        - Missing connection to the Do Not Call list.

Time-Optimized Predictive Algorithm

The time-optimized predictive algorithm provides contact centers with the flexibility to adjust the dialing pace for a specific period. The main benefit of this algorithm is that it allows you to increase the Busy Factor but still keep the overdial rate (ODR) within legislative limits. This improves the daily performance for the Busy Factor and the wait time for each agent.

In releases prior to 8.0, the classical predictive algorithm calculated the required dialing pace only based on the ODR that was set in the Campaign Group object. However, the actual overdial rate might be less at any given time because of factors such as:

- A non-zero waiting time for accepted outbound calls in a queue that the classical predictive algorithm does not take into account.
- A decrease in the hit ratio and in the average call duration and an associated increase in the number of available agents.
- The loss of dialing and queued outbound calls due to errors or transfers to other campaigns.

This time-optimized algorithm optimizes predictive dialing by recalculating the dialing rate so that the overdial rate criteria is met for a specific period of time (day, month, and so on). For example, if the target value for the overdial rate is 3 percent for an eight-hour business day, a new target value can be calculated and the dialing rate can be adjusted every hour so that the overdial rate for the eight-hour period meets the target value for that 8-hour period (as configured in predictive_max_overdial_rate. This increases the predictive dialing efficiency.

Depending on the current overdial rate, this time-optimized algorithm functions as follows:

- If the current overdial rate is less than the target, the acceptable overdial rate can be greater than the target for this time period. Therefore, the new dialing rate can be greater than the dialing rate that is predicted by the classical predictive algorithm. With this new algorithm, OCS increases the dialing rate by the difference between the target and the current overdial rates. This dialing rate increases the agent-busy factor. As a result, outbound campaigns that run under legislative limitations on the overdial rate can be more efficient.
- If the current overdial rate is greater than the target, the acceptable overdial rate must be less than the target in order to average out ultimately to meet the target value.

Therefore, the new dialing rate must be less than the dialing rate predicted by the classical predictive algorithm. In this case, the algorithm evaluates whether it can maintain the higher dialing rate, which increases the agent-busy factor, but at the same time meets the overdial rate target. If it cannot achieve the target, the algorithm chooses the dialing rate that produces the minimum predicted overdial rate. Decreasing the dialing rate decreases the overdial rate; hence, outbound campaigns that run under legislative limitations on overdial rate are more reliable as they take into account the busy factor, the overdial rate, and the changing call load over the course of a time period (such as an eight-hour day).

Note:	If the predicted busy factor for the chosen dialing rate is less than the predicted busy factor for the Progressive dialing mode, the time-optimized algorithm switches to the Progressive dialing mode, because it will result in a zero overdial rate for the remainder of the time period.

Configure the following options for the time-optimized algorithm:

- **predictive_algorithm**--Set the option to time_optimized_odr (a new value that was added in 8.0) that enables the use of this algorithm.
- **pa-odr-interval**--Specify the period of time, in minutes, over which OCS averages the overdial rate that must match the target overdial rate.
- **predictive_max_overdial_rate**--Specify the maximum allowed overdial rate for dialing in Predictive mode.
- **predictive_min_overdial_rate**--Specify the minimum allowed overdial rate for dialing in Progressive mode.

Predictive Algorithm and Engaging Call Dialing Duration

In release 7.6.101 and higher, OCS calculates the average engaging dialing call duration (defined as the time between an agent going ready and becoming ASM Engaged when the engaging call is delivered to the agent) for a given Campaign Group. The predictive algorithm takes this value into account when it calculates dialing pacing. With this enhancement, OCS changes the dialing rate based on the optimization parameters:

- When using the Overdial Rate optimization parameter, OCS can restrict the dialing rate so that its growth will be less than 10 percent.
- When using Average Waiting Time or Busy Factor, OCS can increase the dialing rate.

In addition, be aware of the following:

- As part of the self-diagnostic capability (Predictive Algorithm Self-Diagnostic), if OCS detects that the target value of the optimization parameter cannot be reached or that the dialing rate is restricted, OCS sends the STANDARD 12-50131 event, identifying the impact of the engaging call dialing duration.
- In a primary/backup configuration, OCS passes the value of the average engaging call dialing duration to the backup Predictive Algorithm.
- For the advanced small group predictive algorithm, when using the ASM dialing modes, after OCS receives the agent's ReadyTime notification specifying when the agent will become ready for engaging calls, OCS increases the value by the average engaging call dialing duration to predict when the agent will actually become ready for new outbound calls. OCS uses this corrected ReadyTime value when it compares the value to the value when the agent actually becomes ready to determine the agent's trust factor.

Progressive Dialing Mode

The Progressive dialing mode dials calls from a calling list when an agent is actually available. Progressive mode is recommended for low-volume, high-value campaigns. The number of calls that are dialed depends on the number of Ready agents. The next dialing attempt is triggered by either of the following factors:

- The AGENT_VOICE_MEDIA_STATUS - WaitForNextCall Stat Event message is received from Stat Server
- A non-positive call-progress result is received from the dialer

OCS places one (1) outbound call for one (1) Ready agent.

This dialing mode also supports Progressive Multiplier functionality ("lines-per-agent"), which enables you to change the number of outbound calls dialed for each available agent. This feature is configured using the pa-progressive-multiplier option.

Progressive with Seizing Dialing Mode

In the Progressive with seizing dialing mode (an ASM mode), a call is dialed automatically when an agent is already on the line and ready to take the call. *Progressive with seizing* means Progressive dialing in ASM mode.

In both Progressive with seizing and Predictive with seizing modes, the agent should be on the line before the customer answers; however, there is a distinct difference between these two dialing modes:

- In the Progressive with seizing mode, the agent is already reserved for the call when it is dialed.
- In the Predictive with seizing mode, the agent is *predicted* to be available when the dialer places the call.

This dialing mode also supports Progressive Multiplier functionality ("lines-per-agent"), which enables you to change the number of outbound calls dialed for a single engaged agent. This feature is configured using the pa-progressive-multiplier option.

Preview Dialing Mode

In Preview dialing mode, an agent requests one or several records from OCS, previews each record, and decides whether to dial a call. The agent interacts with the records by using the desktop application that communicates (via T-Server) with OCS. The OCS-desktop communication protocol enables the desktop application to perform the following tasks:

- Request and receive a record for preview.
- Update certain record fields.
- Schedule campaign or personal callbacks.

- Receive a scheduled record as an unsolicited event from Outbound Contact.
- Reject, cancel, or mark a record as Do Not Call.

Preview mode is recommended for low-volume, high-value applications for which individual ownership of accounts is the highest priority.

Preview dialing mode supports treatments based on call results.

Outbound Contact supports simultaneous participation by agents in multiple preview campaigns, including one Predictive/Progressive-mode campaign based on the following behavior:

- Outbound Contact Server distributes Campaign Status Notification messages for every campaign with which an agent is associated.
- If the agent participates in multiple preview campaigns, the record request for a specific campaign may be specified explicitly in the GSW_CAMPAIGN_NAME attribute of the PreviewRecordRequest. If not specified, Outbound Contact Server will select a record from any of the campaigns in a round-robin fashion. In this scenario, the GSW_CAMPAIGN_NAME attribute should have empty string value, or a NULL string value.

Note:	To take effect for the round-robin algorithm, an agent must be configured for the single group that is assigned to multiple campaigns.

- If an agent wants to retrieve preview records from multiple campaigns, and the agent_preview_mode_start option is set with a value of yes, the PreviewDialingModeStart and PreviewDialingModeOver campaign requests should be submitted from the desktop for every campaign.

Note:	The preview_release_nocontact_action option, instructs OCS on what action to apply to an agent's DN when the agent releases an outbound call that was placed manually in the Preview dialing mode but could not be established. For this option to take effect, the attached data of this manually placed call must contain the GSW_RECORD_HANDLE attribute with the record handle, which OCS had passed previously to the agent desktop. In release 8.x, OCS can process records with an Instant Messaging phone type for preview campaigns.

OCS monitors outbound calls that are manually dialed in the Preview mode if they contain UserData attached by OCS. If OCS does not receive `GSW_CALL_RESULT` in UpdateCallCompletionStats and RecordProcessed requests, OCS updates the `call_result` field with the call result received from the dialer. If no dial attempts with the attached UserData have been made, OCS updates the `call_result` field with “Answer” if the record has been retrieved with the `call_result` value of “OK” or “Unknown”. Otherwise, OCS does not change the value of the `call_result` field.

Push Preview Dialing Mode

The Push Preview dialing mode uses Interaction Server to "push" records to agents. The number of records that are "pushed" depends on the value of the Max Queue Size parameter. See Genesys eServices (formerly, Multimedia) for more information about this dialing mode.

Note:	In release 8.x, OCS supports stopping the processing of interactions and also updating records that have not been processed (that is, an agent did not receive the interaction) when the dialing session for an outbound campaign is unloaded. As a result, OCS updates the Calling List with the appropriate status. When the dialing session for a Campaign Group is started again, no customer receives duplicate contacts for the associated campaign.

When an agent requests a personal callback in push preview mode, the `agent_id` field is populated with the place id, instead of the agent id, in the following format: `tenant-dbid@place`. This means that callback is treated as a "place callback" because it is delivered to the same place. Therefore, if you have more than one agent using the same place, then the callback might not be delivered to the agent who made the request.

If the campaign dialing mode is changed from push preview to some other mode, personal callbacks do not work because the `tenant-dbid@place` format is used instead of an agent id.

Direct Push Preview Dialing Mode

Unlike regular Push Preview Dialing Mode, Direct Push Preview Dialing Mode does not require a connection to Interaction Server in order to have records pushed to agent desktops. Instead, OCS pushes preview records directly to agent desktops by sending unsolicited User Events (GSW_USER_EVENT = PushPreviewRecord) to agents that are Push Preview Ready. Refer to the Communication Protocols page in the *Outbound Contact Reference Manual* for a description of user events.

OCS will try to keep n records (where n >= 1) on each Push Preview Ready agent desktop, provided there are sufficient records in the OCS buffers.

Notes:	• Similar to Push Preview Dialing Mode, OCS uses a percentage for the **Buffer Size Coefficients** settings. For Direct Push Preview Dialing Mode, OCS uses a different calculation (`Number_of_agents` * `desktop-chains-limit`, instead of using the value of `maxQueueSize`). Therefore, Genesys recommends that you set the Campaign Group buffer size coefficients to **Minimum** = `100` and **Optimal** = `150`. • Similar to regular Preview Dialing Mode, OCS

	considers Direct Push Preview Dialing Mode non-autodialing. This means that when an agent participates in a Direct Push Preview campaign, the person gets assigned to an inbound activity, unless there is an autodialing campaign running for the same agent group. An agent can participate in multiple Direct Push Preview campaigns at the same time, as in Preview Dialing Mode.

If you are using Direct Push Preview with Workspace Web Edition (Genesys Web Services), refer to this article for additional configuration information.

Push Preview Ready Agents

When distributing Direct Push Preview records to agents, OCS considers the following agents *Push Preview Ready*:

- An agent is configured in the Agent Group.
- An agent is logged in (an agent is not required to be in the Ready state).
- An agent has sent a PreviewDialingModeStart desktop request to OCS (if the agent_preview_mode_start is set to `true`).
- An agent has less interactions on the desktop than the value of desktop-chains-limit option, where an interaction is any chain delivered to the desktop through a User Event:
    - PushPreviewRecord (actually a chain of such records).
    - PreviewRecord (actually a chain of such records).
    - Personal Callback (in any dialing mode).
    - Personal Rescheduled (in any dialing mode).
    - Campaign Callback if it is configured to be delivered as a User Event.

Each second, OCS calculates Push Preview Ready agents for a given Agent Group and places them in the ordered list as follows:

- Agents who do not have any processed Push Preview Records (for example, agents who have just logged in).
- Agents who have one interaction on the desktop. OCS sorts agents by the time when interactions are processed and moves the longest waiting agent to the top of the list.
- Agents who have two interactions on the desktop. OCS sorts agents by the time when interactions are processed and moves the longest waiting agent to the top of the list.

OCS starts from the top of the list and distributes one Push Preview Record to each agent, until agents are no longer Push Preview Ready or until available records are depleted.

Configuration Options

To enable this functionality, configure the following configuration options:

- direct-push-preview—Enables the dialing mode that OCS uses for a Campaign Group that is started in Push Preview dialing mode.
- desktop-chains-limit—Controls how many preview interactions can reside simultaneously at an agent desktop.

Last In First Out Campaign Groups for Record Processing

Starting with 8.1.525.04, OCS supports a special logic for record processing, called Last In First Out (LIFO). With this Campaign Group logic, after a record is added via a Web AddRecord API request to CX Contact, OCS places that record at the top of the dialing buffer (instead of placing it at the bottom of priority buffers), so that the record is dialed immediately. Each newer record placed on the top of the buffer effectively shifts existing records lower in the buffer position.

Campaign Groups with the LIFO logic can be used with any dialing mode. LIFO Campaign Groups do not retrieve records from the contact lists. They only operate on records that are added to the OCS buffer via the AddRecord API request over a Communication DN, originated from CX Contact.

When CX Contact issues an AddRecord notification to primary and backup OCS instances over the Communication DN, they immediately process it, as follows:

- Primary and backup OCSs generate an internal chain of records consisting of one record.
- Primary and backup OCSs add this record at the top of the dialing buffer.
- The primary OCS dials this record when the OCS internal logic allows a call to be placed (for example, an agent is in Ready state in Progressive dialing mode).
- The primary OCS applies a treatment to the record from a limited set of supported treatments as described in the next section.

Treatments

LIFO Campaign Groups support all treatments of type `Apply To Call`, similar to regular Campaign Groups. However, LIFO Campaign Groups support a limited set of `Apply To Record` treatments, as follows:

- Delegate for Processing—allows the execution of a Copy Contact Treatment (in CX Contact terms) over a LIFO record
- Execute SQL statement
- Mark as Agent Error
- Assign to Group
- Update all records in chain—updates records as `record_status = Updated`
- No Treatment

LIFO Campaign Groups ignore all other treatments of type `Apply To Record`.

Limitations

The LIFO Campaign Groups functionality in OCS requires CX Contact and is not supported in a standalone OCS deployment.

Configuration Options

To enable this functionality, configure the following configuration options:

- lifo-mode—Enables a LIFO Campaign Group record processing.
- lifo-record-expiration—Specifies the timeout for record expiration in the dialing buffer.
- lifo-unload-pace—Specifies the number of chains that OCS attempts to process per second while a LIFO Campaign Group is unloading.

Typical Data Flow Scenarios

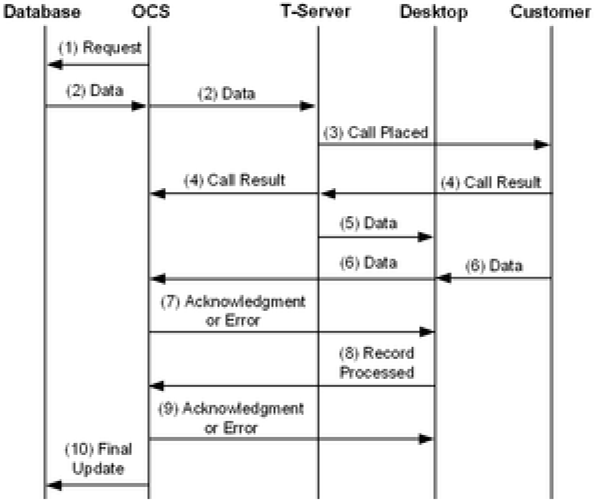
The following sections describes typical dialing scenarios for Predictive mode (or Progressive mode) and Preview mode.

Predictive Mode (or Progressive Mode)

The following is a typical data flow scenario for Predictive mode or Progressive mode:

1. When an outbound dialing session for the Campaign Group is started, OCS places the call.
2. If the call is answered by a "live" voice, it is connected to an agent. User data attached to the call is delivered to the agent's desktop.
3. The agent updates the user data and the call result. The agent then either processes the call or, if requested by the customer, reschedules the call for a later time as a personal or campaign callback.

The following figure illustrates the data flow for a typical Predictive mode or Progressive mode call. In this instance, the switch has Call Progress Detection (CPD) capability, and T-Server requests the switch to dial the customer number. Alternatively, you could configure the system with the CPD Server, which uses a Dialogic board to dial the call. In either case, the agent is already logged in to the system.



Predictive Mode Data Flow

OCS sends a request to the database to retrieve a record.

A record is sent back to OCS, which forwards it to T-Server along with a request to initiate a call.

T-Server initiates the call.

T-Server, or another device, interprets the call and determines that the call should be forwarded to an agent. The call result information goes back to OCS.

At the same time, T-Server transfers the call and customer-specific information to an agent. The agent and the customer are now connected.

The customer provides data to the agent, who updates the record. This information might include a rescheduled call date and time, a do not call back request, or other data. The agent sends the updated record to OCS.

After receiving the request, OCS sends an acknowledgment or error message back to the agent.

Note:	The agent can update the record as many times as necessary. With each update, OCS stores the data to its internal buffer and responds with an acknowledgment or error message.

When the agent and customer have finished, and the call ends, the agent sends the final event to OCS that the record is completed.

OCS responds to the agent desktop with either an acknowledgement that the transaction is complete or with an error message.

At the same time, the final update goes to the database.

Preview Mode

The following is a typical data flow scenario for Preview mode:

1. The agent requests that Preview mode begin.
2. The agent requests a preview record.
3. The agent either rejects the record or makes the call.

- If the agent rejects the record, it is returned to OCS. The agent then has two choices:
    - Request the end of Preview dialing mode and log out.
    - Return to Step 2.
- If the agent makes the call and if the customer requests no further calls for this campaign or no more calls ever, the agent requests OCS to mark the record as RecordCancel or DoNotCall. The agent then has two choices:
    - Return to Step 2.
    - Request the end of Preview dialing mode and log out.

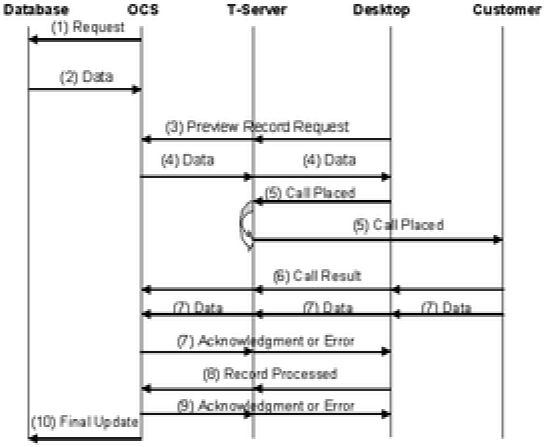
If the call reaches the customer, the agent updates the call results and custom fields. Next, the agent can terminate the call, sending the final call transaction back to OCS, or the customer can ask to reschedule the call.

- The agent then receives the rescheduled record. The agent then has two choices:

    - Return to Step 2.
    - Request the end of Preview dialing mode and log out.

- If the call does not reach the customer, the agent can call an alternate (chained) record.
- If there are additional records in the chain, the agent requests a chained preview record and returns the beginning of this Step 3.
- If there are no additional records in the chain, the agent updates the call results and custom fields, sending the final call transaction back to OCS. The agent then has two choices:
    - Return to Step 2.
    - Request the end of Preview dialing mode and log out.

The data flow for Preview mode starts differently from that of the data flow for Predictive mode or Progressive mode, but is identical once the agent is connected to the customer. The following figure illustrates the data flow for a typical Preview mode call. The agent is already logged in to the system.



Preview Mode Data Flow

1. The agent reports to OCS that the agent is ready to begin work in Preview mode.
2. OCS sends back an acknowledgment or error message.
3. The database sends a record or records to OCS, which forwards the records to the agent. These records might include previously scheduled calls.
4. The agent signals T-Server to initiate a call. T-Server initiates the call.
5. T-Server determines that the call is connected and signals the agent. The agent determines the status of the connection (fax, answering machine, modem, or customer). If the call is connected to a customer, the agent proceeds with the call.
6. The customer provides data to the agent, who updates the record. This information might include a rescheduled call date and time, a Do Not Call request, or other data. The agent sends the updated record to OCS.
7. OCS either sends an acknowledgment or error message back to the agent.

Note:	The agent can update the record as many times as necessary. With each update, OCS stores the data to its internal buffer and responds with an

	acknowledgment or error message (Steps 6 through 8).

8. When the agent and customer have finished, and the call is terminated, the agent sends the final event to OCS that the record is completed.
9. OCS sends the agent either an acknowledgment that the transaction is complete or an error message.
10. At the same time, OCS sends the final update to the database.

Separate Tracking of Engaging CPD Ports

OCS allows you to explicitly set a limitation on the maximum number of ports that will be used for agent engagements. This feature is only applicable to Campaign Groups that run with CPD Server or SIP Server in Progressive with Seizing (see Progressive with Seizing Dialing Mode) or Predictive with Seizing (see Predictive with Seizing Dialing Mode) dialing modes. When this configuration is in effect, the number of CPD ports that are used for dialing outbound calls and the number of CPD ports that are used for agent engagement can be configured and tracked separately by the OCS Predictive Algorithm. This allows more precise control of the dialing resources in the configurations where engaging lines and regular lines are separated. It also allows more flexibility if a number of Campaign Groups (possibly with different logical priorities) simultaneously share the same SIP Server or CPD Server and the same set of hardware lines.

Provisioning

Separate tracking of the engaging ports is controlled by the asm_channel_num option.

Dynamic Port Allocation Between Campaign Groups

Dynamic port allocation allows the optimization of dialing ports usage in configurations where multiple Campaign Groups share the same dialing and call progress detection resources (for example, Switch ports or CPD Server ports). This feature is only available for Campaign Groups running in the Progressive, Predictive, Progressive with Seizing and Predictive with Seizing dialing modes.

> **Important**
>
> Dynamic port allocation requires that Campaign Groups using the same dialer must all be running in either non-ASM mode or ASM mode. It does not support mixed-modes.

When dynamic port allocation is in effect, OCS performs the proportional dynamic allocation of ports, based on the estimated average numbers of ports required for each Campaign Group. OCS periodically recalculates these estimated numbers in accordance with the current dialing parameters, the number of available agents, and historical statistics. OCS calculates the maximum number of ports available for each Campaign Group so that this value is directly proportional to the estimated average number of ports required for this Campaign Group so that the sum of all these numbers for all Campaign Groups equals the total number of available ports.

Provisioning

Dynamic port allocation is configured by several parameters that combine to form a distinct configuration. The dynamic-port-allocation option defines whether dynamic allocation is in effect.

Dynamic Port Allocation in VoIP environments

For the dynamic allocation of ports among Campaign Groups running in a VoIP environment, OCS uses the total number of ports, which is reported by SIP Server in the **total-ports** parameter of EventResourceInfo messages, instead of the **channel_num** option that is defined for a Switch object.

Dynamic Port Allocation with CPD Server

In ASM mode with separated regular and engaging ports, OCS performs a dynamic allocation of ports when the Number of CPD ports property of the Campaign Group or the asm_channel_num option is set to 0.

For ASM mode with separated regular and engaging ports, the asm_channel_num option for these Campaign Groups can also be set to a desired value greater than 0. For these Campaign Groups, OCS will not perform dynamic allocation of ports, regardless of the dynamic-port-allocation option setting. In this case, OCS reserves ports for these running Campaign Groups in accordance with their Number of CPD ports property and the asm_channel_num option setting.

To restrict the maximum number of engaging ports that can be used for a Campaign Group in ASM mode, the asm_channel_num option can be set to a desired value greater than 0. In this case, OCS performs dynamic allocation for this Campaign Group only if the Number of CPD ports property is set to 0.

To restrict the total maximum number of regular and engaging ports that can be used for a Campaign Group in ASM mode with separated regular and engaging ports, the Number of CPD ports property can be set to a desired value greater than 0. In this case, OCS performs dynamic allocation for this Campaign Group only if the asm_channel_num option is set to 0.

Examples

The following three examples describe how ports are allocated dynamically between Campaign Groups that share the same dialing and CPD resources:

Example 1

- OCS runs two Campaign Groups which dial calls in Predictive mode through the same Switch using T-Server as a dialer. The channel_num option for the Switch configuration object is set to 100, and the dynamic-port-allocation option is set to true for the same Switch object. Both Campaign Groups have the Number of CPD Ports property set to 0. OCS estimates the required number of ports for those Campaign Groups to be 20 and 30. Since the total number of ports is configured as 100, the maximum number of ports allowed for the Campaign Groups are 40 and 60, respectively.

Example 2

- OCS runs two Campaign Groups which dial calls in Progressive mode through the same CPD Server. CPD Server is configured to use a total of 25 ports. The dynamic-port-allocation option is set to true at the CPD Server Application level. Both Campaign Groups have the Number of CPD Ports property set to 0. OCS estimates the required number of ports for the Campaign Groups to be 20 and 30. The total number of available ports is 25, therefore, the maximum number of ports that are actually available for the Campaign Groups are 10 and 15.

Example 3

- OCS runs two Campaign Groups which dial calls in Progressive mode through the same CPD Server. CPD Server is configured to use a total of 50 ports. The dynamic-port-allocation option is set to true at the CPD Server Application level. Both Campaign Groups have the Number of CPD Ports property set to 0. OCS estimates the required number of ports for the Campaign Groups to be 5 and 100. OCS considers the number of ports required by the first Campaign Group as *small*, since 5 > (50 / 2 * 0.3). Therefore, the maximum number of ports available for the Campaign Groups are 5 and 45.

Multiple Sites, Users, and Tenants

Multi-Sites

Outbound Contact Server supports multi-site operation. *Multi-site operation* means that you can run several dialing sessions for Campaigns independently in different locations using only one Outbound Contact Server, or you can run one dialing session from different sites at the same time. A site is a location with a contact center and an agent group.

OCS also supports environments in which agent groups or place groups are distributed across multiple sites. In this scenario, dialing is performed by one CPD Server or T-Server. OCS relies on ISCC call distribution to deliver successful calls to agents in multiple sites. If the value of the transfer_to_unknown_dn option is set to true or yes, OCS correctly processes the route of successful outbound calls or transfer calls between agents on different switches.

Note:	The use-data-from option must be configured with the value consult-user-data in the extrouter section of all T-Servers used with OCS. This value enables OCS to correctly process transferred calls between agents/DNs on different switches. When this value is set, AttributeUserData and AttributeConnID values are identified from the consultation call before the transfer or conference is completed. After the transfer or conference is completed, the EventPartyChanged message is generated, and then the AttributeUserData and AttributeConnID values can be taken from the original call. If a Voice Transfer Destination DN and a T-Server, used as a Dialer, belong to one switch in a multi-site environment, but the call is distributed by external routing to an agent who is on a different site, at least one queue from this site must be listed in the Origination DN list of the corresponding Campaign Group. This forces OCS to register regular DNs on the remote site and receive related events.

In multi-site scenarios, OCS generates statistical user events only on the Communication DN that is configured under the same Switch object as the Voice Transfer Destination DN.

OCS makes multi-site operation possible because it connects to the T-Server at each site and tracks the campaign data that is assigned to each agent group at each site.

One example of a multi-site campaign is as follows: A customer has three sites with agents at each site. One campaign is assigned to three agent groups and runs concurrently at all three sites.

Multi-site does not *necessarily* mean that a dialing session for a Campaign Group is started at one site from which calls are distributed to agents in one of three sites; however, with a centralized CPD Server, it is possible to run a dialing session for a campaign from the central location and connect to agents who are at outlying sites.

Multiple Users

OCS can support multiple instances of Genesys Administrator or Outbound Contact Manager that run concurrently.

Multi-Tenant Environment

A *tenant* is typically a contact center (single-site or multi-site). In a multi-tenant site, the administrator can grant permissions to other users to view their own tenant and run their own campaigns. Outbound Contact supports multiple tenants. However, Genesys Administrator (Outbound) works with only one tenant at a time.

Genesys Administrator supports the Environment tenant, which can be used by administrators of multi-tenant sites or by a user with permission that is granted by an administrator. When multiple tenants are set up, an administrator can select any tenant, including the Environment tenant, from the Select Tenant dialog box. The Select Tenant dialog box displays the tenants that are available to each user, based on that user's permissions.

The Environment tenant is created during the installation of the Configuration Layer. Although the Environment tenant is on the same level as all other tenants in the directory, it provides the administrator of a multi-tenant site with the following:

- A single location for all the configuration objects that are shared by sub-tenants (for example, Applications, Application Templates, and so on).
- Preset templates in some folders (for example, the Formats folder includes a preset format).
- Privacy through restricted access to all users except the administrator.

The elements of the Environment tenant simplify the process of adding new tenants.

Centralized Configuration and Shared HMP Resources

In release 7.6.101 and higher, Outbound Contact supports a centralized configuration and sharing of HMP resources in multi-tenant deployment. This functionality can be used for HMP deployment in the ASM mode and the Transfer mode because of rapid ISCC call routing between SIP Servers.

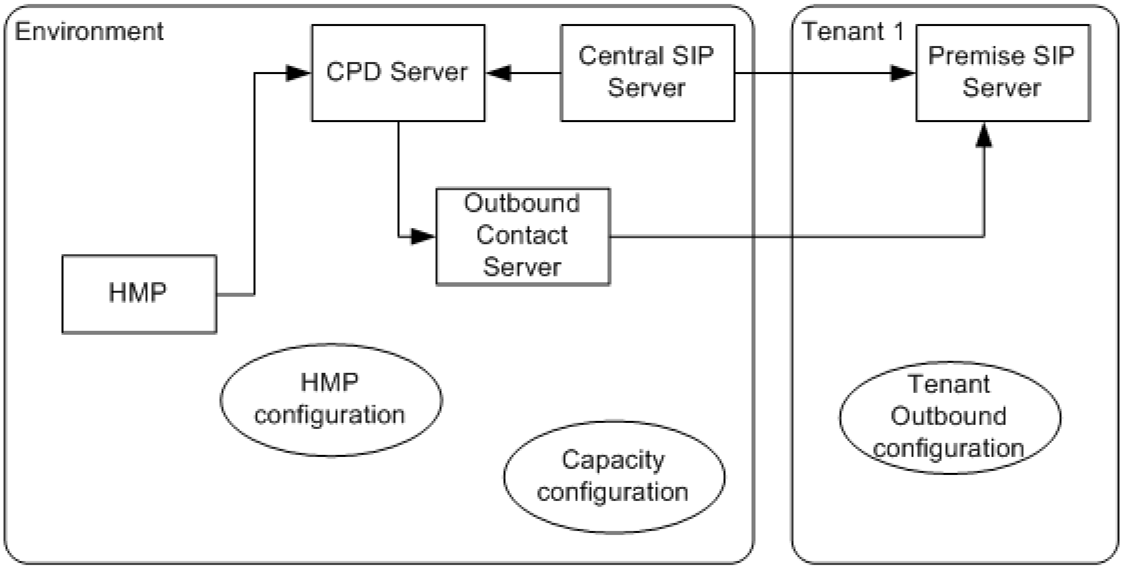
This functionality includes the ability to:

- Have a centralized configuration of HMP in a multi-tenant environment.
- Share HMP resources among multiple Tenants.
- Isolate HMP configuration from Tenant access.
- Configure a capacity-based resource allocation. (This configuration involves dedicating one CPD Proxy Server for each Tenant or any subset of Outbound Campaigns in a Tenant, as described in Managing Dialing Resources.)
- Configure a demand-based resource allocation. (This configuration is based on one CPD Server shared among all Tenants.)

Configuration Requirements:

- All configuration of HMP must be accomplished under Environment/SIP switch and is not accessible by Tenants.
- The Tenant configuration contains only Outbound/agent-specific configurations.

The following figure provides a sample architecture of this configuration environment.



Sample Architecture

Note:	The preceding figure does not show component connections. Be aware that CPD Proxy Server, which is used by the Tenant and connects OCS with CPD Server, is configured under Environment. The Tenant is not able to change its configuration.

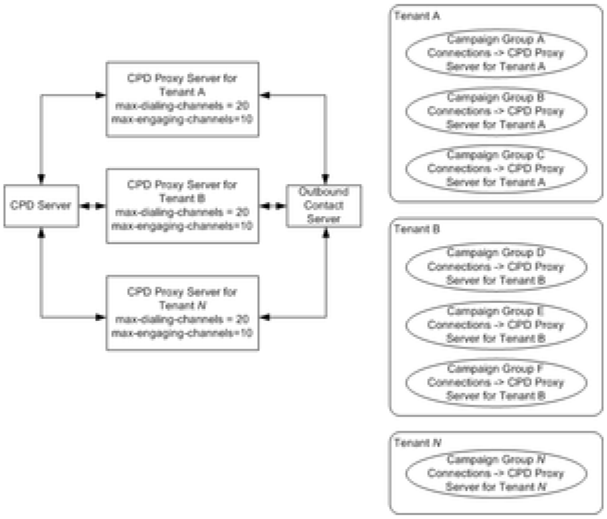
Managing Dialing Resources

To restrict the number of dialing resources available for each Tenant or for any subset of Outbound Campaigns in a Tenant:

- Configure CPD Proxy Server that controls the usage of dialing resources:
  - One for each Tenant that is used by all Campaign Groups for that Tenant, or
  - Multiple CPD Proxy Servers, one for each subset of Outbound Campaigns for the Tenant. For example, CPD Proxy Server 1 may have a maximum of 30 channels to handle Campaign A and B. CPD Proxy Server 2 might have a maximum of 50 channels to handle Campaigns C, D, and E.
- For each CPD Proxy Server, configure the max-dialing-channels option and the max-engaging-channels option, which set the maximum number of dialing and engaging channels for that CPD Proxy Server.

When OCS starts a dialing session for a Campaign Group, it requests and receives the maximum number available for these respective channels from CPD Proxy Server, as configured in these options.

The following figure illustrates how restricted dialing resources might be configured for two or more Tenants.


CPD Proxy Server for Tenant A
max-dialing-channels = 20
max-engaging-channels=10
CPD Server
CPD Proxy Server for Tenant B
max-dialing-channels = 20
max-engaging-channels=10
Outbound Contact Server
CPD Proxy Server for Tenant N
max-dialing-channels = 20
max-engaging-channels=10
Tenant A
Campaign Group A Connections -> CPD Proxy Server for Tenant A
Campaign Group B Connections -> CPD Proxy Server for Tenant A
Campaign Group C Connections -> CPD Proxy Server for Tenant A
Tenant B
Campaign Group D Connections -> CPD Proxy Server for Tenant B
Campaign Group E Connections -> CPD Proxy Server for Tenant B
Campaign Group F Connections -> CPD Proxy Server for Tenant B
Tenant N
Campaign Group N Connections -> CPD Proxy Server for Tenant N

Dialing Resources for Tenant Campaigns

Dialing and Distribution of Engaging Calls

When OCS is running a dialing session associated with a campaign for a particular Tenant, it sends requests to make transfer calls (for HMP Transfer mode) or make engaging calls (for HMP ASM mode) to the Route Point under Environment SIP switch, instead of the Voice Transfer Destination (VTD) on the Tenant SIP switch that is associated with a specific Campaign Group. A routing strategy (loaded on the Route Point)/URS routes these calls to the VTD under the Tenant SIP switch.

This two step process is required because CPD Server cannot manage calls across multiple switches. By using this method, CPD Server works with the Environment/SIP switch only and OCS works with the premise SIP Servers only.

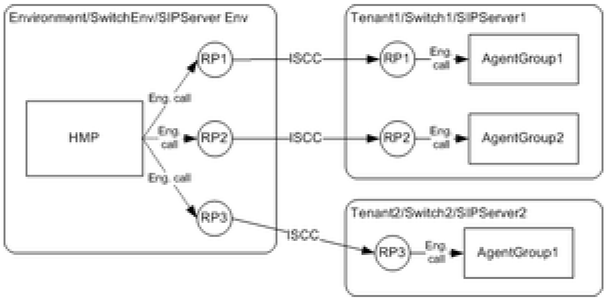
Note:	For additional information about configuring SIP Server to make transfers to external DNs, see the "Multi-Site Support" chapter in the *Framework SIP Server Deployment Guide*.

To accomplish this:

- For every VTD on the Tenant (premise) SIP switch, configure a corresponding Voice Transfer Destination on Environment SIP switch.
- Configure at least one Trunk Group DN for each Tenant to handle engaging calls.
- Create a routing strategy for each Environment/SIP Switch Route Point to route calls to the associated Route Point (VTD) on the Tenant switch.
- Configure the vtd-override option to specify the dialing number that OCS sends to CPD Server and that it will use as the destination number where a transfer call (HMP Transfer mode) or an engaging call (HMP ASM mode) is delivered.

Two Tenant Configuration Example

The following figure provides an example of a two Tenant configuration for handling engaging call distribution in HMP ASM mode.



Engaging Call Dialing and Distribution Diagram

In this configuration:

- Tenant1 includes the following:
    - Switch1, SIPServer1, two Campaign Groups (CampaignGroup1 and CampaignGroup2), and two Agent Groups.
    - Two route points, as VTD.
    - One external route point for the switch used by that Tenant (not shown in the preceding figure).
- Tenant2 includes the following:
    - Tenant2 has Switch2, SIP Server2, one Campaign Group (CampaignGroup3), and one agent group.
    - One route point as the VTD.
    - One external route point (not shown in the preceding figure).
- The Environment SIP switch DNs includes the following:
    - Three route points. Two route points are the central VTDs for the two Campaign Groups for Tenant1. The third route point is the central VTD for the Campaign Group for Tenant2.
    - Two Trunk Group DNs for transferring engaging calls to premise SIP Servers.

Outbound Contact and HTTP Server

OCS supports communication with OCS clients, such as GVP using HTTP requests and responses. This enables an HTTP client to send requests to update records over HTTP or HTTPS.

Starting with release 8.1.516.04, OCS supports custom security-related headers sent by the OCS HTTP Server in HTTP responses. The Application-level option **http-custom-headers** allows you to specify custom headers sent in OCS HTTP Server responses. The following HTTP headers are supported:

- X-Frame-Options: SAMEORIGIN
- X-XSS-Protection: 1; mode=block
- X-Content-Type-Options: nosniff
- Content-Security-Policy: script-src 'self'; object-src 'self'
- Strict-Transport-Security: max-age=31536000; includeSubDomains

Configuring HTTP/HTTPS

To configure HTTP support, see the procedure Configuring OCS for HTTP.

Support for HTTPS

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 Protection of Data in Transit in the *Genesys Security Deployment Guide*.

For the installation procedure of Genesys Security Pack on UNIX, see the Installing Genesys Security Pack in the *Genesys Security Deployment Guide*. On Windows platforms, support for SSL/TLS is integrated into the operating system.

How Outbound Contact Uses HTTP/HTTPS

The HTTP server-side interface in OCS, HTTP Proxy, is a gateway between HTTP version 1.1 and outbound protocols (the Desktop protocol and the Third Party protocol).

At its startup, when OCS determines that an HTTP port is configured, it automatically starts HTTP Proxy. As the end user, you do not need to start this child process manually.

In OCS, this HTTP Proxy translates:

- HTTP requests into OCS proprietary protocols.
- OCS proprietary protocols responses into HTTP responses.

HTTP Proxy supports UTF-8 encoding. OCS sends the value of the encoding option value to HTTP

Proxy, which can transcode (or convert) the JSON body of HTTP requests or responses (such as `AddRecord` and `PreviewRecord` events, or pre-dial validation requests) according to the value specified. If no value is specified, no conversion takes place.

At OCS startup, OCS starts HTTP Proxy, and HTTP Proxy opens the associated port that is configured in the OCS application object. If HTTP Proxy cannot open the port, it shuts down. If HTTP Proxy terminates unexpectedly, OCS tries to restart it after the Reconnect Timeout expires. This Reconnection Timeout is specified on the OCS Configuration tab/Server Info section (in Genesys Administrator).

Previously, HTTP Proxy did not have its own log and passed log messages to OCS, which logged them into the OCS log file. Starting with version 8.1.504.14, OCS writes all HTTP Proxy-related activity traces to a separate HTTP Proxy log. This log file is stored in the same location as the OCS log, but has the suffix **_HTTPPRX** added to the filename.

Note:	Be aware that using NMAP or similar port scanners can cause OCS to close HTTP ports.

Configuring OCS for HTTP

Purpose

- To configure the OCS application object to communicate with clients using HTTP requests/responses.

Start

1. In Genesys Administrator, go to Provisioning > Environment > Applications and double-click your OCS application object.
2. Configure a separate listener port on the Configuration tab/Server Info section with the Connection Protocol set to `http`.

End

Configuration Changes

Dynamic changes to this port are not supported. As a result, any such changes are not communicated to an HTTP Proxy that is already running. However, any HTTP Proxy started that is after that port change will reflect the change.

HTTP terminates if the HTTP listener port is removed from the OCS Configuration tab/Server Info section or if the connection is lost between OCS and HTTP Proxy.

Primary and Backup OCS and HTTP Proxy

If you have configured primary and backup OCS applications, the primary instance starts an associated HTTP Proxy and opens a listening port. The backup OCS only starts HTTP Proxy and opens a listening port when it is switched into primary mode.

Other Supporting Solutions

This topic describes other Genesys applications that enable and support Outbound Contact. They include the following solutions:

- Framework 8.x
- Genesys eServices (formerly, Multimedia)
- Genesys Voice Platform
- Reporting

Framework 8.x

Outbound Contact 8.x is built on Framework 8.x, which includes DB Server, Configuration Server, the Management Layer, T-Server, and Stat Server.

DB Server

Outbound Contact relies on the DB Server for access to data in the Calling List database tables, any Do Not Call list, and any other additional lists that are stored in the database.

> **Important**
>
> The DB Server is limited to a maximum of 1024 connections for each OCS instance. As Outbound Contact opens a new DB Server connection for every Campaign + Campaign Group + Calling List combination (for example, if three campaigns are loaded that share a single calling list, this opens three DB Server connections; if one campaign is loaded, and it is associated with six calling lists, then six connections are opened), Genesys recommends that you distribute these so that no single OCS instance requires more than 1024 DB Server connections.

Configuration Server

Outbound Contact is configured through the Configuration Layer of Framework. More specifically, you use Genesys Administrator to configure the components (applications and configuration objects) that enable Outbound Contact. OCS reads the configurations from the Configuration Server in order to perform its operations. Outbound Contact supports dynamic reconfiguration—that is, you can make changes to Outbound objects (such as Calling List, Table Access, and Campaign objects) in Genesys Administrator or Outbound Contact Wizard without stopping and restarting Outbound Contact. Messages that notify the user of changes are displayed in Genesys Administrator. For information about configuring Outbound objects, see Manually Define Outbound Configuration Objects/Table Access Object.

Management Layer

The Management Layer of Framework includes Local Control Agent (LCA), Message Server, Log Database, Solution Control Server (SCS), and Solution Control Interface (SCI). LCA is used to start and stop applications. Message Server provides centralized processing and storage of every application's maintenance events. Log Database stores events as log records, so that they are available for further centralized processing. SCS is the central processing center of the Management Layer. SCI displays the status of all installed Genesys solutions and provides information about each active alarm condition.

> **Important**
>
> Genesys Administrator can be used instead of SCI.

T-Server

T-Server is the key element of Genesys computer-telephony integration (CTI) because it handles call traffic. Each T-Server is linked to a switch, which it monitors. T-Server is also a "messenger" that enables communication among Outbound Contact components. For details, see the "Communications Protocol" chapter in the *Outbound Contact Reference Manual*.

Network T-Server

Outbound Contact supports Network T-Servers in deployments that do not include CPD Server. To enable this functionality in Predictive dialing mode:

- The Service Number DN type, in addition to any applicable DNs of type ACD Queue and Route Point in Genesys Administrator must be identified on the appropriate place group or agent group.
- The Campaign object must specify the same Service Number DN as the Voice Transfer Destination within the place group or agent group that is specified on the Group tab.

> **Important**
>
> Outbound Contact Server does not support the load-balancing capabilities of the Network T-Server. Outbound Contact Server should not be configured to connect to multiple Network T-Servers that are associated with the same switch.

Stat Server

Stat Server provides the data source for the CCPulse+ application. It also provides agent-state information to Outbound Contact Server so that it can dial effectively.

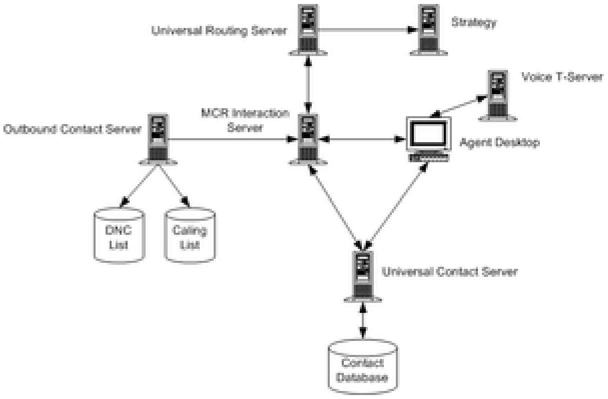
Optional Solution Integration

In addition to the Framework components, Outbound can work with the following Genesys solutions:

- Genesys eServices (formerly, Multimedia)
- Genesys Voice Platform
- Reporting

Genesys eServices (formerly, Multimedia)

Outbound Contact Calling List and Campaign management are now integrated with the Customer Information Management (CIM) platform. This feature extends the Outbound Contact Calling List and Campaign management to multimedia channels, and it improves agent efficiency by pushing preview records to agents without waiting for an agent request. Therefore, it enables the prioritization of outbound interactions with other interaction types for blending purposes. This dialing method can also be referred to as "proactive routing" or "push preview". The following figure illustrates how Outbound Contact integrates with the CIM platform.



CIM Integration with Outbound Contact

The basic process for completing a proactive interaction is as follows:

- OCS submits a preview interaction request to Interaction Server.
- Interaction Server requests that Universal Routing Server execute a strategy to determine which agent to send the interaction to based on skill abilities. The agent must be logged in to Interaction Server and can still process other interactions during this process. The communication between the agent desktop and OCS is through Interaction Server in this scenario.
- The agent processes the preview record appropriately, transfers the call to another agent, or places the call back into the queue.
- Special strategy blocks enable the routing strategy to process the preview call interaction automatically without agent involvement.

Notes:	• Refer to the Universal Routing documentation for more information about configuring

	strategies for proactive interaction routing. • Refer to Campaign Group Object for more information about configuring a Campaign Group object for proactive routing (or push preview) campaign.

ESP Port

In the Push Preview dialing mode, Interaction Server must be specified in the Connections tab of the CampaignGroup object to establish a client-server relationship between OCS (as the client) and Interaction Server (as the server). However in order for OCS to process requests and responses from Interaction Server, OCS must be a server for Interaction Server and receive these responses on a special ESP-type port rather than its default listening port.

Create this additional port PortID—called ESP—on the Server Info tab of the OCS Application object. Use this ESP PortID instead of the default PortID, when you are configuring a connection to the OCS Application object on the Connections tab of the Interaction Server Application object. In effect, this makes OCS a server for Interaction Server and Interaction Server a client for OCS.

Note:	This ESP PortID will be available as a connection in the Interaction Server Application object only after you configure PortID in the OCS Application object.

Interaction Media Type

Interactions that are submitted to Interaction Server must contain a certain media type. By default, OCS creates interactions with the outboundpreview media type. However, OCS can assign other media types to interactions from particular calling list or to interactions that were created for the entire campaign. Refer to the interaction-media-type option on for more information.

Notes:	• Refer to the direct-personal-callback option when you are configuring this solution. • Refer to the "Proactive Interaction Support" section in the Outbound Contact Reference Manual for more information about Proactive Interactions with Outbound Contact.

Genesys Voice Platform

Outbound Contact can integrate with Genesys Voice Platform (GVP) in order to run campaigns in the Power GVP or the Progressive GVP dialing mode.

Proactive Contact with GVP 7.6 Deployment

In a T-Server deployment that is running in the Power GVP dialing mode, OCS uses the Outbound Notification Manager (OBN) application as a connector to GVP to request that GVP dial records. The OBN Manager GVP application should be created in Genesys Administrator to provide its host and port information to OCS. This application must be specified on the Configuration tab/Connections section of the Campaign Group configuration object (in Genesys Administrator) to enable GVP integration. This configuration will enable basic connection capabilities.

If full connection abilities—including Advanced Disconnect Detection Protocol (ADDP) is required — this application also should be present on the Configuration tab/Connections section of the OCS application, where the ADDP connection protocol can be specified.

Note:	For more information about how to configure the Campaign Group configuration object for the Power GVP dialing mode, see Campaign Group Object.

When configured to enable connections, OBN opens a predefined port (for listening purposes) and then accepts requests from OCS in a client-server fashion.

Dialing Algorithm

In a GVP-assisted campaign, dialed outbound calls are dialed from GVP ports, and they can be completely processed and released by GVP. GVP issues information about call processing (such as call results and modified user data) to OBN Manager, which relays this information to OCS. OCS processes this information accordingly, applies any necessary treatments, and updates the records.

Do Not Call Records

OCS issues an EventOBNRecordStopProc message to OBN when it receives a DoNotCall request and determines that records are currently being processed by OBN. OBN immediately attempts to stop processing the specified record.

Calculating the Number of Records

The procedure that OCS uses to calculate the number of records (or chains) to retrieve from the database when a dialing session for a Campaign group is loaded, is similar for both Push Preview and Power GVP modes, but differs from traditional Predictive, Progressive, or Preview modes.

For the Power GVP dialing mode, when a dialing session is loaded for a Campaign Group object configured to use this dialing mode, OCS calculates the number of records for retrieval from the database as a percentage (specified in the Optimal Record Buffer Size dialog box) of the MaxQueueSize parameter for this Campaign Group. If more than one Calling List is used in the Campaign, the number of records retrieved from each Calling List is determined by the specified list weight.

OCS then submits the records to OBN Manager using the following criteria:

- The total number of dialing requests that are submitted is equal to the MaxQueueSize value for the Campaign Group.
- OCS retrieves records from the Calling List table in the database to replenish its buffer when the number of Calling List records in the OCS buffer is less than the number calculated as the percentage (specified in the Minimum Record Buffer Size dialog box) of the MaxQueueSize value for this Campaign Group.

Notes:	• Refer to the dialer-ttl and dialer-num-attempts options when you are configuring this solution. • Refer to the GVP 7.6 documentation for more information about using Outbound Contact with the OBN application.

GVP 8.1 VoIP/SIP Server Deployment

For an description of this deployment, see Overview of the GVP VoIP/SIP Server Deployment.

Reporting

The Genesys Reporting solution contains tools for collecting historical and real time data, and for viewing and analyzing contact center performance. In Outbound Contact, the calculation of some real-time metrics is provided by the OCC Extension. All data that is required for the OCC Extension is directly submitted by OCS through the OCS DataStream protocol. For more information about the Genesys Reporting solution and the templates provided with Outbound Contact, see the *Reporting Technical Reference Guide for the Genesys 7.2* Release.

Outbound-VoIP Deployment

These topics explain the configuration when you use Outbound Contact in a VoIP environment:

- Overview of the GVP VoIP/SIP Server Deployment
- Dialing Modes and VoIP Deployment
- VoIP Dialing Modes
- Outbound-VoIP in ASM Dialing Modes
- Outbound-VoIP in Transfer Modes
- Assured Connection

Note:	Outbound-VoIP deployment is not supported in a multi-tenant environment when it is used with GVP 8.1.1. To support Outbound-VoIP deployment in a multi-tenant environment, GVP 8.1.2 or higher must be used.

Overview of the GVP VoIP/SIP Server Deployment

Outbound Contact supports a VoIP deployment that enables automated outbound dialing when using SIP Server 8.0 and GVP 8.1 (Media Server).

In this environment, Outbound Contact and Universal Routing components function on top of SIP Server, Media Server, and a SIP Media Gateway.

The **VoIP Environment/Outbound Contact Architecture** figure illustrates the architecture. Instead of requiring new dialing modes for a VoIP deployment, Outbound Contact uses the existing dialing modes and runs these as VoIP modes when Outbound Contact is configured as VoIP-ready (see the procedure Configuring Outbound Contact and SIP Server for a VoIP Deployment).

GVP 8.0 uses SIP Server and MCP/Media Server(s) call processing in a VoIP environment. It can perform call-progress detection on outbound calls and can process those calls that are connected to customers, based on the logic that is built into VoiceXML scripts. These scripts allow GVP to manipulate user data and perform call control; for example, the call can be completed or transferred to the live agent for further processing. GVP 8.0 can also communicate with OCS by using HTTP or HTTPS (see Outbound Contact and HTTP Server), which enables automated outbound record processing.

Dialing Algorithm

This deployment supports two dialing modes: Power GVP and Progressive GVP. In this environment, calls originate on the Trunk Group DN and are communicated to the GVP platform after the established CPD is completed using the ApplyTreatment T-Library API.

Calculating the Number of Records

- For the Power GVP (VoIP) dialing mode, the number of records is calculated the same way they are calculated when using the Power GVP dialing mode with GVP 7.6.
- For the Progressive GVP dialing mode: When a dialing session is loaded for a Campaign Group object configured to use this dialing mode, OCS calculates the number of records for retrieval from the database as a percentage (specified in the Optimal Record Buffer Size dialog box) of the total number of channels that are available for this Campaign Group. If more than one Calling List is used in the Campaign, the number of records retrieved from each Calling List is determined by the specified list weight.

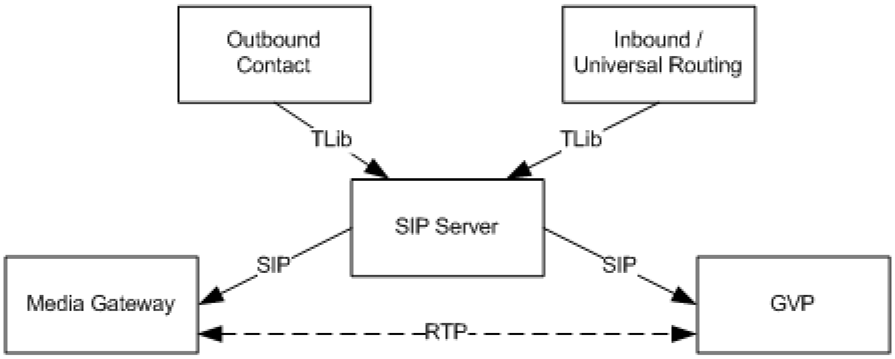
OCS then submits outbound call requests to SIP Server based on the total number of channels that are available, while trying to keep as many calls in progress as there are available channels.

OCS retrieves records from the Calling List table in the database to replenish its buffer when the number of records in the OCS buffer is less than the number calculated as a percentage (specified in Minimum Record Buffer Size dialog box) of the total number of channels that are available for this Campaign Group.

In a VoIP-ready deployment, OCS is responsible for initiation of outbound and engage calls dialing and partial control of call distribution.

Note:	For more information on dialing modes functioning in a VoIP environment, see VoIP Dialing Modes. If the Campaign Group object is not configured with a Trunk Group DN, the dialing modes function as they would in a non-VoIP environment.

The Genesys components/applications function as shown in the following figure.



VoIP Environment/Outbound Contact Architecture

- Outbound Contact components handle the following:
  - Outbound call pacing
  - Outbound call initiation toward the customer (Transfer, ASM, and Proactive Contact modes)
  - Engaging call initiation toward agents (ASM mode)
  - Outbound call distribution toward agents (Transfer and ASM modes) and toward the voice platform (Power GVP and Progressive GVP modes)
- The Universal Routing components handle routing outbound and engaging calls to the agent.
- SIP Server controls all SIP signaling for initiated calls and also acts as an interface between solution components.
- GVP Media Server handles the following:
  - Playing the basic announcement (ASM Mode)
  - Call recording
  - Call-progress detection
  - Media bridging (ASM mode)
  - Playing voice applications (Power GVP and Progressive GVP dialing modes)

In addition to Media Server, the GVP components also include Resource Manager, Fetching Module, and Squid and Reporting Server. For more information on the GVP components, see the Genesys Voice Platform 8.1 Deployment Guide.

Note:	GVP Resource Manager manages Media Server

	resources. If two or more Media Servers are configured to serve a Trunk Group, Resource Manager distributes the engaging calls and customer calls among them.

Supported Deployments

Outbound supports several types of VoIP deployments for agent connectivity including:

- IP agents located on the same SIP Server.
- IP agents who are on multiple SIP Servers.
- Both IP agents on SIP Server and TDM agents who are located on T-Server.

Resources

The resources available for each dialing session/outbound campaign are determined by the Trunk Group DN that is specified in the Advanced tab of the Campaign Group object (see Campaign Group Object).

Note:	For GVP 8.1.1, the Trunk Group DN must be named Environment working in any tenant; in GVP 8.1.2 there is no restriction for Trunk Group DN name.

On behalf of the Trunk Group DN, SIP Server provides OCS with media-resources usage (total ports and available ports). OCS initially receives this information in EventRegistered when it registers this DN with SIP Server. Once OCS registers Trunk Group DNs, SIP Server automatically provides updated resource/port information for Trunk Group DNs, which OCS uses to run dialing sessions/campaigns. In addition to this information, a dialing session limits the number of ports it can use to the number specified in the Number of Channels parameter in the Campaign Group object (Configuration tab/ Advanced section).

Dialing Modes and VoIP Deployment

An Outbound-VoIP environment supports all auto-dialing modes.

The following table provides a comparison of the modes in both a traditional and an IP environment.

Dialing Mode Comparison

Dialing Mode	Traditional	Outbound-VoIP - Ready
Predictive	Predictive	IP-predictive
Predictive GVP	Predictive	IP-predictive
Progressive	Progressive	IP-progressive
Predictive with seizing	Predictive ASM using CPD Server	IP-ASM (predictive)
Progressive with seizing	Progressive ASM using CPD Server	IP-ASM (progressive)
Power GVP	Power GVP using OBN	IP-VoiceXML (based on the calls per second setting)
Progressive GVP	Not applicable	IP-VoiceXML (based on port availability)

Configuration

Configuring Outbound Contact and SIP Server for a VoIP Deployment

Purpose

- To configure Outbound Contact and SIP Server so that a Campaign Group is *VoIP-Ready.*

Start

1. Configure a Trunk Group DN on the SIP Server switch object. For more information, see *Framework Genesys Administrator Help.*
2. Configure a Campaign Group object with the Trunk Group DN. For more information, see Campaign Group Object.
3. Configure a SIP Server application object that controls the Trunk Group DN.
4. In the OCS application object, add this SIP Server on the Configuration tab/Connections section.

End

VoIP Dialing Modes

Outbound Contact supports the following VoIP dialing modes: Predictive, Progressive, Predictive with seizing, Progressive with seizing, Power GVP, Predictive GVP, and Progressive GVP. In these VoIP dialing modes, the call control/flow is as follows:

- The call is dialed from the Trunk Group DN.
- OCS controls the call.
- In Transfer modes, OCS transfers the call to VTD.
- In ASM modes, OCS bridges the call with engaging call.
- In GVP modes, OCS also controls the Treatment VoiceXML application.

Note:	If a Campaign Group is not Outbound-VoIP-ready, these dialing modes function like the non-VoIP dialing modes (with the exception of the Progressive GVP mode which is not available in non-VoIP dialing modes). For information on how OCS works with VoiceXML applications in these modes, see VoiceXML Applications. For information on how OCS handles Apply to Call treatments in these modes, see Apply to Call Treatments. For more information on Outbound in a VoIP environment, see OutboundVoIPDeployment.

Progressive, Predictive, Predictive GVP, Progressive with Seizing, and Predictive with Seizing Dialing Modes

When a Campaign Group/environment is Outbound-VoIP-ready (see VoIP Environment Configuration), these modes function like their standard, non-VoIP counterparts, but in a VoIP environment with its associated components. See their associated descriptions.

Power GVP Dialing Mode

The Power GVP dialing mode uses SIP Server to dial outbound calls, to detect call results, and to further process successfully connected calls. This Power GVP mode dials calls so that the number of calls in progress equals the Max Queue Size setting of the Campaign Group. Like the non-VoIP usage, this dialing mode can be very effective when running "agent-less" campaigns, because it enables you to use custom-created VoiceXML scripts to automate call processing and allow for self-service of the contacted outbound customers. See Genesys Voice Platform for more information.

Progressive GVP Dialing Mode

The Progressive GVP dialing mode dials calls from a Calling List when a GVP port is available. The dialing pace is calculated based on port availability; so this mode dials calls equal to the number of Call Progress Detection ports configured for the Campaign Group. This dialing mode requires SIP Server for placing outbound calls, instead of T-Server, and uses GVP Voice XML applications for call processing.

Note:	In this dialing mode, the predictive algorithm does not calculate the dialing rate based on agent availability. Instead, it calculates the number of dial requests to maintain the total number of customer calls in any state (for example, dialing or established) that is not greater than the number of ports available for the Campaign Group, in accordance with Resources assignment (see Resources). Warning: OCS considers a port available when it is notified as such by SIP Server.

VoiceXML Applications

When running Outbound Contact using the VoIP dialing modes, OCS identifies which VoiceXML application that GVP should start by including the URI of that application in the TApplyTreatment function.

- (All call results) OCS determines the URI based on the value set in the initial-page-url option of the Voice Platform Profile configuration object and whose DBID is specified in the IVR Profile ID attribute for the Campaign Group.
- (Answering Machine call results only) If you want to use a dedicated VoiceXML application for Answering Machine call results, OCS determines the URI based on the value set in the am-initial-url option of the Voice Platform Profile configuration object and which is associated with the Campaign Group. If a URI is not specified in the am-initial-url option, OCS uses the URI set by the initial-page-url option.

Note:	The URI Answering Machine VoiceXML application is never passed in ivr-profile-id pair to SIP Server, which means that it cannot be prefetched by GVP.

In addition, for Answering Machine Detection call results, OCS can instruct GVP on how to handle beep detection. The am-beep-detection option specifies whether GVP is forced to detect an answering machine beep before starting the VoiceXML application. This information is passed on to GVP in the Extensions KV list of the same TApplyTreament function.

Note:	For information on the initial-page-url and am-initial-url options, see the *Genesys Voice Platform 8.1 User's Guide.* For information on the IVR Profile ID attribute for the Campaign Group, see IVR Profile.

Apply to Call Treatments

In a VoIP environment, after an outbound call is established on the Trunk Group DN, OCS can apply Transfer/Connect or Drop treatments to the call. These treatments—which you configure by using the Treatment configuration object with an Apply to Call action—always have a higher priority than VoiceXML application call handling.

This approach is helpful in the following scenarios:

- A Connect treatment is configured for an Answering Machine call result. A Destination DN is also configured. OCS receives EventEstablished with an AM call state. Despite the configured IVR Profile ID, OCS issues RequestSingleStepTransfer to the Destination DN and completes call/record processing.

- A Drop treatment is configured for an Answering Machine call result. OCS receives EventEstablished with an AM call state. Despite the configured IVR Profile ID, OCS issues RequestReleaseCall and completes call/record processing.

Apply to Call treatments offer an alternative mechanism for call handling. Although it is possible to perform the same call processing in the VoiceXML application (transfer or drop), treatments can be used to improve performance, so that GVP does not need to fetch (or retrieve), parse and execute the VoiceXML application in these basic scenarios.

Outbound Call Transfer by Treatment Application after Processing by the VoiceXML Script

OCS can now apply a treatment of type Transfer or Connect as a step of processing of an outbound call when running in VoIP environment in Power GVP, Progressive GVP, or Predictive GVP dialing modes, if requested by the VoiceXML script. Whenever the processing of a successful outbound call is completed by a VoiceXML application, and further processing of this call is still required, the call can be transferred by OCS to a pre-configured arbitrary destination DN. For example, an outbound call can be transferred to an ACD queue for delivery to a live agent for assisted service after the call has been self-serviced by VoiceXML script execution.

In the Predictive GVP dialing mode, to deliver a call to the Voice Transfer Destination that is configured for the Campaign Group, the VoiceXML application can also send to OCS a `ContinueProcessing` request without the GSW_TREATMENT attribute instead of a `RecordProcessed` request with the GSW_TREATMENT attribute.

The advantage of this feature is that the transfer is handled by OCS after receiving a specific message from the VoiceXML application. There is no need to implement the transfer by means of the VoiceXML application itself or by any other system component.

Provisioning

Enabling this functionality requires that the Calling List reference the Treatment configuration object. This object should have the Apply to Call action set to Transfer or Connect, a destination DN specified (for example, ACD queue), and the Number in sequence property set to 1. The Apply to Record action of the Treatment configuration object can be set to either No Treatment or Update all records in chain.

Note:	A treatment of type Transfer or Connect can be configured for certain call results only. It is supported for Answer, Answering Machine, Pager, Fax, and OK call results. Genesys recommends that you configure it to the Answer call result, since other call results might affect how OCS makes outbound call on the first stage of call processing (when requesting SIP Server to make an outbound call and when processing EventEstablished on the Trunk Group DN for a successful call results).

Processing

The following is an example of a RecordProcessed request which will be delivered to OCS by the

VoiceXML application so that OCS can attempt a Transfer treatment application:

```
POST http://host1.domain1:8080/records/15?req=RecordProcessed HTTP/1.1
Host: host1.domain1:8080
User-Agent: GVP/8.0 Banking self-service #4
Content-type: application/json
Content-length: 116
{
"GSW_CALL_RESULT":33,
"GSW_TREATMENT":"RecordTreatCampaign",
"CUSTOMER_CODE": 22,
"DATE_LAST_SERVED": "10/30/2008"
}
```

If the VoiceXML application delivers the RecordProcessed notification to OCS by using HTTP with the GSW_TREATMENT attribute that is specified and the treatment is configured as described in the example above, OCS uses a single step transfer to deliver the call to the destination DN that is specified in the Treatment configuration object.

The Apply to Record property of the treatment controls how OCS treats the chain of records after the transfer. If this property is set to Update all records in chain, OCS finalizes the chain of records in the Calling List and the transferred call processing can no longer affect the chain. If the Apply to Record property is set to No Treatment, OCS does not finalize the chain and awaits further updates for this chain through HTTP from the VoiceXML application or through Desktop protocol from the agents desktop.

Note:	If the chain of records is not finalized by this treatment and the call is passed to an agent for the further processing, OCS only processes the Desktop protocol requests from the agent if this agent belongs to a Group that is associated with the current Campaign Group and is properly logged in.

Outbound-VoIP in ASM Dialing Modes

In VoIP environments, when Campaigns are running in ASM modes, agents are engaged by the RTP stream from Media Server. Media Server bridges the engaging and outbound calls, connecting the two RTP streams.

Campaign Group Configuration

In the Outbound-VoIP ASM modes, configure a Campaign Group object the same way that you would for the standard Predictive with seizing or Progressive with seizing dialing modes, but with the addition of the Trunk Group DN configuration. For more information, see Campaign Group Configuration.

Note:	If the Trunk Group DN is not configured in the Campaign Group object, OCS processes that Campaign Group by dialing on behalf of the Voice Transfer Destination (VTD), as it did in 7.*x* releases.

Engaging Calls

In a VoIP environment, OCS requests that an engaging call be dialed. OCS sends this request to SIP Server (using T-Library) for it to make a call from the Trunk Group DN to VTD. Similarly to the standard ASM mode, the engaging call is then delivered from VTD to the agent's DN.

Note:	For engaging calls, the TMakeCall request includes the Trunk Group DN and the Voice Transfer Destination DN.

The following options affect how the engaging call is dialed:

- beep-on-merge--Controls whether a beep is played to the agent immediately before the agent is bridged to the customer call.
- call_wait_agent_connected_timeout--Specifies the timeout, in seconds, between when the engaging call is dialed and when the agent answers the call.

For a successful engaging call (that is, one that is established with an agent), when SIP Server sends EventEstablished to OCS, it includes the Media Server ID (GSW_MEDIA_SRV_ID) as a user data attribute. This enables OCS to bridge only engaging and customer calls that reside on the same Media Server.

For an unsuccessful engaging call (that is, one that is not established with an agent), note the following:

- If an engaging-call request cannot be made, SIP Server issues an EventError on the Trunk Group DN identifying the reason for the failure. Error codes include 50 (Unknown error), 53 (Invalid attribute), and 415 (Invalid destination).

In a VoIP environment, the most likely cause of the failure is the unavailability of resources required to make the call (for example, a lack of available ports on Media Server or the Media Gateway).

- If an engaging-call request is successfully placed but the engaging call is released because the agent is busy, SIP Server issues EventDestinatationBusy and EventReleased on the Trunk Group DN.
- If the engaging-call request is successfully placed but the agent does not answer within the timeout specified by the call_wait_agent_connected_timeout option, OCS releases the call.

Note:	No announcements are supported for engaging calls.

Customer Calls

For customer calls, OCS uses the TMakePredictiveCall request to initiate customer calls. This request includes the Trunk Group DN and the Voice Transfer Destination DN.

Call Progress Analysis

Call progress analysis (CPA) occurs based on the setting of the call_answer_type_recognition option, which specifies what has to be detected on the pre-connect (SIT tones) and post-connect (for either a fax or an answering machine) phases of the call, as well as answering machine detection sensitivity.

Note:	If a record is configured with a call_answer_type_recognition value (see Per-Record Basis) that is not supported by SIP Server (for example, ISDN messages that are based on answering-machine detection only), or if the record does not specify a value, OCS applies instead the no_am_detection value.

The following options affect CPA for VoIP dialing modes:

- cpd-recording, which allows you to specify whether the call progress detection phase of the call should be recorded.
- call_timeguard_timeout option, which specifies the maximum time that is allowed for CPA after the call is connected.
- cpd-on-connect option. which specifies that CPA begins after the call is connected. Set this option to false or no.

Notes:	• Setting this cpd-on-connect option to false or no enables the use of Color Ring Back Tones. If cpd-on-connect is not configured or its value is set to false or no, CPA starts as soon as the media stream is available.

	• A *Color Ring Back Tone* (CRBT) refers to the ability to play other audio sounds (music, voice, and so on) instead of a standard ring tone while a call is connecting.

OCS receives the results of CPA (performed by the Media Gateway or the GVP Media Server) in the CallState attribute of EventEstablished (the call has been connected), EventDestinationBusy (the call has been rejected) or EventReleased (call was not answered or it was released by SIP Server) and assigns a call result.

Note:	When CPD Proxy is used in front of multiple CPD Servers, there is a probability of abandoned calls in both Progressive ASM and Predictive ASM modes.

For information on attributes that are specific to VoIP environment, see the *Outbound Contact Reference Manual.*

Successful Call Flows

For a successful customer call, SIP Server passes the Media Server identifier to OCS.

Unsuccessful Call Flows

For an unsuccessful customer call, note the following:

- If a customer call (RequestMakePredictiveCall) cannot be placed, SIP Server issues an EventError on the Trunk Group DN, providing the reason for the failure in the ErrorCode and the ErrorMessage attributes. Error codes include 50 (Unknown error), 53 (Invalid attribute), and 415 (Invalid destination).

In a VoIP environment, the most likely cause for the failure is the unavailability of resources required to make a call (for example, a lack of available ports on Media Server or the Media Gateway).

- If the call is initiated but is released because of a negative call-progress result (usually due to the detection of a SIT tone or a busy response), SIP Server issues EventDestinationBusy and EventReleased on the Trunk Group DN with the associated CallState attribute.
- If the call is initiated successfully, but the call is released because it was not answered or is answered by a fax or answering machine, according to the SIP Server configuration, SIP Server issues EventReleased on the Trunk Group DN with the associated CallState attribute.

Merging Engaging and Customer Calls

For all dialing sessions, OCS maintains information about the distribution of engaging and customer calls among Media Servers. There are two methods — bridging and transfer — for merging an engaging call with a customer call when these calls might be on different Media Servers.

You specify the merge method (bridging or transfer) using the merge-method option at the Campaign Group Level or the OCS Application Level.

Note:	In a multi-site environment, to maintain the correct Events and ensure user data propagation when merging these calls, configure the SIP Server **use-data-from** option with a value of `current`. For more information about this option, see the *Framework SIP Server Deployment Guide*.

Important

Call Supervision functionality for engaging calls in ASM dialing modes in VoIP environments is not supported. If OCS receives an `EventEstablished` on the supervisor's DN after receiving `EventEstablished` on the agent's DN, the engaging call leg will not be merged with an outbound call.

Bridging Method

OCS controls when and how to bridge the engaging and customer calls. As the default method, when a customer call is connected and call-progress analysis is completed, OCS attempts to find an established engaging call on the same Media Server as the customer call and responds as follows:

- If OCS finds an established engaging call, it bridges the two calls.
- If OCS does not find an established engaging call (for example, in the Predictive mode) on the same Media Server, OCS releases the customer call, with the option to play a configured announcement before releasing it, as configured in either the asm_drop_announcement_data option or the asm_drop_am_announcement_data option.

Tip

While the **asm_drop_announcement_data** option allows you to play the same announcement for all call results, the **asm_drop_am_announcement_data** option allows you to play a different announcement for Answering Machine call results than the one used for Answer call results.

For SIP Server, if the **asm_drop_announcement_data** option and/or the **asm_drop_am_announcement_data** is configured, OCS releases the call only after the announcement is played, which occurs after OCS receives EventTreatmentEnd from SIP Server.

You can configure OCS to handle a scenario in which it cannot find an established engaging call on the same Media Server by setting the value of the on-bridging-unable option to `transfer`. By doing so, OCS connects the customer call to the first found established engaging call, regardless of the Media Server on which the engaging call is established. You configure this option at the Campaign Group Level OCS Application Level.

Tip

> If the value of the merge-method option is set to `bridging`, then the on-bridging-unable option is taken into account in scenarios where bridging is not possible; if the value of the **on-bridging-unable** option is set to `transfer`, then OCS attempts to use the Transfer method; if the value is set to `drop`, then the call is dropped.

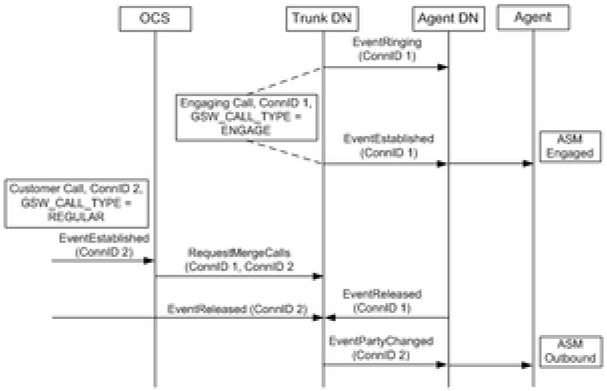
> ### Tip
>
> When the bridging method is used, no SIP messaging is provided to OCS if the agent's SIP endpoint fails and the calls are not merged. As a result, OCS is unable to identify the call with a Transfer error.

Transfer Method

For the transfer method, OCS merges each established outbound call with the longest-engaged agent across all Media Servers, as follows:

- If the first attempt to merge an outbound call with the engaging call is failed, OCS keeps considering that agent as the longest-waiting agent until the second attempt to merge the next outbound call with this engaging call.
- If the second attempt is failed, OCS sends a request to release the engaging call and stops considering that agent as the longest-waiting agent.
- If an engaging call is released, that agent becomes the shortest-waiting agent when a new engaging call is established on the agent DN.

The following figure illustrates the event flow for the merging of the call legs.


OCS
Trunk DN
Agent DN
Agent
EventRinging (ConnID 1)
Engaging Call, ConnID 1, GSW_CALL_TYPE = ENGAGE
EventEstablished (ConnID 1)
ASM Engaged
Customer Call, ConnID 2, GSW_CALL_TYPE = REGULAR
EventEstablished (ConnID 2)
RequestMergeCalls (ConnID 1, ConnID 2
EventReleased (ConnID 1)
EventReleased (ConnID 2)
EventPartyChanged (ConnID 2)
ASM Outbound

Event Flow--ASM Engaged and Outbound Call

Transferring a Customer Call

If OCS determines that the customer call must be transferred to another destination (for example, a treatment with a transfer call action), OCS uses a single-step transfer. OCS considers the transfer complete when the customer call is transferred from the Trunk Group DN to the VTD. Then, OCS finalizes the chain processing with a call result.

Using the Opt-Out Feature With SIP Server in VoIP Environments

The OCS opt-out feature enables the call recipient to opt-out from any further outbound calls in ASM mode. This feature addresses legislative requirements and enables call recipients to opt-out by pressing certain buttons on the touch tone phone if there are no agents available to speak to them. A typical supported scenario is described in the section, Opting Out of Outbound Calls in ASM Mode.

When SIP Server places outbound calls in VoIP environments and there are no available agents to handle the answered outbound call, the post-connect outbound call processing is handled by Genesys Voice Platform (GVP). GVP runs VoiceXML applications that can play announcements, collect DTMF (or speech) input, compare the input that is provided by the call recipient with the pre-configured opt-out pattern, and deliver the DoNotCall request to OCS by using HTTP or HTTPS. No pre-defined VoiceXML application exists that can handle all of these tasks. Therefore, you must create your own ad-hoc VoiceXML application.

OCS activates the VoiceXML application on GVP in this type of ASM mode when it cannot merge an outbound call with an engaged call, and only if all of the following are true:

- The Campaign Group object in the configuration has an IVR Profile configured that defines which VoiceXML application to activate.
- The OCS digits-detection-pattern configuration option is set to a non-empty string.
- The OCS digits-detection configured option is set to one of the following values:
  - all
  - answer and the detected call result is Answer
  - am and the detected call result is Answering Machine

When OCS activates a VoiceXML application for the record being called, it makes available to the VoiceXML application that is specified in the SIP Headers, the values of the following options:

- digits-detection-timeout
- digits-detection-pattern
- digits-reaction

This process enables the VoiceXML application to build its own logic to process the opt-out request and adapt to the configuration changes in OCS. However, OCS must receive only correctly-formed DoNotCall requests from the VoiceXML application by using HTTP or HTTPS, so that the specific call recipient's phone number can be marked as DoNotCall. All of the underlying logic, such as playing the message, receiving, and processing the called-party input (in the form of DTMF and/or speech) contained in the VoiceXML application.

For more information about how OCS works with GVP and VoiceXML applications in VoIP Environment, see VoiceXML Applications. For information about how to create VoiceXML applications, see the *Genesys Voice Platform 8.1 Deployment Guide*.

Overdialed calls in Progressive ASM or Predictive ASM modes

If you are running campaigns in Progressive ASM or Predictive ASM mode, OCS can optionally transfer overdialed calls (outbound calls that cannot be bridged immediately to an established engaging call) to a Voice Transfer Destination DN (typically a Routing Point) rather than terminate the calls. Those calls can then be handled by the routing strategy and delivered to queues with available agents or to IVR ports.

After an overdialed call is transferred by OCS, an agent or IVR application can still finalize the call within OCS and update the record in a chain.

> Tip
>
> The distribution of overdialed calls to queues being monitored by OCS or to any of the agents working on active campaigns within OCS can have a negative impact on call pacing. Keep this in mind when planning your overdialed call distribution strategy.

> Warning
>
> You must ensure legislative compliance if using this type of transfer. If the router is unable to locate an available agent, it can take longer than two seconds to transfer the call to a new queue.

SIP Server and High Availability Limitation

When you are running a dialing session in an ASM mode for an outbound IP campaign and SIP Server is part of a Network Load Balancing cluster that is operating in hot-standby mode, use the Switchover command instead of the Stop command in Solution Control Interface (SCI) to switch from the primary to the backup SIP Server. If you use the Stop command, the primary SIP Server will generate EventLinkDisconnect and of all its associated clients will re-register all DNs on the backup SIP Server. If the primary SIP Server had established calls for Outbound VoIP agents and the agents received an Engaged status, the re-registration process will change the agent status to Busy Unknown or Ready, depending on whether the agent was engaged before starting the backup SIP Server.

Call Supervision Limitation

Call Supervision functionality for engaging calls in ASM dialing modes in VoIP environments is not supported. If OCS receives an EventEstablished on the supervisor's DN after receiving EventEstablished on the agent's DN, the engaging call leg will not be merged with an outbound call.

Outbound-VoIP in Transfer Modes

For Outbound-VoIP functioning in Transfer modes (Predictive or Progressive), the Campaign Group also needs to be Outbound-VoIP-ready (see Dialing Modes and VoIP Deployment).

In these dialing modes, OCS dials outbound calls using SIP Server on behalf of the Trunk Group DN and controls the transfer of successful outbound calls (that is, those calls answered by a person) to VTD for further delivery to agents.

Customer Calls

In an Outbound-VoIP Transfer mode, the dialing and transfer of a customer call are handled in the same way as Outbound VoIP ASM mode for the following:

- Call progress analysis
- Unsuccessful call flow handling (see Unsuccessful Call Flows)
- Call transfers (Transferring a Customer Call)

For transferring to a Voice Transfer Destination DN, see the next section.

Transferring to a Voice Transfer Destination DN

You can use either a single-step or a two-step transfer. Upon a positive call-progress detection, OCS transfers an outbound call to a Voice Transfer Destination (VTD) DN according to the setting of the call_transfer_type option. You define the transfer process by setting the option to one_step for a single-step transfer and to two_step for a two-step transfer.

Single-Step Transfer

A single-step transfer is quicker than a two-step transfer, but it does not give the user control over how the transfer occurs. When the call_transfer_type option is set to one_step, OCS looks to the call_wait_in_queue_timeout option, which determines how long the call should wait (in the queue) before an agent answers. If the timeout expires, OCS aborts the transfer by dropping the call, and the record is marked with the Dropped call result.

Note:	Some switches do not support this type of transfer.

Two-Step Transfer

A two-step transfer is slower than a single-step transfer, but it creates a consult call that allows the user more control over the transfer (for example, to cancel it). This type of transfer depends on the following:

- The SIP Server consult-user-data option must be set to inherited. For more information on the SIP Server consult-user-data option, see the *Framework SIP Server Deployment Guide.*

- The OCS call_wait_in_queue_timeout option must be specified.

Notes:	• For the consult call, if the timeout expires, OCS aborts the transfer, tries to reconnect to the call and releases the call if unsuccessful. OCS also assigns the Transfer Error call result. • If the timeout expires while OCS is waiting for the call to be established at the Trunk Group DN, OCS aborts the transfer, releases the call, and assigns the Dropped call result.

OCS applies a treatment to the chain-call result, if the treatment is configured. It does not attempt to play any voice file for an outbound call before it is released.

Outbound calls that do not reach a destination (for example, an agent DN or an IVR port) are considered abandoned and reported as overdialed to the predictive algorithm.

Assured Connection

Note:	Assured connection is available in the Predictive and Progressive dialing modes when dialing with SIP Server in a VoIP environment.

Sometimes, it may be beneficial to dial an outbound call in an automated dialing mode that is guaranteed to be connected to an agent. For example, this may be desirable for outbound customers of high importance, or for calling some numbers as required by government regulations.

Neither the Predictive or Progressive dialing mode can typically guarantee connection of the call to an agent. In Predictive dialing mode, the number of calls initiated by OCS is always larger than the number of agents available to handle those calls once they are answered. Therefore, it is probable that some calls may be abandoned due to a long waiting time in the queue.In the Progressive dialing mode, an agent targeted for an outbound call may become unavailable or receive an inbound call. Normally, these are acceptable risks, and in some countries, government regulations set an allowable level of abandoned calls.

For Assured Connection functionality, when using Progressive mode and skill-based routing functionality, OCS provides the ability to engage the most suitable agent to handle a conversation. Agent selection criteria (for example, skill requirements) are defined by the data in the mandatory and user-defined fields of a calling list record. OCS attaches all user data that are related to the record being processed to the engaging call that was dialed for this record. All the data that are associated with the customer contact are made available to URS in the attached data of the engaging call. OCS attaches all pairs that are attached to the outbound call that was dialed for the record, except for GSW_RECORD_HANDLE pair. The following types of pairs are attached:

- Mandatory pairs, such as GSW_PHONE and GSW_CHAIN_ID
- User-defined pairs, as defined by the `send_attribute` option configuration of the mandatory and user-defined fields
- Auxiliary pairs, such as GSW_APPLICATION_ID and GSW_SWITCH_DBID
- Pairs as defined by using Script configuration object(s) that are associated with Calling List, Campaign, and Campaign Group objects to which the outbound record belongs

This feature enables the selection of the most suitable agent to handle the assured connection conversation. It also enables you to meet specific business requirements, for example, to avoid call transfers in the following scenario: callback requests are added to a single calling list which is used to guide a Progressive outbound campaign and an agent must be available when a contact call is established and that the agent is able to communicate with the contact in the language that is specified by the contact, as well as being able to service the specific contact request.

Starting in release 8.1.1, OCS can guarantee connection of pre-selected calls to agents in the Predictive or Progressive dialing mode when running campaigns in a VoIP environment. To ensure connection to an agent, OCS verifies each call before sending it to be dialed, to determine if this call should be dialed with a guaranteed connection. If so, OCS first dials an engaging call and waits until it establishes on an agent. After the agent is engaged, which guarantees that the agent will not log out, receive another call, or be reassigned to another activity, the outbound call is dialed. After the outbound call is answered by the customer, it is immediately merged with the engaging call. If the outbound call does not reach the customer, the engaging call is released and the agent is returned back to handle regular campaign calls.

There are two ways to define if the outbound call for a specific record needs to be dialed in assured connection mode:

By selecting a field in the calling list and a set of values for this field using the assured-connect-field option that indicates that the record should be dialed in assured connection mode.

When it retrieves the record, OCS reads the specified field and checks its value. If the value falls within the pre-configured set of values for the field, OCS marks the record for assured connection.

By using SCXML treatments, with the set_flex_attr custom action, by setting the assured-connect option to true for a given record or chain of records. This setting has a higher priority than a definition using the assured-connect-field option.

The following configuration options are used to implement assured connection:

- assured-connect---Enables or disables the assured connection feature.
- assured-connect-field--Specifies the field name and set of values which nominate the record for assured connection.

Assured Connections and Predictive Algorithm

This section describes how the Predictive Algorithm (PA) handles assured calls. The port usage for each assured call is as follows:

- One port is used for an engaging call from the request to dial the engaging call, until this call is merged with an outbound call or released.
- One port is used for an outbound call from the request to dial the outbound call, until this call is merged with an engaging call or released.
- Two ports are used for an established call if the merging method is bridging.
- No ports are used for an established call if the merging method is transfer.

The PA takes into account the extended port usage related to assured calls for the following operations:

- To restrict the outbound dialing rate so that the total number of busy ports does not exceed the total number of ports available for Campaign Group. This latter value is specified by the Number of CPD ports for the Campaign Group or by the dynamic allocation of ports.
- To perform the dynamic allocation of ports between running Campaign Groups.
- To perform the dynamic reassignment of agents in accordance with the effective size of the Agent Pool.

The PA cleans up an assured call when the pa-dial-expire option has expired since the most recent dialing of engaging or outbound call associated with the assured call.

For the calculations of Agent Busy Factor and Average Waiting Time, the engaged agents without outbound calls are considered Ready. The PA adds all of the engaged time intervals (from when an engaging call is connected to an agent to when that call is merged with an outbound call or is released) to waiting times.

For the Predictive dialing mode, the PA adjusts the outbound dialing rate to achieve the optimization goal, taking into account that the engaged agents can be connected only to individual assured outbound calls. This increases the actual traffic load and decreases the useful efficiency of agents, especially if many assured calls are not answered by live persons. When the optimization goal is not achieved, the PA causes OCS to generate a self-diagnostic log event (12-50131) for call-related factors, as follows:

- "Assured Calls" when there are assured calls.
- "Assured Dialing Influence" when the Optimization Goal can not be achieved due to the presence of assured calls.

Notes:	• A standard practice is to have just a few assured outbound calls during the lifetime of a campaign. • A considerable percentage of assured calls may significantly affect dialing performance of the campaign and produce a high Abandon Rate as well as a low Busy Factor.

Assured Connection vs. ASM (Engaging) Dialing Mode

Assured connections and ASM (engaging) dialing mode differ in the following ways.

In ASM dialing mode, engaging and regular call dialing are unrelated processes, especially in predictive ASM. You cannot precisely define which regular call will be bridged with what engaging call. Also, there is no guarantee that there will be an engaging call with which to complete the bridge transfer.

In ASM dialing mode, OCS does not release an engaging call that is not bridged while the Campaign Group is running, regardless of the result of regular call dialing. In the *assured-connect* dialing sequence, an engaging call is released if the regular call dialing has a negative result.

The purpose of an assured connection is to guarantee the presence of an agent for a specific call in predictive dialing, where the agent's presence is not guaranteed by definition. The assured connection can be used for only a limited number of contacts in the Campaign; otherwise the quality of predictive dialing is compromised.

ASM is supported in IP and TDM environments (with restrictions for certain line protocols and Dialogic board types).

Assured connections are supported in Outbound IP environments (with SIP Server and Media Server) only.

OCS Load Balancing in SIP Cluster

Starting with release 8.1.522.05, when operating in SIP Cluster deployments with multiple SIP Cluster nodes, OCS supports load distribution in the following dialing modes: Progressive, Predictive, Progressive with Seizing, Predictive with Seizing, Progressive GVP, Predictive GVP, and Power GVP. OCS distributes the load by issuing requests to dial outbound and engaging calls to multiple SIP Cluster nodes. OCS uses one of the SIP Cluster nodes (called the *prime* node) to subscribe and receive all DN-related events.

See the Configuring Outbound Contact topic about configuring OCS in the SIP Cluster environment.

Regular (non-ASM) Dialing Modes (Progressive, Predictive, Progressive GVP, Predictive GVP, Power GVP)

When load balancing is enabled, in the regular (non-ASM) dialing modes, OCS evenly distributes TMakePredictiveCall requests for outbound calls between all SIP Cluster nodes in round-robin fashion. OCS maintains a list of active (connected) primary/backup SIP Cluster nodes and sends each new TMakePredictiveCall request to the subsequent active node. OCS checks the state of an inactive node on the next round and attempts to send a TMakePredictiveCall request to it when it becomes active.

OCS receives all DN-level T-Events from the prime SIP Cluster node.

OCS attempts to reconnect to a backup SIP Server Application if a connection with the primary SIP Server is lost. OCS considers a SIP Cluster node inactive if connections to both the primary and backup SIP Server Applications are lost. OCS stops dialing if a connection is lost to the prime SIP Cluster node (both the primary and backup SIP Server Applications are disconnected).

ASM Dialing Modes (Progressive with Seizing, Predictive with Seizing)

In ASM dialing modes, OCS balances the load per Campaign Group, where each Campaign Group can use a different SIP Cluster node for dialing. OCS always uses the same SIP Cluster node to dial outbound and engaging calls for the same Campaign Group. You designate a SIP Cluster node to be used for dialing for a Campaign Group in the Agent Group level option **sip-cluster-asm-dial-node**. OCS sends TMakePredictiveCall (for outbound calls) and TMakeCall (for engaging calls) requests to that SIP Cluster node. If this option is not configured or a SIP Server Application with the configured DBID is not present in the OCS Application connections, OCS uses the prime SIP Cluster node for dialing for that Campaign Group.

OCS attempts to reconnect to a backup SIP Server Application if a connection with the primary SIP Server is lost. OCS considers a SIP Cluster node inactive if connections to both the primary and backup SIP Server Applications are lost. OCS stops dialing for a Campaign Group if a connection is lost to the prime SIP Cluster node or to the *dialer* node that is used for the Campaign Group.

Enabling Load Balancing Mode

To enable load balancing mode:

1. Add primary SIP Server Applications of the SIP Cluster nodes (all that will be included in load balancing) to the **Connections** tab of the primary OCS Application object. OCS can be connected to T-Controller ports (**TCport**) or Smart Proxy ports (**SmartProxy**) of SIP Cluster nodes.
2. Designate one SIP Custer node as the prime node to distribute DN-related events to OCS.
3. (For ASM modes only) Designate a SIP Cluster node for each Campaign Group to use as a dialer node. Genesys recommends distributing Campaign Groups equally between SIP Cluster nodes, so that the final load on SIP Cluster is balanced.
4. Configure the following configuration options:
   - sip-cluster-lb-mode—Set this option to `round-robin`.
   - sip-cluster-prime-node—Set this option to the valid DBID of the primary SIP Server Application object of the SIP Cluster node that is designated as the prime one for outbound dialing.
   - sip-cluster-asm-dial-node—(For ASM modes only) Set this option to the valid DBID of the primary SIP Server Application object of the designated dialer SIP Cluster node at the Agent Group object that is associated with the Campaign Group. This option must be set for every Agent Group associated with Campaign Groups which will run in ASM modes.
5. Restart OCS for the configuration changes to take effect.

Deployment Planning

These topics provide an overview of the deployment process for Outbound Contact, and includes tools for deployment planning:

- About the Deployment Process
- How to Use The Deployment Planning Topics
- System Requirements
- Worksheet Definitions
- Outbound Deployment Planning Worksheet
- Other Resources

Note:	The Outbound Contact installation uses information that was set up during Framework configuration and installation. You can install Outbound Contact only after Framework is installed. For more information, see Outbound Object-Creation Sequence.

About the Deployment Process

The deployment process involves the configuration and installation of the applications and components that are needed for a functional setup of Outbound Contact.

Before deploying Outbound Contact, review the System Requirements, as well as the following documents that contain information about licensing requirements, supported platforms, and sizing specifications:

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

In addition, use the Worksheet Definitions to gather information for the configuration and installation of Outbound Contact.

- The Outbound Contact deployment process in a traditional voice environment involves the following components:
  - Outbound Contact Server (OCS)
  - Genesys Administrator
  - CPD (Call Progress Detection) Server (optional)
  - CPD Proxy Server (optional)

Note:	Outbound Contact 8.1 supports Outbound Contact Manager 7.6.

- The Outbound Contact deployment process in an IP environment involves the following components:
  - Outbound Contact Server (OCS)
  - Genesys SIP Server
  - Genesys Administrator
- *Either* GVP Media Server *or* CPD (Call Progress Detection) Server (optional)
  - CPD Proxy Server (optional)
- In a blended environment, with both IP and traditional telephony, all the components listed above are required.

During installation, test each component before you continue to the next.

Make sure that you install all Framework components, including Configuration Server, Configuration Manager, T-Server, the Management Layer, DB Server, and Stat Server. In addition, install Genesys Administrator, which can be used instead of Configuration Manager to handle all configuration requirements.

For information about configuring and installing Framework products, see the Framework documentation.

If necessary, install Dialogic boards and software for call-progress detection. For more information about installing Dialogic boards, see Introduction to Dialogic Boards.

For additional information about Formats, Calling Lists, Campaigns, Campaign Groups, and how to start or stop a Dialing Session, see Framework Genesys Administrator Help.

How to Use The Deployment Planning Topics

Before beginning Outbound Contact configuration and installation, review the System Requirements, and then complete the Worksheet Definitions.

The worksheet provides space for your notes, and multiple boxes (cells) provide room to write additional values, if needed. If you need to provide additional information, you can make copies of the worksheet.

The structure of these topics is based on the configuration sequence found in the Outbound Configuration Wizard. Outbound properties for Genesys Administrator are also included.

After completing the installation, keep the worksheet as a reference. In the future, if you need help from Genesys Technical Support, you can fax the worksheet to a support engineer in order to provide the complete system configuration.

System Requirements

This section describes some of the system sizing requirements for the installation of Outbound Contact.

In addition, the following documents on the Technical Support website provide detailed information about the operating systems and databases supported by Outbound Contact:

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

Environmental Sizing

The information provided in this subsection helps you create the most effective environment for running Outbound Contact in your contact center.

The following table shows the recommended sizing requirements for various contact centers.

Outbound Contact Environmental Sizing for Components

Outbound Contact Component	CPU	Memory	Network	Transactions per Second
Outbound Contact Manager	Pentium III, 400 MHz (recommended)	128 MB (Windows 2000 or Windows XP)	10 mbps (minimum)	Not applicable
Outbound Contact Server	Pentium III, 400 MHz (recommended)	512 MB (minimum)	10 mbps (minimum)	0.5—1 transactions per agent per second
CPD Server Supported interfaces: • LSI • Line-side (E1/ T1) • PRI • HMP The CPD Server and the Dialogic cards and drivers it controls must be installed on the same computer.IP version 6 addressing schema (IPv6) is not supported by Dialogic	Single or dual processor, 1GHz	512 MB (minimum)	10 mbps (minimum)	0.5—1 transactions per agent per second

HMP.				
Environmental recommendation for 240 Ports	Dual processor Pentium III, 600 MHz	256 MB	100 mbps	0.5—1 transactions per agent per second
Number of Agents				
25—100	Pentium III, 400 MHz	128 MB	10 mbps (minimum)	0.5—1 transactions per agent per second
10—200	Pentium III, 400 MHz	256 MB	10 mbps (minimum)	0.5—1 transactions per agent per second
20—600	Pentium III, 400 MHz (minimum) + 100 MHz for every 200 agents	256 MB	10 mbps (minimum)	0.5—1 transactions per agent per second
60—1000	Single/dual processor,1 GHz	512 MB	100 mbps	0.5—1 transactions per agent per second
100—2000	Dual processor, 1GHz	512 MB (minimum)	100 mbps	0.5—1 transactions per agent per second
200—4000	Dual processor, 1.5 GHz	512 MB (minimum)	100 mbps	0.5—1 transactions per agent per second

Note:	The list of Dialogic boards might be extended based on their support by the CPD Server. Refer to the *Genesys Supported Media Interfaces Guide* for the current list of supported boards.

Database Sizing

Outbound Contact Server supports IBM's DB2 database, in addition to the previously supported database management systems (Oracle, Microsoft SQL Server, Informix, and Sybase).

The sizing of your database can affect the performance of the Outbound solution. The term sizing pertains to how the database is tuned, the available memory, and the number of processors. The term tuning pertains to the indexing of the calling list. The higher the number of transactions that the database processes, the more the database performance is affected, unless it is properly sized.

When sizing your database, consider the factors shown in the following table.

Database Sizing

Format	Estimated Size for Each

Record	1 KB
Transaction	1 KB
Database	100 MB per 100,000 records per campaign

If the size of one record is equal to the size of one transaction (1 KB) and if the size of the calling list is 100,000 records, the estimated size of the database is 100 MB. See the *Outbound Contact Reference Manual* for more information about database tuning.

Network Traffic

The following table provides basic data about network traffic produced by Outbound specific activities among various Framework and Outbound components. This information can help you determine the optimal component location on the network.

Network Traffic Among Framework and Outbound Components

Primary Data Types	Average Message Length	Messages Per Transaction	Elements Determining Total Message	Traffic Total Traffic Volume
OCS <-> DB Server				
Record from Calling List	0.25 KB + total length of Field Names + average Record Data length	1	Number of Records requested per Select = <Number of Agents in Campaign Group> * <Optimal Buffer Size Coefficient> - <Minimum Buffer Size Coefficient>	<Number of Records from all Calling Lists in Campaign> * <Average Number of Attempts per Record>
Update of Record Attributes	0.5 KB	1 if OCS is not configured to save intermediate results, otherwise, 1 per treatment	Hit Ratio, Treatments	<Number of Records from all Calling Lists in Campaign> * <Average Number of Attempts per Record>
OCS <-> CPD Server				
Dialing Request	0.25 KB + User Data	1 per Dial Attempt	Hit Ratio, Treatments	<Number of Records from all Calling Lists in Campaign> * <Average Number of Attempts per Record>
Agent Seize Request (ASM mode only)	0.5 KB	1 per AgentReady TEvent	Conversation Duration	Total Number of successful Contacts

Call Progress Result	0.25 KB	1 per Dial Attempt	Hit Ratio, Treatments	<Number of Records from all Calling Lists in Campaign> * <Average Number of Attempts per Record>
OCS <-> Agent Desktop				
Record	0.25 KB + User Data	1 per Answered Call or Preview Record Request	Number of Records processing Desktop sessions	Total Number of successful Contacts, in `Predictive` or `Progressive` mode, <Number Records from all Calling Lists in Campaign> * <Average Number of Attempts per Record>, in `Preview` mode
Desktop Protocol Requests	0.25 KB	Typically 2	Number of Records processing Desktop sessions	Total Number of successful Contacts in `Predictive` or `Progressive` mode, <Number Records from all Calling Lists in Campaign> * <Average Number of Attempts per Record>, in `Preview` mode
OCS <-> T-Server				
Requests (no CPD Server)	0.25 KB + User Data	1 per Dial Attempt	Hit Ratio, Treatments	<Number of Records from all Calling Lists in Campaign> * >Average Number of Attempts per Record>
TEvents	0.25 KB + User Data	5 for simple call-distribution scenario	Hit Ratio, Treatments	<Number of Records from all Calling Lists in Campaign> * <Average Number of Attempts per Record>

CPD <->ET-Server				
Dialing Request	0.25 KB + User Data	1 per Dial Attempt	Hit Ratio, Treatments	<Number of Records from all Calling Lists in Campaign> * <Average Number of Attempts per Record>
Transfer Request	0.25 KB	1 per Dial Attempt	Hit Ratio, Treatments	<Number of Records from all Calling Lists in Campaign> * <Average Number of Attempts per Record>
Agent Seize Request (ASM mode only)	0.5 KB + User Data	1 per AgentReady TEvent	Conversation Duration	
TEvents	0.25 KB + User Data	10 for simple call-distribution scenario, 1 for unsuccessful dial attempt	Hit Ratio, Treatments	<Number of Records from all Calling Lists in Campaign> * <Average Number of Attempts per Record>

Worksheet Definitions

For more information about the following items, see "Other Resources".

Logging In to Genesys Administrator

- User name--Enter the user ID authorized to log in to Configuration Server.
- User password--Enter the password for the user ID authorized to log in to Configuration Server.
- Application--Enter the name of the Genesys Administrator `Application` object.
- Host--Enter the host name of the computer on which the Configuration Server is installed.
- Port--Enter the port that the Configuration Server uses for client connections.

License Manager

- Path to License Manager--Enter the path to the License Manager. License Manager must be available for licensed application.

Outbound Solution Object

- Name--Enter a name for the `Outbound solution` object.
- Solution Control Server--Enter an existing Solution Control Server.

Tenant

- Name(s)--If this a multi-tenant environment, enter the names of the tenant to be used for Outbound Contact.

Switches

As part of the Framework configuration and installation, you will have already assigned agent names, agent logins, and DNs (see DNs in the Planning Worksheet) to switches.

In most systems, the telephone directly notifies the switch when the telephone set (*teleset*) status changes--for example, when agents begin or end a call. By default, Outbound Contact registers teleset states internally. Older PBX software does not enable the teleset to notify the switch directly when its status changes. If the contact center's PBX software does not enable direct notification, you must customize the contact center's teleset state disposition. Use configuration settings that instruct Outbound Contact to send Framework's T-Server a notice for each teleset status change. For more information, see Outbound Contact Configuration Options.

The following refer to the capabilities of a switch:

- Name(s)--A switch set up during Framework configuration. Other switches can include numbers and types of all telephony devices (DNs) to be monitored and controlled through the CTI (computer-telephony integration) link. Enter the switch name(s) to be used for Outbound Contact. If new switches are needed, they must be added through Framework.
- CPD capability--Does the switch have call-progress detection (CPD) capabilities? If the switch does not provide call-progress detection, and you wish to use CPD Server as a part of Outbound Contact for the contact center, collect the following information about the Dialogic boards:
    - Call transferring--Does the switch have this capability?
    - Call conferencing--Does the switch have this capability?
    - Call forwarding--Does the switch have this capability?
    - Do Not Disturb--Does the switch have this capability?

Special DNs

Communication DN

Outbound Contact reports on outbound activity by using a DN that is set up as a Communication DN.

- Number--Enter the name or number of the Communication DN.
- Type--Always enter "Communication DN".

Overflow DN

Outbound Contact monitors calls that cannot be handled by available agents. It does this through a DN set up as an Overflow DN.

- Existing DN name--Enter the name or number of the overflow DN.

Applications

Each application name must be unique, and it can contain alphanumeric characters and underscores. the properties for most applications are the same; therefore, definitions are given only once for the "Applications" section of the worksheet.

- T-server name(s)--Enter the name of the T-Server.
- Stat Server name(s)--Enter the name of the Stat Server.
- DB Server name(s)--Enter the name of the DB Server.
- OCS name--Enter the name of the Outbound Contact Server.
- CPD Server name(s)--Enter the name of the CPD Server.
- Genesys Administrator name--Enter the name of Genesys Administrator.
- Reporting--See the Reporting documentation.

Tenants

- Tenant name(s)--If this is a multi-tenant environment, select the name of the tenant.

Server Info

- Host--Select the host computer on which your application will be running.
- Communication port--Enter the port number that the application's clients will use to establish connections to the named server.
- Back up Server--Select the back up server to be contacted if connection to the primary server fails.

Options

- Option names--Options are changed according to the user's needs. For detailed information about options, see Outbound Contact Configuration Options.

Database Access Point(s)

- Name--Enter the name of the `Database Access Point (DAP)` object.

Server Info

- DB Server name--Select the name of an existing DB Server.
- DB Server host--Select the host computer on which the DB Server will be running.

Tenants

- Tenant name(s)--If this is a multi-tenant environment, select the name of the tenant.

DB Info

- DBMS name--Select the name of the host computer on which the database resides.
- DBMS type--Select the type of database.
- Database name--Enter the name of the database.
- User name for login to DB--Enter the user name for accessing the database.
- Password for login to DB--Enter the password for access the database.

User-Defined Fields

A calling list must contain Genesys mandatory fields, and it can also contain user-defined fields. For more information about mandatory fields, see Field Object and Format Object. The user must create additional fields, containing business information that is used for sorting and chaining records--for example, `Last_Name`.

- Name--Enter the name of the user-defined field--for example, `cust_id`.
- Data type--Enter the type of data, for example, the data type for `cust_id` would be `int`, and must always be numeric.
- Length--Enter the number of characters needed for field contents--for example, `cust_id` can be limited to five numeric characters.
- Default value (optional)--Enter the value that will be assigned to the field unless it is changed.

Formats

A *format* contains a collection of Genesys mandatory fields and user-defined fields. It defines the layout of a calling list. More than one format can be created and named for the specific needs of particular calling lists.

Identify the number of formats that you need, assign names to them, and identify the user-defined fields required for each.

Note:	Plan and create your formats and all custom fields carefully. You cannot modify existing fields or add new fields after the calling list table is created in Genesys Administrator.

- Name--Enter the name of format.
- User-defined field names--Enter the names of the user-defined fields to each format.

Table Access

- Name--Select the name of the `Table Access` object.
- Table type--Select "Calling List."
- DB access point name--Select the name of the database access point through which the table can be accessed.
- Format name--Enter the name of the format applied to this table.
- Database table name--Enter the name as specified in the database.

Dialing Filters

Dialing filters select records from a calling list for dialing, based on specified criteria. They can be created in Configuration Manager or in Genesys Administrator and stored in the `Filters` view in `Genesys Administrator > Provisioning tab > Outbound Contact`. For more information, see Filter Object.

- Name--Enter the name of the dialing filter.
- Criteria SQL statement--Enter the SQL statements that give the criteria for the filter.
- Order by SQL statement--Enter the field on which the list will be sorted.

Treatments

Decide on treatments to use for unsuccessful call results. For more details and examples, see Call Handling Treatments.

- Name--Enter the name of the treatment.
- Call result--Enter the result of dialing a calling list record.
- Apply to record--Enter the action to be applied to a calling list record, based on the call result.
- Apply to call (very limited use)--Enter the final outcome of attempts to dial a record.
- Destination DN (if "Apply to call")--Enter the DN to which a treatment will forward a call.
- Number in sequence--In a treatment sequence, each treatment contains a unique number that determines the order in which it will be applied to the same call result, on calls from the same calling list.
- Cycle attempt (only for redial)--Enter the maximum number of consecutive times that the treatment will be performed on the record.
- Interval--Enter the time interval, in minutes, that the application will wait between the first dialing attempt and the first treatment attempt, or between consecutive attempts.
- Increment--Enter the number of minutes that the application will add to the interval (see the previous item) after the first treatment attempt, and after each consecutive attempt.

- Date/Time--Enter the date/time when another treatment attempt will be performed.

Calling Lists

Calling lists contain dialing records. Decide where you will store the calling lists. They can be located either in the Genesys Configuration Database or in a separate database. The size of the calling lists and the ease of maintenance are the deciding factors when identifying the appropriate location. Very large calling lists should be maintained in a database other than the Genesys Configuration Database.

- Name--Enter the name of the new calling list.
- Table Access Point name--Enter the name of the `Table Access` object to which the calling list refers.
- Dialing filter name (optional)--Enter the name of default dialing filter to apply to the calling list.
- Script property--The Script property in the campaign, in the calling list, and in the Campaign Group defines the Script object, which contains all of the attributes that are required by Agent Scripting.

Treatments

- Treatment names (optional)--Identify or create treatments that will be applied to the records on the calling list if calls are unsuccessful.

Campaigns

Campaigns are collections of callings lists and agent or place groups. Agent skills that might be considered during the processing of customer interactions include language or marketing knowledge.

- Name--Enter the name of the campaign.
- Calling list name(s)--Enter the names of the calling lists to be used by each campaign.
- List weight (if more than one list)--Enter the percentage of call records to be retrieved from a calling list for a campaign. For more information, see *Framework Genesys Administrator Help.*

Campaign Group

- Group name(s)--Enter the names of the agent or place groups assigned to each campaign.
- Group type--Enter the type (agent or place) of each group assigned to each campaign.
- Dialing mode--Decide on a dialing mode to use as the default for the agent groups in a campaign. The dialing modes are `Progressive, Predictive, Preview, Push Preview`, and `Power GVP`.
- Voice Transfer Destination ("Origination DN" in previous releases)--Enter the queue or Routing Point to which a call is transferred after a "live" answer.

- Optimization method--Sets the default dialing optimization method for the Campaign Group and applies only to Predictive dialing mode.
- Target value--Enter the percentage value of the optimization method.
- Buffer size coefficient minimum--Enter the minimum number of chains per active agent that OCS can keep on hand for dialing.
- Buffer size coefficient optimal--Enter the optimal number of chains per active agent that OCS can keep on hand for dialing.
- Number of CPD ports--Enter the number of dialing ports that can be used for dialing on behalf of a Campaign Group. If you are using `ASM` mode, at least one port is required for each agent for the engaging calls, in addition to ports for outbound dialing.
- CPD Server name (if more than one)--Enter the name of the CPD Server to use for this Campaign Group.
- Script property--The Script property in the campaign, calling list, and Campaign Group defines the Script object, which contains all of the attributes that are required by Agent Scripting.

Note:	For an Outbound-IP environment, configure Trunk Group DNs on the `Configuration` tab/Advanced section. For more information, see Outbound VoIP Deployment.

Reporting (Optional)

- Real-time reporting--If you install real-time reporting (CCPulse+), see also the *Reporting 8.0 Deployment Guide* and the *Reporting 8.0 CCPulse+ Administrator's Guide*.

Note:	You can also use CCPulse+ to view historical reports.

- CCPulse+ name--Enter the name of the CCPulse+ to be used.
- Stat Server name--Enter the name of the Stat Server to be used with CCPulse+.
- Historical reporting--If you install historical reporting (ICON and/or CCPulse+), see the *Reporting 8.0 Deployment Guide* and/or the *Interaction Concentrator Deployment Guide.*

Outbound Deployment Planning Worksheet

This planning worksheet provides a single place in which to record the information that is required to configure and install Outbound Contact. The cells on the worksheet provide room to write additional values. After it is completed, the planning worksheet expedites the installation and configuration process. It serves as a "look-up" sheet for critical information that you will need in order to complete the process. The information about the items in the Sample Planning Worksheet is important to have on hand.

Sample Planning Worksheet

The following table is a sample planning worksheet in which to record information that you need to have on hand during the configuration process.

T-Server(s)

Planning Worksheet

	Name	Item	Item	Item	Item
Logging In to Genesys Administrator					
1.	User name				
2.	User password				
3.	Application				
4.	Host				
5.	Port				
License Manager					
6.	Path to License Manager				
Outbound Solution Object					
7.	Name				
Tenant					
8.	Names				
Switches					
8.	Name(s)				
9.	Type of switch				

10.	Internal or External CPD capability				
11.	T-Server connected to each switch (one-to-one relationship)				
12.	Special DNs.19for example, Communication DN and Overflow DN				
DNs					
13.	ACD positions				
14.	Extensions				
15.	Call processing Ports (CPP)				
16.	ACD Queues				
17.	Routing Points				
18.	Virtual Queues				
19.	Communication DNs				
20.	Trunk Group DNs (for Outbound-IP dialing modes) **Note:** For more information, see Outbound VoIP Deployment.				
Applications					
21.	T-Server name(s)				
22.	Host				
23.	Port				
24.	Back-up				
25.	Designated switch (one-to-one relationship)				
Stat Server(s)					

26.	Stat Server name(s)				
27.	Host				
28.	Port				
29.	Back-up server				
DB Server					
30.	DB Server name(s)				
31.	Host				
32.	Port				
33.	Back-up server				
34.	Options (see Database Access Point(s).)				
35.	DBMS type				
Outbound Contact Server (OCS)					
36.	OCS name				
37.	Host				
38.	Communication port				
39.	Back-up server				
39.	Name(s) of T-Server(s) to which OCS connects				
40.	Name of Message Server to which OCS connects (optional)				
CPD Server					
41.	CPD Server name(s)				
42.	Host				
43.	Communication Port				
44.	Back-up server for CPD Server				
45.	Name of T-Server to which CPD Server connects				
46.	Type of				

	Dialogic board(s)				
47.	DTI Ports (DNs for Dialogic board). (See DNs, near the top of the table, starting at row 13.)				
Genesys Administrator					
48.	Genesys Administrator name (application)				
49.	Name of OCS to which Genesys Administrator connects				
Objects					
Database Access Point(s)					
50.	Name				
51.	DB Server name. (See row 30.)				
52.	DB Server host. (See row 31.)				
53.	DB Server communication port. (See row 32.)				
54.	DBMS name				
55.	DBMS type. (See row 35.)				
56.	Database name				
57.	User name for login to DB				
58.	Password for login to DB				
59.	Case Conversion				
User-Defined Fields					
60.	Name				

61.	Data type				
62.	Length				
63.	Default value (optional)				
Formats					
64.	Name				
65.	User-defined field names. (See above.)				
Table Access					
66.	Name				
67.	Table type				
68.	DB Access Point name. (See row 50.)				
69.	Format name. (See row 64.)				
70.	Database table name				
Dialing Filters					
71.	Name(s)				
72.	Criteria SQL statement				
73.	Order by SQL statement				
Treatments					
74.	Name				
75.	Call result				
76.	Apply to record				
77.	Apply to call				
78.	Destination DN (if "Apply to call")				
79.	Number in sequence				
80.	Cycle attempt (only for redial)				
81.	Interval				
82.	Increment				
83.	Date/Time				
84.	Range				

Calling Lists					
85.	Name				
86.	Table Access Point name				
87.	Dialing filter name (optional)				
88.	Script property (optional)				
89.	Treatment names (optional). (See Treatments.)				
Campaigns					
90.	Name				
91.	Calling list name(s)				
92.	List weight (if more than one list)				
Campaign Group					
93.	Session name(s)				
94.	Session type				
95.	Dialing mode				
96.	Voice Transfer Destination ("Origination DN" in previous releases)				
97.	Optimization method				
98.	Target value				
99.	Buffer size coefficient minimum				
100.	Buffer size coefficient optimal				
101.	Number of CPD ports				
102.	CPD Server name (if more than one) (optional)				
103.	Script property				

	(optional)				
104.	Stat Server name				
105.	OBN Manager name (optional component)				
106.	Interaction Server (optional component)				
107	Trunk DNs (for Outbound-IP dialing modes) **Note:** For more information, see OutboundVoIPDeployment.				
Reporting (Optional)					
108.	Real-time reporting; if yes, see Reporting documentation.				
109.	Historical reporting; if yes, see Reporting documentation.				

Other Resources

You may want to use the following additional resources for reference.

- *Framework Genesys Administrator Help*--This Help system contains details about setting up application and component objects, and performing other important configuration tasks.
- *Framework*--You can obtain information for the Outbound Deployment Planning Worksheet from Framework. For example, Tenants and Switch objects usually are created in the Framework Wizard, but sometimes they are created in Genesys Administrator. Outbound Contact uses some Framework `Application` objects that are created in the Framework Wizard or Genesys Administrator--for example, the Stat Server and DB Server. For some objects, you only need to identify their names in Framework and write those names on the worksheet.
- *Outbound Contact 8.1 Reference Manual*--This document explains the constants and communication protocols for Outbound Contact.
- Wizard Advisories for Outbound Contact, Framework, and Reporting, also contain useful information, and are located on each specific solution CD. All other documents are provided on the documentation library software CD that is specific to Outbound Contact as well as to other Genesys software products.

Note:	Refer to the appropriate GVP and Interaction Server documentation if you are implementing `Power GVP` or `Push Preview` dialing modes.

Manually Configure and Install Outbound Applications

These topics describe how to configure and install Outbound Contact manually:

- Overview
- Configuration
- Installing On Windows
- Installing On UNIX

Overview

These topics provide instructions for a manual, first-time configuration and installation of the Outbound Contact components, which include Outbound Contact Server (OCS), CPD (Call Progress Detection) Server, CPD Proxy Server, and Outbound solution objects. This topic also describes an Outbound-specific aspect of Stat Server.

Note:	These topics include instructions for installing Outbound Contact Manager (OCM), even though Genesys Administrator replaces OCM as the preferred configuration interface in release 8.1. OCM 7.6 can be used with Outbound Contact 8.1. For installation instructions for Genesys Administrator, see the *Framework Genesys Administrator Deployment Guide.*

Use the completed Worksheet Definitions in this guide, as well as the other relevant chapters in this guide, to configure and install Outbound Contact applications (components) and objects.

Note:	Genesys does not recommend installing its components via a Microsoft Remote Desktop connection. You should perform the installation locally.

The overall process of configuring and installing Outbound Contact components involves the following steps:

Configuring and Installing Outbound Contact Components

Objective	Related Procedures and Actions
1. Configure application objects.	1. Configure and install all Framework components, which include Configuration Server, T-Server, the Management Layer, DB Server, and Stat Server. 2. Configure and install Genesys Administrator. 3. Start Configuration Server. 4. Start Genesys Administrator. 5. Configure OCS as described on Outbound Contact Server. 6. Configure CPD Server as described on CPD ServerCPD Server. 7. Configure Genesys Administrator. See the *Framework Genesys Administrator Deployment Guide.*

2. Install Outbound Contact components.	1. Install OCS as described for Windows (Installing OCS on Microsoft Windows 2003 or 2008) or for UNIX (see Installing OCS on UNIX). 2. Install CPD Server as described on Installing CPD Server on Microsoft Windows 2003 or 2008. 3. Install Genesys Administrator. See the *Framework Genesys Administrator Deployment Guide.*
3. Configure and install real-time and historical reporting.	Refer to Real-Time and Historical Reporting.

Configuration

Note:	All of the information in this section is based on using Genesys Administrator to configure and install your Outbound applications. However, you can still use Configuration Manager.

Before you create an `Application` object, make sure that an application template exists. The template provides most of the application's configuration options and default values.

To locate an existing template, open the `Environment` view in Genesys Administrator and then the `Application Templates` view. If the template is absent, import one from the `Templates` folder on the Outbound Contact Center CD.

Importing a template

Start

1. Select the `Application Templates` view.
2. Open the Tasks panel, if necessary, and click `Import Application Template.`
3. In the file-selector dialog box, do the following:
4. If necessary, navigate to where the template is stored.
5. Select the appropriate `.apd` file.
6. Click `Open.`
7. On the `Configuration` tab, enter or modify information as required.
8. On the `Options` tab, enter or modify information as required.
9. If the `Templates` folder also contains a metadata file, import that metadata into the Application Template.
10. To save the new Application Template and register it in the Configuration Database, perform one of the following:
    a. Click `Save and Close` to return to the list of Application Templates.
    b. Click Save to continue configuring the Application Template.
    c. Click `Save and New` to save the new Application Template and start creating another one.

End

Configuration Advisory

Application objects represent software components in Genesys Administrator. This topic describes four Outbound Contact application objects:

- Outbound Contact Server (OCS)
- CPD Server (optional)
- CPD Proxy Server (optional)
- Outbound Contact Manager (optional if you are using Genesys Administrator)

Outbound Contact also requires the following applications:

- Genesys Administrator--Allow you to configure all Outbound Contact applications and objects, run dialing sessions, and monitor all related activities
- Stat Server (Reporting)--Determines what agents are assigned to which Agent or Place Groups, identifies the current Agent State, and for reporting purposes.
- CCPulse+ (Reporting)--Calculates real-time and historical Outbound-specific statistics
- ICON (Reporting)--9Calculates historical Outbound statistics

The person who will run or change an Outbound dialing session/campaign requires access permissions that include Read access privileges to all campaign-related configuration objects. Click the `Permissions` tab of each configuration object to view the `Access Groups` that have permission to read it.

> ### Warning
> Starting with release 8.1.527.14, OCS supports Stat Server that has the **accept-clients-in-backup-mode** option set to `true`. See the stat-server-ha option description for details. Earlier versions of OCS do not support Stat Server that has the **accept-clients-in-backup-mode** option set to `true`. In Stat Server version 8.5.1, this is the default value for this option when it is not configured.

Client-Side Port Security Configuration

To increase security, you can define a fixed port for the connection between an Outbound Contact component and another server that is behind a firewall. The client-side port definition feature allows a server application to control the number of client connections, preventing the server from an excessive number of malicious requests to the same server-side port.

For configuration instructions, see the "Client-Side Port Definition" chapter of the *Genesys Security Deployment Guide*. The following table identifies which Outbound Contact-specific components support this configuration.

Component Support of Client-Side Port Security

Server Clients	Configuration Server/Configuration Server Proxy	T-Server
OCS	Yes	Yes
CPD Server	Yes	Yes
CPD Proxy Server	Yes	Not Applicable

Connections to Configuration Server

During the application installation, new command-line arguments are added to connect to Configuration Server using the port designated during installation, which are then used during application startup. The arguments include:

`-transport-address` and `-transport-port`.

The command-line format for starting a connection to Configuration Server would be:

```
cm_server.exe -host <host_name> -port <port_number> -transport-address <IP address>
-transport-port <port number> -app <application_name>
```

After installation, these arguments appear in three places:

- Configuration Server `Application` object > `Configuration` tab > `Server Info` section.
- (Windows operating system) `startServer.bat` file; (UNIX operating systems) `run.sh`.
- In the Registry when the application (OCS, CPD Server, or CPD Proxy Server) is started as a Service

Note:	To support reconnection to Configuration Server, you must still create or update the existing connection to Configuration Server in the OCS `Configurations` tab of the OCS `Application` object, following the standard procedure for configuring connections to other servers. For specific instructions associated with client-side port connections, see the *Genesys Security Deployment Guide*.

Outbound Contact Server

Outbound Contact Server (OCS) is the main component of Outbound Contact.

Configuring OCS

Start

1. In Genesys Administrator, open the `Environment` view and then select the `Applications` view.
2. From the toolbar, click `New`.
3. In the `Browse` dialog box, select the Application Template for this application (navigating to the appropriate folder if necessary), and click `OK`.

Note:	If an Application Template for OCS is not available, click `Cancel` and import or create a template. Then restart this procedure to create the new template.

4. On the `Configuration` tab, enter the following information for the OCS application:

- Name: Enter the name of the OCS application
- Connections: Click Add, and then add a T-Server application. Optionally, you can also add a Message Server application.

Note:	Do not name the OCS application OCServer because this is a section name in the Options tab of a configuration object. See Section Names for more information.

The only servers for which OCS searches in the Connections section of its application at startup are T-Server and Message Server (optional). Connection to other servers that are required to run an outbound dialing session (such as CPD Server, Stat Server, and Interaction Server) can be specified on the Configuration tab of a Campaign Group object.

Notes:	• If you are configuring OCS to support TLS, see the *Genesys Security Deployment Guide* for configuration instructions; also, see Transport Layer Security Connections for information on which Outbound Contact components support TLS connections for this release. • If you are configuring a connection to Interaction Server, create a port called ESP. Use this ESP PortID instead of the default PortID, when you configure a connection to the OCS Application object of the Interaction Server Application object. This makes OCS a server for Interaction Server and Interaction Server a client for OCS. • If you are configuring TLS between OCS and DB Server, add a Database Access Point (DAP) to DB Server on the Configuration tab/Connections section of the OCS Application object. • If you are configuring TLS between OCS and CPD Server/CPD Proxy Server, add a connection on the Configuration tab/Connections section of the OCS Application object to CPD Server/CPD Proxy Server respectively.

5. In the Server Info section of the Configuration tab, do the following:
   a. Enter the host name of the computer on which this application will be installed, and enter the port that the application will use for client connections.
   b. If you plan to use another server as a backup, in the event that the primary server fails, enter the name of the backup server and adjust the value of the Reconnect Timeout option.
   c. If you need OCS to support communication with clients when using HTTP or HTTPS requests (for example, to update records over HTTP/HTTPS), configure the interface by specifying a separate listener port for HTTP Proxy Server with the Connection Protocol set to http. For more information about HTTP and HTTPS requests and HTTP Server, see Outbound Contact and HTTP Server and the *Proactive Contact 8.0 Solution Guide.*

Note:	Dynamic changes to this port are not supported. As a result, any such changes are

	not communicated to an HTTP Proxy that is already running. However, any HTTP Proxy started after that change will reflect the change.

6. Define the `Working Directory` and `Command-Line` properties - for example:
```
Working Directory: C:\GCTI\Outbound Contact ServerCommand Line: cm_server.exeCommand
Line Arguments: -host <host_name> -port <port_number> -app <Capplication_name>
```

Note:	If you are implementing the client-side port security feature, specify the additional arguments described in Client-Side Port Security Configuration.

7. Click the `Options` tab, and change the values of the configuration options according to your configuration.

For detailed information about OCS options, see the Outbound Contact Server Options.

End

Genesys Administrator

For configuration and installation procedures for Genesys Administrator, see the *Framework Genesys Administrator Deployment Guide.*

CPD Server

Outbound Contact requires hardware that recognizes call results, a capability known as call progress detection (CPD). Outbound Contact works with a CPD device provided by the switch vendor, or with CPD hardware from Dialogic.

If the switch does not have call progress detection capability, you must install switch-extending equipment (a Dialogic board) to perform this function. Install the GlobalCall Dialogic package before installing the CPD Server.

Note:	Beginning with release 7.5, you can also install the Dialogic Host Media processing (HMP) software. Only one instance of CPD Server using HMP software can be executed on a host computer.

Before you configure CPD Server, have on hand the Worksheet Definitions. Refer also to CPD Proxy Server and Dialogic Board Setup.

Configuring CPD Server

Start

1. In Genesys Administrator, open the `Environment` view and then select the `Applications` view.
2. From the toolbar, click `New`.
3. In the `Browse` dialog box, select the Application Template for this application (navigating to the appropriate folder if necessary), and click `OK`.

Note:	If an Application Template for OCS is not available, click `Cancel` and import or create a template. Then restart this procedure to create the new template.

4. On the `Configuration` tab, enter the following information for the CPD Server application:
   a. Name: Enter the name of the CPD Server application.
   b. Connections: Click `Add`, and then add a `T-Server` application and a `Message Server` application to which this server will connect.
5. In the `Server Info` section of the `Configuration` tab, perform the following,
   a. Enter the host name of the computer on which this application will be installed, and enter the port that the application will use for client connections.
   If you plan to use another server as a backup in the event that the primary server fails, and if a Dialogic board is installed, enter the name of the backup server and adjust the value of the `Reconnect Timeout` option.

Note:	If you are configuring CPD Server to support TLS, see the *Genesys Security Deployment Guide* for configuration instructions. Also, see Transport Layer Security Connections for information on which Outbound Contact components support TLS connections for this release.

   b. Define the `Working Directory` and `Command Line` properties—for example:
   ```
   Working Directory: C:\GCTI\cpd_server
   Command Line: cpdserver.exe
   Command Line Arguments:
   -host <host_name> -port <port_number> -app <application_name>
   ```

Note:	If you are implementing the client-side port security feature, specify the additional arguments described in Client-Side Port Security Configuration.

6. Click the `Options` tab, and change the values of the configuration options according to your configuration.
   For detailed information about the options for CPD Server, CPD Server Options.

End

CPD Proxy Server

CPD Proxy Server is the core load-distribution component of CPD Server. The main task of this optional component is to distribute client requests among various CPD Servers, in order to balance the call load and optimize the use of hardware resources within the system. You must configure CPD Proxy Server before you install it.

Configuring CPD Proxy Server

Start

1. In Genesys Administrator, open the `Environment` view and then select the `Applications` view.
2. From the toolbar, click `New`.
3. In the `Browse` dialog box, select the Application Template for this application (navigating to the appropriate folder if necessary), and click `OK`.

Note:	If an Application Template for OCS is not available, click `Cancel` and import or create a template. Then restart this procedure to create the new template.

4. On the `Configuration` tab, enter the following information for the CPD Proxy Server application:
   a. Name: The name of the CPD Proxy Server application.
   b. Connections: Click `Add`, and then add the CPD Server(s) and the Message Server to which CPD Proxy Server will connect.
5. In the `Server Info` section of the `Configuration` tab, perform the following,
   a. Enter the host name of the computer on which this application will be installed, and enter the port that the application will use for client connection.
   If you plan to use another server as backup, in the event that the primary server fails, enter the name of the backup server and adjust the value of the `Reconnect Timeout` option. See Connection and Reconnection for more information.

Note:	If you are configuring CPD Proxy Server for TLS, see the *Genesys Security Deployment Guide* for configuration instructions. Also, see Transport Layer Security Connections for information on which Outbound Contact components support TLS connections for this release.

   b. Define the `Working Directory` and `Command Line` properties--for example:
   `Working Directory: C:\GCTI\cpd_proxyCommand Line: cpdproxy.exe Command Line Arguments: -port <port_number> -host <host_name> -app <application_name>`

Note:	If you are implementing the client-side port security feature, specify the additional arguments described in Client-Side Port Security Configuration.

6. Click the `Options` tab, and change the values of the configuration options according to your configuration.

For detailed information about these options, see Log Files.

Note:	If you have a multi-tenant environment and want to share HMP resources, you can also configure options to restrict dialing and engaging channels. For more information, see Centralized Configuration and Shared HMP Resources.

End

DM3 Boards with CPA Functionality

If you are using ASM mode, and if you are using the DM3 families of boards with CPA functionality that CPD Server supports, you must install the GlobalCall Dialogic package before you install CPD Server. These families include the following:

- DM/V1200-4E1 (number of ports = 120)
- DM/V600-2E1 (number of ports = 60)
- DM/V960-4T1 (number of ports = 96)
- DM/V480-2T1 (number of ports = 48)

Outbound-Specific Statistics for Stat Server

In addition to determining which agents are assigned to which Agent or Place Groups and identifying the current Agent State, Stat Server requires Outbound statistic definitions in order to monitor the Outbound activities of Campaigns, Calling Lists, Dialing Sessions, ACD Queues, and agents.

Importing Outbound-Specific Statistics into the Stat Server Object

Start

1. In Genesys Administrator, open the `Environment` view and then open the `Applications` view.
2. On the Details panel, double-click the `Stat Server` object.
3. Click the `Options` tab.
4. Locate and import `StatProfile.cfg`.
    - The latest version of this file is available on `IP_CCPOCTemp_8000003b1_ENU_multi`, in the following folder: `multi\b1\ip\realtime`.

Note:	Stat Server is usually configured using the Framework Configuration Wizard or in Configuration Manager. In release 8.x, you can also configure it in Genesys Administrator and using the Outbound Contact Wizard.

End

The following sections provides information on Outbound-specific real-time statistics that were introduced after Release 7.2.

New Real-Time StatServer Java OCCExtensions Statistics as of Release 7.6 or higher

Note:	These statistics require Stat Server version 7.6.1 or higher with OCC Extension 7.6.1 or higher.

CurrentTrustFactor

- Category: JavaCategory
- JavaSubCategory: OCCStatExtension.jar:CurrentTrustFactor
- Objects: Agent, Place

This metric, with values ranging from 0% to 100%, defines the trust factor for a specific agent. For more information about how this factor is calculated and when it is used, see the time-to-ready-tolerance option. This statistic is calculated based on the following formula: A_{indiv} = (1 - trust factor) * 100 = $N_{correct\ notifications}$/ $N_{total\ notifications}$ * 100%

CurrentFeedbackAccuracy

- Category: JavaCategory
- JavaSubCategory: OCCStatExtension.jar:CurrentFeedbackAccuracy
- Objects: CampaignGroup

This metric, with values ranging from 0% to 100%, defines the current accuracy of the agent's ready time feedback for a session.

This statistic is calculated based on the following formula:

A_{sess} = (1 - Total Number of False Notifications/ Total Number of Notifications) * 100%

New Real-Time StatServer Java OCCExtensions Statistics as of Release 7.5 or higher

CurrentAgentAssignment

- Category: JavaCategory
- JavaSubCategory: OCCStatExtension.jar:CurrentAgentAssignment
- Objects: Agent, Place

CurrentCampaignGroupDBID

- Category: `JavaCategory`
- JavaSubCategory: `OCCStatExtension.jar:CurrentCampaignGroupDBID`
- Objects: `Agent`, `Place`

CurrentNumberAgentsAssigned

- Category: `JavaCategory`
- JavaSubCategory: `OCCStatExtension.jar:CurrentNumberAgentsAssigned`
- Objects: `CampaignGroup`

New Real-Time StatServer Java OCCExtensions Statistics as of Release 7.2 or higher

CurrentNumberChainsReady

- Category: `JavaCategory`
- JavaSubCategory: `OCCStatExtension.jar:CurrentNumberChainsReady`
- Description: Current number of chains ready.
- Objects: `CampaignGroup`, `CallingList`, `CampaignCallingList`

CurrentHitRatio

- Category: `JavaCategory`
- JavaSubCategory: `OCCStatExtension.jar:CurrentHitRatio`
- Description: Current hit ratio.
- Objects: `CampaignGroup`

CurrentNumberDialPortsUsed

- Category: `JavaCategory12:31 PM 2/17/2010`
- JavaSubCategory: `OCCStatExtension.jar:CurrentNumberDialPortsUsed`
- Description: Current number of dial ports used.
- Objects: `CampaignGroup`

CurrentNumberEngPortsUsed

- Category: `JavaCategory`
- JavaSubCategory: `OCCStatExtension.jar:CurrentNumberEngPortsUsed`

- Description: Current number of engaged ports used.
- Objects: `CampaignGroup`

CurrentBusyFactor

- Category: `JavaCategory`
- JavaSubCategory: `OCCStatExtension.jar:CurrentBusyFactor`
- Description: Current number of busy factor.
- Objects: `CampaignGroup`

CurrentOverdialRate

- Category: `JavaCategory`
- JavaSubCategory: `OCCStatExtension.jar:CurrentOverdialRate`
- Description: Current number of overdial rate.
- Objects: `CampaignGroup`

CurrentAverageWaitingTime

- Category: `JavaCategory`
- JavaSubCategory: `OCCStatExtension.jar:CurrentAverageWaitingTime`
- Description: Current average waiting time.
- Objects: `CampaignGroup`

CurrentNumberChainsFinalized

- Category: `JavaCategory`
- JavaSubCategory: `OCCStatExtension.jar:CurrentNumberChainsFinalized`
- Description: Current number of chains finalized.
- Objects: `CampaignGroup`, `CallingList`, `CampaignCallingList`

CurrentNumberChainsRetrieved

- Category: `JavaCategory`
- JavaSubCategory: `OCCStatExtension.jar:CurrentNumberChainsRetrieved`
- Description: Current number of chains retrieved.
- Objects: `CampaignGroup`, `CallingList`, `CampaignCallingList`

The Outbound Solution Object

You can start Outbound Contact through Solution Control Interface (SCI), a Framework application. SCI uses the Outbound Solution object to start and stop the Outbound Contact components.

Note:	You can also start Outbound Contact through Genesys Administrator. For more information, see *Framework Genesys Administrator Help.*

In Genesys Administrator > Provisioning tab > Environment, select the Solutions folder, and determine if the Outbound Solution object is in the list. If it is not, you must import it from the product CD and then configure it.

Importing and Configuring the Outbound Solution Object

Start

1. In Genesys Administrator, select Provisioning > Environment > Solutions.
2. From the Tasks panel, select Create > Upload solution.
3. In the Click Add and choose... dialog box, click Add and navigate to the Template folder on the product CD.
4. Select the OCSolution_80.sd file, click Open and then Upload.
5. Double-click the solution to open the Outbound Solution object.
6. On the Configuration tab:
   - In the Name field, enter a name.
   - In the Assigned Tenant field, select a tenant.
   - In the Solution Control Server field, select a server.
7. Click the Components section and then add the applications that Outbound Contact needs in order to run (see the following table).
8. Click Save and Close.

End

The following table shows the applications that you could add to the Components section for the Outbound Solution Object. The table heading definitions are as follows:

- Application--The name of the application that Outbound Contact needs in order to run.
- Optional--The requirement status for the application: False means that the application is mandatory in order for Outbound Contact to be able to run; True means that the application is optional, and that it is not mandatory for Outbound Contact to be able to run.
- Priority--The order in which the application starts, relative to the other applications.

Outbound Solution Object Components Tab

Application	Optional	Priority
T-Server	False	1
Interaction Server	True	2
DB Server	False	2
Stat Server	False	2
Data Sourcer	True	2
ETL Proxy	True	2
Outbound Contact Server	False	3
Message Server	True	4
CPD Server	True	4
Outbound Contact Manager or Genesys Administrator	True	5

Installing on Windows

The Genesys Outbound CD contains the installation packages for the OCS and OCM applications. The CPD Server installation package is located on the Genesys Outbound CPD Server CD.

The setup files for these components are located in the following folders:

- OCM: `\\solution_specific\outbound_contact_mgr\windows`

Note:	The OCM release 7.6.x is packaged with Outbound Contact 8.1.

- OCS: `\\solution_specific\outbound_contact_server\windows`
- CPD Server: `\\solution_specific\cpd_server\windows`

If you downloaded Outbound Contact from an FTP site, the installation package is contained in a ZIP file and you must extract it using the WinZip utility. After you extract the files, the setup files for the applications will be in the `Windows folder`.

Installing OCS on Microsoft Windows 2003 or 2008

Note:	The OCS `Application` object must already be configured before you begin the installation.

Start

1. From the product CD, open the following folder: `solution_specific\outbound_contact_server\windows and run setup.exe`

Note:	If you are installing a new version of the same release, you will need to select `Install new instance of the application` as the Setup `Type.`

2. If you are setting up client-side port configuration for the initial connection to Configuration Server, select the `Use Client Side Port` check box and then specify the following parameters:
   - `Port`--Enter the port number for the client, as preconfigured on your firewall. (This is not the Listening port of the OCS application.)
   - `IP Address`--Enter the IP Address of the computer on which you are installing and running the OCS application.

Notes:	• After selecting this option, the installation process will add the necessary command line arguments `-transport-address` and `-transport-port`) for connecting to Configuration Server at the application startup.

	• 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.

3. Enter the `host name, port, username` and password of the Configuration Server.
4. Select the `Application` object that you created in Genesys Administrator.
5. Select the license options:
   - `License Manager`--Enter the port and name for the license server.
   - `License File`--Use the `Browse` button to locate the license file.
6. To select the destination directory, either click `Next` to accept the default (displayed) or click `Browse` to locate another directory.
7. Click `Install` to start the installation.
   On Windows operating systems, all servers are installed as services. Therefore, during installation, you will not be asked whether you want to install OCS as a Windows service. See Using Windows Services, for more information.
8. Click `Finish` to complete the installation.

End

> Tip
>
> To maximize OCS performance, Genesys recommends that you set the power profile of your Windows operating system to **High Performance**.

Installing CPD Server on Microsoft Windows 2003 or 2008

CPD Server is an optional application.

Notes:	• It is not advisable to mix board technologies. • Do not use different protocols (such as ISDN and Melcas) in one box. • The CPD Server Installation Wizard may not recognize that the Dialogic drivers had been already installed if its version is different from the version of the drivers that are packaged on the Genesys CPD Server CD. • CPD Server supports only the Dialogic drivers (for Service Release 6) that are provided on the CPD Server CD.

Start

1. From the product CD, open the following folder: `solution_specific\cpd_server\windows` and run `setup.exe`
2. As the `Maintenance Setup Type`, select either `Install new instance of the application` or `Maintenance of the existing installation`.
3. Select `Call Progress Detection Server` as the component to install.
4. If you are setting up client-side port configuration for the initial connection to Configuration Server, select the `Use Client Side Port` check box and then specify the following parameters:
   - `Port`--Enter the port number for the client, as preconfigured on your firewall. (This is not the Listening port of the CPD Server application.)
   - `IP Address`--Enter the IP Address of the computer on which you are installing and running the CPD Server application.

Note:	After selecting this option, the installation process will add the necessary command line arguments (`-transport-address` and `-transport-port`) for connecting to Configuration Server at the application startup.

5. Enter the `host name, port, username`, and `password` of Configuration Server.
6. Select the license options:
   - `License Manager`--Enter the port and name for the license server.
   - `License File`--Use the `Browse` button to locate the license file.
7. To select the destination directory, either click `Next` to accept the default (displayed) or click `Browse` to locate another directory.
8. Click `Install` to start the installation.
   On Windows operating systems, all servers are installed as services. Therefore, you will not be asked whether you want to install CPD Server as a Windows service. For more information, see Using Windows Services.
9. Click `Finish` to complete the installation.

End

Installing CPD Proxy Server on Microsoft Windows 2003 or 2008

CPD Proxy Server is an optional application. CPD Proxy installation does not require Dialogic hardware to be installed on the target computer. **Start**

1. From the product CD, open the following folder: `solution_specific\cpd_server\windows`
2. As the `Maintenance Setup Type`, select either `Install new instance of the application` or `Maintenance of the existing installation`.
3. Select `Call Progress Detection Proxy` as the component to install.
4. If you are setting up client-side port configuration for the initial connection to Configuration Server, select the `Use Client Side Port` check box and then specify the following parameters:

- Port--Enter the port number for the client, as preconfigured on your firewall. (This is not the Listening port of the CPD Proxy Server application.)
- IP Address--Enter the IP Address of the computer on which you are installing and running the CPD Proxy Server application.

Note:	After selecting this option, the installation process will add the necessary command line arguments (-transport-address and -transport-port) for connecting to Configuration Server at the application startup.

5. Enter the host name, port, username, and password of the Configuration Server.
6. Select the Application object that you created in Genesys Administrator.
7. To select the destination directory, either click Next to accept the default (displayed) or click Browse to locate another directory.
8. Click Install to start the installation.

   On Windows operating systems, all servers are installed as services. Therefore, during installation, you will not be asked whether you want to install CPD Proxy Server as a Windows service. See Using Windows Services for more information.

9. Click Finish to complete the installation.

End

Real-Time and Historical Reporting

The *Reporting Technical Reference Guide* describes the Reporting templates.

For information about importing templates, refer to the following resources:

- *Reporting 7.5 CCPulse+ Help*, "Using the Import/Export Utility"
- *Reporting 7.2 Data Modeling Assistant Help*, "Importing and Exporting Templates"

For information about actions the user should take if the CCPulse+ Import/Export Utility is disabled, refer to the *Reporting 7.5 CCPulse+ Administrator's Guide*.

Using Windows Services

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

Stopping or Modifying the Startup Type for an Application Running as a Windows

Service

Start

1. On the Windows taskbar, click `Start` and then select `Control Panel > Administrative Tools > Services.`
2. Double-click the application to open its `Property` dialog box window.
3. Under `Service Status,` click the appropriate button to change the operational status of the service.
4. From the `Startup` type drop-down list, select the desired startup type.
5. Click `Apply`.

End

Installing on UNIX

The only Outbound Contact component that can be installed on a UNIX operating system is Outbound Contact Server.

Installing OCS on UNIX

Start

1. On the Outbound Contact 8.1 CD, locate the directory for your UNIX vendor: `solution_specific/outbound_contact_server/<UNIX directory>`
2. Copy the directory into a temporary directory.
3. In the temporary directory, locate the `install.sh` shell script.

Note:	All files require permission to execute.

4. Run the script by executing the following command: `sh install.sh`
5. Type the host name of the computer on which Outbound Configuration Server will be installed, or press `Enter` to accept the default.
6. When/if prompted, enter the `host, network port, user name,` and `password` of Configuration Server.
7. When prompted, specify whether you want to use the Client Side Port option. Then specify the following parameters:
    - `Port`--Enter any free port number. (This is not the Listening port of the OCS application.)
    - `IP Address`--Enter the IP Address of the computer on which you are installing and running the OCS application. (The installation script will propose an IP Address for you to use.)

Note:	After selecting this option, the installation process will add the necessary command line arguments (`-transport-address` and `-transport-port`) for connecting to Configuration Server at the application startup.

8. From the list of applications configured for this host, select a specific OCS application.
9. Select the destination directory into which OCS will be installed.
10. When prompted, enter either 32 or 64 to specify the version that will be used.
11. Enter the full path to the license file.
12. After the installation process is complete, a message appears, indicating that the installation was successful.

End

Next Steps

- Set the Environment Variable for the xerces-c shared library.

Setting the Environment Variable for the xerces-c Shared Library

Purpose

- OCS on UNIX uses the `xerces-c` shared library. This library is copied to the same directory as the OCS executable by the installation program. Before OCS can be successfully started, you must provision the path to this library in your UNIX system.

Start

1. Review the following table to locate the name of the Environment Variable to be used with your operating system.

Environment Variables

Operating System	Environment Variable Name
AIX	LIBPATH
HP-UX 32-bit	SHLIB_PATH
HP-UX 64-bit	LD_LIBRARY_PATH
Linux	LD_LIBRARY_PATH
Solaris 32-bit	LD_LIBRARY_PATH
Solaris 64-bit	LD_LIBRARY_PATH_64

2. Set the Environment Variable.
   The following examples show how to set the Environment Variable if you are using C-Shell on Linux:
   - `setenv LD_LIBRARY_PATH /home/OCS:$LD_LIBRARY_PATH`
   - `setenv LD_LIBRARY_PATH .:$LD_LIBRARY_PATH`
3. If you use LCA and Solution Control Server to start OCS, you can use the following approach:
   a. After you install OCS, replace the Command Line property on the `Start Info` tab of the OCS `Application` object with the env UNIX command.
   b. At the beginning of Command Line Arguments field, insert the following string (see the table above for the correct Environment Variable name) followed by the executable file:
   `<Environment Variable Name>=<value>`
   The rest of the line should remain as it was set up during installation.
   Example 1:
   - Working Directory: `/home/OCS`
   - Command Line: env
   - Command Line Arguments: `LD_LIBRARY_PATH=.:$LD_LIBRARY_PATH ./cmserver -host HostName -port 8100 -app "OCS_Name" -l "LicensePort@LicenceHost"`

   Example 2:
   - Command Line Arguments: `LD_LIBRARY_PATH=/home/OCS:$LD_LIBRARY_PATH ./cmserver -host HostName -port 8100 -app "OCS_Name" -l "LicensePort@LicenceHost"`

End

Outbound Solution Wizard

This topic explains how to use the Outbound Solution Wizard, also known as the Outbound Contact Configuration Wizard.

Note:	Genesys recommends that you use Genesys Administrator instead of the Outbound Contact Configuration Wizard for configuring your enterprise, because much of the wizard functionality is integrated into Genesys Administrator. Warning: This Wizard was not updated for release 8.1. The Outbound Contact Configuration Wizard 7.5 is included in the Outbound Contact 8.1 package.

This Wizard creates a standard configuration of the Outbound Contact components (Outbound Contact Server, Outbound Contact Manager, and Call Progress Detection Server). The wizard creates all mandatory options with their default values. You can easily redefine these default values based on the specific needs of your contact center.

The Wizard also provides a method for setting up the Dialogic board channels (ports) for CPD Server.

Note:	Use this Wizard for standard configurations only, instead of for complex configurations (for example, in an environment that has two Dialogic boards for the same CPD Server or when configuring two or more DNs on a switch that has the same dtiB number).

Before You Begin

First, complete the Framework installation and configuration. Be sure to refer to the Worksheet Definitions. New changes for the Outbound Configuration Wizard include:

- Introduces new component definitions in the `Solution` object for the GVP `OBN Manager` and `Interaction Server` applications.
- Supports the `Campaign Group` object. The `CampaignGroupInfo` object is no longer supported.
- Dynamically adds new components in the `Solution` object.
- Supports the HMP configuration in the CPD Server Wizard.

Note:	Only one instance of CPD Server using HMP software can be executed on a host machine.

Creating an Outbound Solution

As mentioned, the Outbound Solution Wizard enables you to create an Outbound solution. The wizard is accessible through the Genesys Wizard Manager.

Using the Outbound Solution Wizard to create a solution

Start

1. Insert the Genesys 8.1 Outbound CD.
2. In the root directory, double-click `setup.exe` to install the Outbound Contact Configuration Wizard.
3. Open the Genesys Wizard Manager.
4. On the left pane of the Genesys Wizard window, select `Outbound Contact`.
   The Genesys Wizard Manager displays existing solutions, if any, and a prompt to deploy a new Outbound Contact Solution.
5. Create an Outbound Contact Solution or reconfigure an existing one:
   - To create a new Outbound Contact Solution, click `Outbound Contact` on the right pane of the Genesys Wizard Manager page.
     The Outbound Solution Wizard launches and leads you through the process of creating a new Outbound Contact Solution. Skip the remaining steps of this procedure.
   - To reconfigure an existing Outbound Contact Solution object, click the object name on the right pane of the Genesys Wizard Manager page.
     The `Properties` dialog box opens for the `Outbound Contact Solution object`. Proceed to Step 6.

Note:	Complete Steps 6 through 10 only if you are reconfiguring an existing Outbound Contact Solution object.

6. Click the `Components` tab and select a specific component.
7. Click `Properties`.
8. Click `More`.
   The `Properties` dialog box opens for the component that you selected in Step 7.
9. Change the configurations, as needed.
   For full explanations of the options in this dialog box, see Outbound Contact Configuration Options. Also refer to the information in your Sample Planning Worksheet.
10. Click `Apply`.

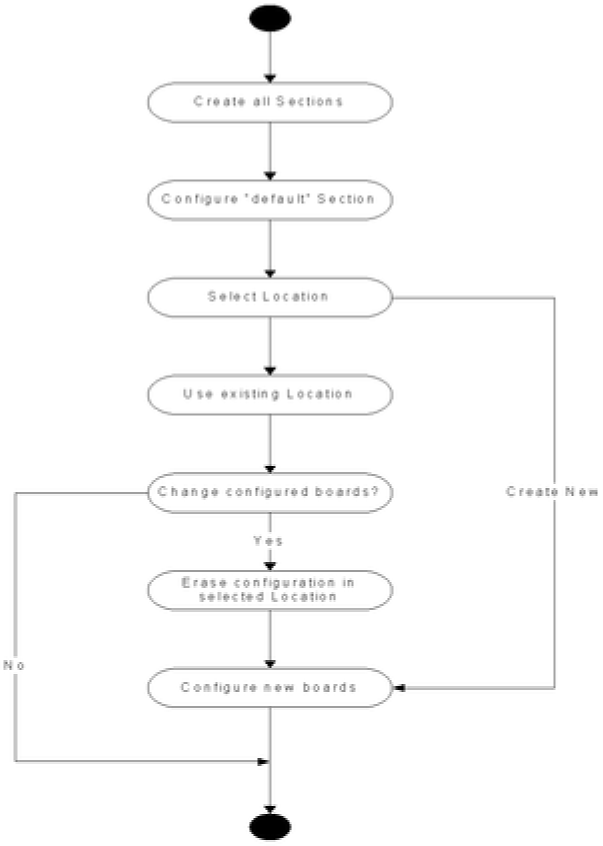
End

Recommended Configuration for CPD Server

The Wizard will not allow you to create an incomplete configuration. If you abort the configuration, the Wizard removes the incomplete configuration.

Resources Configuration Activity

The Wizard implements a list of supported board types for selection. As more than one board may be installed, the Wizard allows selection of a few boards for configuration, instead of one-by-one configuration. When configuring a virtual board, the Wizard assigns a DN to every channel of the board. The Wizard assigns the DNs in sequence, one by one. The following figure illustrates the recommended configuration process for the `CPD Server Application` object.


Create all Sections
Configure 'default' Section
Select Location
Use existing Location
Change configured boards?
Create New
Yes
Erase configuration in selected Location
No
Configure new boards

Recommended Configuration for CPD Server Application Object

Notes:	• Do not mix different board technologies. • Do not use different protocols (such as ISDN and Melcas) in one box. • The CPD Server Installation Wizard will not recognize that the Dialogic drivers had been already installed if its version is different from the version of the drivers that are packaged on the Genesys Dialogic CD included with Outbound Contact. • The CPD Server Installation Wizard will not create the Dialogic DN configuration if you do

	not create a `CPD Server` application. • CPD Server supports only the Dialogic drivers (for Service Release 6) that are provided on the Dialogic CD included with Outbound Contact.

Deleting a previously configured Dialogic board

If you are creating a new configuration, you must first delete the previously configured Dialogic boards.

Start

To delete a previously configured board:

1. When the Wizard displays a list of configured boards, select the boards you want to delete.
2. If a board is associated with a CPD Server application, then the Wizard will warn you and ask for a confirmation.

If the Wizard receives a confirmation, it clears the value of the location option in all applications that match the board name.

End

CPD Server Application Sections and Options

For information about the sections referenced in CPD Server Options.

Silent Setup

InstallShield Silent allows for an automated electronic software distribution, also known as a *Silent Setup.* InstallShield Silent only works on Windows operating systems. With InstallShield Silent, you do not have to monitor the setup or provide input via dialog boxes. Once this information is stored in a *response file,* an InstallShield Silent setup runs on its own, without any intervention by the end-user. An installation procedure for a server application differs slightly from an installation procedure for a GUI application. Both, however, require that you create a response file with the necessary parameters and then use it for the actual installation. Outbound Contact Manager supports Silent Setup installation.

Note:	Genesys Administrator also supports silent installation.

Creating the Response File

To select setup options and automatically record the InstallShield Silent response file, run your setup with the following command line:

```
setup -r
```

Your responses to the dialog boxes are recorded and used to create a response file. By default, the response file is named `Setup.iss,` and is stored in the Windows directory of your computer. To specify a different directory or file name for the response file, add `/f1--[full_path to iss file\]<FileName.3E"` to the setup command. Include the double quotes and do not put a space between /f1 and the path.19for example:

```
setup --r /f1--C:\GCTI\silent_response_files\mySetup.iss--
```

In the optional argument, the `/f1 portion uses the numeral one (1), not the letter l.`

Subsequently, use the response file any time you need to install an application with the configured parameters.

Sample Response File (setup.iss)

```
[InstallShield Silent]
Version=v7.00
File=Response File
[File Transfer]
OverwrittenReadOnly=NoToAll
[{C660F232-38A4-45AF-AE08-A332091BDC91}-DlgOrder]
Dlg0={C660F232-38A4-45AF-AE08-A332091BDC91}-gctiGdWelcome-0
Count=4
Dlg1={C660F232-38A4-45AF-AE08-A332091BDC91}-gctiGdAskPath-0
Dlg2={C660F232-38A4-45AF-AE08-A332091BDC91}-gctiGdReadyToInstall-0
Dlg3={C660F232-38A4-45AF-AE08-A332091BDC91}-gctiGdFinish-0
[{C660F232-38A4-45AF-AE08-A332091BDC91}-gctiGdWelcome-0]
Direction=1
```

```
[{C660F232-38A4-45AF-AE08-A332091BDC91}-gctiGdAskPath-0]
Direction=1
Path=C:\Program Files\GCTI\Outbound Contact Manager
[{C660F232-38A4-45AF-AE08-A332091BDC91}-gctiGdReadyToInstall-0]
Direction=1
[{C660F232-38A4-45AF-AE08-A332091BDC91}-gctiGdFinish-0]
Reboot=0
[{45805BA8-94DF-4C49-A49E-9CC2179FDB8D}-DlgOrder]
Count=0
```

The response file contains saved information about the number of dialog boxes displayed, the order in which the dialog boxes were displayed, the values of any data entered or selected by the end user, and which button the user clicked to close the dialog box.

Running the Silent Installation

Launch the InstallShield Silent Installation with this command line:

```
Setup.exe -s /f1--<full path to Setup.iss>-- /f2--<full path to setup log file>--
```

Where:
`<full path to Setup.iss>`

The full path to the `Setup.iss` file put within double quotation marks. For example:

`c:\windows\setup.iss.`

By default, `Setup.exe` looks for a response file called `Setup.iss` in the same directory as `Setup.exe.`
`<full path to setup log file>`

The full path to the setup log file put within double quotation marks. For example: `c:\windows\setup.log`

By default, `setup.log` generated in the same directory as the response file being used.

A silent installation program does not display a dialog if an error occurs. The status information for the silent installation is recorded (by default) in a file called `setup.log`.

Note:	Do not enter a space between the `f1` or `f2` parameter and its value in double quotation marks.

The log file generated as a result of the Silent Setup procedure is described in the following section.

About the Silent Setup Log File

InstallShield Silent prints installation results into a `Setup.log` file. The default name for the silent setup log file is `Setup.log`, and its default location is the same folder as `Setup.iss.` You can specify a different name and location for your setup log file using the `f2` switch when launching `Setup.exe.` The `Setup.log` file contains three sections:

- In the first section, the first entry, [InstallShield Silent], identifies the version of InstallShield Silent used in the silent setup. The second entry identifies the file as a log file.
- Entries in the second section, [Application], identify the installed application's name and version and the company name.
- The third section, [ResponseResult], contains the result code indicating whether the silent setup has succeeded. The following table describes the integer return values that is assigned to the ResultCode key name.

Silent Setup Result Codes

Result Code	Description
0	Success
1	General error
2	Invalid mode
3	Required data not found in the Setup.iss file
4	Not enough memory
5	File does not exist
6	Cannot write to the response file
7	Unable to write to the uninstallation log file
8	Invalid path to the InstallShield Silent response file
9	Not a valid list type (string or number)
10	Data type is invalid
11	Unknown error during setup
12	Dialog boxes are out of order.
51	Cannot create the specified folder
52	Cannot access the specified file or folder
53	Invalid option selected

Sample Setup Log File

An example of a Setup.log file is as follows:

```
[InstallShield Silent]
Version=v8.100.00
File=Log File
[Application]
Name=Genesys Outbound Contact Manager
Version=7.6
Company=GCTI
Lang=0009
[ResponseResult]
ResultCode=0
```

Manually Define Outbound Configuration Objects/Table Access Object

These topics explain how to create and configure each of the Outbound-related objects in Genesys Administrator:

- Overview
- Outbound-Specific Configuration of Objects
- Table Access Object
- Field Object
- Format Object
- Calling List Object
- Filter Object
- Time Zones
- Treatment Object
- Campaign Object
- Campaign Group Object
- Outbound Schedule Object

Important

- You can also use Configuration Manager or Genesys Administrator to configure these objects.
- Because the `Dependencies` and `Permissions` tabs for these configuration objects do not contain any information that is specific to Outbound Contact, they are not described in this chapter. For information about these tabs, see *Framework Genesys Administrator Help.*

Overview

After you configure and install the component applications for Outbound Contact, you configure additional objects in Genesys Administrator to support and run dialing sessions for campaigns.

The following Framework objects should already exist before you create the Outbound-specific objects: `Tenants, Persons`, and telephony configuration objects (`Switches, Agent Logins, DNs`), `Places, Place Groups` and/or `Agent Groups`.

Outbound Object-Creation Sequence

Genesys recommends that you create the Outbound-specific objects in the following order:

1. `Fields`. Create user-defined fields (if necessary) before you create the `Format` object that will contain them.
2. `Format`. Create a `Format` before you create the `Table Access Point` object to which it will be applied.
3. `Table Access Point`. Create the `Table Access Point` before your create the `Calling List` object to which it will be applied.
4. `Filters`.
5. `Treatments`.
6. `Calling Lists`.
7. `Campaigns`.
8. `Campaign Groups`.

Permissions in Campaign-Related Objects

For users to run or change a dialing session for an outbound campaign, they must be assigned to an `Access Group` object that has special access privileges to all campaign-related configuration objects. `Special access` is the default value.

Verifying or Changing Permissions for an Access Group Object

Start

1. Open each campaign-related object.
2. Select the `Permissions tab`.
3. Make the necessary changes.

   For more information about setting permissions, see Framework Genesys Administrator Help.

End

Outbound-Specific Configuration of Objects

The following objects require specific configurations for Outbound Contact:

- Special DNs
- Place Groups or Agent Groups
- Places

This section discusses the configuration of each of these objects in turn.

Special DNs

The DNs that require Outbound-specific configurations include Communication DNs and Overflow DNs.

Communication DN

Outbound Contact reports on outbound activity through a DN set up as a Communication DN. This type of DN enables communication between Outbound Contact Server (OCS) and Stat Server (which determines what agents are assigned to which Agent or Place Groups, identifies the current Agent State, and provides the statistics for Outbound real-time and historical reporting). This DN type also enables communication between OCS and other parties, using a Communication DN API. It is a virtual DN, meaning that it does not exist in a switch. The name of this DN type can be either text or numeric.

Creating a Communication DN

Start

1. Open the `Switch` object and select the `DNs` tab.
2. Click `New`.
3. On the `Configuration` tab of the `DN`, define the fields as follows:

- Name
- Required; default value is `[DN]`. Specifies the name of the DN. The name that you enter must be unique within the tenant and it must not match any actual directory number in the switch.
- Type
- Required; default value is `[Unknown DN Type]`. Specifies the type of DN. Select the `Communication DN` type from the drop-down menu.

- On the `Options` tab, create a `default` section.
- In the `default` section, create an `outbound_contact_server` option by entering the `Option Name` and

Option Value in the Edit Option dialog boxes.

- Click Save and Close.

End

Overflow DN

A predictive campaign sometimes overdials and, as a result, has outbound calls waiting in the queue. Some contact centers define (by using Virtual DN or routing strategy) business rules to send these outbound calls to another queue. This other queue is commonly known as an overflow queue. There are two types of overflow DNs:

- ACD Queue or Routing Point; used to mark the record as dropped.
- Extension or ACD Position; used to send RequestReleaseCall to T-Server.

Designating an Overflow DN

- To designate an existing DN as an Overflow DN, you must configure an overflow_dn option on the Options tab of a particular DN object.

Start

1. On the Switching tab, select Switches > DNs > <specific DN>.
2. On the Options tab, create a section that is named default.
3. Create the overflow_dn option with a value of true.
4. After you configure the overflow_dn option--depending on your overflow DN type--do the following:
   - For a ACD Position or Extension type DN, create a Place and add this overflow DN into this Place.
   - For an ACD Queue or Routing Point type DN, add your overflow DN to the list of origination DNs of the group(s) that will be used when you run your dialing session for campaigns.

End

Agent and Place Group Objects

Agents, Places, and Agent/Place Groups are configured in Framework; however, Agent Groups and Place Groups still need to be configured specifically for Outbound Contact. The following sections pertain to these Outbound-specific configurations.

Agent or Place Group Object--Advanced Tab Fields

An Agent/Place Group object needs at least one DN (directory number) associated with it in an Origination DN list (see Origination DNs). An ACD (automatic call distribution) queue is a required DN for an Agent/Place Group. Other DN(s) in the list could be VQ (virtual queue) or Routing Point.

Outbound Contact Server (OCS) monitors queues to determine how to pace outbound calls and to determine which group an agent logs into for a particular campaign. OCS monitors the telephony events `EventQueued`, `EventDiverted`, and `EventAbandoned` to determine the number of calls that are waiting in the queue. Stat Server notifies OCS about agent logins, that provides information about which queue or group an agent exists.

Note:	As of OCS version 7.6.101.18, OCS no longer considers the number of inbound calls in a queue when it calculates the outbound dialing pace in the `Progressive with seizing` and the `Predictive with seizing` dialing modes for the Small Group and Advanced Small Group predictive algorithms.

Origination DNs

Found in Genesys Administrator: Agent/Place Group object > `Configuration` tab, `Advanced` section, Origination DNs.

The primary role of an Origination DN is for OCS to monitor it for `EventQueued` and `EventDiverted` so that calls can be properly tracked as they are dialed and moved to agents (or Abandoned). For this reason, if a Route Point is used, it may be necessary to use a Virtual Queue also if the Route Point on the T-Server does not support the correct events.

The origination DNs can be of the following types:

- ACD Queue
- Routing Point
- Virtual Queue
- Service Number

Any DN of these types will be registered in the relevant T-Server on start up or when they are added to the Origination DN (if OCS is already started). OCS will not allow a dialing session/campaign group to be loaded unless the Origination DNs for a Campaign Group are monitored.

Agents log into an ACD queue that has a DN number. OCS recognizes that DN as an identifier for an Agent/Place Group and might use that DN as the point of origin for Outbound calls. OCS might use an alternate DN (for instance, a Routing Point) as the point of origin for Outbound calls.

Note:	Do not share DNs in the Origination DNs List among Agent/Place Groups. Assign each Agent/Place Group to a unique Origination DN.The pacing of outbound calls in the Predictive dialing mode can cause unpredictable results if DNs are shared.

Adding DN(s) to the Origination DNs List

Start

1. In Place Group (on the `Provisioning` tab > `Switching`) or Agent Group object (on the `Provisioning` tab > `Accounts`) > `Configuration` tab > `Advanced/Origination DNs` section, click `Add`.

2. Select (that is, highlight) the DN(s).
3. Click `OK` in the `DN` dialog box.
4. Click `Save and Close`.

End

Place Object

Telesets are assigned to Place objects. A Place might contain more than one DN. Prepare an actual layout of the numbering plan to configure the Places and assign DNs to them.

OCS works only with DNs that are assigned to places. Genesys recommends that you do not assign more than two DNs (`Extension` and `ACD Position`) to one place and that you do not assign DNs from several switches to one place.

Note:	Beginning with release 7.6, OCS does not take into consideration or monitor the configuration state (State Enabled or State Disabled) of the DNs included in Places where agents log in to both an Agent Group and a Place Group that are used for a campaign. OCS considers an agent valid and available if the Place and Person configuration objects indicate State Enabled. In release 7.5, OCS checked the state of the first DN only in the DNs list configured for the Place where an agent logged in.

Table Access Object

A Table Access object defines the relationships among calling list(s), formats, and the database. This object points to a database table of a specified format by providing a Database Access Point.

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

Creating a New Table Access Object

Start

1. In Genesys Administrator, go to Provisioning > Outbound Contact > Table Access.
2. Click New.
3. On the Configuration tab, define the fields as described in the following subsection.
4. Click Save and Close.

End

Table Access Object--Configuration Tab Fields

Name

Required; the default value is [Table Access]. Specifies the name of the Table Access object. The name that you enter must be unique within the tenant.

Note:	You can define the Table Access names for calling lists but the Table Access object for the Do Not Call table has a gsw_donotcall_list fixed name.

Table Type

Required; the default value is [Unknown Table Type]. Specifies the type of Table Access in the database. Select either Calling List or Log Table for the Do Not Call list.

Note:	After you define and save the table type for a Table Access object, you cannot change it.

Description

Optional; the default value is [Blank]. A brief description of the Table Access object.

DB Access Point

Required; the default value is [None]. Specifies the Database Access Point through which the table is accessed. Enter a value by locating an existing `Database Access Point` object.

Format

Required; the default value is `[None]`. Specifies the name of the format to be applied to the table. Enter a value by locating an existing `Format` object.

Note:	After you define and save the format, you cannot change it. The `Do Not Call` table has a fixed structure, and therefore, it does not require format or field configurations.

Database Table

Required; the default value is `[Blank]`. Specifies the name of the table in the database.

Note:	The name of table must conform to SQL language rules, which may vary for each supported database management system (DBMS).

State Enabled

Required; by default, this checkbox is selected. Indicates whether a check box that indicates a customer interaction can be directed to this target. For a more complete description of this check box, see *Framework Genesys Administrator Help*.

Field Object

The Field object defines a field in a Calling List database table. Fields are single pieces of data (for example, a telephone number) within a record.

There are two types of fields in a calling list:

- Genesys mandatory fields--These fields must exist in all Calling List tables. See "Identifying Mandatory Fields" in the "Formats" topic in *Framework Genesys Administrator Help* (Provisioning Your Environment < Outbound Contact Objects > Formats). This topic contains a list of mandatory fields and associated data types, as well as a short description for each field.
- User-defined fields--These custom fields contain business-related data.

Creating a New Field Object

Start

1. In Genesys Administrator, go to Provisioning > Outbound Contact > Fields.
2. Click New.
3. On the Configuration tab, define the fields as described in the following subsection.
4. On the Options tab, define option(s) in the default section.
5. Click Save and Close.

End

Field Object--Configuration Tab Fields

Name

Required; the default value is [Field]. The field name is the name of the column in the database table. This value must be unique within the tenant. Depending on the database management system (DBMS), the field name in this object has to conform to the same rules as the field's name in the database table.

Tenant

Automatically populated by the system.

Data Type

Required; the default value is [Unknown Data Type]. You must select a value other than default for this object. The type of data stored in this field corresponds to a data type supported by the relational database management system.

The **Campaign Data Type Values** table provides information on data type values. The **DATETIME**

formats table provides information on DATETIME formats.

Campaign Data Type Values

Data Type Value	Description
char	Character string with a fixed length
datetime	Date and time1 Data for the `DATETIME` field type is presented in the specific format of the Calling List when it is populated. This format is database dependent.
float	Real number
int2	Integer; the maximum value for this field with this data type that OCS can handle is 4294967295, which is 2^{32}-1.
varchar	String of variable length
[Unknown Data Type]	Default value. You must change this default value and replace it with another value from this table.

DATETIME formats

DBMS	Data type for Date/Time User Field	Required Format	Example
MSSQL	DATETIME	YYYY.MM.DD hh:mm:ss	2003.04.15 11:32:56
Sybase	DATETIME	YYYY.MM.DD hh:mm:ss	2003.04.15 11:32:56
Oracle	DATE	MM/DD/YYYY HH24:MI:SS	/15/2003 23:32:56
Informix	DATETIME YEAR TO SECOND	YYYY-MM-DD hh:mm:ss	2003-04-15 12:32:56
DB2	TIMESTAMP	yyyy-mm-dd-hh.mm.ss.nnnnnn	2003-04-15-12.32.56.123456

Description

Optional; the default value is `[Blank]`. A brief description of the field.

Length

Required for `char` and `varchar` data types only; not used for any other data types. The default value is `[0]`. The length of the field in the database.

Field Type

Required; the default value is `[Unknown Field Type]`. You must select a different value than the default for this object. Outbound business-specific information is stored in this field. The **Field Type Values** table provides information on field type values.

Field Type Values

Field Type Value	Description
Agent	Login ID of the last agent who handled an outbound call associated with this record.
Application	Unique Application identifier (DBID).
Call Time	Time of the actual dialing attempt.
Campaign	Campaign database identification number (DBID) with which this record is associated.
Chain	ID of records that are linked in a chain. All records associated with one customer account have the same chainID. **Important:** When OCS retrieves a record from a database, it converts a string with a `chain_id` value to an integer value. The maximum integer value that OCS supports is 2,147,483,647.
Contact Info	Contact telephone number (home, work, cell) or e-mail address. `Contact Info` in 7.1 was Phone `Number` in 6.5.1.
Contact Info Type	Type of contact information, such as telephone or e-mail address. `Contact Info Type` in 7.1 was `Phone Type` in 6.5.1.
Dialing Result	Result of a dialed call.
From	Earliest time after midnight to dial a call.
Group	Unique Agent Group or Place Group identifier (DBID).
Number in Chain	Priority assigned to a given record in the chain. The smallest number is processed first.
Number of Attempts	Number of dialing attempts.
Record ID	Unique ID number for the current record.
Record Status	Status of the record.
Record Type	Type of record, such as General, Scheduled, Personal CallBack, NoCall
Scheduled Time	Time at which scheduled call should be dialed
Switch ID	Unique Switch identifier (DBID).
Time Zone	Time-zone database identification number (DBID) of the record.
To	Latest time after midnight to dial a call.
Treatments History	History of applied treatments for the given chain (see the `treatment_sched_threshold` option).
User-Defined Field	An optional field that contains custom business data.
[Unknown Field Type]	Default value. You must replace this value with another value from this table.

Default Value

Optional; the default value is `[Blank]`. The default value for the field. All formats that include the field will use this default value. The value should be consistent with the data type of the field. For example, if the data type is `integer`, the default value should be an integer value only. When Genesys Administrator imports records from an ASCII file in which a field is missing or has no value, Genesys Administrator populates the field with this default value--if the `nullable` checkbox is *not* selected. If the field is nullable, Genesys Administrator accepts a blank value. See also Framework Genesys Administrator Help (`Provisioning Your Environment > Outbound Contact Objects > Calling Lists > Overview`).

Primary Key

Optional; the default is `unchecked`. The checkbox indicates whether the field is a primary key in a database table. In Outbound Contact, the only primary key fields are `chain_id` and `chain_n`. For more information about these fields, see *Framework Genesys Administrator Help*.

Unique

Optional; the default is unchecked. The checkbox indicates whether the field value is unique within the table.

Nullable

Optional; default is unchecked. The checkbox indicates whether the field value can be set to NULL.

State Enabled

Required; default is `checked`. Unchecking this checkbox disables the entire format object. Do not change the default.

Editing Fields in the Field Object

`Default` and `Description` are the only fields in the Field object that you can edit after you save the object. To modify the `Default` and `Description` of a field, double-click the Field object. A field cannot be deleted as long as it is associated with at least one format. Field objects are closely related to Format, Table Access, and Calling List objects. Refer to Calling List Objects and Format Objects for more information.

Format Object

Format objects contain fields that form a structure for a database table. Outbound Contact provides two default Format objects: `Default_Outbound_ 6` (for backward compatibility) and `Default_Outbound_70`. By default, both of these objects contain Genesys mandatory fields only.

Creating a New Format Object

Start

1. In Genesys Administrator, go to `Provisioning > Outbound Contact > Formats`.
2. Click `New`.
3. On the `Configuration` tab, define the fields, as in the following subsection.
4. On the `Options` tab, define option(s) in the `default` section.
5. Click `Save and Close`.

End

Format Object--Configuration Tab Fields

Name

Required; the default is value `[Format]`. The name must be unique within the tenant. When the name is specified, it cannot be changed.

Tenant

Automatically populated by the system.

Description

Optional; the default value is `[Blank]`. A brief description of the named format.

State Enabled

Required; default is checked. A check box that indicates a customer interaction can be directed to this target. A more complete description can be found in *Framework Genesys Administrator Help*.

Populating Format with Fields

- To populate a newly created `Format` object (currently empty) and with fields (all mandatory and perhaps some custom fields).

Start

1. After creating and saving a Format object, in the Details panel, double-click the format.
2. In the Navigation panel, click the Fields tab.
3. In the Details panel, click Add.
4. From the Field dialog box, select one or more fields that you want to include in the Format object.

Note:	These fields were created by default as part of the Default_Outbound_6 and Default_Outbound_70 formats.

5. To customize this format with user-defined field(s), in the Field dialog box, click New.
6. Define the field(s) following the procedures in the section Creating a New Field Object.

Note:	Do not remove mandatory fields. Warning: You cannot delete a format if it is associated with a Table Access object or Filter object. You cannot modify existing fields after you associate the format with a Table Access object.

7. Click OK in the Field dialog box.
8. Click Save and Close.

End

Calling List Object

Calling Lists are tables that contain customer contact information, including telephone numbers to be called during an outbound campaign. Calling Lists are also configuration objects in Genesys Administrator. Define a Calling List object after you have created the Format and Table Access objects. If you need user-defined fields in your Calling List, create those fields and add them to the Format object before you create the Table Access object. When a format is associated with a Table Access object, you cannot add or delete fields in that format. You can create Filter and Treatment objects before, while, or after you define the Calling List object and add or remove these objects at any time.

Note:	Only calling lists based on a 7.1 or higher format can be migrated to Outbound Contact 8.x. Warning: Calling lists that are associated with formats that do not contain the switch_id, group_id, and treatments fields (such as those from version 7.0 or earlier), or that contain the phone and phone_type fields instead of the contact_info and contact_info_type fields (respectively) must be migrated to Outbound Contact solution 7.1 first, and then migrated to release 8.x. See the *Genesys Migration Guide* for more details.

Create the Calling List object *before* the Calling List table exists in the relational database. This approach enables Genesys Administrator to create the Calling List table according to the properties of the Calling List configuration object. If the table does not exist yet, OCM displays this message The table [table name] does not exist. Create it? Click Yes to create the Calling List table.

If the Calling List table already exists in the database, the properties of the Calling List object must exactly match the structure of the table (as defined by its Format and Table Access objects) in the database.

Creating a New Calling List Object

Start

1. In Genesys Administrator, select Provisioning > Outbound Contact > Calling List.
2. In the Details pane, click New.
3. Define the field values in the associated tabs, including the Configuration tab, the Options tab, the Campaigns tab, the Format tab, and the Treatments tab.

Note:	For other tabs, see the *Framework Genesys Administrator Help.*

4. Click Save and Close.

End

Calling List Object--Configuration Tab Fields

General Section

Name

Required; the default value is `[Calling List]`. The name of the calling list. This value must be unique within the tenant.

Description

Optional; the default value is `[Blank]`. A brief description of the calling list.

Tenant

Automatically populated by the system.

State Enabled

Required; default is `checked`. A check box indicating that a customer interaction can be directed to this target. You can find a more complete description in *Framework Genesys Administrator Help*.

Calling Time From

The default value is `[8:00:00 AM]`. Not used at the list level, at this time.

Calling Time To

The default value is `[6:00:00 PM]`. Not used at the list level, at this time.

Maximum Attempts

Required; the default value is `[10]`. The maximum number of attempts to call a single record in this Calling List during one campaign. You cannot set this value to zero.

Note:	This parameter is taken into account by Outbound Contact Server when applying treatments only. If the record is being processed again as a Personal or Campaign Callback, this parameter is not used by OCS.

Script

The Script Property in the Campaign, Calling List, and Campaign Group defines the Script object, which contains all of the attributes that are required by Agent Scripting. For more information, see the section, "Attaching Script Information to OCS User Events and Telephony Events", in the Outbound Contact Reference Manual.

Note:	For descriptions of the fields in the Data Base and Query sections of the `Configuration` tab, see *Framework Genesys Administrator Help.*

Format

The Format for the Calling List.

Data Base and Query Sections/Calling List object

These entries are associated with using SQL queries to create a Calling List.

Table Name

The name of the table in the database.

Data Access Point

Required; the default value is `[None]`. Specifies the `Database Access Point` through which the table is accessed. Enter a value by locating an existing `Database Access Point` object.

DB Server

Required; the default value is `[None]`. Specifies the name of the DB Server associated with the database.

SQL Server

Required; the default value is `[None]`. Specifies the name of the SQL Server used for running the SQL query.

Type

Required; the default value is `[None]`. Specifies the type of database: MSSQL, Informix, DB2, Oracle, or Sybase.

DB Name

Required; the default value is `[None]`. Specifies the name of the database in which the table exists.

Login

Required; the default value is `[None]`. Specifies the login name for the database.

Password

Required; the default value is `[None]`. Specifies the password for the database.

Case Conversion

Case conversion method for key names of key-value lists coming from DB Server. This value specifies whether and how a client application converts the field names of a database table when it receives data from DB Server. If you select upper, field names are converted into uppercase; if you select lower, field names are converted into lowercase; and if you select any, field names are not converted. This value corresponds to the `dbcase` option in a configuration file.

Notes:	• This setting does not affect the values of key-value lists coming from DB Server. That is, actual data is presented exactly as it appears in the database tables. • Use the default value (any) unless directed to do otherwise by Genesys Technical Support.

SELECT FROM

Required; the default value is `[None]`. Specifies the table name from which to select records for the Calling List.

WHERE

Required; default value is `[None]`. Specifies the SQL query to identify the records.

ORDER BY

Required; the default value is `[None]`. Specifies how to order the records identified, based on the query.

Calling List Advanced Entries

If you use the Calling List Advanced object instead of the Calling List object, you create lists based on specified Table Access and Filter objects rather than a SQL query.

Table Access

Required; the default value is `[None]`. The Table Access object to which the Calling List refers. The Table Access object defines the format of the Calling List and identifies the DB Access Point used to access the Calling List table. Use the Browse button to locate an existing value.

Log Table Access

The default value is `[None]`. Not used at this time.

Filter

Optional; The default value is `[None]`. The filter to be applied to this calling list. A filter defines which call records within the Calling List table will be dialed by the campaign.

Note:	The Filter field becomes enabled only when the Table Access field is populated.

Calling List Object--Options Tab

Use this tab to define Outbound Calling List--related options.

Calling List Object--Campaigns Tab

This tab lists the Campaigns that include this Calling List.

Calling List Object--Format Tab

On the Calling List object only (not Calling List Advanced object), use this tab to add the fields that define the Format for this calling list.

Calling List Object--Treatments Tab Fields

You can add an existing treatment to the Calling List by clicking the Add button and selecting an available treatment. You can delete an applied treatment by selecting a displayed treatment and clicking Remove. Click Save and Close to the changes. This tab displays the following information for applied treatments.

Name

Name of applied treatment.

State

Indicates whether the Treatment is enabled or disabled. See Treatment Object for more information about setting up treatments for unsuccessful calls.

Changing the Structure of a Calling List

The database administrator can make changes directly to the Calling List table using the database management system (DBMS) tools; however, Genesys Administrator will no longer be able to access this table. Genesys recommends that you create a new Calling List object, which will match the new physical structure of the table, as described on Creating a New Calling List Object.

Fine-Tuning a Calling List

When Genesys Administrator creates a physical table for the Calling List, It also creates appropriate indexes on the table; however, if the structures of the dialing filters require additional indexes, the customer must create them. The customer also must perform tune-up procedures, such as updating index statistics and maintaining transaction logs. To assign a List Weight to a calling list, see List Weight (in Configuration Manager); Share (in Genesys Administrator).

Record Cancellation Requests and Customer ID

In addition to making record cancellation requests according to the record handle and the phone number, they can now be made according to the customer ID, which is currently leveraged by the Do Not Call feature.

Both the Desktop protocol and the 3rd party protocol have been extended to include (optionally) the `GSW_CUSTOMER_ID` attribute in the `RequestRecordCancel` and the `CM_ReqCancelRecord` requests respectively.

With this ability, record cancellation requests can identify sets of records not only by the record handle and the telephone number, but also by the customer ID. If more than one record identifier is included in the same request, the identifiers are prioritized as follows: record handle (highest), telephone, and customer ID (lowest). OCS processes `RequestRecordCancel` by customer ID in the same way that the other types of `RequestRecordCancel` are processed. The only difference is that the records that are subject to cancellation are determined by the value of the customer ID field in the Calling List table.

Configuring Objects to Identify Records by Customer ID for Record Cancellation Requests

- To configure object to enable identification of record cancellation requests by customer ID.

Note:	If you have already use the Customer ID for Do Not Call requests, you do not need to configure anything else.

Start

1. Create a new user-defined field object. On the `Configuration` tab, define the fields as follows:
   - `Name` = `<user-specific name>`
   - `Data Type` = `varchar`
   - `Length` = `64`
   - `Field Type` = `User-Defined`

   Also select the `Nullable` and `State Enabled` options.
2. On the `Options` tab for the `Field` object, assign the `send_attribute` to it by adding a `default` section. On the `Option` tab, define the fields as follows:
   - `Option Name` = `send_attribute`
   - `Option Value` =`GSW_CUSTOMER_ID`
3. Designate the new user-defined field as the `customer_id` option in the OCS Application object. In Genesys Administrator > `Provisioning > Environment > Applications > Outbound folder >`

`OCS application > Options > OCServer` section, create and define the `customer_id option`. Use the name of the new user-defined field as the value of `customer_id`.

- `Option Name` = `customer_id`
- `Option Value` = `<name of new user-defined field>`

4. Add the new field (defined in Step 1) to a new `Format` object.
   In `Genesys Administrator > Provisioning > <Tenant> > Format` object, create a new format for a `Calling List` table that will include the new user-defined field.
5. Create a `Calling List` object with the new format. In `Genesys Administrator > Provisioning > <Tenant> > Calling Lists`, configure a `Calling List Advanced` object with the following:
   In the `Configuration` tab, specify the following:
   - Table Access: `<New Calling List>`

   This is a new Calling List, formatted with the `customer_id` field.

End

Customized Stored Procedures for Calling List and Campaign Objects

OCS uses stored procedures to get the values of different counters from the calling list database table (for example, the number of retrieved records), and to enable end users to customize them by defining or modifying the procedures code.

Customizing your stored procedures enables you to fine-tune a stored procedure for the following purposes:

- To adjust statistical calculations for pre-defined statistical counters.
- To define up to five custom database-related counters.

Starting in release 8.1.2, you can develop your own customized stored procedures by completing the following steps:

- Determine if the default procedure can be extended or should be completely rewritten.
- Create and debug code for the customized stored procedure, by using third-party DBMS tools.
- Store the body of the customized stored procedure in the Genesys configuration, by using Genesys Administrator.
- Configure OCS to completely replace the standard procedure code (using the `instead` option) or extend the code of the standard procedure with the customized procedure (using the `before` or `after` options).

Although you are limited by the predefined names, and input and output parameters in the custom procedure, there are no limitations within the body of the custom procedure.

Configuring Custom Procedures

Custom procedures are configured in Genesys Administrator, in the Annex (Options) section of Campaign and Calling List objects. The section names for OCS options are standard and the calling list level options take precedence over the Campaign level options.

Reporting procedures are defined by configuring this pair of options: `report-procedure-body` and `report-procedure-location`.

Genesys Administrator has a simple, built-in text editor that enables you to define the body of the customized procedure. After the code has been defined and the location object for the code determined, Genesys Administrator stores the custom procedure code in the option value field as a binary string.

Define custom procedures in Genesys Administrator by navigating to one of the following locations in the interface:

- `Provisioning > Outbound Contact > Campaigns > Campaign object name > Configuration tab > Stored Procedures`
- `Provisioning > Outbound Contact > Calling Lists > Calling list object name > Configuration > Stored Procedures`

Custom Reporting Procedures

Each DBMS has its own requirements for stored procedures. The following table describes the requirements for customization of stored reporting procedures for each DBMS, whether you are adding custom counters to existing procedures, or creating completely new user-defined stored procedures.

Customizing Reporting Procedures

If Using Before/After for Custom Counters	If Using for Reporting Procedure, Use this Template for Procedure Body
MS SQL, Sybase	

Use variables @CustomCounter01Num - @CustomCounter05Num predefined as NUMERIC in procedure body for counters. Then use variables @CustomCounter01 - @CustomCounter05 defined as VARCHAR(20) to return values.
For example:
SELECT @CustomCounter01Num = (SELECT COUNT(distinct record_id) FROM $list_tbl_name WHERE $dial_filter_where)
SELECT @CustomCounter01 = CONVERT(VARCHAR(20), ISNULL(@CustomCounter01Num, 0))
	CREATE PROCEDURE $sr_proc_name(
	@CTime VARCHAR(20),
	@ReadyCount VARCHAR(20) OUTPUT,
	@RetrievedCount VARCHAR(20) OUTPUT,
	@FinalCount VARCHAR(20) OUTPUT,
	@TotalCount VARCHAR(20) OUTPUT,
	@TotalRecsCount VARCHAR(20) OUTPUT,
	@ReadyRecsCount VARCHAR(20) OUTPUT,
	@CustomCounter01 VARCHAR(20) OUTPUT,
	@CustomCounter02 VARCHAR(20) OUTPUT,
	@CustomCounter03 VARCHAR(20) OUTPUT,
	@CustomCounter04 VARCHAR(20) OUTPUT,
	@CustomCounter05 VARCHAR(20) OUTPUT)
	AS
	BEGIN
	-- TODO: Add custom code here
	END

Oracle	
Use variables v_CustomCounter01 -	CREATE OR REPLACE PROCEDURE $sr_proc_name(

v_CustomCounter05 predefined as NUMBER in procedure body for counters. Then use variables p_CustomCounter01 - p_CustomCounter05 defined as VARCHAR2 to return values.
For example:
SELECT COUNT(*) INTO v_CustomCounter01 FROM (select DISTINCT record_id FROM $list_tbl_name WHERE $dial_filter_where);
p_CustomCounter01 := TO_CHAR(NVL(v_CustomCounter01, 0));
	p_CTime VARCHAR2,
	p_ReadyCount OUT VARCHAR2,
	p_RetrievedCount OUT VARCHAR2,
	p_FinalCount OUT VARCHAR2,
	p_TotalCount OUT VARCHAR2,
	p_TotalRecsCount OUT VARCHAR2,
	p_ReadyRecsCount OUT VARCHAR2,
	p_CustomCounter01 OUT VARCHAR2,
	p_CustomCounter02 OUT VARCHAR2,
	p_CustomCounter03 OUT VARCHAR2,
	p_CustomCounter04 OUT VARCHAR2,
	p_CustomCounter05 OUT VARCHAR2)
	AS
	BEGIN
	-- TODO: Add custom code here
	END;
	END $sr_proc_name;

DB2	

Use variables iCustomCounter01 - iCustomCounter05 predefined as INTEGER in procedure body for counters. Then use variables CustomCounter01 - CustomCounter05 defined as CHAR(32) to return values.
For example:
SELECT COUNT(DISTINCT record_id) INTO iCustomCounter01 FROM $list_tbl_name WHERE $dial_filter_where;
SET CustomCounter01= CHAR(VALUE(iCustomCounter01, 0));
	CREATE PROCEDURE $sr_proc_name(
	IN CTime INTEGER,
	OUT ReadyCount CHAR(32),
	OUT RetrievedCount CHAR(32),
	OUT FinalCount CHAR(32),
	OUT TotalCount CHAR(32),
	OUT TotalRecsCount CHAR(32),
	OUT ReadyRecsCount CHAR(32),
	OUT CustomCounter01 CHAR(32),
	OUT CustomCounter02 CHAR(32),
	OUT CustomCounter03 CHAR(32),
	OUT CustomCounter04 CHAR(32),
	OUT CustomCounter05 CHAR(32))
	LANGUAGE SQL
	BEGIN
	-- TODO: Add custom code here
	END

Informix	
Use variables v_CustomCounter01 - v_CustomCounter05 predefined as INT in the procedure body for counters. Procedure returns these variables in the RETURN statement. For example: SELECT COUNT(DISTINCT record_id) INTO v_CustomCounter01 FROM $list_tbl_name WHERE $dial_filter_where;	CREATE PROCEDURE $sr_proc_name(p_CTime CHAR(32)) RETURNING INT, INT, INT, INT, INT, INT, INT, INT, INT, INT, INT; -- TODO: Add custom code here RETURN v_ReadyCount, v_RetrievedCount, v_FinalCount, v_TotalCount, v_TotalRecsCount, v_ReadyRecsCount, v_CustomCounter01, v_CustomCounter02, v_CustomCounter03, v_CustomCounter04, v_CustomCounter05; END PROCEDURE
PostgreSQL	
Use variables v_custom01 - v_custom05 declared as INTEGER in procedure body for counters. Then use variables p_CustomCounter01 - p_CustomCounter05 declared as VARCHAR(32) to return values.	CREATE OR REPLACE FUNCTION $sr_proc_name(c_time numeric(18,0), OUT p_ReadyCount varchar(32), OUT p_RetrievedCount varchar(32), OUT p_FinalCount varchar(32),

For example:
`DECLARE v_custom01 INTEGER := 0;`
`SELECT COUNT(DISTINCT record_id) INTO v_custom01 FROM $list_tbl_name WHERE ($dial_filter_where);`
`p_CustomCounter01 := CAST(COALESCE(v_custom01, 0) AS VARCHAR);`
	`OUT p_TotalCount varchar(32),`
	`OUT p_TotalRecsCount varchar(32),`
	`OUT p_ReadyRecsCount varchar(32),`
	`OUT p_CustomCounter01 varchar(32),`
	`OUT p_CustomCounter02 varchar(32),`
	`OUT p_CustomCounter03 varchar(32),`
	`OUT p_CustomCounter04 varchar(32),`
	`OUT p_CustomCounter05 varchar(32))`
	`AS $$`
	`--TODO: Declare variables here`
	`BEGIN`
	`--TODO: Add custom code here`
	`RETURN;`
	`END;`
	`$$ LANGUAGE plpgsql;`

Customized Stored Procedures Macro Expressions

To simplify the creation of customized procedures, OCS supports the use of macro expressions in custom procedures. OCS substitutes this macro expression for the actual values when the SQL code for the custom procedure is generated, depending on the calling list for which the code generation is executed. By using macro expressions, OCS enables you to define a custom procedures template on a Campaign level that can be generated for each calling list that is configured under the Campaign. See the **Macro Expressions for Custom Stored Procedures** table for macro expressions for custom stored procedures. Refer also to the **OCS Supported Macro Expressions** table that also contains other macro expressions supported by OCS.

Macro Expressions for Custom Stored Procedures

Parameter name	To be submitted for:	Usage example in the custom procedure body
`$list_tbl_name`	Name of the calling list table	`UPDATE $list_tbl_name SET ...`
`$dial_filter_where`	WHERE clause of the dialing filter	`UPDATE $list_tbl_name SET ..." WHERE $dial_filter_where`
`$dial_filter_order_by`	ORDER BY clause of the dialing filter	`SELECT TOP 10 chain_id FROM $list_tbl_name ORDER BY $dial_filter_order_by`
`$list_dbid`	DBID of the Calling List	
`$camp_dbid`	DBID of the Campaign	
`$group_dbid`	DBID of the Group	
`$camp_group_dbid`	DBID of the Campaign Group (session)	

Monitoring Dynamic Modification to Custom Procedures

OCS monitors dynamic modifications to the dialing filter and recreates customized procedures if the filter is updated, if the procedure contains the `$dial_filter_where` and/or `$dial_filter_order_by` macro expression.

OCS also monitors the options that define the body and location of the customized procedure and regenerates these procedures if these options are modified.

Implementation of Custom Procedures

There are three possible configurations:

1. Before--A call to the custom code is executed before any code execution in the hosting procedure, immediately following the BEGIN clause.
2. After--A call to the custom code is executed after all of the code in the hosting procedure is executed.
3. Instead--The body of the standard stored code is fully replaced with the custom procedure code.

Logging and Debugging

To assist in debugging custom procedures, OCS logs all SQL code for custom procedures when it is created and any errors in the custom procedures when they are created.

OCS logs the SQL code for procedures by using the `DEBUG` log level. Genesys recommends that any errors that occur when procedures are created be logged with the `TRACE` log message severity to ensure the proper alarms on Message Server are configured.

Omitting Verification of the Campaign DBID for Callbacks and Rescheduled Records Retrieval

Stored procedures take the Campaign DBID (and optionally the Group DBID as configured in the OCS `callback-observe-group` option) into account when any record type, other than General, is retrieved.

Use the `callback-observe-campaign` option to control whether Campaign DBID is considered by the OCS record-retrieval mechanism.

Filter Object

Filters specify the criteria for data selection from a database table. Outbound filters are applied to Calling Lists. See the `Filter` field under "Calling List Object".

Creating a New Filter Object

Start

1. In Genesys Administrator, select `Provisioning > Outbound Contact > Filters`.
2. In the Details pane, select `New`.
3. Define the fields in the `Configuration` tab.
4. Define options in the `Options` tab.
5. Click `Save and Close`.

End

Filter Object Configuration Tab Fields

Name

Required; the default value is `[Filter]`. The name of the filter. This value must be unique within the tenant.

Tenant

Automatically populated by the system.

Description

Optional; the default is `[Blank]`. A brief description of the filter.

Format

Required; the default value is `[None]`. The format to which this filter is applied. When it is specified, it cannot be changed. You assign a Filter object to a Calling List object with the same format. Use the Browse button to locate an existing value.

State Enabled

Required; the default value is `checked`. A check box that indicates that a customer interaction can be directed to this target. A more complete description is in *Framework Genesys Administrator Help*.

Filter Object Options Tab Fields

Two options on the Options tab define a Dialing Filter: criteria and order_by. Create a section called default on the Filter object's Options tab; then add the options criteria (see the procedure Configuring the criteria Option below) and order_by. (see the procedure Configuring the order by Option to Sort Records).

Configuring the criteria Option

The criteria option defines the where clause of a SQL statement. It defines the restricting conditions for the records to be dialed by a campaign. The syntax for values in the criteria option is SQL syntax.

Note:	The Outbound Contact Wizard was not updated for release 8.x.

Start

1. Double-click the filter and select the Options tab.
2. Click Add.
3. In the New Option dialog box, enter default in the Section field and criteria in the Name field.
4. Add a SQL statement in the Value field.
   For example, to restrict the dialing records to contact only platinum credit card holders, type card_type='platinum' in the Value field.

Note:	When using Genesys mandatory fields in SQL statements, the value must be in enum format. Refer to "Genesys Enumeration Tables" in the "Communication Protocols" chapter of the *Outbound Contact Reference Manual*. (When creating viewing filters in Genesys Administrator, this only applies on the Expert Mode tab.)

5. Click OK to save and close the option to this Filter object.

End

Configuring the order_by Option to Sort Records

The order_by option defines the sort order of the records that meet the restrictions defined in the criteria option. **Start**

1. Click Add.
2. In the New Option dialog box, enter default in the Section field and order_by in the Name field.
3. In the Value field, type the name of the field by which you want to sort.
   For example, to order records by telephone number, enter phone in the Value field.
4. Click OK to save and close the option to this Filter object.

Note:	If you use a filter on a large Calling List with more

	than 1,000 records, Genesys recommends that the Administrator create an index using DBMS tools. This dramatically improves data-retrieval performance.

End

Defining a Dialing Filter that Exceeds 255 Characters for Configuration Server 7.6.x and Earlier

If you are using Configuration Server 7.6.x or earlier and the dialing filter exceeds 255 characters, you must divide the long string into several shorter strings. To do so, use configuration options with the same name and add sequential numbers at the end of the option's name. OCS, OCM, and Genesys Administrator builds a dialing filter by concatenating the values of these options.

Note:	Configuration Server 8.x supports configuring options with values greater than 255 characters. If you are using Configuration Server 8.0 or higher, you can configure a single option with a dialing filter that exceeds 255 characters.

Sample Configuration:

In the following example, the values of these configuration options:

- `criteria`
- `criteria1`
- `criteria2`

are the first, second, and third parts, respectively, of the dialing filter's `WHERE` clause.

In the following example, the values of these configuration options:

- `order_by`
- `order_by1`
- `order_by2`

are the first, second, and third parts, respectively, of the dialing filter `ORDER BY` clause.

Note:	When parts of the filter are concatenated, a single space is added between the parts. As a rule, when configuring these options, divide your SQL clause only at points where insertion of the white-space is permitted by the SQL standard, otherwise the result of concatenation will yield an invalid SQL clause.

Example 1

```
criteria = phone like '415%' AND tz_dbid IN (116, 118,
criteria1 = 122) AND customer_priority IN (10, 9, 4,
criteria2 = 3, 2, 18) AND customer_type = 1
order_by = phone_type ASC, tz_dbid DESC,
order_by1 = customer_priority ASC, customer_policy_n DESC,
order_by2 = customer_id DESC
```

Resulting Filter 1

The options configured as shown above result in the following dialing filter: WHERE phone like `'415%' AND tz_dbid IN (116, 118, 122) AND customer_priority IN (10, 9, 4, 3, 2, 18) AND customer_type = 1 ORDER BY phone_type ASC, tz_dbid DESC, customer_priority ASC, customer_policy_n DESC, customer_id DESC`

Filters that Break a Chain of Records

You can write a filter so that it breaks a chain of records into two logical chains: one chain that satisfies the filtering expression and a second chain that does not. For example:

List:

```
Record 1:phone = '111', phone_type = 1, chain_id = 1, chain_n = 0
Record 2:phone = '222', phone_type = 2, chain_id = 1, chain_n = 1
Record 3:phone = '333', phone_type = 1, chain_id = 1, chain_n = 2
Record 4:phone = '444', phone_type = 3, chain_id = 1, chain_n = 3
```

Example: The following filter `where phone_type = 1`
will break the given chain into two logical chains:

- Logical chain 1 includes records 1 and 3, which satisfy the filter condition.
- Logical chain 2 includes records 2 and 4, which do not satisfy the filter condition.

You should avoid such filtering conditions, if possible.

Record Cancel for Broken Chain

Under particular conditions, records with the same `chain_id` are not all canceled by a cancel by phone request (RequestRecordCancel with `GSW_PHONE` attribute).

This behavior occurs when four conditions coexist:

- The `RequestRecordCancel` has the `All Chain` attribute, which means to update the whole record chain, not just the single record,

but

- a filter applied to the calling list breaks a single chain into two logical chains

while

- the dialing session for a campaign group with a campaign (to which the calling list belongs) is loaded or running,

and

- a phone number specified in the `GSW_PHONE` attribute is filtered out.

The records that are *not* filtered out are also not canceled even though the chain attribute is `All Chain`.

Take, for example, these conditions applied to the list that follows:

- Filter: `where phone_type = 1`
- RequestRecordCancel: Phone = `'222'`
- ChainAttr = `'AllChain'`

List:

```
Record 1: phone = '111', phone_type = 1, chain_id = 1, chain_n = 0
Record 2: phone = '222', phone_type = 2, chain_id = 1, chain_n = 1
Record 3: phone = '333', phone_type = 1, chain_id = 1, chain_n = 2
Record 4: phone = '444', phone_type = 3, chain_id = 1, chain_n = 3
```

Records 1 and 3 meet the filter condition `where phone_type = 1`. Records 1 and 3 compose logical chain 1. These two records (with `chain_n = 0` and `chain_n = 2`) are retrieved from the database and dialed.

Record 2 (with `chain_n = 1`) and record 4 (with `chain_n = 3`) compose the second logical chain. Record 2 is subject to the RequestRecordCancel where `Phone = '222'`. Because RequestRecordCancel has the `All Chain` attribute (ChainAttr = `'All Chain'`), both records 2 and 4 are canceled.

Full Chain Update

The full chain update feature applies to configurations in which chains of records are separated by the dialing filter and, therefore, are partially processed by OCS. After OCS finishes processing this type of chain, by default it marks the records that were filtered out by the dialing filter (such as Record 2 in the following example) as `record_status = Ready`.

Ordinarily, if a dialing filter is changed and the chain processing is resumed, such an update will cause repetitive dialing of other records that belong to the same chain. This could lead to contacting the same customer again within that dialing period. Using the full chain update feature prevents this unwanted behavior of OCS.

The following provides an example of the configuration when a chain of records is separated by the dialing filter:

Chain:

- Record 1: Phone = 4155551212, phone type = Home Phone, Chain ID = 1, Chain N = 0
- Record 2: Phone = 6505551212, phone type = Direct Business Phone, Chain ID = 1, Chain N = 1

Dialing Filter Applied:

criteria = '`phone_type` = 1'

Outbound Contact provides the option `update_all_records` to enable control over how OCS updates the status of records after the chain processing has been completed, in the following way:

- If the value for the `update_all_records` option is set to `true`: OCS updates all records in the chain, regardless of the dialing filter applied. All records outside the parameters of the dialing filter are updated with the same status that the chain receives after OCS has finished processing it.
- If the value for this option is set to `false` (default): Records that were filtered out by the dialing filter are marked as `record_status = Ready`.

Note:	This option can be configured both on the Application- and individual Calling List-levels. The Calling List-level option takes precedence over the Application-level.

When in use, Full Chain Update ensures that all records in the chain are updated with the same status as the chain itself, regardless of the chain's currently applied dialing filter, which might separate the chain by filtering out some records. For example, if the record in the chain has been cancelled with the flag 'AllChain', then all records, regardless of the dialing filter, will be marked as `Cancelled`.

Time Zones

Outbound Contact uses time zones in call records to determine the contact's time zone. Genesys Administrator populates the tz_dbid field with the international three-letter abbreviation for the time zone parameter when it imports or exports a calling list. Call time, dial schedule time, and valid dial time (dial from and dial till) are based on the record's time-zone. For more information about the time zone abbreviations see Framework Genesys Administrator Help.

Note:	If Daylight Savings Time (DST) is configured for time zones located below the equator using the Current Year or Fixed Date (Local) properties, define both the Start Date and End Date in the DST definition as the current year and make the Start Date later than the End Date.

Outbound Contact dynamically updates time changes from winter to summer and summer to winter. The default set of Time Zones created during Configuration Server installation is located in the Time Zones folder under the Environment (for a multi-tenant environment) or under the Resources folder (for a single tenant).

Creating a Custom Time Zone for a Tenant

- To create a custom Time Zone for a Tenant, if necessary.

Start

1. In Genesys Administrator, select Provisioning > Environment > Time Zones.
2. Click New.
3. Define the fields in the Configuration tab. For more information, see *Framework Genesys Administrator Help.*
4. Click Save and Close.

End

Time Zones Object--Configuration Tab Fields

Name

Required; default is [Time Zone]. The name of a Time Zone object. This value must be unique within the tenant.

Tenant

Automatically populated by the system.

Description

Optional; the default value is `[Blank]`. A brief description of the `Time Zone` object.

Time Zone

Required; the default value is `[GMT]`. Greenwich Mean Time (GMT) plus the amount of time (in hours and minutes) to add to that standard.

DST Select

Optional; A drop-down list to select Daylight Savings Time definition (GMT) to indicate that the time zone observes Daylight Savings Time (DST) or Current Year or Fixed Date (local) to indicate that DST is not observed.

Netscape (in Specific Time Zone section)

Required; Enter the time zone as it appears in Netscape Navigator.

Microsoft Internet Explorer (in Specific Time Zone section)

Required; Enter the time zone as it appears in Microsoft Internet Explorer.

State Enabled

Required; default is `checked`. A check box indicating whether or not a customer interaction can be directed to this target. A more complete description can be found in *Framework Configuration Manager Help*.

Time Zones and Time-Related Calculations

OCS uses the operating system time zone information for time-related calculations. For example, when a record is rescheduled by the agent or by using a treatment. On UNIX platforms, OCS recognizes time zone information in POSIX format only. OCS does not support any other format.

Treatment Object

Treatment objects are assigned to Calling List objects. Treatment objects define what OCS should do with a call that does not reach the intended party. A treatment, for example, could be that Outbound Contact redials a number that returns a `Busy` call result. If no treatments are assigned to a call result, OCS changes the record status to `Updated`.

You can add or remove treatments from a Calling List object while a dialing session for a campaign is running. However, when a treatment is in progress, it cannot be interrupted. Changes made to that treatment will be applied to the *next* record that receives the call result that prompts the treatment.

Creating and Applying Treatments

Start

1. In Genesys Administrator, select `Provisioning > Outbound Contact > Treatments`.
2. Click `New`.
3. Define the fields in the `Configuration` tab.
4. Click `Save and Close`.

End

Treatment Object--Configuration Tab Fields

Name

Required; the default value is `[Treatment]`. Type the name of the treatment. It must be unique within the tenant.

Tenant

Automatically populated by the system.

Description

Optional; the default is `[Blank]`. A brief description of the selected treatment.

Call Result

Required; the default value is `[Unknown Call Result]`. Select a value (the result of a dialing attempt) for this object. The following table provides information about call result values.

Call Result Values

Call Result Value	Description
[Unknown Call Result]	Default value. You must change this default value

	and replace it with another value from this table. OCS does not apply treatments to the Unknown call result.
Abandoned	Call dropped and will not be redialed.
Agent CallBack Error	OCS generates this call result when a call record is rescheduled according to a personal callback request from the desktop application, but OCS cannot find the designated agent to receive the callback when the scheduled time arrives.See the "Communication Protocols" chapter in the Outbound Contact Reference Manual for details on the personal callback request.
All Trunks Busy	No free trunks on the switch for dialing a call.
Answer	Customer was reached at the dialed phone number.
Answering Machine Detected	Answering machine was detected at the dialed phone number.
Bridged	Reserved for future use.
Busy	Dialed phone number was busy.
Call Drop Error	Reserved for future use.
Cancel Record	Record has been marked with `Cancel` by the agent desktop application or a third-party application.
Cleared	Reserved for future use.
Conferenced	Reserved for future use.
Consult	Reserved for future use.
Converse-On	Reserved for future use.
Covered	Reserved for future use.
Deafened	Reserved for future use.
Dial Error	Hardware error from a Dialogic board or a call progress detection (CPD) board on the switch.
Do Not Call	Record has been marked with `DoNotCall` by the agent desktop application or a third-party application.
Dropped	Call dropped by dialer after call dialed. Call dropped if/when the `call_wait_in_queue_timeout` option expires.
Dropped on No Answer	Reserved for future use.
Fax Detected	Fax machine was detected at the dialed phone number.
Forwarded	Reserved for future use.
General Error	General error occurs when a call is not completed, possibly caused by an invalid telephone number in the record or a wrong number according to the switch.
Held	Reserved for future use.

No Answer	Ring without answer at destination.
No Dial Tone	Absence of dial tone based on an error returned by the Dialogic board or the call progress detection (CPD) board on the switch.
No Established Detected	Reserved for future use.
No Port Available	No port is available to place call.
No Progress	Reserved for future use.
No RingBack Tone	Reserved for future use.
NU Tone	A special Public Switched Telephone Network (PSTN) code valid only in Europe.
OK	OCS treats the call result `OK` as undefined, therefore, it disregards any treatments created for this call result.
Overflowed	Reserved for future use.
Pager Detected	Pager was reached at the dialed phone number.
Pickedup	Reserved for future use.
Queue Full	Reserved for future use.
Redirected	Reserved for future use.
RemoteRelease	Reserved for future use.
Silence	Call was dialed, but there was no call progress detection (CPD).
SIT Detected	Any type of network tone.
SIT IC (Intercept)	Only applies if the network supports this specific standard information tone (SIT). Check with the switch vendor for confirmation.
SIT Invalid Number	Only applies if the network supports this specific SIT. Check with the switch vendor for confirmation.
SIT NC (No Circuit)	Only applies if the network supports this specific SIT. Check with the switch vendor for confirmation.
SIT RO (Reorder)	Only applies if the network supports this specific SIT. Check with the switch vendor for confirmation.
SIT Unknown Call State	Only applies if the network supports this specific SIT. Check with the switch vendor for confirmation.
SIT VC (Vacant Code)	Only applies if the network supports this specific SIT. Check with the switch vendor for confirmation.
Stale	Call result is marked as `stale` if the following timer has expired: `stale_clean_timeout`OCS will also mark the call result as `stale` for call records that have not received a `RecordProcessed` request from the desktop application when a dialing session/campaign group is being unloaded.
Switch Error	No dial tone received.
System Error	Dialing software error from the Dialogic driver or CPD from the switch.
Transfer Error	Dialer has a problem transferring calls based on

	call action.
Transferred	Reserved for future use.
Wrong Number	Call is answered but the desired person(s) could not be reached at this number; this call result is sent by the agent desktop application and not detected by the dialer.
Wrong Party	Call is answered but by a wrong party; this call result is sent by the desktop application and not detected by the dialer.

Apply to Record

Required; default [Unknown Action]. Select a value from the drop-down list. The action to apply to a Calling List record based on the call result. Applies to all call results in the Call Results Values table except for call result `Answer`. The following table provides information on Apply to Record values.

Apply to Record Values (Unanswered Call Results)

Apply to Record Value	Description
Assign to Group	Used for the `Agent CallBack Error` call result. The call record is treated as a scheduled call of type Campaign. The record type is set as `CampaignCallBack`. The call record is sent to one of the desktops in the Campaign Group. The desktop must be programmed with a communication protocol to use this feature. See the "Communication Protocols" chapter in the Outbound Contact Reference Manual.
Execute SQL Statement	Used to execute pre-configured SQL statements when certain call results are received. There is no limitation on the type or complexity of the SQL statement that can be configured. Macro expressions can also be used to form the SQL statement. To execute this statement, OCS connects to the Calling List table from which the associated dialing record originated.
Mark as Agent Error	Used for the `Agent CallBack Error` call result. The record status is set to `Missed Callback`. See also the `predictive_callback` option in this guide. **Note:** Although you can set this value to other negative call results in Configuration Manager, you should only use it for `Agent CallBack Error` call result.
Next in chain	Used to set the total number of times that the next-in-chain treatment cycle occurs, and set the time period to wait before beginning the cycle again.
Next in chain after	Used to set the interval until the next record in chain is dialed, and until the next pass through the chain, after the last record in the chain has been dialed.
Next in chain at specified date	Used to set the date/time to dial the next record in chain.

No Treatment	No treatment will be applied.
Redial	Redial number within a specified number of minutes (Interval) for a specified number of times (Cycle Attempt).
Retry at specified date	Record will be redialed at specified times within the time interval specified in the Range property of the treatment.
Retry in	Retry once after the specified number of minutes (Interval).
Send HTTP Request	Sends an HTTP request with the JSON body to a remote Web/Application Server.
Update all records in chain	No more dial attempts are made for the chain and all records in the chain are marked as `Updated.`
[Unknown Action]	Default value. You must change this default value and replace it with another value from this table.

Apply to Call

Optional; applicable only when call result is Answer, Answering Machine Detected, or Fax Detected; the default value is `[Unknown Action Code]`. Allows an alternate path when a dialing attempt is answered or reaches an answering machine or fax machine. Select a value from the drop-down menu. The following table provides information about Apply to Call values.

Apply to Call Values (Answering Machine/Fax Results)

Apply to Call Value	Description
Connect	Call connected to the DN defined as the Destination DN.
Drop	Drop = Unknown. Call is dropped or disconnected.
Mute Transfer	Reserved for future use.
Play a Message	Reserved for future use.
Route	Reserved for future use.
Send a fax	Reserved for future use.
Send a page	Reserved for future use.
Send an e-mail	Reserved for future use.
Transfer	Call transferred to the DN defined as the Destination DN.
[Unknown Action Code]	[Unknown Action Code] = Dropped. Default value. You must change this default value and replace it with another value from this table.

Destination DN

Required for the `Apply to Call` actions, `Connect` and `Transfer`; default is `[None]`. The DN to which the call will be connected or transferred. Browse for an existing DN. If a `Destination DN` is not supplied, OCS will ignore the treatment.

Number in sequence

Required; the default value is [0]. Assign the value 1 for a stand-alone treatment or for the first treatment of a treatment sequence. In a treatment sequence, each treatment contains a unique number that determines the order in which each treatment is applied to the same call result.

Cycle Attempt

Required to be other than 0 when Apply to Record is set to Next in chain, Next in chain after, Next in chain at specified date, Redial; the default value is [0]. The maximum number of consecutive attempts to execute the treatment on the record.

Interval (minutes)

Required when Apply to Record is set to Next in chain, Next in chain after, Redial, or Retry in; default is [0]. A time interval, in minutes, that OCS waits between the first dialing attempt and the first treatment attempt.

Increment (minutes)

Required when Apply to Record is set to Next in chain, Next in chain after, Retry in, Redial; the default value is [0]. Specifies the number of minutes added to the previous redial time interval.

Date

Required if the Apply to Record action is set to Next in chain at specified date or Retry at specified date; the default value is [current date]. The date when another treatment attempt will be performed. Select a calendar date from the drop-down list.

Time

Required if the Apply to Record action is set to Next in chain at specified date or Retry at specified date; the default value is [current time]. The time of day that another treatment attempt will be performed. Select a time from the drop-down list.

Range

This is reserved for future use.

Treatment Sequences

Note the following:

- You can define treatment sequences (sometimes called linked treatments) for the same call result by using the Number in sequence field on the Configuration tab of the Treatment object. The first treatment in the sequence must have a value of 1 in the Number in sequence field. The values for this field must be numbered sequentially (1,2, 3, *etc.*). The sequence will not work if a treatment sequence is broken, for example, numbered 1, 2, and 4.
- You can create two Busy treatments and specify the order (sequence) in which to apply them. For

example, create one Busy treatment that has an action of Redial and the Number in sequence field set to 1; create a second Busy treatment that has an action of Retry in and the Number in sequence field set to 2; then assign both Busy treatments to the same Calling List.

- You can also use the sequence of treatments in combination with cycling through the chain. To do this, you need to place one of the treatments of the type Next-In-Chain, Next-In-Chain After, or Next-In-Chain at Specified Date, at the end of the sequence of treatments.

Properly configuring one of these treatments to be the last in the sequence will result in the following: When the chain is completed, OCS will jump from the last record to the first record in the chain to begin the cycle again.

Campaign Object

A Campaign object contains information about the Calling List(s) and, if needed, defines campaign-level options.

Creating a New Campaign Object

Start

1. In Genesys Administrator, select Provisioning > Outbound Contact > Campaigns.
2. Click New.
3. Define the fields in the Configuration tab.
4. In the Calling Lists section, click Add to add calling lists to the campaign.
5. Click Save and Close.

End

Campaign Object--Configuration Tab

Name

Required; the default value is [Campaign]. Type or choose the name of the campaign. This name must be unique within the tenant.

Tenant

Automatically populated by the system.

Script

Optional; the default value is [None]. Defines the Script object that contains all of the attributes that are required by Agent Scripting. For more information, see the "Attaching Script Information to OCS User Events and Telephony Events" section in the *Outbound Contact Reference Manual* for more information.

State Enabled

Required; default is checked. A check box that indicates that customer interaction can be directed to this target. A more complete description can be found in Framework Genesys Administrator Help.

Calling List

Required; the default value is [None]. The names of Calling Lists that will be dialed during this Campaign. A Campaign can have zero or more Calling Lists. Select from defined Calling Lists.

List Weight (in Configuration Manager); Share (in Genesys Administrator)

Required; default is `[10]`. This property is applicable for a campaign with more than one active Calling List. It is the percentage of call records to be retrieved from a Calling List for a Campaign. Specify what percentage of that volume will be drawn from each calling list. See the "Understanding Weights" topic in the *Framework Genesys Administrator Help*.

Note:	The `List Weight` field is the `Share` field in the `Calling List` tab.

Active

Required; default is `checked`. A check box that indicates if the selected Calling List is involved in this campaign. The check box is used to activate or deactivate the Calling List dynamically in a running dialing session for a campaign.

Campaign Object--Options Tab

Use this tab to define outbound campaign-related options. An option defined in this tab fine-tunes the system on the individual campaign level.

Campaign Group Object

A Campaign Group is the main configuration object in Outbound Contact. It can be found in the Campaign Groups tab within the Campaign object in Genesys Administrator > Provisioning tab > Outbound Contact. The Campaign Group is defined as a Campaign (a set of calling lists) that is assigned to work resources such as an Agent Group or a Place Group.

The following sections describe how to configure a Campaign Group object:

Creating a New Campaign Group Object

Start

1. In Genesys Administrator, select Provisioning > Outbound Contact > Campaigns.
2. In the Details pane, double-click a Campaign.
3. Select the Campaign Groups tab.
4. Click New.
5. Define the fields in the Configuration tab.

Note:	The following sections (General Section, Advanced Section, and Connection Section) describe how to configure this tab.

6. Configure the Options tab.
7. Configure the Permissions tab.
8. Click Save and Close.

End

General Section

Name

Required; The format is [Campaign@CampaignGroup]. Automatically populated by the system.

Tenant

Required; Automatically populated by the system.

Campaign

Required; The name of the Campaign that will form the Name field above. Automatically populated by the system.

Group Type

Required; The type of group (agent or place). Automatically populated by the system.

Group

Required; The name of the group that will form the `Name` field above.

Description

Optional; the default value is `[Blank]`. A brief description of the campaign.

State Enabled

Required; default is `checked`. A check box that indicates that customer interaction can be directed to this target. A more complete description can be found in *Framework Genesys Administrator Help*.

Advanced Section

Dial Mode

Required; the default value is `[Predictive]`. Any of the modes that are described in Dialing Modes.

Voice Transfer Destination

Required; the default value is `[None]`. Identifies where calls are to be delivered for handling. In addition to the `Voice Transfer Destination` DN, any `ACD Queue`, `Virtual Queue`, `Routing Point`, or `Service Number` DN that might be part of a call flow that involves this group of agents or places must be specified as an `Origination DN` in the Advanced section of the `Agent Group` or `Place Group` objects in Genesys Administrator. OCS needs to monitor these DNs to identify "foreign" inbound or outbound calls that are delivered to agents but were not issued by OCS on behalf of the Campaign Group, or if the `ACD Queue` DN is configured as an "overflow" DN.

Notes:	• You must avoid situations in which the same DN is configured as an Origination DN for more than one Agent Group or Place Group that is part of a predictive or progressive campaign dialing mode because the pacing algorithm may give incorrect statistics that may result in poor dialing ratios. • In addition to the actual Voice Transfer Destination DN, other DN types such as ACD Queue, Virtual Queue, Routing Point, or Service Number must be configured as Origination DNs in the Agent Group object or the Place Group object. These types of DNs need to be specified so that OCS can monitor them in order to recognize *foreign* inbound or outbound traffic (such as calls that seize agents but were not

	issued by OCS on behalf of the Campaign Group) or to recognize that the ACD Queue DN is configured as an overflow DN.

Operation Mode

Required; the default value is [Manual]. Not used at this time.

Optimization Method and Optimization Method Value

Required in Predictive mode; the default value is [Agent Busy Factor] for Optimization Method, and the default value is [80] for the Optimization Method Value. These parameters work together to determine how busy the campaign will be. They are not applicable to Push Preview and Power GVP dialing modes.

Maximum Queue Size

Optional; the default value is [0]. Determines the number of records to keep in the dialer's queue. It is an internal counter that is used when OCS performs with Power GVP or Push Preview modes. OCS tries to always keep dialer's buffer full with the specified Maximum Queue Size.

IVR Profile

Optional; the default value is [Blank]. Specifies the IVR Profile that is specified in the GVP EMPS IVR Profile. The ivr-profile-name option uses the value specified in this field.

Notes:	• In an VoIP environment, OCS provides this IVR Profile to GVP in the TMakePredictiveCall request before the GVP VoiceXML application is started for dialing the customer call. • In an VoIP environment (Power GVP or Progressive GVP modes), User Data associated with the customer call is delivered through SIP messages.

Interaction Queue

Optional; the default value is [None]. Specifies a configuration object that is created in the Scripts section in Genesys Administrator > Provisioning > Routing/eServices by Interaction Routing Designer when developing a Business Process to process outbound Preview mode interactions. For more information, see the *Universal Routing Business Process User's Guide*. An Interaction Queue is used in Push Preview mode only.

Script

Optional; the default value is [None]. Defines the Script object that contains all of the attributes that are required by Agent Scripting. For more information, see the "Attaching Script Information to OCS User Events and Telephony Events" section in the *Outbound Contact Reference Manual* for more information.

Trunk Group DN DBID

Required only if you are using Outbound Contact in an Outbound VoIP environment. If specified, OCS uses this DN as the DN from on behalf of which outbound and engaging calls originate. The name of the Trunk Group DN is the same as the partition-id value specified in SIP Server, which determines the resource allocation for calls made from this Trunk Group DN.

Note:	For additional information about configuring an Outbound-IP environment, see Outbound-VoIP in ASM Dialing Modes.

Minimum Record Buffer Size/Optimal Record Buffer Size

Required; the default value is [4] for Minimum and the default value is [6] for Optimal. Acts as a multiplier for the number of agents that are available for a campaign. Together, the Minimum and Optimal values determine how many records OCS keeps in memory when performing traditional campaigns. Because there might be no agents for the Push Preview and Power GVP dialing modes, these parameters are used differently by OCS as a percentage of the desired Maximum Queue Size value. For example, you can set them to Min: 100%, Opt: 150-200%.

Number of Channels

Identifies the maximum number of channels (or ports), that a Campaign Group can use to place calls. This max number is used in: Progressive, Predictive, Progressive with seizing, Predictive with seizing, and Progressive GVP dialing modes.

Connection Section

The Connections section enables you to specify all server connections (besides T-Server) that are required to run a dialing session for an outbound campaign in a certain dialing mode for a particular Campaign Group.

Note:	This type of connection provides basic host/port connectivity only. It does not provide features such as Advanced Disconnect Detection Protocol (ADDP). Also, you can add your application to the Connections tab of the OCS Application object in order to configure any additional connection parameters.

Outbound Schedule Object

Each Schedule is represented in the Genesys Configuration Layer as a Script object, of the Outbound Schedule type. OCS only considers schedules which are Enabled in configuration.

The following section briefly describes how to create an Outbound Schedule object. For complete information on configuring and using Schedules, see the *Framework Genesys Administrator Help.*

Creating a New Outbound Schedule Object

Start

1. In Genesys Administrator, go to Provisioning > Outbound Contact > Schedules.
2. Click New. The Outbound Schedule Properties window opens, and then click Add under the Schedule Items to add an existing Campaign Group or create a new Campaign Group.
3. On the Configuration tab, enter the following information (indicates a mandatory field)
   Name--The name of the Schedule. You must specify a value for this property, and that value must be unique within the Configuration Database (in an enterprise environment) or within the Tenant (in a multi-tenant environment).
   Tenant--In a multi-tenant environment, the Tenant to which this Schedule belongs. This value is set automatically, and you cannot change it.
   Script Type--Select Outbound Schedule.
   State--Indicates if this Schedule is enabled (default) or disabled.
4. On the Options tab, enter information as required.
5. On the Permissions tab, enter information as required.
6. To save the new Schedule object and register it in the Configuration Database, do one of the following:
   - Click Save and Close to return to the Scripts view. To configure the new object, double-click on the name of the object to open the Schedule Properties dialog box. Refer to Genesys Administrator Help for information about configuring the properties of the Schedule.

Note:	When defining sequential execution points for a Schedule, the At time condition specified for a Stop action must not be earlier than the At time condition specified for the corresponding Start action. OCS does not assume that the earlier Stop time is on the next calendar day; instead, it starts the action at the specified time and then immediately stops it. **Warning:** This is not an issue when defining Stop actions with the After condition; OCS supports execution on the following day in this case.

   - Click Save to continue configuring the Schedule.
   - Click Save and New to save the new Schedule and start creating another one.

End

Outbound Contact Configuration Options

These topics describe each of the options that you can configure to ensure that Outbound Contact performs as needed in your environment:

- Overview
- Outbound Contact Server Options
- CPD Server Options

Note:	In Genesys Administrator, all Outbound Contact options that were located on the Annex tab of the respective Outbound Contact objects (with the exception of SCXML options configured at the OCS application level), are located on the `Options` tab.

Overview

In Outbound Contact, you define options at the Outbound Contact Server (OCS) application level. You can also define options at the object level for Switch, DN, Agent, Place Group, Calling List, Campaign Group, and Campaign objects. Defining the settings at the object level enables flexibility and fine-tuning for a campaign setup.

OCS first reads the configuration options in the objects. If an option is not defined at the object level, OCS checks the configuration at the application level. If the option is not specified at the application level, OCS uses the default value for that option.

Note:	When you make changes to an option, the changes take effect immediately, unless otherwise noted in the option description.

Depending on the needs of the contact center configuration, the Outbound Contact Administrator can choose to:

- Use default OCS options, if the contact center has no specific requirements.
- Set options at the OCS application level, if the contact center requires overriding of OCS settings.
- Set options at the level of a specific object, if the contact center has specific needs for individual Switch, Agent, Place Group, or Campaign objects.

OCS stores configuration options in section folders on the Options tab. The Options tab contains the section folders and options for application objects in OCS and CPD Server.

This topic lists the section folders and options according to their general function, and the name of the object.

Notes:	• Logged data can be truncated to hide sensitive data in the log by configuring either the default-filter-type option or any of the log-filter-data section options to the value hide. The truncation is represented in the log by asterisks. For more information about these options, see the "Common Configuration Options" chapter in the *Framework Configuration Options Reference Manual*. • Starting in OCS 8.1.2, the tag option value is supported and built into the Log Library and, in user data printouts, covers all events received from external servers, such as T-Server, SIP Server, and Interaction Server. OCS will continue to print SQL statements and HTTP BODY JASON pairs in full when the default-filter-type logging option is set to copy and hides all sensitive data by replacing it with asterisks (****) when this option is set to any

	other value, including tag.

Outbound Contact Server Options

The Outbound Contact Server options are defined in various configuration objects. These options are summarized by logical group, that is, the operation for which they are required. The detailed descriptions of the options are then listed in alphabetical order.

- Section Names
- OCS Options by Logical Group
- OCS Option Descriptions

Section Names

The Options tab of the configuration object needs to contain a section that is named in one of the following ways:

- OCS application object name, which specifies the name of the current OCS Application object.
- default
- OCServer

Note:	Genesys recommends that you define all OCS options in the OCServer section folder, with the exception of default record priorities options, which are defined in the Outbound section.

The Outbound section is used for defining the default record priorities in Outbound Contact Campaign Group objects at the OCS application level. This section contains parameters specific to record priorities. See Flexible Prioritization of Record Types for more information about these options.

There is also an additional section, called log_call_stats, which is created only at the OCS application level. This section contains parameters specific to Audit Logging. See Section log call stats for descriptions of the options that this section contains.

When OCS looks for the defined value of an OCS option, it searches the sections in a prescribed order until it finds the option in one of the sections.

The prescribed search order is the following:

1. OCS searches the designated sections on the Options tab of the configuration object, as follows:
   a. <OCS application object name> section
   b. default section
   c. OCServer section
2. If OCS does not find the option, it follows the same procedure to search in the Options tab of its own Application object.
3. If OCS does not find the option, it uses the option's default value.

Example of an OCS Option Search

In this example, two different OCS applications (OCS1 and OCS2) work with the same switch. The options for each OCS application are defined on the Options tab of the switch configuration object or on the Options tab of the OCS Application object.

OCS1 is searching for the outbound_release_action option.

1. Options tab of the Switch object:
   a. OCS1 searches for the outbound_release_action option in the OCS1 section.

   b. If the option is not in the OCS1 section or if there is no section named OCS1, OCS1 searches for the option in the section named `default`.

   c. If the option is not in the `default` section, or if there is no section named `default`, OCS1 searches for the option in the section named `OCServer`.

2. `Options` tab of OCS:
   If OCS1 does not find the `outbound_release_action` option in any of the three sections on the `Options` tab of the Switch object, it searches the `Options` tab of its own `Application` object, repeating the search process that it performed on the `Options` tab of the Switch object.

3. Other:
   If OCS1 does not find the `outbound_release_action` option on either `Option` tab, it uses the option's default value (`false`).

Note:	Most Outbound Contact Server options are dynamic. This means that the application does not have to be restarted in order for option changes to take effect, unless otherwise noted in the option description.

How OCS Prioritizes If Different Configuration Objects Contain the Same Section

- If there are different configuration objects that contain options in the same named sections with the same key and different values, OCS chooses only one of them, according to the following priorities:
  - Calling List (highest priority)
  - Campaign Group
  - Campaign
  - Agent Group or Place Group
  - Switch
  - OCS application (lowest priority)
- If there are several configuration objects that are used in a particular campaign with options that contain different keys, OCS attaches all of these options as a list of key-value pairs in the `UserData` attribute.
- If you want to indicate that a particular outbound campaign has a higher priority than other campaigns, you can attach a `campaign_priority = high` key-value pair to every call or preview record that OCS generates while running the dialing session for this campaign. As an example, you can perform the following steps if you want to configure a Campaign Group to run a dialing session/campaign called `Collections90days` that uses an OCS application called `OCS Western Region`:

1. Choose a section name to store the data that you want to attach, such as `collection_options`.
2. Create a section with this name on the `Options` tab of a `Campaign Group` configuration object. In this section, add an option specifying the key as `campaign_priority` and the value as `high`.

Note:	If more than one dialing session/Campaign Group is going to run this campaign, configuration objects representing all these Campaign Groups should have this configuration.

3. Configure the `user_data_section_name` option and set the value to `collection_options` in the `OCServer` section of the `Options` tab of the `OCS Western Region` application object.

Section log_call_stats

This section contains the configuration options used to configure Audit Logging, and takes effect only if the configuration option `log_call_stats` is set to `yes` or `true`.

In addition to the option `all` (see log_call_stats), this section can also contain the common options `expire` and `segment`. Refer to the Framework Configuration Options Reference Manual for full descriptions of these options.

OCS Options by Logical Group

The following table lists the OCS options related to a given functionality. Some options might belong to more than one logical group.

For a full description of each option, use the link provided to find the option in the section OCS Option Descriptions.

OCS Options by Logical Group

Logical Group	Options
Agent Desktop	agent_logout_preview_call_result agent_preview_mode_start campaign-callback-distr cancel-on-desktop desktop-chains-limit direct-personal-callback engaged_answer_action engaged_release_action hard_request_to_login_dn outbound_answer_action outbound_release_action preview_release_nocontact_action record_processed stale_clean_timeout
ASM Dialing	asm_channel_num asm_drop_am_announcement_data asm_drop_announcement_data assured-connect assured-connect-field beep-on-merge engaged_answer_action engaged_release_action merge-method on-asm-overdial on-bridging-unable vtd-override
Call Processing	divert_to_unknown_dn ivr_group ivr_update_on_release on-asm-overdial overflow_dn predictive_callback remote_release_action transfer_to_unknown_dn
CX Contact	campaigngroup-preload-uri campaigngroup-states-interval campaigngroup-states-uri customer_id digital-dialer-uri disposition-code-field history-uri history-wait-condition-threshold http-connection-pool-size http-custom-headers

Logical Group	Options
	http-proxy http-response-timeout http-retry-applies-to http-retry-max-attempts pre-desktop-validation-uri pre-dial-validation protocol.add-record.uri schedule-states-uri send-campaigngroup-states send-history send-schedule-states validation-uri
Dial Log	dial_log_buffer dial_log_delimiter dial_log_destination dialer-num-attempts log_call_stats
Dialing	am-detection-map am-drop-field assured-connect assured-connect-field call_answer_type_recognition call_timeguard_timeout call_transfer_type call_wait_agent_connected_timeout call_wait_connected_timeout call_wait_in_queue_timeout call_wait_original_establish_timeout caller-id-sets caller-id-rotation campaign-callback-distr channel_num check_dnc_callback check_dnc_list cpd-on-connect cpd-recording CPNDigits CPNDisplayName CPNPlan CPNPresentation CPNScreening CPNType customer_id dialer-ttl dialing_rate_limit lifo-mode lifo-record-expiration lifo-unload-pace public_network_access_code user_data_section_name
Dialing Regulations	all asm_drop_am_announcement_data asm_drop_announcement_data call_timeguard_timeout check_dnc_callback check_dnc_list CPNDigits digits-detection digits-detection-pattern digits-detection-timeout

Logical Group	Options
	digits-reaction dnc-default-duration dnc-purge dnc-reread dnc-use-duration log_call_stats pa-abandon-rate-limit pa-amd-false-positive-rate pa-amd-test-percentage pa-odr-period-start-time pa-safe-dialing predictive_max_overdial_rate predictive_min_overdial_rate
Dynamic Agent Assignments	agent-assignment agent-assignment-max-num agent-assignment-min-num agent-assignment-priority agent-reassignment-if-waiting-ports agent-reassignment-if-waiting-ports-timeout agent-reassignment-if-waiting-records agent-reassignment-if-waiting-records-timeout inbound-agent-assignment-min-num inbound-agent-assignment-priority ocs-urs-broadcast ocs-urs-interact
GVP	am-beep-detection dialer-num-attempts dialer-ttl gvp-modes-use-switch-rate-limit ivr-profile-name ivr_record_processed predictive_patience_time
Historical Reporting	conversion icon_attribute right_person snapshot_interval
HTTP-based Notifications	campaigngroup-states-interval campaigngroup-states-uri contact-processed-notification contact-processed-notification-uri encoding history-uri history-wait-condition-threshold history-wait-recovery-timeout http-connection-pool-size http-body http-custom-headers http-headers http-method http-proxy http-response-timeout http-retry-applies-to http-retry-max-attempts treatment-action-uri send-campaigngroup-states send-history validation-race-condition-call-result

Logical Group	Options
Interactions Processing in Push Preview Dialing Modes	direct-personal-callback direct-push-preview interaction-media-type recall-on-unload
Licensing	license-file num-of-licenses
Outbound Contact VoIP Dialing Modes	am-beep-detection beep-on-merge cpd-on-connect cpd-recording merge-method on-asm-overdial on-bridging-unable predictive_patience_time
Outbound Schedule	campaigngroup-preload-uri on-schedule-disable schedule-states-uri send-schedule-states
Pre-dial Validation	http-connection-pool-size http-response-timeout http-retry-applies-to http-retry-max-attempts pre-desktop-validation-expiration-timeout pre-desktop-validation-type pre-desktop-validation-uri pre-dial-validation validation-keys validation-propagate-user-data validation-timeout-call-result validation-uri validation-wait-history
Predictive Algorithm	dynamic-port-allocation history_length inbound_agent_outlier_limit outbound_agent_outlier_limit pa-abandon-rate-limit pa-amd-false-positive-rate pa-amd-test-percentage pa-dial-expire pa-exclude-long-dialing pa-handle-expire pa-handle-time-consider pa-hitratio-min pa-inbound-ignore pa-odr-interval pa-odr-period-start-time pa-progressive-multiplier pa-queue-expire pa-safe-dialing pa-selfcheck-adt-threshold pa-selfcheck-awt-threshold pa-selfcheck-bf-threshold pa-selfcheck-interval pa-selfcheck-odr-threshold predictive_algorithm predictive_hit_ratio

Logical Group	Options
	predictive_hot_start predictive_inbound_call_duration predictive_inbound_rate predictive-longcalls-truncation predictive_max_overdial_rate predictive_min_overdial_rate predictive_outbound_call_duration predictive_patience_time progressive_blending_reserved_agents progressive_blending_reserved_status small_group_size stat-hit-ratio-count time-to-ready-tolerance
Predictive Algorithm Self-Test	pa-selfcheck-adt-threshold pa-selfcheck-awt-threshold pa-selfcheck-bf-threshold pa-selfcheck-interval pa-selfcheck-odr-threshold
Predictive Hot Start	predictive_hit_ratio predictive_hot_start predictive_inbound_call_duration predictive_inbound_rate predictive_outbound_call_duration
Prioritization of Record Types	default_campaigncallback_priority default_campaignrescheduled_priority default_general_priority
Real-Time Reporting	db-stat-recalc-interval outbound_contact_server record-count-use-timeframe report-procedure-body report-procedure-location stat-server-ha
Record Processing	callback-observe-campaign callback-observe-group campaign_name_field campaigngroup-auto-complete disposition-code-field force-unload-wait-db ignore-empty-group protocol.add-record.uri record_save_intermediate_results send_attribute stale_clean_timeout treatment_sched_threshold update_all_records send-history history-uri history-wait-condition-threshold history-wait-recovery-timeout validation-race-condition-call-result
SIP Cluster	sip-cluster-asm-dial-node sip-cluster-lb-mode sip-cluster-prime-node

Logical Group	Options
SCXML-based Treatments	action-queue-quota local_file_maxage maxAge maxStale scxmli.persistence.max_active scxmli.transcoder.default treatment-holidays-table treatment-preferred-contact-field treatment-uri treatment-weekdays-table

OCS Option Descriptions

action-queue-quota

- Default Value: `10000`
- Valid Values: A positive integer, greater than `10`
- Changes take effect: After restart
- Configuration Level: OCS Application
- Logical Group: SCXML-based Treatments

Specifies the number of action items to process from an SCXML action queue before returning control to OCS.

agent-assignment

- Default Value: `no/false`
- Valid Values: `yes/true, no/false`
- Configuration Level: Campaign Group
- Logical Group: Dynamic Agent Assignments

Controls the ability of the Campaign Group to be used in agent reassignment. The value `no/false` is used for Campaign Groups that have calls distributed by the switch and not distributed by Universal Routing Server.

agent-assignment-max-num

- Default Value: `0`
- Valid Values: `0` to `n`
- Configuration Level: Campaign Group
- Logical Group: Dynamic Agent Assignments

Defines the maximum number of agents in the Campaign Group when determining agent reassignment. When the value is set to `0`, there are no restrictions for the maximum amount of agents in the Campaign Group. The `agent-assignment-max-num` option and dynamic agent reassignment, in accordance with the availability of ports, is applied irrespective of the value that is specified for the agent-assignment-min-num option (before OCS 8.1.3, these were not applied if the value of the **agent-assignment-min-num** was set to zero).

agent-assignment-min-num

- Default Value: `0`
- Valid Values: `-1, 0 ... n`
- Configuration Level: Campaign Group
- Logical Group: Dynamic Agent Assignments

Defines the minimum number of agents in the Campaign Group when determining agent reassignment.

- When the value is set to `0`, the Campaign Group is always considered to be understaffed. Agents are assigned to this Campaign Group even when the lowest priority Campaign Groups are understaffed.

- When the value is greater than `0`, the Campaign group is considered understaffed until the number of assigned agents is at least equal to the specified value. For instance, if you provision a campaign group with the value 100, the campaign group is considered understaffed until it has 100 agents. Agents are reassigned to the campaign group accordingly.
- When the value is set to "`-1`", the Campaign Group is considered always staffed, even if there are no assigned agents. This setting prevents assignment to the Campaign Group and OCS assigns agents to this Campaign Group only if the following items are true:
    - These agents can not be assigned to any running activities (inbound activity and running associated Campaign Groups) with higher priority due to the restrictions imposed by the options regarding the agent assignment.
    - Any running activity with lower priority is staffed; that is, the number of assigned agents is not less than the non-zero minimum values specified in the **agent-assignment-min-num** or **inbound-agent-assignment-min-num** option.
    - The **inbound-agent-assignment-min-num** option is not set to `zero`.
    - The **agent-assignment-min-num** options for all active associated Campaign Groups are not set to `zero`.

agent-assignment-priority

- Default Value: 1
- Valid Values: `1... n`
- Configuration Level: Campaign Group
- Logical Group: Dynamic Agent Assignments

Defines the priority of the Campaign Group when determining agent reassignment. As the value increases, the priority for the specified Campaign Group also increases.

agent_logout_preview_call_result

- Default Value: `-1`
- Valid Values: `-1, 0, 1,...n`

- Configuration Level: Campaign Group, Application
- Logical Group: Agent Desktop

Enables OCS to change the call result for `Preview, Personal CallBack`, or `Campaign CallBack` records that are on an agent's desktop after receiving an `EventAgentLogout` message from Stat Server.

- When the value is `-1`, OCS updates the record with the previous call result.
- When the value is greater than or equal to `0`, OCS updates the record in the database table with the integer value of this option. For Genesys Administrator to display the call result as a string, the integer value of this option must match a call result enumeration value listed in the Defined Constants topic of the *Outbound Contact Reference Manual*.

agent_preview_mode_start

- Default Value: `no/false`
- Valid Values: `yes/true, no/false`
- Configuration Level: Campaign Group, Application
- Logical Group: Agent Desktop

Controls whether agents must send the `PreviewDialingModeStart` event before they can receive preview records or callbacks. If set to `true`, before sending a personal or group callback, OCS checks the agent's status to see whether his or her desktop sent `PreviewDialingModeStart`.

agent-reassignment-if-waiting-ports

- Default Value: `no/false`
- Valid Values: `yes/true, no/false`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Dynamic Agent Assignments

When set to `true`, OCS calculates the "effective pool" of agents, which is a number of agents that can effectively work with the given number of available ports. If this effective pool is smaller than the currently-assigned number of agents and less than the **agent-assignment-max-num**, then the remainder of the agents is reassigned to other activities. To prevent too frequent reassignments, OCS initiates a new reassignment only when the time since a previous reassignment is longer than the timeout configured in the **agent-reassignment-if-waiting-ports-timeout** option. This option is applicable to dialing sessions with agent-assignment enabled.

agent-reassignment-if-waiting-ports-timeout

- Default Value: `30`
- Valid Values: A positive integer

- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Dynamic Agent Assignments

Specifies the timeout, in seconds, after which an agent can once again be reassigned from a dialing session with no available dialing ports to a different activity. This option is applicable to dialing sessions with agent-assignment enabled and with the **agent-reassignment-if-waiting-ports** option set to `true`.

agent-reassignment-if-waiting-records

- Default Value: `no/false`
- Valid Values: `yes/true, no/false`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Dynamic Agent Assignments

When set to `true`, OCS reassigns all agents from this Campaign Group to other activities when there are no available dialing records for the Campaign Group, and the duration of the "waiting records" condition has surpassed the time set in the **agent-reassignment-if-waiting-records-timeout** option. This option is applicable to dialing sessions with agent-assignment enabled.

agent-reassignment-if-waiting-records-timeout

- Default Value: `30`
- Valid Values: A positive integer
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Dynamic Agent Assignments

Specifies the timeout, in seconds, during which OCS waits to reassign agents from a Campaign Group that has no available dialing records to some other activity. This timeout is not applicable if all agents have been reassigned away from the Campaign Group and new records are retrieved into the dialing buffers. This option is applicable to dialing sessions with agent-assignment enabled and with the **agent-reassignment-if-waiting-records** option set to `true`.

all

- Default Value: `CallStats`
- Valid Values: A valid path and file name for the audit log
- Configuration Level: Application
- Changes Take Effect: Immediately

- Logical Group: Dialing Regulations

Specifies a full path to the Audit Log flat file, including the filename without the extension.

Note:	This option is configured in the log-call-stats section.

am-beep-detection

- Default Value: `no/false`
- Valid Values: `yes/true, no/false`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: GVP, Outbound Contact VoIP Dialing Modes

Specifies whether GVP is forced to detect an answering machine beep tone before playing music or starting the VoiceXML application in certain Outbound VoIP dialing modes, specifically Power GVP, Progressive GVP, Progressive with Seizing (ASM) and Predictive with Seizing (ASM).

- When set to `true`, GVP delays playing music or starting the VoiceXML application until the beep tone is detected.
- When set to `false`, GVP starts playing music or executing the VoiceXML application immediately after the answering machine is detected. This may result in the message being played before the beep tone, which is the indication for the answering machine to start recording. Depending on the length of the message, the answering machine may not record all or any of the message.

am-detection-map

- Default Value: An empty string (`""`)
- Valid Values: A name of a `Business Attribute` value configuration object, or `default`
- Configuration Level: Calling List, Application
- Logical Group: Dialing

Specifies the name of the `Business Attribute` value configuration object that contains the AM-detection map to be used for a particular Calling List or to be used application-wide. See Per-Record Basis for more information.

Dependency: If the CX Contact option **custom-format** is configured, OCS does not support the **am-detection-map** option for all custom CX Contact fields (`Other21+`) and fields `c_other1,..,c_other20`.

am-drop-field

- Default Value: An empty string
- Valid Values: * (asterisk) OR `<field name>:<value1>[,<value2[,value3]...]`, where:

  - * (asterisk) is any value
  - `<field name>` is the name of the field in the Calling List
  - `<value1>[,<value2[,value3]...]` is an optional comma-separated list of values for this field

    For example:

  - **am-drop-field** = *
  - **am-drop-field** = `cd_device_index:2,3,6`

- Configuration Level: Calling List (higher priority), Campaign Group, Application
- Changes Take Effect: When the next record is retrieved from the Calling List table
- Logical Groups: Dialing
- Introduced: 8.1.517.04

Specifies how OCS handles an outbound call on a particular device when the call is placed in any auto-dialing mode and Answering Machine (AM) is detected. Any existing field from the Calling List (including user-defined) can be used. If the field with the configured field name exists in the Calling List, OCS checks the value of the field for each record. A record is marked to be handled in accordance with the configured value. The specified field is not required to have `send_attribute` configured. If the option is set to * (asterisk), all calls to the detected AM will be dropped.

Dependency: If the CX Contact option **custom-format** is configured, OCS does not support the **am-drop-field** option for all custom CX Contact fields (`Other21+`) and fields `c_other1,..,c_other20`.

asm_channel_num

- Default Value: `0`
- Valid Values: `0` or a positive integer less than the `Number of CPD Ports` property of the Campaign Group
- Configuration Level: Campaign Group
- Changes Take Effect: Immediately
- Logical Group: ASM Dialing

Specifies the maximum number of engaging ports that can be used by the given Campaign Group. This option always works in pair with the `Number of CPD Ports` property which is defined on the `Advanced` tab of `Campaign Group` configuration object. OCS considers the `Number of CPD Ports` to be the total number of ports available to the Campaign Group, for example, the sum of engaging ports and ports for outbound dialing. The following table summarizes OCS behavior for different variations of the CPD Ports setting:

OCS Behavior and CPD Port Configuration

CPD port configurations	Number of CPD ports = 0	Number of CPD ports > 0
`asm_channel_num = 0`	No restrictions.	No restriction on the number of engaging ports (although they must not be greater than 'Number of CPD Ports' property).

`asm_channel_num > 0`	Only the number of engaging ports is restricted.	Both the engaging ports and the ports for outbound dialing are restricted.

This option can be used with hardware configurations that use a separate pool of engaging lines (ports for agent engagement). See the CPD Server **use-engaging-lines** option which controls engaging mode of CPD Server. This option can also be useful for blended environments to limit the number of agents being engaged and thus allocate some portion of the agents to always be available to handle inbound calls.

Note:	Assigning the **asm_channel_num** option with a value of less than the number of agents available for the Campaign Group might cause an excessive overdial rate if the classical Predictive Algorithm is being used. See the predictive_algorithm option.

asm_drop_am_announcement_data

	CPD Server	SIP Server
Default Value	An empty string ("")	none
Valid Values	Path to the file in the VOX format (string)	A prompt ID for an announcement, as configured in GVP Media Control Platform
Configuration Level	Campaign Group, Application	Campaign Group, Application
Changes Take Effect	Immediately	Immediately
Logical Groups	Dialing Regulations, ASM Dialing	Dialing Regulations, ASM Dialing

- (For ASM modes only) Specifies the message to be played if an answering machine is detected before releasing the established customer call in the ASM modes.

The values differ depending on whether you use CPD Server or SIP Server, as follows:

- (For CPD Server) The full name (including the path) to the file (in the VOX format). CPD Server plays this announcement and then releases the established customer call because either of the following occurs:
  - The customer leg could not be bridged with the engage call leg
  - No destination DN is specified for bridging answering machine detected calls.
- (For SIP Server) The Prompt ID for the message, as configured in GVP Media Control Platform. For more information on the Prompt ID, see the GVP documentation.

This option can be used with the asm_drop_announcement_data option. While the **asm_drop_announcement_data** option instructs OCS to play the same announcement for all call results, using both options allows CPD Server to play a different announcement for Answering Machine call results than for Answer call results.

Note:	This option, added in OCS 7.6.101.04, can only be used in the Active Switching Matrix (ASM) mode.

asm_drop_announcement_data

	CPD Server	SIP Server
Default Value	An empty string ("")	none
Valid Values	Path to the voice file (including the name) in the VOX format (string)	A prompt ID for an announcement, as configured in GVP Media Control Platform
Configuration Level	Campaign Group, Application	Campaign Group, Application
Logical Groups	Dialing Regulations, ASM Dialing	Dialing Regulations, ASM Dialing

Specifies the message to be played before releasing the established customer call in the ASM modes.

If the established customer leg of the call has to be released by CPD Server or SIP Server (in an Outbound-IP environment) because it could not be bridged with the agent engaging call leg, a predefined announcement to the customer can be used before the call is released.

The values are as follows: For CPD Server--The full name (including the path) to the file (in the VOX format) with the announcement that OCS provides to CPD Server for all call results.

Note:	CPD Server must be able to access this network path so that this file can be used.

If no value is specified, OCS does not add the `OCS_DROP_ANNOUNCE_DATA` attribute to the extensions parameter of the Outbound Contact Server `RequestMakePredictiveCall` event.

If a value is specified, the full path to the voice file will be used by CPD Server in the following extensions parameter of the Outbound Contact Server `RequestMakePredictiveCall` event:

`OCS_DROP_ANNOUNCE_DATA`
Data Type: String
Contains the full name of the voice file in `.vox` format.

CPD Server will play the specified file before releasing the established customer call. If CPD Server successfully opens the file specified in the `OCS_DROP_ANNOUNCE_DATA` attribute, it prints a `dx_playvox success` message to the log file. If there was an error, a `dx_playvox failed` message is printed in the log file.

For SIP Server--The Prompt ID for the announcement, as configured in GVP Media Control Platform. For more information on the Prompt ID, see the GVP documentation.

This option can be used with the asm_drop_am_announcement_data option. The **asm_drop_am_announcement_data** option instructs OCS to play the same announcement for Answering Machine call results, while the **asm_drop_announcement_data** option instructs the component to the same announcement for all other call results.

Note:	When using Trunk Group DNs in an Outbound-IP environment, OCS uses the standard `TApplyTreatment` event with a `MUSIC_DN` parameter, according to SIP Server requirements. In this scenario, OCS releases the call only after the

	announcement is played, which occurs when it has received EventTreatmentEnd from SIP Server.

assured-connect

- Default Value: `false`
- Valid Values: `false`, `true`
- Configuration Level: Individual record and chain of records (via SCXML), Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Dialing, ASM Dialing

At the Campaign Group or Application level, this option enables/disables assured (guaranteed) connection for a specific Campaign Group or the entire OCS application:

- When set to `false`, OCS does not apply assured connection to any record.
- When set to `true`, OCS applies assured connection to an individual record or a chain based on the **assured-connect-field** option and/or the **assured-connect** option that is set using the `set_flex_attr` custom action of the SCXML treatment. See Setting Options for Individual Records or Chain of Records for more information about custom actions.

assured-connect-field

- Default Value: Empty string
- Valid Values: `field name:[:list of values]`, where:

field name	Name of a field in the Calling List.
list of values	An optional comma-separated list of values for this field.

- For example: `is_assured:yes,true` or `is_assured:1,2,4`
- Configuration Level: Calling List, Campaign, Application
- Changes Take Effect: When record is retrieved from the Calling List table
- Logical Groups: Dialing, ASM Dialing

If the field with the configured field name exists in the Calling List, OCS checks the value of the field for each record. A record is marked for assured connection if either of the following conditions are met:

- The value in the field matches exactly one of the values given in the list of values.
- The value in the field is not NULL or zero (`0`) and the list of values is empty.

Dependency: If the CX Contact option **custom-format** is configured, OCS does not support the **assured-connect-field** option for all custom CX Contact fields (`Other21+`) and fields `c_other1,..,c_other20.`

beep-on-merge

- Default value: `false`
- Valid Values: `yes/true, no/false`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Outbound Contact VoIP Dialing Modes, ASM Dialing

Enables the playing of a beep tone to the agent on the engaging call immediately before the agent is bridged to the customer call when running in Outbound VoIP dialing modes.

call_answer_type_recognition

- Default Value: `no_am_detection`
- Valid Values: `no_progress_detection, no_am_detection, positive_am_detection, full_positive_am_detection, accurate_am_detection, telephony_preset`
- Configuration Level: Individual record and chain of records (via SCXML), Campaign Group, Application
- Logical Group: Dialing

Specifies answer, answering machine, and fax detection settings when dialing using CPD Server, SIP Server, or OBN Manager.

Note:	This option does not apply when OCS uses T-Server for dialing and PBX equipment for call progress detection.

The values are as listed in the following table:

Option Value Description

Option Value	Non-IP Outbound Environment Description	Outbound-IP Environment Description
`no_progress_detection` **Note:** The **call_answer_type_recognition** option cannot be set to `no_progress_detection` on DM3 hardware because of line-side PDK protocol restrictions.	Disables call progress detection, and the call is transferred as soon as it is established.	No SIT, answering machine, or fax tone is detected; the call is connected, as if it were answered by an actual person. **Note:** If a SIP error message contains the call result with SIT tone information (for example, if a carrier sends the SIT `Invalid number` call result in the SIP error message “404 Not Found”), the information can be used as a detected tone despite the `no_progress_detection` option setting.
`no_am_detection`	Disables answering machine detection but detection of all other devices is still enabled.	SIT and fax tones are detected but answering machines tones are not; if no SIT or fax tone is

		detected, the call is connected as if it were answered by an actual person.
positive_am_detection **Note:** This value works only with tape-based answering machines. It does not work with modern digital answering machines.	Enables standard answering machine detection (Positive Answering Machine [PAM] mode).	SIT, answering machine, and fax tones are detected; answering machine detection should be performed with default parameters for Media Gateway or Media Server.
full_positive_am_detection	Enables full positive answering machine detection (Full Positive Answering Machine [FPAM] mode).	SIT, answering machine, and fax tones are detected; answering machine detection is performed with the highest probability of live voice detection, if supported by Media Gateway or Media Server. If it is not supported, this value functions in the same way as the positive_am_detection value.
accurate_am_detection	Enables or disables detection, based on an analysis of the duration of the greeting.	SIT, answering machine, and fax tones are detected; answering machine detection is performed with the highest probability of answering machine detection, if supported by Media Gateway or Media Server. If it is not supported, this value functions in the same way as the positive_am_detection value.
telephony_preset	Causes the same behavior as the no_am_detection value. If the user selects telephony_preset, this value is automatically converted to the no_am_detection value.	Tones are detected in accordance with the default configuration for SIP Server and Media Gateway.

call_timeguard_timeout

- Default Value: 0 (milliseconds)
- Valid Values: 0...<n> (milliseconds)
- Configuration Level: Campaign Group, Application
- Logical Groups: Dialing Regulations, Dialing

Enables a user to set a timeout for post-connect call progress detection. The call is transferred to a queue when the timeout expires, regardless of the call result or the completion of call progress detection. The timeout is calculated from the moment that CPD Server receives an EventEstablished message. If this option is set to 0 (zero) or if it is not present, CPD Server does not break call progress detection.

Note:	This option applies when OCS uses SIP Server, CPD Server or OBN Manager for dialing. It does not

	apply when OCS uses T-Server for dialing and PBX equipment for call progress detection. You must specify a value greater `0` for this option when configuring Trunk Group DNs in an Outbound-IP environment. For more information about Outbound-IP environment requirements, see Outbound VoIP Deployment.

call_transfer_type

- Default Value: `one_step`
- Valid Values: `one_step, two_step`
- Configuration Level: Campaign Group, Application
- Logical Group: Dialing

Specifies the transfer type that the dialer (CPD Server or OCS) use for outbound calls. This option applies when OCS uses CPD Server or for two-step transfers in Outbound IP environment (see Outbound VoIP Deployment). It does not apply when OCS uses PBX equipment for call progress detection.

For an environment using CPD Server as the dialer, CPD Server translates the `one_step` value as either `mute` or `single_step` depending on the switch that is being used; check your switch documentation.

The `two_step` value is commonly used to overcome some switch transfer issues such as *ring splash* on a Meridian switch. Ring splash occurs when a dialer transfers a connected call to an agent, and the caller hears a ringback before he or she hears the agent answer. For some switches, this ringback cannot be turned off, either by a command in the CTI link or by a switch setting.

As a workaround to this situation, Genesys uses a *two-step* transfer. In a two-step transfer, the switch puts the original leg of the call on hold and starts to initiate a transfer. For a moment, the caller hears silence or music depending on whether the `music on hold` feature is enabled on the trunk or queue. For the second leg of the call, the Origination DN (usually a queue), as defined in the Campaign Group, is dialed. The transfer process is complete when an agent answers the call. The call_wait_in_queue_timeout option is a timer and the value determines how long the call should wait (in the queue) before an agent answers. If the timer expires, the dialer drops the call, and the record is marked with the `Dropped` call result.

Note:	Genesys recommends turning off the `music on hold` feature on the trunk or queue when using a two-step transfer.

call_wait_agent_connected_timeout

- Default Value: 6 (seconds)
- Valid Values: A positive integer
- Configuration Level: Campaign Group, Application
- Logical Group: Dialing

Specifies the timeout, in seconds, from the time the engaging call is dialed to the time the call is answered by an agent. This timeout applies only to engaging calls in ASM mode; that is, calls that engage an agent before the customer answers. If the agent answers the call, the agent is seized for that call. The value of this option indicates how long to wait for an agent to be seized after dialing that agent.

Note:	This option applies when OCS uses CPD Server for dialing. It does not apply when OCS uses T-Server for dialing and PBX equipment for call progress detection.

call_wait_connected_timeout

- Default Value: 120 (seconds)
- Valid Values: 0-7200 (seconds)
- Configuration Level: Campaign Group, Application
- Logical Group: Dialing

Specifies the timeout upon of expiration of which dialer should consider that call is not answered by call party. When OCS is dialing calls through CPD Server or OBN Manager:

- For ISDN: Specifies the timeout, in seconds, between dialing and the determination that the called party is not answering.
- For analogue and line-side DNs: Specifies the timeout between the first ring and the determination that the called party is not answering.

When OCS is dialing calls through T-Server or SIP Server:

- The value of this option is passed to the T-Server as an AttributeTimeout in the MakePredictiveCall message. The timeout value received in the request has a higher priority than T-Server's own options, and it is used to set up the duration of the No Answer interval on the switch.
- Set the value between 6 and 99 when using the Aspect switch classification board. The default value of 120 causes the RequestMakePredictiveCall requests sent by OCS to fail because the switch accepts only two digit values.
- For a VoIP environment in which SIP Server is used for dialing, the maximum value of this option is 32 seconds for all dialing modes. It can be set to a higher value if the **predictive-timerb-enabled** option is disabled on SIP Server, but additional settings might need to be applied to SIP Server and Genesys Media Server (MCP) to accommodate the higher value. For more information, see the "Increasing Ringing Period for Predictive Calls" chapter in the *SIP Server Deployment Guide*. To ensure accurate reporting of T-Library attributes, set the **call_wait_connected_timeout** option to the value of less than 32 seconds in the VoIP environment.

call_wait_in_queue_timeout

- Default Value: 10 (seconds)
- Valid Values: A positive integer
- Configuration Level: Campaign Group, Application

- Logical Group: Dialing

Specifies the maximum amount of time, in seconds, that an outbound call is allowed to stay in a queue. This option applies when OCS uses CPD Server or for two-step transfers in an Outbound IP environment (see Outbound VoIP Deployment).

Note:	For CPD Server usage, after this timeout expires, CPD Server can release the call only when the two-step transfer is used (see the call_transfer_type option, and CPD Server controls the consult leg of the transferred call.

call_wait_original_establish_timeout

- Default Value: 4 (seconds)
- Valid Values: A positive integer or `0`
- Configuration Level: Campaign Group, Application
- Logical Group: Dialing

Specifies the timeout, in seconds, between the end of a positive call progress detection response from Dialogic—such as `Answer, Answering Machine (AM), or Cadence Break`—and an `EventEstablished` from T-Server.

This option applies only when OCS uses CPD Server. It does not apply when OCS uses PBX equipment for call progress detection.

callback-observe-campaign

- Default Value: `true`
- Valid Values: `true`/yes, `false`/no
- Configuration Level: Campaign, Application
- Changes Take Effect: When Campaign Group is loaded
- Logical Group: Record Processing

When set to `true`, OCS will use the Campaign DBID when retrieving callbacks and rescheduled records. That is, only records scheduled by the given Campaign will be retrieved. When set to `false`, OCS will retrieve records previously scheduled by any campaign.

callback-observe-group

- Default Value: no/`false`
- Valid Values: yes/`true` or no/`false`
- Configuration Level: Campaign, Application
- Changes Take Effect: Upon loading of the Campaign Group

- Logical Group: Record Processing

Specifies whether OCS should select Personal Callbacks and Personal Rescheduled records from the Calling Lists based on both the Campaign DBID and the Group DBID or just the Campaign DBID. If these records are selected based on both DBIDs, then it is guaranteed that they will be processed by the same group to which the agent belonged when these records where initially scheduled.

- When set to `true`, OCS selects them based on the Campaign DBID and the Group DBID.
- When set to `false`, OCS selects them based on the Campaign DBID only.

Note:	In a scenario in which one or more Campaign Groups share the same Campaign, and thus the same Calling Lists, OCS updates the `group_id` field in the list with the current Group DBID, in addition to the `campaign_id`, when records are marked as `Retrieved`. The ability to identify the Campaign Group name associated with personal records is only available for Calling Lists that have the `group_id` field included in the format. To maintain backward compatibility with Calling Lists using format 6, OCS checks the format for the `group_id` field and the **callback-observe-group** option. If format 6 is being used, the `group-id` field is not updated and OCS ignores this option value.

caller-id-sets

- Default Value: An empty string
- Valid Values: A comma-separated list of Script object names (Caller ID sets)
- Configuration Level: Campaign Group
- Changes Take Effect: Upon Campaign Group activation, for example, when the Campaign Group is loaded
- Logical Groups: Dialing

Specifies a comma-separated list of Script object names (Caller ID sets). It may contain zero, one, or more names of Script objects. Processing is similar to how OCS processes treatment options at the Campaign Group level.

caller-id-rotation

- Default Values: `asc`
- Valid Value: `asc, desc, random`
- Configuration Level: Campaign Group
- Changes Take Effect: Upon Campaign Group activation, for example, when the Campaign Group is loaded
- Logical Groups: Dialing

Specifies the method of Caller ID rotation. The Campaign Group rotates through the available pool of Caller IDs for each phone number. Each time an attempt is made per number, the next available Caller ID from the pool is selected and used as CPNDigits for the call attempt.

- When set to `asc`—sequential, in ascending order—OCS selects the next Caller ID per ASC order defined by option names (integer values).
- When set to `desc`—sequential, in descending order—OCS selects the next Caller ID per DESC order defined by option names (integer values).
- When set to `random`, OCS selects a random Caller ID from the pool.

campaign-callback-distr

- Default Value: `random`
- Valid Values: `random, equal-from-zero`, or `equal-from-avg`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Agent Desktop, Dialing

Specifies how campaign callbacks are distributed to agents when the predictive_callback option is set to `false` and campaign callback notifications are distributed as UserEvents. The option values that define this distribution are as follows:

- For the `random` value, OCS randomly distributes callbacks between eligible agents, as it did previously.
- For the `equal-from-zero` or `equal-from-avg` values, OCS monitors the actual number of campaign callbacks distributed to each agent in the campaign group from the moment the dialing session/campaign group is loaded until it is unloaded. When OCS needs to distribute new campaign callback, it selects an agent who has processed the fewest number of campaign callbacks since the dialing session/campaign group was loaded. If two or more agents have processed the same minimum number, an agent is selected randomly from among them.

If an agent logs out of the group, OCS clears the agent's history, including the number of callbacks processed. When an agent logs into the group, OCS assigns the number of callbacks processed by this agent, as follows:

- If you set the value to `equal-from-zero`, OCS assigns `0` as the number of callbacks processed.
- If you set the value to `equal-from-avg`, OCS assigns the average number of callbacks processed by agents in the group. You might use this value in a scenario in which an agent logs in for the first time at mid-day and want to ensure that all agents receive callback requests. Otherwise, this new agent would receive all requests until he or she reached the group average.

campaigngroup-auto-complete

- Default Value: An empty string
- Valid Values: `no-records`
- Configuration Level: Campaign Group, Application

- Changes Take Effect: Immediately
- Logical Group: Record Processing
- Introduced: 8.1.528.25

If set to `no-records`, OCS automatically completes (stops and unloads) a Campaign Group when all dialing records are depleted for it.

campaigngroup-preload-uri

- Default Value: An empty string
- Valid Values: A valid URI
- Configuration Level: Application
- Changes Take Effect: At the next activation of the schedule item
- Logical Groups: CX Contact, Outbound Schedule

Specifies the URI of the external system (Campaign Manager) to which OCS sends Campaign Group pre-load requests.

campaigngroup-states-interval

- Default Value: `10`
- Valid Values: `0` (zero), a positive integer
- Configuration Level: Campaign Group, Application
- Logical Group: CX Contact, HTTP-based Notifications
- Changes Take Effect: Next attempt to send notification to Web or Application server

Defines the time interval (in seconds) for sending periodic Campaign Group state notifications. If this option is set to zero (`0`), no periodic notifications are sent. (Notifications for Campaign Group state changes continue to be sent). For more information about this feature, refer to Campaign Group Status Reporting.

campaigngroup-states-uri

- Default Value: No default value
- Valid Values: A valid URI
- Configuration Level: Campaign Group, Campaign, Application
- Logical Group: CX Contact, HTTP-based Notifications
- Changes Take Effect: Next attempt to send notification to Web or Application server

Specifies the URI of the Web or Application server to which the Campaign Group status information is being sent. For more information about this feature, refer to Campaign Group Status Reporting.

campaign_name_field

- Default Value: An empty string ("")
- Valid Values: A valid field name from the Calling List table with an appropriate data type to store the campaign name.
- Configuration Level: Calling List, Application
- Logical Group: Record Processing

All calling list records in Genesys Administrator and the Calling List table include the campaign name and the campaign ID. OCS stores the campaign name in a user-defined field in the Calling List table. This option specifies the name of that field in the Calling List table. If the value of the **campaign_name_field** option is any string (not an empty string), when OCS updates the `campaign_id` field with the DBID of the campaign, it also updates the field specified by this value with the name of the campaign.

Note:	Misuse of this option can cause statements from OCS to fail on a SQL Server.

cancel-on-desktop

- Default Value: `true`
- Valid Values: `true, false`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Agent Desktop

Specifies how OCS behaves when it receives a `RequestRecordCancel` request but finds records on an agent's desktop that have the specified phone or the customer ID.

- When set to `true`, OCS finalizes these records as cancelled when it receives a `RequestRecordCancel` request.
- When set to `false`, OCS does not finalize records on an agent's desktops as cancelled when it receives `RequestRecordCancel` request. As a result, agents can complete calls and their associated record that are still in progress, rather than being locked out the record.

OCS also notifies the `RequestRecordCancel` requester (agent desktop or third party application) in the `RecordCancelAcknowledge` response about the inability to handle cancellation request completely. The `GSW_MESSAGE` attribute displays the following message: "Incomplete processing: record(s) on desktop."

Note:	This functionality does not affect `RequestRecordCancel` requests made by the record handle or any `DoNotCall` requests. GSW_MESSAGE is not a new attribute to the desktop protocol, but it is for third-party protocol.

channel_num

- Default Value: `0` (Switch Level); `40` (Application Level)
- Valid Values: `0` or a positive integer
- Configuration Level: Switch, Application
- Logical Group: Dialing

Specifies the total number of available channels (CPD ports) to be used by a Campaign Group or on the switch when using the Predictive or Progressive dialing mode.

- If OCS dials a call through T-Server, the value of this option is taken from the `Options` tab of the `Switch` object.
- If OCS dials through CPD Server, the option is not used. OCS derives the proper number of CPD ports directly from CPD Server through the protocol events.

This option works as follows:

1. Before requesting the next call from T-Server, OCS calculates the number of busy channels on the switch.
2. OCS dials an outbound call when the number of busy channels is less than the value specified for the **channel_num** option.

If the default value `0` (zero) is assigned to **channel_num**, there are no restrictions on the switch, and OCS dials continuously.

Note:	For the dynamic allocation of ports among Campaign Groups running in a VoIP environment, OCS uses the total number of ports, which is reported by SIP Server in the `total-ports` parameter of EventResourceInfo messages, instead of the **channel_num** option that is defined for a Switch object.

check_dnc_callback

- Default Value: `no/false`
- Valid Values: `yes/true`, `no/false`
- Configuration Level: Application
- Changes Take Effect: Immediately
- Logical Groups: Dialing Regulations, Dialing

Enables or disables a special predial check of *callback* records against the Do Not Call (DNC) List. This option applies to the Personal CallBack, Campaign CallBack, and Personal Rescheduled record types. The value set for this option determines whether a scheduled call is processed or not processed, depending at which time the DNC restriction was applied.
If the option is set to `yes` or `true`, OCS identifies the record type and, after comparing the timestamps, determines whether the DNC restriction was applied *after* or *before* a customer agreed

to a scheduled call from an agent:

- If *after*, OCS processes the call.
- If *before*, OCS does not process the call.

See also DNC Rule for Scheduled Calls.

The **check_dnc_callback** option works only if the **check_dnc_list** option is set to `true`. If you set **check_dnc_callback** to `true`, it may override **check_dnc_list** = `true` for the Personal CallBack, Campaign CallBack, or Personal Rescheduled records, based on the timestamp comparison.

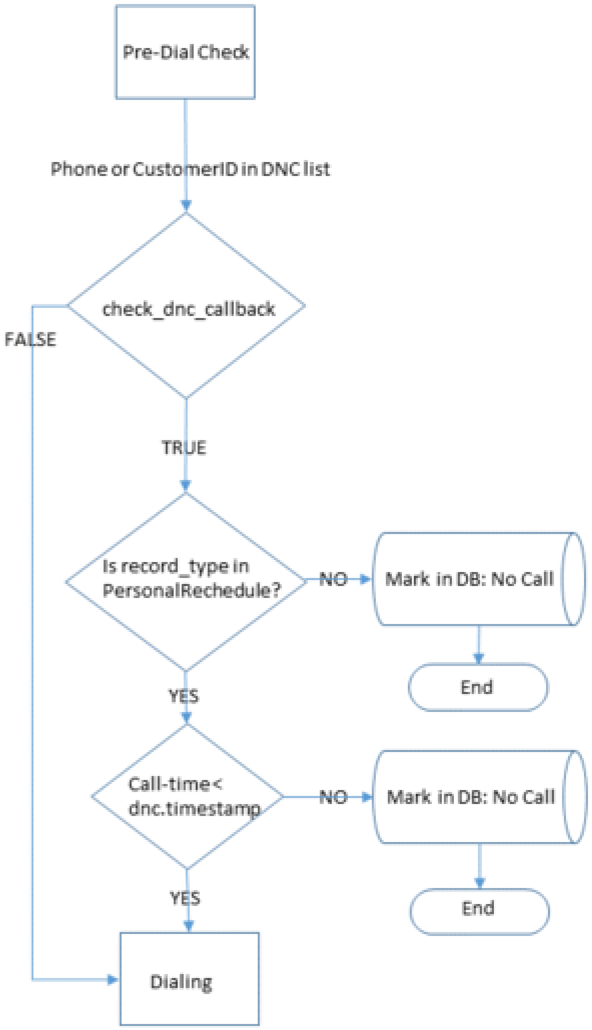
DNC Rule for Scheduled Calls

The purpose of the Do Not Call (DNC) rule for scheduled calls is to determine whether a Personal CallBack, Campaign CallBack, or Personal Rescheduled record can be dialed even though the phone number or customer ID on the record is on the DNC List. The callback might be permissible, for example, if both of the following conditions are met:

- The **check_dnc_callback** option is set to `true`, which enables OCS to perform a predial check of callback type records.
- The customer had requested a callback *before* submitting a `DNC` request.

The algorithm for the predial check is as follows:

1. OCS checks the `DNC` List to determine whether the record to be dialed is on it.
2. If `yes`, OCS reads the value of the `check_dnc_callback` option:
    - If **check_dnc_callback** = `false`, OCS dials the record.
    - If **check_dnc_callback** = `true`, OCS continues the predial check.
3. OCS checks the record type:
    - If the record type is not a callback, OCS marks the record DNC. End of predial check.
    - If the record type is Personal CallBack, Campaign CallBack, or Personal Rescheduled, OCS continues the predial check.
4. OCS checks the time stamp:
    - If the timestamp of the original call (when the callback was scheduled) precedes the timestamp of the DNC request, OCS will dial the callback record. End of predial check.
    - It the timestamp of the original call does not precede the timestamp of the DNC request, OCS will not dial the callback record. End of predial check. The following figure shows the algorithm for the DNC rule for scheduled calls.



DNC Rule for Scheduled Calls

check_dnc_list

- Default Value: `yes/true`
- Valid Values: `yes/true, no/false`
- Configuration Level: Campaign, Application
- Changes Take Effect: Immediately
- Logical Groups: Dialing Regulations, Dialing

Enables or disables a predial check against the Do Not Call list for all types of records.

- If you set this option to `yes` or `true` or if the option is not defined in the configuration, OCS performs a predial check against the `Do Not Call` (DNC) List.
- If this option is set to no or `false`, OCS dials without performing a predial check.

Note:	When set to false, OCS also does not check any records that are already delivered to an agent's desktop against the phone number (or customer_id) in the newly submitted request (by phone or customer_id respectively). Therefore, OCS does not issue a RecordCancel notification to the desktop.

contact-processed-notification

- Default Value: false
- Valid Values: true/false, yes/no
- Configuration Level: Application, Campaign Group
- Changes Take Effect: Immediately
- Logical Group: HTTP-based Notifications
- Introduced: 8.1.528.21

Controls whether OCS generates contact-processed notifications for the chains in a given Campaign Group.

contact-processed-notification-uri

- Default Value: An empty string
- Valid Values: The URI of the remote Web or Application Server with HTTP or HTTPS scheme
- Configuration Level: Application, Campaign Group
- Changes Take Effect: Immediately
- Logical Group: HTTP-based Notifications
- Introduced: 8.1.528.21

Specifies the URI of the remote Web or Application Server with an HTTP or HTTPS scheme. If set to an empty string, OCS does not attempt to send any notification, similar to pre-dial validation.

conversion

- Default Value: ""
- Valid Values: Any value that can be stored in the field for which this option is configured
- Configuration Level: Field
- Changes Take Effect: Immediately
- Logical Group: Historical Reporting

Marks the field that indicates that the answered call was a successful transaction. If it is updated with a value equal to what is configured in this option's value, it will be recognized by Genesys Info Mart as a conversion indicator.

Note:	This option should be specified for only one field within a calling list.

cpd-on-connect

- Default Value: `false`
- Valid Values: `yes/true, no/false`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Outbound Contact VoIP Dialing Modes, Dialing

Specifies when call progress analysis is started in Outbound VoIP dialing modes. If value is set to `no/false`, OCS instructs Media Server to start call progress analysis as soon as the media stream is available. If set to `yes/true`, OCS instructs Media Server to delay call progress analysis until the call is answered.

cpd-recording

- Default Value: `false`
- Valid Values: `yes/true, no/false`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Outbound Contact VoIP Dialing Modes, Dialing

(For VoIP dialing modes) Enables or disables the recording of the call progress detection phase of the call. If value is set to `no` or `false`, the recording is disabled; otherwise it is enabled.

CPNDigits

- Default Value: Empty string
- Valid Values: String of characters, according to the formats specified in the appropriate numbering/ dialing plan.
- Configuration Level: Individual record and chain of records (via SCXML), Calling List, Campaign Group, Application
- Logical Groups: Dialing Regulations, Dialing

Specifies the Calling Party Number (CPN).

Notes:	The SCXML setting at the chain/record level has a higher priority.

	• Starting in release 8.0.001, this option can be configured for individual records or chain-of-records to enable fine-tuning of record processing. Unlike other options, which can be configured at the individual record level in the Business Attributes object, this option is defined using SCXML treatments, which provides greater flexibility in the calculation of the option value and enables the definition of complicated business logic, based on that calculation. For example, record property, time of day, day of the week, or any other parameter that can be calculated inside the SCXML treatment executable content (JavaScript) can now affect the value of this option. • Starting in release 7.6, this option can be applied to a campaigns running in Proactive Contact (Power GVP or Progressive GVP) mode. Its value can be passed to OBN Manager if GVP 7.5 or 7.6 is used or SIP Server if GVP 8.0 or later is used.

For more information about the Caller ID Support feature and the CPN options, see the following resources:

- Extensions related to `TMakePredictiveCall()` in the "Extensions" section of the "Unstructured Data" chapter in the *Genesys Developer Program T-Library SDK Developer's Guide*
- *Framework T-Server for Alcatel A4400/OXE Deployment Guide*
- *Framework T-Server for Avaya Communication Manager Deployment Guide*
- Q.931 ISDN user-network interface layer 3 specification for basic call control

CPNDisplayName

- Default Value: An empty string (`""`)
- Valid Values: Any string
- Configuration Level: Individual record and chain of records (via SCXML), Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Dialing

Defines the value of the `DisplayName` attribute used by SIP Server, for which its behavior is identical to its own **display-name** option setting. This option takes effect only when OCS is dialing using SIP Server or using CPD Server in HMP Transfer mode.

CPNPlan

- Default Value: `0`

- Valid Values: `0-15`
- Configuration Level: Campaign Group, Application
- Logical Group: Dialing

Specifies the decimal representation of the Numbering Plan code.

CPNPresentation

- Default Value: `0`
- Valid Values: `0-3`
- Configuration Level: Campaign Group, Application
- Logical Group: Dialing

Specifies the decimal representation of the Presentation Indicator code.

CPNScreening

- Default Value: `0`
- Valid Values: `0-3`
- Configuration Level: Campaign Group, Application
- Logical Group: Dialing

Specifies the decimal representation of the Screening Indicator code.

CPNType

- Default Value: `0`
- Valid Values: `0-7`
- Configuration Level: Campaign Group, Application
- Logical Group: Dialing

Specifies the decimal representation of the Type of Number code.

customer_id

- Default Value: An empty string (`""`)
- Valid Values: `<The name of a user-defined field in the calling list to serve as a customer identifier>`
- Configuration Level: Campaign, Application
- Logical Group: CX Contact, Dialing

Specifies a user-defined field in the Calling List table that will serve as a customer identifier. When this option is not present in the Application object, or if the value of this option is the name of a field that does not exist in the Calling List table, OCS uses the phone number in order to determine which records cannot be dialed.

Notes:	• Configuring customer ID at the Campaign level allows you to define distinct customer ID fields for different campaigns. • The Field configuration object that is used as the value for the **customer_id** option must specify the send_attribute key-value pair listed on its Options tab. The section name is created according to the section naming convention that is described in the Section Names topic. The value of the send_attribute key-value pair must be GSW_CUSTOMER_ID. For more information, see Attaching Record Information to Desktop and OCS User Events in the *Outbound Contact Reference Manual*. • This option can be used to identify the customer for both Do Not Call requests and Record cancellation requests. For more information about its use for Do Not Call requests, see User-Defined Field for Do Not Call. For more information about its use for Record Cancellation requests, see Record Cancellation Requests and Customer ID.

db-stat-recalc-interval

- Default Value: 180
- Valid Values: Integers from 5 to maximum integer
- Changes take effect: Immediately
- Configuration Level: Campaign, Application
- Logical Group: Real-Time Reporting
- Introduced: 8.1.300.18

Specifies the frequency, in seconds, of Reporting Stored Procedure execution. In cases where the Reporting Stored Procedure is still executing when the time comes to execute it again, that execution is skipped.

Note:	Overly frequent executions of the Reporting Stored Procedure might affect the performance of the database, which stores calling lists, and so might affect OCS dialing.

default_campaigncallback_priority

- Default Value: `1;1`
- Valid Values: Any two non-negative integers separated by a semicolon.
- Changes Take Effect: After Campaign is reloaded and restarted.
- Configuration Level: Application, Campaign, or Campaign Group
- Logical Group: Prioritization of Record Types

Specifies the priority of record types and the number of records to fetch for dialing. The two numbers that make up the value of the option are divided by a semicolon and represent the following two parameters:

- `priority`: 1 is the highest priority; 0 (zero) means do not dial. Recommended values are 0 (zero), 1, 2, or 3.
- `n_records`: A valid value is any positive number or 0 (zero). 0 (zero) means do not dial.

See Flexible Prioritization of Record Types for more information.

default_campaignrescheduled_priority

- Default Value: `1;1`
- Valid Values: Any two non-negative integers separated by a semicolon.
- Changes Take Effect: After Campaign is reloaded and restarted.
- Configuration Level: Application, Campaign, or Campaign Group
- Logical Group: Prioritization of Record Types

Specifies the priority of record types and the number of records to fetch for dialing. The two numbers that make up the value of the option are divided by a semicolon and represent the following two parameters:

- `priority`: 1 is the highest priority; 0 (zero) means do not dial. Recommended values are 0 (zero), 1, 2, or 3.
- `n_records`: A valid value is any positive number or 0 (zero). 0 (zero) means do not dial.

See Flexible Prioritization of Record Types for more information.

default_general_priority

- Default Value: `1;2`
- Valid Values: Any two non-negative integers separated by a semicolon.
- Changes Take Effect: After Campaign is reloaded and restarted.

- Configuration Level: Application, Campaign, or Campaign Group
- Logical Group: Prioritization of Record Types

Specifies the priority of record types and the number of records to fetch for dialing. The two numbers that make up the value of the option are divided by a semicolon and represent the following two parameters:

- `priority`: 1 is the highest priority; 0 (zero) means do not dial. Recommended values are 0 (zero), 1, 2, or 3.
- `n_records`: A valid value is any positive number or 0 (zero). 0 (zero) means do not dial.

See Flexible Prioritization of Record Types for more information.

desktop-chains-limit

- Default Value: 1
- Valid Values: Integer, from 1 to `MAXINT`
- Configuration Level: Agent Group, Application
- Changes Take Effect: Next Campaign Group activation
- Logical Group: Agent Desktop

Controls how many preview interactions can reside simultaneously at an agent's desktop. A preview interaction is a chain of records delivered to an agent's desktop by a User Event, such as in Preview mode, Direct Push Preview Mode, Personal Callback (in any mode), Personal Rescheduled (in any mode), or Campaign Callback (if configured to be delivered as a User Event).

Note:	This option affects only delivery of interactions in Direct Push Preview mode. This option does not limit interactions delivery in other modes, such as Preview. This option also does not limit callbacks and rescheduled records delivery.

dial_log_buffer

- Default Value: 16384 (bytes)
- Valid Values: `2-32768`
- Configuration Level: Calling List, Application
- Changes Take Effect: When the next dial log file is opened
- Logical Group: Dial Log

Specifies the size of the buffer used for the `Record History` Log file output, and the frequency within which OCS updates the `History` Log file.

The smaller the setting of the size limit for the OCS buffer, the more frequently OCS must empty the buffer by unloading the data into the file.

dial_log_delimiter

- Default Value: An empty string ("")
- Valid Values: Any character combination
- Configuration Level: Calling List, Application
- Changes Take Effect: When the next dial log file is opened
- Logical Group: Dial Log

Specifies the delimiters that are used between the fields of the log for the calling list (the Record History Log text file). Because the flat files are produced for this option, delimiters must be added to the log. The default value is an empty string. If the value of the option is an empty string (string length of 0), the ASCII TAB character (code 9) is used.

You can place this option in the OCS `Application` object or in a specific list. OCS first looks for the option in the `OCServer` section on the `Options` tab of a list. If it does not find the option there, it looks on the `Option` tab (`OCServer` section) of the application object. For additional information, see Log Options Defined.

dial_log_destination

- Default Value: An empty string ("")
- Valid Values: `<directory path>`
- Configuration Level: Calling List, Application
- Changes Take Effect: When the next dial log file is opened
- Logical Group: Dial Log

Specifies the path to the directory that stores a log for the calling list (the Record History Log text file).

You can put this option in the OCS `Application` object or in a specific list. OCS first looks for the option in the OCServer section on the `Options` tab of a list. If it does not find the option there, it looks on the `Option` tab (OCServer section) of the `Application` object. If OCS still does not find the option, then OCS does not maintain a log for the list. For additional information, see Log Options Defined.

dialer-num-attempts

- Default Value: 3
- Valid Values: 1-25
- Configuration Level: Campaign Group, Application
- Logical Groups: GVP, Dial Log

Specifies the number of attempts that OBN Manager will attempt to pass a request to the GVP dialer (IPCS or VCS) for a dial from it's internal dialing queue.

Note:	This option only applies when OCS uses OBN Manager for dialing. It does not apply when OCS uses T-Server for dialing and PBX equipment for call progress detection.

dialer-ttl

- Default Value: 5
- Valid Values: `3-1440`
- Configuration Level: Campaign Group, Application
- Logical Groups: GVP, Dialing

Specifies the Time To Live (in minutes) during which OBN Manager will attempt to pass a request to the GVP dialer (IPCS or VCS) for a dial from it's internal dialing queue.

Note:	This option only applies when OCS uses OBN Manager for dialing. It does not apply when OCS uses T-Server for dialing and PBX equipment for call progress detection.

dialing_rate_limit

- Default Value: `100`
- Valid Values: `0-N`
- Configuration Level: Campaign Group, Switch, Application
- Logical Group: Dialing

If configured at the Campaign Group or Application level, this option specifies the maximum number of dialing requests per second. If set to `0` (zero), OCS does not dial at all.

If configured at the Switch level, this option specifies the maximum number of calls per second that are shared between all Campaign Groups running in the auto dialing modes (Progressive, Predictive, Progressive with seizing, Predictive with seizing, and Predictive GVP) that use the switch. The option does not apply to engage call legs. If the option is set to `0` (zero), OCS does not dial for any of the Campaign Groups that use the switch.

> **Important**
>
> The values of this option set at the Switch level do not affect the values set at the Campaign Group and Application levels.

digital-dialer-uri

- Default Value: An empty string

- Valid Values: Any valid URI (for example: `http://ccs-ui.usw.gen.com/dialmanager/v2/`)
- Configuration Level: Application
- Changes Take Effect: Immediately
- Logical Group: CX Contact

Specifies the URI of the CX Contact Dial Manager, which will be used as a Dialer for SMS and Email campaigns in Power dialing mode. OCS will expand the URI to a particular endpoint automatically. For example:

-
    - `http://ccs-ui.usw.gen.com/dialmanager/v2/dial/sms/`
    - `http://ccs-ui.usw.gen.com/dialmanager/v2/dial/email/`

digits-detection

- Default Value: `none`
- Valid Values: `none, all, answer, am`
- Configuration Level: Individual record and chain of records (via SCXML), Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Dialing Regulations

Controls whether or not the DTMF detection takes place on the CPD Server side. When set to `true`, OCS instructs the CPD Server to perform the DTMF detection.

Note:	OCS only instructs the CPD Server if: the dialing mode is ASM, and either one or both of the **asm_drop_announcement_data** and **asm_drop_am_announcement_data** OCS options for the specific Campaign Group are set to a non-empty value (that is, the message that will be played is defined before digit detection occurs).

For SIP Server and VoIP Environment, this option instructs OCS to activate the VoiceXML application for the opt-out feature.

digits-detection-pattern

- Default Value: Empty
- Valid Values: Any string of keys (digits)
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Dialing Regulations

Specifies the string of keys (digits) that represents the signal for which OCS awaits before configuring an `Opt-Out` request.

Note:	If the value of this option is an empty string, OCS behaves as if digits-detection = none. OCS does not require an exact match of the configured keys to those received from CPD Server. Instead, OCS treats any string starting from the digits that are configured in this option as a match. This means that user inputs like 9, 99, 9#, and similar inputs match the specified pattern of 9 and cause the configured reaction.

For SIP Server and VoIP Environment, the value of this option is passed to the VoiceXML application using the SIP Headers.

digits-detection-timeout

- Default Value: `0`
- Valid Values: Any non-negative integer
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Dialing Regulations

Specifies the timeout limitation for digit detection. A value of `0` means *do not wait*.

Note:	OCS passes the `OCS_DIGITS_DETECTION_TIMEOUT` pair to CPD Server only if the `OCS_DIGITS_DETECTION` pair is set to a value of 1 in the `req_MakePredictiveCall` configuration option.

For SIP Server and VoIP Environment, the value of this option is passed to the VoiceXML application using the SIP Headers.

digits-reaction

- Default Value:`DoNotCall|AllChain`
- Valid Values: Any string in the following format: `<digit(s)>|<protocol request>[|<flag>]`, consisting of three sections separated by a | symbol, the last of which, is optional. For example, `1|DoNotCall|AllChain, 2|DoNotCall, 3|DoNotCall|RecordOnly`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Dialing Regulations

Specifies how OCS reacts to the call when it receives the opt-out signal. OCS uses the value of this option to build the desktop protocol request after receiving (from the CPD Server) a string of detected digits that matches the string that is specified in the **digits-detection-pattern** option.

This request always targets the phone number or record handle of the record that is being processed. Only `DoNotCall` and `RequestRecordCancel` requests are supported. The `Chain/Record` request is applicable, only if it can be controlled by the end user by using the `AllChain` or `RecordOnly` value in the **digits-reaction** option.

To detect a match for the configured digits received from CPD Server, OCS does not need an exact match. Instead, OCS treats any string as a match, starting from the digit or digits configured in this option. This means that user inputs like 9, 99, 9#, or similar inputs can match and cause the configured reaction.

For SIP Server and VoIP Environment, the value of this option is passed to the VoiceXML application using the SIP Headers.

direct-personal-callback

- Default Value: `true`
- Valid Values: `true/yes, false/no`
- Configuration Level: Campaign Group, Application
- Logical Groups: Interactions Processing in Push Preview Dialing Mode, Agent Desktop

Controls how `personal callback` records of type are processed in the `Push Preview` dialing mode.

- When the value is set to `true`, OCS will submit the personal callback interaction directly to the agent.
- When the value is set to `false`, OCS will submit the personal callback interaction to an interaction queue assigned to the Campaign Group. This allows the Universal Routing Server strategy to distribute this interaction.

Note:	This option is used with Interaction Server in `Push Preview` dialing mode only.

direct-push-preview

- Default Value: `false`
- Valid Values: `true`, `false`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Next Campaign Group activation
- Logical Groups: Interactions Processing in Push Preview Dialing Mode

Controls the dialing mode that OCS uses for a Campaign Group that is started in Push Preview dialing mode. A value of `false` indicates that the Campaign Group will run in Push Preview mode with Interaction Server. A value of `true` indicates that the Campaign Group will run in Direct Push Preview mode, even if Interaction Server is present in the Campaign Group connections.

disposition-code-field

- Default Value: An empty string
- Valid Values: A valid name of the field in the calling list

- Configuration Level: Calling List, Application
- Changes Take Effect: Immediately
- Logical Group: Record Processing, CX Contact
- Introduced: 8.1.528.25

When the option is set to a non-empty string, OCS does the following:

- Uses the specified field name to store a disposition code value provided in the GSW_DISPOSITION_CODE key-value pair to OCS from an agent desktop or from a negative pre-dial validation when updating the record (if **record_save_intermediate-results** = `true`) or when finalizing the record. The disposition code value is stored in the calling list only if it is explicitly provided to OCS in the GSW_DISPOSITION_CODE key-value pair.
- Updates the specified field as set in #1 regardless of the presence of the Field object in the Format object.
- Delivers the field to ICON in record details in GO Chain Events. (Note that the **disposition_code** field is configured without **send_attribute**).

divert_to_unknown_dn

- Default Value: `no/false`
- Valid Values: `yes/true`, `no/false`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Call Processing

Specifies OCS's reaction to an `EventDiverted` with either an unknown `ThirdPartyDN` or without an agent logged in to it.

If set to no or `false`, immediately upon receiving this type of `EventDiverted`, OCS updates the record with the `Stale` call result and the `Agent Error` record status.

If set to yes or `true`, OCS does not update the record, and continues to monitor the outbound call.

Notes:	• If a call is diverted to a DN, on which OCS has not been registered, then the corresponding record will not be updated in a calling list upon call release. Instead, it will stay in a `Retrieved` status until the timer set by the `stale_clean_timeout` option expires. • In release 7.5 and higher, an "unknown DN" is a DN that is either excluded from any `Place` object, or is included in a `Place` object but the `Place` object does not have an associated agent. An agent is associated to a `Place` object when a Campaign Group that is configured with this agent is activated within OCS.

dnc-reread

- Default Value: An empty string ("") (Do Not Call rereads are not active)
- Valid Values: Time and frequency of the Do Not Call rereads in DAYS@HH24:MM:SS format.
- Configuration Level: Table Access Point, Application
- Changes Take Effect: Immediately
- Logical Group: Dialing Regulations

You can set this option in the Annex tab of the Do Not Call lists' Table Access configuration object or the OCS Application configuration object. If the option is set for a specific Table Access object, only the Do Not Call table accessible through this Table Access object is affected. Setting the option at the OCS Application level affects all Do Not Call tables, unless you also configure different values at the Table Access level. In that case, the Table Access setting takes priority for the associated table.

At the configured time, OCS rereads all loaded Do-Not-Call lists into memory. Configure the rereads as follows:

- DAYS is the frequency of updates in days (each Nth day). The value for DAYS must be greater than 0. The option value following the @ sign defines at what time the Do Not Call reread occurs.
- HH (hours) defines the hour when the reread should happen. Format is 24 hours. and can be any value between 0 (midnight) and 23.
- MM (minutes) defines the minutes after the specified hour when the reread should begin. You can enter any value between 0 and 59.
- SS (seconds) defines the seconds after the specified minute when the reread should begin. You can enter any value between 0 and 59.
- OCS calculates the time for the next reread using as a base time the time when last reread occurred or the time when Do Not Call list was first read into memory (if no rereads have yet occurred). For example, if the frequency of rereads is configured at 7 days, OCS rereads the Do Not Call list every 7 days following the initial read.

Related Documentation: Deploying DNC Functionality

dnc-default-duration

- Default Value: 0 (indicates that the DNC duration is indefinite)
- Valid Values: 0 to MAX_INT
- Configuration Level: Campaign Group
- Changes Take Effect: For subsequent DNC requests with the GSW_RECORD_HANDLE attribute
- Logical Group: Dialing Regulations

This option specifies the DNC default duration, in seconds. OCS will only refer to this option if no value is specified in the GSW_DURATION attribute and if GSW_RECORD_HANDLE is specified. If a value is specified for both GSW_DURATION and **dnc-default-duration**, the GSW_DURATION attribute value takes priority.

Related Documentation: Deploying DNC Functionality

dnc-purge

- Default Value: `0`
- Valid Values: `0` to `MAX_INT`
- Configuration Level: Table Access Point, Application
- Changes Take Effect: Immediately
- Logical Group: Dialing Regulations

This option specifies a delay, in seconds, before an expired DNC entry is purged.

- If the value is set to `0`, OCS does not purge expired DNC entries from DNC tables.
- If the value is set to a positive integer, OCS purges expired DNC entries that have expired before the time calculated as the current UTC time minus this value.

The purge occurs after OCS reads the DNC table after restart or after OCS re-reads the DNC table in accordance with the dnc-reread option.

Related Documentation: Deploying DNC Functionality

dnc-use-duration

- Default Value: `false`
- Valid Values: `true`, `false`
- Configuration Level: Application
- Changes Take Effect: After OCS restart
- Logical Group: Dialing Regulations

A value of `false` means that Do Not Call expiration is off. A value of `true` means that Do Not Call expiration is on. In this case, OCS looks for the `expires_at` value in the Do Not Call table.

Related Documentation: Deploying DNC Functionality

dynamic-port-allocation

- Default Values: `false/no`
- Valid Values: `false/no, true/yes`
- Configuration Level: Switch, OCS Application, CPD Server Application
- Changes Take Effect: Immediately
- Logical Group: Predictive Algorithm

Defines whether the dynamic ports allocation is in effect for the Campaign Groups that dial through a specific Switch (when OCS sends dial requests to T-Server) or through a specific CPD Server or Proxy (when OCS sends dial requests to CPD Server or Proxy). The level of the configuration is very important for this option. For Campaign Groups that use a T-Server associated with a specific Switch, it should be configured at that Switch or the OCS Application level. For Campaign Groups that share the same CPD Server/Proxy, it should be configured at the CPD Server or Proxy Application level.

When the value of this option is set to `true`, OCS performs a dynamic allocation of ports for all the Campaign Groups that use the corresponding Switch or CPD Server/Proxy and have the `Number of CPD ports` property set to `0`. This property is defined on the `Advanced` tab of the `Campaign Group` configuration object. When the value of this option is set to `false`, OCS does not perform dynamic port allocation among any Campaign Groups and all ports are assigned statically.

OCS dynamically allocates the total number of ports only among running Campaign Groups. To retain static allocation and reserve ports for specific Campaign Groups, the `Number of CPD ports` property for these Campaign Groups should be set to a value that is greater than `0`. As such, the total number of dynamically allocated ports for Campaign Groups is the total number of ports available for the corresponding Switch or CPD Server or Proxy minus the sum of ports reserved for running Campaign Groups with a static allocation of ports.

If OCS dials through T-Server and the `dynamic-port-allocation` option is defined at the Switch level, the total number of available ports is taken from the channel_num option that is specified in the `Annex` tab of the `Switch` object. If OCS dials through CPD Server/Proxy, OCS receives the total number of available ports directly from CPD Server or Proxy. In the latter case, the number of available ports is defined by the CPD Server or Proxy configuration.

To protect Campaign Groups with small estimated numbers of required ports relative to other Campaign Groups, OCS uses these small estimated values as the maximum number of available ports, without further scaling these numbers. OCS considers the estimated number of ports to be small if this number is less than 30 percent (%) of the average ports distributed between all participating Campaign Groups.

When the **dynamic-port-allocation** option is set to `false`, OCS uses a static allocation of ports in accordance with the `Number of CPD ports` Campaign Group property and the **asm_channel_num** option for all Campaign Groups that share the corresponding Switch or CPD Server/Proxy. For more information, see the **asm_channel_num** option and Separate Tracking of Engaging CPD Ports topic.

encoding

- Default Value: none
- Valid Values: (See the ICU Home Converter Explorer page for a list of valid values.)
- Configuration Level: Application
- Changes Take Effect: After OCS restart
- Logical Group: HTTP-based Notifications

OCS sends this option value to an HTTP proxy, which can then transcode the JSON body of HTTP requests or responses according to the settings specified. If the option value is not specified, no conversion takes place.

To change the default, set this option to the name of a converter that can translate UTF-8 data to the local character set. Converters can be found at ICU Home Converter Explorer.

engaged_answer_action

- Default Value: `soft_answer`
- Valid Values: `hard_not_ready, hard_ready, soft_answer`
- Configuration Level: Switch, Application
- Changes Take Effect: Immediately
- Logical Groups: Agent Desktop, ASM Dialing

Determines the agent's place state after an engaging call is established.

- When set to `hard_not_ready`, OCS sends a request to T-Server to force the teleset to a `Not Ready` state.
- When set to `hard_ready`, OCS sends a request to T-Server to force the teleset to a Ready state.
- When set to `soft_answer`, OCS uses the agent state provided by Stat Server.

Note:	When using this option, OCS takes the value of the hard_request_to_login_dn option into account. In other words, if **hard_request_to_login_dn is set to `true`, OCS sends `RequestAgentReady` to the login DN instead of the DN where the call was answered.**

engaged_release_action

- Default Value: `soft_previous`
- Valid Values: `soft_previous, hard_ready, hard_not_ready`
- Configuration Level: Switch, Application
- Changes Take Effect: Immediately
- Logical Groups: Agent Desktop, ASM Dialing

Determines the agent's place state after an engaging call has been released.

- When set to `soft_previous`, OCS uses the agent state provided by Stat Server.
- When set to `hard_ready`, OCS sends a request to T-Server to force the teleset to a Ready state.
- When set to `hard_not_ready`, OCS sends a request to T-Server to force the teleset to a `NotReady` state.

Note:	When using this option, OCS takes the value of the hard_request_to_login_dn option into account. In other words, if **hard_request_to_login_dn** is set to `true`, OCS sends `RequestAgentReady` to the login DN instead of the DN where the call was answered.

force-unload-wait-db

- Default Value: `false`
- Valid Values: `true/yes, false/no`
- Configuration Level: Campaign, Application
- Logical Group: Record Processing
- Changes Take Effect: The next time that the dialing session for this campaign is unloaded forcefully

Controls how OCS handles requests for a records update sent to the database when a dialing session for the campaign is unloaded forcefully. When set to `false/no`, OCS does not wait for a database response for the records update request and terminates the connection to DB Server immediately after the last update request is sent. This ensures that the dialing session/campaign group is forcefully unloaded almost immediately. However, this may cause some records to be left in the `Retrieved` state, even though OCS has requested that those records be returned to either the Ready state or marked as `Stale`.

When set to `true/yes`, OCS waits for database responses for all update requests that were sent before terminating the connection to DB Server. This ensures that all records are updated properly in the database. However, this may slow down the forced unloading process and keep the dialing session/campaign group in an `Unloading` state longer, depending on the number of calling list records in the OCS buffers. OCS holds these records in the buffers to properly mark them when unloading a dialing session/campaign group.

gvp-modes-use-switch-rate-limit

- Default Value: `no/false`
- Valid Values: `yes/true, no/false`
- Configuration Level: OCS Application
- Logical Group: Dialing

When set to `true/yes`, Progressive GVP and Power GVP dialing modes are added to the list of auto dialing modes, which can use dialing rate limits defined at Switch level.

hard_request_to_login_dn

- Default Value: `no/false`
- Valid Values: `yes/true, no/false`
- Configuration Level: Switch, Application
- Changes Take Effect: Immediately
- Logical Group: Agent Desktop

Determines which DN receives the `AgentReady` or `AgentNotReady` request when an agent's `Place` in the database has more than one DN associated with it.

You can use this option when Outbound Contact is configured as follows:

- An agent's Place is configured with an ACD Position and an Extension.
- The outbound_release_action option is set to hard_ready or hard_not_ready.

When both of these conditions are met, OCS sends an AgentReady or AgentNotReady request after receiving an EventReleased message on either of the agent's DNs, depending on the value of this option.

- When set to yes or true, OCS sends an AgentReady or AgentNotReady request to the DN where the agent is logged in.
- When set to no or false, OCS sends an AgentReady or AgentNotReady request to the DN that received the EventReleased message.

history_length

- Default Value: 30
- Valid Values: Any positive integer from 30 to 300
- Configuration Value: Campaign Group, Application
- Logical Group: Predictive Algorithm

Specifies the point at which OCS switches from the Progressive mode to Predictive mode. As soon as OCS starts dialing in Progressive mode, it begins to collect call flow statistics or call history to create predictive statistics.

OCS fills the call history in the order in which calls are established (answered by dialed party), meaning if the **history_length** option has value of 40, the *first* 40 established calls (not 40 randomly-chosen established calls) must be completed before the Campaign switches to predictive mode.

For large agent groups (approximately 1000 agents), Genesys recommends that you set the value of **history_length** to 100 or higher, and that you also control the number of CPD ports.

These settings help maintain the target busy factor and minimize the number of abandoned calls in large agent groups.

history-uri

- Default Value: none
- Valid Values: URI
- Configuration Level: Calling List, Campaign, Application
- Logical Groups: Record Processing, HTTP-based Notifications, CX Contact
- Changes Take Effect: Next dial attempt, or next attempt to send the Call Report Record to Web or Application Server, if the previous attempt failed

Specifies the URI of the Web or Application Server.

Related Documentation: Submitting Call Report Records

history-wait-condition-threshold

- Default Value: `0`
- Valid Values: Positive Integer
- Minimum Value: `0`
- Configuration Level: Campaign Group, Application
- Logical Groups: Record Processing, HTTP-based Notifications, CX Contact
- Changes Take Effect: Next dialing session

Specifies the number of calls for which at least one call report record failed, which triggers a wait condition for the OCS dialing session (see Notes).

Notes:	• A record is considered failed if it posted to Web or Application Server but failed to save as a result of error response or timeout. • A value of `0` means that any call for which at least one call report record failed triggers a wait condition. • When a wait condition occurs, a campaign remains in running state but dialing is suspended. • A wait condition is cleared when all call report records marked for re-submission are successfully processed.

Related Documentation: Submitting Call Report Records

history-wait-recovery-timeout

- Default Value: `30` (seconds)
- Valid Values: Positive integer, starting from 5
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Next dialing session
- Logical Groups: Record Processing, HTTP-based Notifications

Specifies the time to wait before re-submitting the failed call report record.

Related Documentation: Submitting Call Report Records

http-body

- Default Value: An empty string
- Valid Values: JSON string (and support for macro expressions)
- Configuration Level: Treatment
- Changes Take Effect:
- Logical Groups: HTTP-based Notifications

Specifies the HTTP body. All macro expressions are expanded in the same manner as for the Execute SQL Statement treatment, especially record-level macro parameters, such as $GSW_PHONE and $GSW_CUSTOMER_ID. See Send HTTP Request Action Type for details.

Example:

```
{
"data": {
"listid": 5711,
"appendOnly": false,
"fields": {
"client_id": "$GSW_CUSTOMER_ID"
}
}
}

{
"data": {
"listid": 5711,
"appendOnly": true,
"fields": {
"device": "$GSW_PHONE"
}
}
}
```

http-connection-pool-size

- Default Value: 64
- Valid Values: Positive integer
- Configuration Level: Host, Application
- Changes Take Effect: Next dial attempt
- Logical Groups: HTTP-based Notifications, Pre-dial Validation, CX Contact

Defines the maximum number of simultaneous connections to the Web or Application Server running on the specified host at the specified port number.

Note:	OCS parses URI and detects the name of the host at which Web or Application Server is to be contacted. OCS then attempts to find the host configuration object with the name that exactly matches the host setting from the URI. If found,

	OCS attempts to read this option from the Annex of the host configuration object. If not found, OCS reads this option from the OCS Application object.

http-custom-headers

- Default Value: An empty string
- Valid Values: Custom HTTP headers in the following format:
  `<header1_name>: <header1_value>[|<header2_name>: <header2_value>...]`
- Configuration Level: Application, Host
- Changes Take Effect: After OCS Application restart
- Logical Groups: HTTP-based Notifications, CX Contact
- Introduced: 8.1.516.04

Specifies the custom HTTP headers. In the valid values format, the header name and its value are delimited with a colon followed by a space (:), and headers are delimited with a pipe (|). A semicolon and colon could be part of the header value. When set at the Host level, it must be configured on the Host object representing the host of an external Web or Application Server. The Host-level setting applies only to outgoing HTTP requests and takes precedence over the Application-level setting.

Example: **http-custom-headerst** = `X-Frame-Options: SAMEORIGIN|X-XSS-Protection:1; mode=block|X-Content-Type-Options: nosniff|Content-Security-Policy: script-src 'self'; object-src 'self'|Strict-Transport-Security: max-age=31536000; includeSubDomains`

http-headers

- Default Value: An empty string
- Valid Values: Exactly as specified in http-custom-headers
- Configuration Level: Treatment
- Changes Take Effect:
- Logical Groups: HTTP-based Notifications

Specifies the HTTP headers. All macro expressions are expanded in the same manner as for the Execute SQL Statement treatment. Note that the value of this option should be combined with the Application-level option **http-custom-headers**, in case if it's present. See Send HTTP Request Action Type for details.

http-method

- Default Value: POST
- Valid Values: POST | PUT | GET | DELETE
- Configuration Level: Treatment

- Changes Take Effect:
- Logical Groups: HTTP-based Notifications

Specifies the HTTP method. No support for macro expressions. See Send HTTP Request Action Type for details.

http-proxy

- Default Value: An empty string
- Valid Values: A string representing the URI of the HTTP tunneling proxy server supporting the HTTP CONNECT method, in the following format: `http://<host>[:<port>][/<path to HTTP proxy server resource>]`
- Configuration Level: Host, Application
- Changes Take Effect: At the next HTTP connection (existing connections will keep using the previous settings)
- Logical Group: HTTP-based Notifications, CX Contact
- Introduced: 8.1.528.05

Specifies the URI of the HTTP tunneling proxy server supporting the HTTP CONNECT method.

http-response-timeout

- Default Value: 3
- Valid Values: `1-3600` (seconds)
- Configuration Level: Host, Application
- Changes Take Effect: Next dial attempt
- Logical Groups: HTTP-based Notifications, Pre-dial Validation, CX Contact

Defines maximum time (in seconds) that OCS will wait for the response from the Web or Application Server for its request for pre-dial validation or SCXML document. The maximum valid value is `3600` seconds. If a higher value is configured, OCS applies the default value of 3.

Note:	OCS parses the URI and detects the name of the host at which Web or Application Server is to be contacted. OCS then attempts to find a host configuration object with the name that exactly matches the host setting from the URI. If found, OCS attempts to read this option from the Annex of the host configuration object. If not found, OCS reads this option from the OCS Application object.

http-retry-applies-to

- Default Value: An empty string

- Valid Values: `validation, contact-processed, pre-load`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Next retry attempt
- Logical Groups: HTTP-based Notifications, Pre-dial Validation, CX Contact
- Introduced: 8.1.528.21

Specifies a comma-separated list of HTTP service types to which the functionality of retrying to send failed HTTP requests are applied. An empty string means that OCS does not resend failed HTTP requests if there are any connectivity errors.

http-retry-max-attempts

- Default Value: `0`
- Valid Values: `0-30`
- Configuration Level: Host, Application
- Changes Take Effect: Next retry attempt
- Logical Groups: HTTP-based Notifications, Pre-dial Validation, CX Contact
- Introduced: 8.1.528.21

Specifies the maximum number of attempts for OCS to resend failed HTTP requests. A value of `0` (zero) means that OCS does not resend failed HTTP requests if there are any connectivity errors.

icon_attribute

- Default Value: `0`
- Valid Values: `0,1,2`
- Configuration Level: Field
- Logical Group: Historical Reporting

Defines how the field's value will be stored by ICON.

- If set to `0`, ICON will not store the value of this field.
- If set to 1, ICON will use a non-secured table.
- If set to 2, ICON will use a secured table.

Note:	For ICON to store a field’s value correctly, you must configure the `icon_attribute`. Depending on the KVP, you might also need to configure the send_attribute option in order for ICON to receive the KVP to store. For more information about configuring the send_attribute option, refer to the Outbound section of the ICON documentation.

Genesys Info Mart requires the following two options to be configured in order to calculate metrics: right_person and conversion.

ignore-empty-group

- Default Value: `false`
- Valid Values: `true/yes, false/no`
- Configuration Level: Campaign, Application
- Changes Take Effect: Immediately
- Logical Group: Record Processing

Specifies how to handle record processing when no agents are logged into group.

- When the value is set to `false`, OCS functions as it did previously and retrieves callback/rescheduled records when the scheduled time approaches, regardless of agent availability, and does not return any records back to the database when a group becomes empty.
- When the value is set to `true`, OCS behaves as follows:
    - When the last agent logs out of the group, OCS returns all records retrieved for the Campaign Group back to the calling list with the Ready status.
    - When there are no agents logged into the group, no records of any type are retrieved for the Campaign Group.

Note:	If agent assignment is activated for the Campaign Group and the agent-reassignment-if-waiting-records option is set to `true` for this Campaign Group, the Campaign Group will behave as follows: • If all agents are reassigned out of the group, all records will be flushed from the buffers. • Whenever records need to be retrieved (each 10 seconds for General records, each 100 seconds for rescheduled records), before doing the actual retrieval, Campaign Group will determine the number of agents which are both: • Configured for this Agent Group; and • Logged In. **Warning:** If this number is 0, Campaign Group will consider itself empty and will not retrieve any records. If this number is greater than 0, Campaign Group will consider itself not empty and retrieve records. Agents may then be naturally assigned to this Campaign Group since it is no longer empty.

ignore-not-monitored

- Default Values: `false`

- Valid Values: `yes/true, no/false`
- Configuration Level: Application
- Changes Take Effect: Immediately
- Logical Groups: Agent Desktop

Specifies whether OCS accepts or ignores a TargetRemoved event with the agent status of NotMonitored that is generated by Stat Server. When set to `false`, OCS processes a TargetRemoved event with the agent status of NotMonitored in the same way as a TargetUpdated event. When set to `true`, OCS ignores a TargetRemoved event with the agent status of NotMonitored as the previous OCS versions did.

Note: Genesys recommends keeping this option set to its default value unless explicitly instructed otherwise by Genesys Customer Care.

inbound-agent-assignment-min-num

- Default Value: `0`
- Valid Values: `-1`, `0` -- `n`
- Configuration Level: Application
- Logical Group: Dynamic Agent Assignments

Defines the minimum amount of agents to perform inbound call activities. It is used when determining agent reassignment.

When the value is set to `0`, there are no restrictions for the minimum amount of agents performing inbound call activities. Agents are assigned to inbound call activities even when the highest priority Campaign Groups are understaffed.

To prevent assignment to the inbound activity, the *inbound-agent-assignment-min-num*' option should be set to special value "`-1`". In this case, OCS assigns agents to the inbound activity only if the agents cannot be assigned to any running associated Campaign Groups due to the restrictions imposed by the options regarding the agent assignment (maximum number of assigned agents, waiting records, and so on.)

inbound-agent-assignment-priority

- Default Value: `0`
- Valid Values: `0` -- `n`
- Configuration Level: Application
- Changes Take Effect: Immediately
- Logical Group: Dynamic Agent Assignments

Defines the priority of inbound call activities. It is used when determining agent reassignment. As the value increases, the priority for the specified agent also increases.

inbound_agent_outlier_limit

- Default Value: `600 (seconds)`
- Valid Values: Any non-negative integer
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Predictive Algorithm

Specifies the minimum amount of inbound call time, in seconds, before OCS changes an agent's status changes to `Not Available`. If an agent is busy with an inbound call longer than the specified value (including the `After Call Work` state), then the predictive algorithm does not consider this agent available when calculating the number of predictive calls.

interaction-media-type

- Default Value: `outboundpreview`
- Valid Values: Any valid value for the media type business attribute
- Configuration Level: Calling List, Application
- Changes Take Effect: When the next interaction is created
- Logical Group: Interactions Processing in Push Preview Dialing Mode

Defines the media type of the interaction submitted to Interaction Server in the `Push Preview` dialing mode.

ivr_group

- Default Value: `false`
- Valid Values: `true/yes`, `false/no`
- Configuration Level: Place Group
- Changes Take Effect: After OCS reloads a dialing session for a campaign group
- Logical Group: Call Processing

Specifies whether OCS provides simplified resource availability management for IVR groups.

When this option is set to `true` or `yes`, OCS uses the simplified resource availability model when running a campaign for a Campaign Group linked to this Place Group. Places in that group can contain DNs of type `ACD Position, Extension`, or `Voice Treatment Port`.

ivr-profile-name

- Default Value: An empty string ("")
- Valid Values: `1-n`

- Configuration Level: Campaign Group, Application
- Logical Group: GVP

Populated after a nine-digit value is entered in the IVR Profile field in the Advanced tab of Campaign Group configuration object. The value itself is the IVR Profile DBID that is specified in the GVP EMPS IVR Profile.

ivr_record_processed

- Default Value: false
- Valid Values: true/false, yes/no
- Configuration Level: Campaign Group, Application
- Logical Group: GVP
- Changes Take Effect: Immediately

This option controls how OCS processes chain finalization in GVP dialing modes. It works similar to the record_processed option, but enables you to configure different OCS processing options for agent-related dialing modes and GVP modes.

Note:	In Progressive GVP dialing mode, if the **ivr_record_processed** option is not set, OCS uses the value in the **record_processed** option.

ivr_update_on_release

- Default Value: false
- Valid Values: true/false, yes/no
- Configuration Level: Place Group
- Changes Take Effect: After set to true or yes, when the **ivr_group** option is set to true or yes.
- Logical Group: Call Processing

Enables OCS to update the calling list record with values from the outbound call's UserData.

When this option is set to true or yes, OCS updates the fields from the calling list record with values from the corresponding UserData key-value pairs that are received in the EventReleased message. This is similar to UpdateCallCompletionStats UserEvent processing.

This option is also used in Power GVP dialing mode with OBN Manager in environments running GVP 7.6. In this dialing mode, when this option is set to true or yes, OCS updates the fields of the calling list record with values from the corresponding UserData key-value pairs that are received in the eventOBNRecordProcessed message from OBN Manager for unsuccessful call results. In this dialing mode, the **ivr_group** option has no effect on the described functionality.

license-file

- Default Value: None
- Valid Values: Any string
- Configuration Level: Application
- Logical Group: Licensing

Specifies the license address. This option is not OCS-specific. It is specified in the license section, which is optional. This option, along with the **num-of-licenses<** option, pertains to OCS license control. The license address format can be entered in either of the following formats:

- The host name and port of the license server, as specified in the SERVER line of the license file, in the port@host format; for example, `7260@ctiserver`
- The full path to, and the exact name of, the license file - for example, `/opt/mlink/license/license.dat`

Note:	Changes that you make to this option take effect after an application is restarted.

lifo-mode

- Default Value: `false`
- Valid Values: `true, false`
- Configuration Level: Campaign Group
- Changes Take Effect: At the next Campaign Group activation
- Logical Group: Dialing

When set to `true`, designates a Campaign Group as a Last In First Out (LIFO) Campaign Group, with a special logic for record processing.

lifo-record-expiration

- Default Value: `3600`
- Valid Value: Any positive integer
- Configuration Level: Campaign Group
- Changes Take Effect: At the next Campaign Group activation
- Logical Groups: Dialing

Specifies the timeout, in seconds, for how long a record is kept in the dialing buffer when Last In First Out (LIFO) mode is enabled (lifo-mode is set to `true`). If the timeout expires but the dialing of a given record is not initiated, the following sequence of events happens:

1. The record is removed from the dialing buffer.
2. The record is assigned a Queue Full (18) call result.
3. A treatment is applied to that call result from a set of supported treatments by LIFO Campaign Groups.
4. The record is finalized in the calling list table.

When this option is set to `0`, records will not be expired in the dialing buffer.

lifo-unload-pace

- Default Value: `30`
- Valid Values: Any positive integer
- Configuration Level: Campaign Group
- Changes Take Effect: At the next Campaign Group activation
- Logical Groups: Dialing

Specifies the number of chains that OCS attempts to process per second while a LIFO Campaign Group is unloading when Last In First Out (LIFO) mode is enabled (the lifo-mode option is set to `true`).

When this option is set to `0`, all chains are processed in the buffers in one cycle and 1-second intervals are not established.

local_file_maxage

- Section: `scxml` in OCS application/Annex
- Default Value: `10000`
- Valid Values: Any integer greater than or equal to `0`
- Changes Take Effect: During startup. Changes to this option are not applied dynamically.
- Logical Group: SCXML-based Treatments

Defines time in milliseconds for how long a SCXML treatment fetched from local file (*file:// type of URL*) is cached. When the same SCXML treatment is to be used for the new session within the configured timeout, the cached version of the SCXML treatment will be used instead of fetching it from local file. If both **maxAge** (or **local_file_maxage** for local files) and **maxStale** are set to `0`, SCXML treatments will not be fetched from cache.

log_call_stats

- Default Value: `no`
- Valid Values: `yes/true, no/false`
- Configuration Level: Application
- Changes Take Effect: Immediately

- Logical Groups: Dialing Regulations, Dial Log

Specifies whether to create a separate logging subsystem. If you set this option to yes or `true`, OCS creates a separate logging subsystem for Audit Logging, in order to capture additional statistics on telephony events. OCS does not overwrite the data in any existing audit log or replace it with new data. The Audit Logging function *add*s data to a cumulative log.

If you are running OCS 8.1.0 or earlier, the separate log file is created in the same directory where the OCS application resides. When the size of this log file reaches 10 MB, a new log file of the same type is created.

If you are running OCS 8.1.1 or later, the location of the log file and the maximum size of a log segment are defined by the log options `all` and `segment`. You can also specify an `expire` option, which defines a maximum number of log file segments or a time interval after which the segments of the log file are deleted. These options are contained in the configuration section log_call_stats.

Note:	Be careful not to confuse this option **log_call_stats**, which activates Audit Logging, with the section **log_call_stats**, which contains the options used to configure Audit Logging.

If you set the option to `no, false`, or if the option is not present, OCS does not create the separate logging subsystem.

maxAge

- Section: **scxml** in OCS application/Annex
- Default Value: `0`
- Valid Values: Any integer greater than or equal to `0`
- Changes Take Effect: During startup. Changes to this option are not applied dynamically.
- Logical Group: SCXML-based Treatments

Defines time in seconds for how long a http-fetched SCXML treatment is cached. When the same SCXML treatment is to be used for the new session within the configured timeout, the cached version of the document will be used instead of fetching it from the application server.

If both **maxAge** (or **local_file_maxage** for local files) and **maxStale** are set to `0`, SCXML treatments will not be fetched from cache.

maxStale

- Section: **scxml** in OCS application/Annex
- Default Value: `0`
- Valid Values: Any integer greater than or equal to `0`
- Changes Take Effect: During startup. Changes to this option are not applied dynamically.
- Logical Group: SCXML-based Treatments

Defines the time in seconds to extend the lifetime of a cached SCXML treatment. For example, if the value of the maxAge option is set to `60` seconds and the value of the maxStale option is set to `30` seconds, then the fetched SCXML treatment will be cached for the timeframe of `90` seconds. If both **maxAge** (or **local_file_maxage** for local files) and **maxStale** are set to `0`, SCXML treatments will not be fetched from cache.

merge-method

- Default Value: `bridging`
- Valid Values: `bridging`, `transfer`
- Configuration level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Outbound Contact VoIP Dialing Modes, ASM Dialing

Defines the OCS merge method for outbound and engaging calls in the ASM mode when dialing using SIP Server.

- When set to `transfer`, OCS connects the customer call to the longest-waiting engaging call across all Media Servers. See Transfer Method for details.
- When set to `bridging`, OCS attempts to find an established engaging call on the same Media Server as the customer call.

If there is no established engaging call on the same Media Server, OCS releases the customer call, with the option to play a configured announcement before releasing it, according to the asm_drop_announcement_data option.

If bridging the customer call is not possible in this scenario, you can customize this setting using the on-bridging-unable option at the Campaign Group-Level. Using this option, OCS connects the customer call to the first established engaging call, regardless of the Media Server on which the engaging call is established using the transfer method in `TMergeCall`.

num-of-licenses

- Default Value: `0 (`all available licenses`)`
- Valid Values: The string max or an integer from `0` to `(9999 + num-sdn-licenses)`
- Configuration Level: Application
- Logical Group: Licensing

Specifies how many licenses OCS checks out initially. This option is not OCS-specific. It is specified in the **license** section. This option, along with the **license-file** option, pertains to OCS license control. When the value increases, OCS will apply the change immediately. When the value decreases, it will take effect the next time OCS is started.

ocs-urs-interact

- Default Value: `false`

- Valid Values: `true, yes, no, false`
- Configuration Level: DN (only Communication DN type)
- Changes Take Effect: Next time a "Claim Agent" request is received by OCS
- Logical Group: Dynamic Agent Assignments

When set to `true` or yes, the specified Communication DN will be used for the "Claim Agent" request delivered by URS to OCS. OCS ignores "Claim Agent" requests if they are distributed for a DN that is not configured for this option or if this option is set to the value `false` or no for the specified DN.

on-asm-overdial

- Default Value: `drop`
- Valid Values: `transfer`, `drop`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Outbound Contact VoIP Dialing Modes, ASM Dialing, Call Processing

Defines OCS's behavior in Progressive ASM and Predictive ASM dialing modes in a VoIP environment when an outbound call cannot be bridged immediately to an established engaging call. If configured as `drop`, the outbound call is dropped, optionally with an announcement (see asm_drop_announcement_data and asm_drop_am_announcement_data options). If configured as `transfer`, OCS attaches the `GSW_ASM_OVERDIAL=1` key-value pair to the outbound call and uses single-step transfer to deliver the call to the Voice Transfer Destination as defined by the Campaign Group configuration. However, no announcement will be played, even if the announcement options are configured.

Note:	If the transfer attempt returns any errors, OCS drops the outbound call and finalizes the record.

> **Tip**
>
> The `GSW_ASM_OVERDIAL` key-value pair can be used in the routing strategy to distinguish overdialed outbound calls and process them in a separate logical branch of the strategy.

on-bridging-unable

- Default Value: `drop`
- Valid Values: `transfer`, `drop`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Outbound Contact VoIP Dialing Modes, ASM Dialing

Defines OCS's behavior when there is no established engaging call on the same Media Server and it is not possible to use the bridging method.

on-schedule-disable

- Default Value: `next_cycle`
- Valid Values: `next_cycle`, `disable_actions`, `terminate`
- Configuration Level: Outbound Schedule (`default` section only), OCS Application
- Changes Take Effect: The next time an Outbound Schedule is disabled.
- Logical Group: Outbound Schedule

Specifies how OCS responds to a dynamic change to the Outbound Schedule, specifically, when an Outbound Schedule configuration object is marked as disabled.

- When set to `next_cycle`, OCS continues executing Schedule actions; disabling takes effect on the next execution cycle of the Schedule (it is not activated).
- When set to `disable_actions`, OCS stops executing further schedule actions without changing the states of Schedule items.
- When set to `terminate`, OCS stops executing further Schedule actions, and sends the Complete command (stop, if required, followed by unload) to all active or running Schedule items.

Regardless of the option value, OCS does not activate a disabled Outbound Schedule at the next activation time. For more information, see Dynamic Disabling of Schedules.

Note:	OCS applies any other dynamic changes to schedule parameters at the next activation.

ocs-urs-broadcast

- Default Value: `false`
- Valid Values: `yes/true`, `no/false`
- Configuration Level: Application
- Changes Take Effect: Immediately
- Logical Groups: Dynamic Agent Assignments

Controls how OCS handles ClaimAgentsFromOCS requests received from URS over a Communication DN. If this option is set to `true`, OCS disregards the Target Application ID parameter of a ClaimAgentsFromOCS request and processes this request as if a Target Application ID of the request equals the Application ID of the processing OCS. If this option is set to `false`, OCS processes a ClaimAgentsFromOCS request only if the Target Application ID parameter of the request matches the actual Application ID of the processing OCS.

outbound_agent_outlier_limit

- Default Value: `600` (seconds)
- Valid Values: Any non-negative integer, from `0` to the maximum integer supported by your operating system
- Configuration Level: Campaign Group, Application
- Logical Group: Predictive Algorithm

Specifies the minimum amount of outbound call time, in seconds, before OCS changes an agent's status changes to `Not Available`. If an agent is busy with an outbound call longer than the specified value (including the `After Call Work` state), then the predictive algorithm does not consider this agent available when calculating the number of predictive calls.

Note:	OCS takes this option into account regardless of the setting for the predictive-longcalls-truncation option.

outbound_answer_action

- Default Value: `soft_answer`
- Valid Values: `hard_not_ready, hard_ready, soft_answer, hard_acw`
- Configuration Level: Switch, Application
- Changes Take Effect: Immediately
- Logical Group: Agent Desktop

Determines the agent's place state after an outbound call is established.

- When set to `hard_not_ready`, OCS sends a request to T-Server to force the teleset to a `Not Ready` state.
- When set to `hard_ready`, OCS sends a request to T-Server to force the teleset to a Ready state.
- When set to `soft_answer`, OCS uses the Agent State provided by Stat Server.
- When set to `hard_acw`, OCS sends a request to T-Server to force the teleset to the After Call Work state after an outbound call is established on an agent's DN.

Note:	When using this option, OCS takes the value of the hard_request_to_login_dn option into account. In other words, if **hard_request_to_login_dn** is set to `true`, OCS sends `RequestAgentReady` to the login DN instead of the DN where the call was answered. The hard_acw value was added in OCS version 7.6.101.29.

outbound_contact_server

- Default Value: `undefined`
- Valid Values: yes/`true, no/false`, `undefined`
- Configuration Level: DN
- Changes Take Effect: Immediately
- Logical Group: Real-Time Reporting

Controls which Communication DNs are used by OCS.

Applying this option to the Communication DNs (CommDNs) reduces the number of DNs used for OCS. OCS uses CommDNs to communicate with Stat Server and third-party applications. Set this option to `true` or `yes` if you want OCS to communicate with a third-party application through this DN. The value `undefined` has the same effect as if this option is absent.

outbound_release_action

- Default Value: `soft_previous`
- Valid Values: `hard_ready, hard_not_ready, soft_previous, hard_acw`
- Configuration Level: Switch, Application
- Changes Take Effect: Immediately
- Logical Group: Agent Desktop

Determines the agent's place state after an outbound call is released.

- When set to `hard_ready`, OCS sends a request to T-Server to force the teleset to the Ready state.
- When set to `hard_not_ready`, OCS sends a request to T-Server to force the teleset to the `Not Ready` state.
- When set to `soft_previous`, OCS uses the Agent State provided by Stat Server.
- When set to `hard_acw`, OCS sends a request to T-Server to force the teleset to the After Call Work state after an outbound call is released from an agent's DN.

Note:	When using this option, OCS takes the value of the hard_request_to_login_dn option into account. In other words, if `hard_request_to_login_dn` is set to `true`, OCS sends `RequestAgentReady` to the login DN instead of the DN where the call was answered. The hard_acw value was added in OCS version 7.6.101.29.

overflow_dn

- Default Value: `false`

- Valid Values: `yes/true, no/false`
- Configuration Level: DN
- Logical Group: Call Processing

Designates a DN to which the switch reroutes overflow calls. Outbound calls that are answered by the called party but remain in an ACD Queue too long before an agent answers are recognized by the switch as overflow calls. If this option is set to `true` or `yes`, Outbound Contact treats such call as overflow calls. Outbound Contact supports two different methods of handling these overflow calls. The DN configuration in Genesys Administrator determines which method OCS uses. The two methods are as follows:

- If the overflow DN is an `Extension/ACD Position` DN, OCS instructs T-Server to release the call. To use this method, set the **overflow_dn** option to `true` on the `Options` tab of the `Extension/ACD Position` DN to which the switch delivers overflow calls. When OCS receives an `EventRinging` message on this DN, it automatically sends a request to T-Server to answer the call, and then updates the record with the `Dropped` call result. After receiving the `EventEstablished` response from T-Server, OCS sends a request to T-Server to release the call.
- If the overflow DN is a Queue or a Routing Point, the call is transferred to a Destination DN such as voicemail or IVR for a prerecorded message. To use this method, set the **overflow_dn** option to `true` on the `Options` tab of the ACD Queue or Routing Point from which the overflow calls are distributed to the destination DNs. When OCS receives an `EventQueued` on this DN, it automatically destroys the call in its memory, and then updates the record with the `Dropped` call result.

If you set the value of this option to `false` or no, OCS does not treat calls as overflow calls if they remain in an ACD Queue (waiting for an agent) beyond the timeout period. OCS does not update the record as `Dropped`, and the call is not released as an overflow call.

pa-abandon-rate-limit

- Default Value: `0`
- Valid Values: Any number between `0` and `100` in decimal format (for example, 3.55)
- Configuration Levels: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Predictive Algorithm, Dialing Regulations

Specifies the maximum allowed value of the expected Abandon Rate, as a percentage. If the Abandon Rate, which is expected for the outbound dialing pace calculated in accordance with a given Target Value of the optimization method, exceeds this value, the predictive algorithm restricts the outbound dialing pace so that the expected Abandon Rate matches this value.

This option is used in all Predictive dialing modes and for all optimization methods. If the option is set to `0` (zero) or `100`, the restriction is not applied.

Note:	If the Answering Machine Detection (AMD) false-positive rate is configured by using the **pa-amd-false-positive-rate** configuration option, OCS includes AMD false-positive rates in the expected Abandon Rate.

pa-amd-false-positive-rate

- Default Value: 0 (zero)
- Valid Values: Any value from 0 to 100, in decimal format (for example, 1.5)
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Predictive Algorithm, Dialing Regulations

Specifies, as a percentage, the AMD (Answering Machine Detection) false positive rate as a proportion of total calls answered by live individuals.

pa-amd-test-percentage

- Default Value: 0 (zero)
- Valid Values: Any integer from 0 to 100
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Predictive Algorithm, Dialing Regulations

Specifies, as a percentage, the number of randomly selected AMD calls passed to agents during the test for the AMD false positive rate. A value of zero (0, the default) means that the test is not to be run.

OCS can account for the AMD false positive rate when running Predictive (or Predictive with Seizing) campaigns in a VoIP environment. OCS can also use the AMD false positive rate when running Progressive (or Progressive with Seizing) campaigns for reporting purposes.

Note:	This feature is not available for deployments using CPD Server.

pa-dial-expire

- Default Value: 2
- Valid Values: 1-6000
- Configuration Level: Campaign Group, Application
- Logical Group: Predictive Algorithm

Specifies the timeout (in minutes) that the predictive algorithm uses to clean up calls that are in a Dialed or in a Queued state.

If an outbound call is dialed or queued, and no further events about this call are received by the predictive algorithm within the specified timeout, the call is removed from the predictive algorithm's memory buffer.

Note:	This option is used for queued calls only if the **pa-queue-expire** option is not configured or is set to `0`.

These removed calls are excluded from all the types of calculations performed by the predictive algorithm including the predicting next calls.

Note:	For Push Preview and Power GVP dialing modes, be aware of the following: The `Dialed` state means that the interaction was submitted to Interaction Server but was not delivered to the agent desktop or finalized by the routing strategy (Processes block). Set the value significantly higher than the default value of 2, because the distribution time for multimedia interactions is significantly higher than the distribution time for calls.

pa-exclude-long-dialing

- Default Value: `false`
- Valid Values: `true, false`
- Configuration Level: Campaign Group
- Changes Take Effect: Immediately
- Logical Group: Predictive Algorithm

The option enables or disables the exclusion of current outbound calls with long dialing durations from pacing calculations when OCS uses the Advanced Small Group predictive algorithm.

- If the option is set to `true`, OCS excludes current calls with long dialing times from pacing calculations.
- If it is set to `false`, OCS includes these calls in pacing calculations.

This option takes effect only when the Advanced Small Group predictive algorithm is in effect (see the predictive_algorithm). For Campaign Groups using this new feature, OCS logs "Long Dialing" statistics string in the PA Session Info section of the OCS log. OCS does not apply this option in the Predictive GVP dialing mode.

pa-handle-expire

- Default Value: `30`
- Valid Values: `1-6000`
- Configuration Level: Campaign Group, Application
- Logical Group: Predictive Algorithm

Specifies the timeout (in minutes) that the predictive algorithm uses to clean up calls that exist on an agent's desktop. If an outbound call is dialled and is delivered to an agent, and no further events are received about this call are received by the predictive algorithm, the call will not be included when predicting the next call.

pa-handle-time-consider

- Default Value: `1800 (seconds)`
- Valid Values: Any positive integer (from 1 to the maximum integer supported by your operating system)
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Predictive Algorithm

Specifies the maximum call duration for all call types when calculating the average call duration. If a call's duration (including the `After Call Work` state) is more than the specified value, then the predictive algorithm does not include this call duration when calculating the average call duration.

pa-hitratio-min

- Default Value: `5`
- Valid Values: Between `0.0000000001` (`1.0e-10`) and `100.0` (in decimal format, for example: `0.55, 2, 3.5, 1.0e-9`)
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Predictive Algorithm

Specifies the minimum value (in percent) of the hit ratio that is used to predict the dialing rate by the Classical Predictive algorithm. If the current estimated hit ratio is less than the value set by this option, OCS uses the value set by this option for pacing calculations.

Notes:	• Zero (`0.0`) is not a valid value for this option. This option only takes effect for Classical Predictive algorithm. • Setting the value lower than the default value for this option can cause significant overdialing.

pa-inbound-ignore

- Default Value: `none`
- Valid Values: `none, outbound, engage, all`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Predictive Algorithm

Specifies which method OCS uses to ignore the sampled value of inbound traffic in dialing pace calculations. This enables users to specify how to handle dialing-pace calculations for outbound and engaging calls in the `Predictive with seizing` dialing mode with respect to inbound traffic. The values function as follows:

- none—OCS does not ignore the inbound traffic and uses it to calculate the dialing pace for outbound and engaging call dialing.
- `outbound`—OCS ignores the inbound traffic for the dialing pace of outbound calls and uses it only for the dialing pace of engaging calls.
- engage—OCS ignores the inbound traffic for the dialing pace of engaging calls and uses it only for the dialing pace of outbound calls.
- `all`—OCS ignores the inbound traffic and does not use it for calculating the dialing pace for outbound and engaging calls.

Note:	This option was added to OCS, version 7.6.101.18.

pa-odr-interval

- Default Value: `480` (in minutes; that is, 8 hours for the standard length of a working day)
- Valid Values: `240` to the maximum integer supported by your operating system
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Predictive Algorithm

Specifies the time interval, in minutes, that OCS uses to match the target value of the Overdial Rate. In other words, if set this option to the default value, the average overdial rate calculated at the end of an 8-hour day must meet the overdial rate specified in the predictive_max_overdial_rate option.

Note:	OCS defines the beginning of a time period to be the moment when the session is started or restarted or when the optimization method is changed to Overdial Rate. When the time period ends, the algorithm starts a new time period.

pa-odr-period-start-time

- Default Value: `-1`
- Valid Values: Any integer from `-1` to 86399 (the number of seconds in a 24-hour period)
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Predictive Algorithm, Dialing Regulations

Specifies, in seconds after midnight, the beginning of the 24-hour period for the calculation of the Abandon Rate. At this second, OCS reinitializes the calculation by setting to zero (`0`) the following three parameters: total number of outbound calls connected to agents, total number of abandoned outbound calls, and total number of disconnected AMD calls. If this option is set to `-1` or an invalid value, or is not present, OCS does not reinitialize the calculation of the Abandon Rate.

pa-progressive-multiplier

- Default Value: 1
- Valid Values: 0 to N (maximum integer or floating point number; decimal point is delimiter, regardless of the locale)
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Predictive Algorithm

Specifies the multiplier that OCS uses to calculate pacing based on the number of outbound calls that are simultaneously in progress and the number of agents that are in ready status.

If this option is set to 0 (zero), no dialing takes place.

This option only applies to Campaign Groups that are explicitly configured to run in Progressive and Progressive with Seizing dialing modes. OCS calculates the number of new outbound calls to be dialed as the largest integer (rounded down) that does not exceed the number of available agents multiplied by the value of the progressive multiplier.

For example:

- Progressive Multiplier = **2.4**
- Number of ready agents = **5**
- Number of queued calls = **1**
- Number of dialed calls = **3**

OCS calculates the number of new outbound calls as follows:

4 = [2.4 * (5 – 1 – {3/2.4})] = [2.4* (5 – 1 – 2)] = 4.8 (rounded down) = **4**

OCS supports both dynamic port allocation and dynamic reassignment of agents by ports when running Campaign Groups in Progressive with Progressive Multiplier or Progressive with Seizing with Progressive Multiplier dialing modes.

If this option is set to a value greater than 1, OCS applies the predictive_max_overdial_rate option in the same way as for Predictive mode. That is, if the real overdial rate of a running dialing session exceeds a non-zero value of the **predictive_max_overdial_rate** option, OCS switches to regular Progressive mode. In other words, OCS starts applying 1 instead of the value specified by the **pa-progressive-multiplier** option.

Note: Configuring this option does not impact any other options that control the pace of dialing, such as the maximum dialing rate or the number of CPD ports. OCS will continue to apply all other pacing-related settings.

pa-queue-expire

- Default Value: 0
- Valid Values: 0 to N (maximum integer)

- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Predictive Algorithm

Specifies the timeout, in seconds, that the predictive algorithm uses to clean up calls that are in a Queued state. If an outbound call is queued, and no further events about this call are received by the predictive algorithm within the specified timeout, the call is removed from the memory buffer belonging to the predictive algorithm. If this call was queued with the `Answering Machine Detected` call result, the predictive algorithm considers the call as a `No Contact` call and decreases the Hit Ratio.

If the option is set to `0` or is not present, OCS uses the timeout value set in the **pa-dial-expire** option to determine when to clean up calls that are in a Queued state. In this case, the predictive algorithm does not check for the `Answering Machine Detected` call result.

pa-safe-dialing

(Added: 8.1.3)

- Default Value: `true/yes`
- Valid Values: `true/yes`, `false/no`
- Configuration level: Campaign Group, OCS Application
- Change takes effect: Immediately

This option supports the Safe Dialing feature.

If the value of this option is set to `true` or `yes`, OCS stops outbound dialing if there are 30 or more abnormal abandoned calls among the last 100 outbound calls that were answered by live persons, including calls that were transferred to agents but then abandoned.

Abandoned calls are considered abnormal if they were abandoned or dropped while there were available agents in the Ready state (for Transfer mode) or Engaged state (for ASM mode). OCS restricts the number of abnormal abandoned calls detected to the number of available agents during each 1-second interval.

OCS resumes outbound dialing and re-initiates the counting of last abnormal abandoned calls when the Campaign Group is stopped and started, or if the value of this option is set to a different valid value.

If the value of this option is set to `false` or `no`, OCS does not stop outbound dialing because of abnormal abandoned calls.

Note: OCS does not support Safe Dialing for IVR groups when the value of the **ivr_group** option is set to `true`, or when one or more Places that are associated with the Campaign Group contain Voice Treatment Port DNs.

pa-selfcheck-adt-threshold

- Default Value: `10`

- Valid Values: Any integers from 1
- Configuration Levels: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Predictive Algorithm, Predictive Algorithm Self-Test

Specifies the percentage of the target Average Distribution Time that the Predictive algorithm uses as the threshold to report the dialing performance degradation. If the difference between the current and the target Average Distribution Time is less than the percentage of the expected Average Distribution Time, the Predictive algorithm does not generate a report. The PA reports performance degradation only if a suspicious condition is detected.

pa-selfcheck-awt-threshold

- Default Value: `10` (percent)
- Valid Values: integer >=1
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Predictive Algorithm, Predictive Algorithm Self-Test

Specifies the percentage of the target Average Waiting Time that the predictive algorithm (PA) uses as the threshold to report a dialing performance degradation. If the difference between the current and the target Average Waiting Time is less than the percentage of the expected Average Waiting Time, the PA does not make a report. Otherwise, the PA reports the degradation only if a suspicious condition is detected for the Average Waiting Time.

For more information what a suspicious condition might be, see Predictive Algorithm Self-Diagnostic.

pa-selfcheck-bf-threshold

- Default Value: `10` (percent)
- Valid Values: `integer >=1`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Predictive Algorithm, Predictive Algorithm Self-Test

Specifies the percentage of the target Busy Factor that the PA uses as the threshold to report a dialing performance degradation. If the difference between the current and the target Busy Factor is less than the percentage of the expected Busy Factor specified in this option, the PA does not make a report. Otherwise, the PA reports the degradation only if a suspicious condition is detected for the Busy Factor.

For more information what a suspicious condition might be, see Predictive Algorithm Self-Diagnostic.

pa-selfcheck-interval

- Default Value: `20` (minutes)
- Valid Values: 2 to the maximum integer supported by your operating system
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Predictive Algorithm, Predictive Algorithm Self-Test

Specifies the time interval, in minutes, that the PA uses to calculate the current values of the optimization parameters and to track any suspicious condition for reporting dialing performance degradation.

pa-selfcheck-odr-threshold

- Default Value: `20` (percent)
- Valid Values: 1 to the maximum integer supported by your operating system
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Predictive Algorithm, Predictive Algorithm Self-Test

Specifies the percentage of the target Overdial Rate that the PA uses as the threshold to report about a dialing performance degradation. If the difference between the current and the target Overdial Rate is less than the percentage of the expected Overdial Rate, the PA does not make a report. Otherwise, the PA reports the degradation only if a suspicious condition is detected for the Overdial Rate.

For more information what a suspicious condition might be, see Predictive Algorithm Self-Diagnostic.

pre-desktop-validation-expiration-timeout

- Default Value: `3600`
- Valid Values: `0-3600`
- Configuration Level: Calling List, Campaign, Application
- Changes Take Effect: At the next pre-desktop validation check
- Logical Groups: Pre-dial Validation

Specifies the number of seconds during which the received validation result (positive or negative) remains actual. When set to `0` (zero), the result remains actual for indefinite period of time.

pre-desktop-validation-type

- Default Value: `off`
- Valid Values:

  - `off`: Pre-desktop validation is turned off.
  - `info`: Pre-desktop validation is turned on and is used as a supplementary data provider for the agent desktop. Records are delivered to the desktop regardless of validation outcome.
  - `relaxed`: Pre-desktop validation is turned on and is used as a preliminary relaxed filter.
  - `strict`: Pre-desktop validation is turned on and is used as a preliminary strict filter.
- Configuration Level: Calling List, Campaign, Application
- Changes Take Effect: At the next chain chosen for delivery to the agent desktop
- Logical Groups: Pre-dial Validation

Specifies whether re-desktop validation is enabled on and is used as a supplementary data provider for the agent desktop, as a preliminary relaxed filter, or as a preliminary strict filter.

pre-desktop-validation-uri

- Default Value: No default value
- Valid Values: The valid URI of the external validation system
- Configuration Level: Calling List, Campaign, Application
- Changes Take Effect: At the next pre-desktop validation request
- Logical Groups: Pre-dial Validation, CX Contact

Specifies the URI of the external validation system.

pre-dial-validation

- Default Value: `false`
- Valid Values: `true/yes`, `false/no`
- Configuration Levels: Individual record and chain of records (via SCXML), Calling List, Campaign, Application
- Changes Take Effect: At the next dialing attempt
- Logical Group: Pre-dial Validation, CX Contact

Specifies whether pre-dial validation is in effect or not for the given record. When set to `true`, OCS performs pre-dial validation of the record, for example, it delivers an HTTP POST request to the Web or Application Server, waits for the response and then, processes the response. When set to `false`, OCS starts dialing the record without the pre-dial validation step.

Note:	Setting this option at the individual record level requires the use of an SCXML treatment script. For a complete description, see SCXML-based Treatments. See also the `sample06.scxml` sample script, which can be found in the OCS installation folder, in the `scxml_samples` subfolder.

predictive_algorithm

- Default Value: small_group
- Valid Values: classical, small_group, advanced_small_group, time_optimized_odr
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Predictive Algorithm

Controls which predictive algorithm (PA) OCS uses for dialing outbound calls when a dialing session/campaign is running in the Predictive or Predictive with seizing dialing mode.

Note:	If the optimization method is Busy Factor or Average Waiting Time, OCS uses the classical predictive algorithm regardless of the value specified for this option. A new valid value, time_optimized_odr, was added to release 8.0.

The values function as follows:

- classical—OCS uses the classical predictive algorithm based on classical queuing theory, regardless of the current number of available agents and the optimization parameter that are used.
- Small group values: These two values are optimized for small groups of agents. Either of these values can be used if the Overdial Rate is specified as an optimization parameter and the number of available agents does not exceed the number set in the small_group_size option. If the number of agents exceeds the number set in the **small_group_size** option, OCS uses the classical predictive algorithm. Also see Predictive Algorithm for Small Groups.

The two small group values include:

- small_group—OCS uses a small group predictive algorithm, which waits until all dialed calls are completed before dialing new calls.
- advanced_small_group—OCS uses an advanced predictive algorithm. This option value activates the new and improved predictive algorithm for small groups of agents. The advantages of this new algorithm include:
  - It better tracks inbound calls, because it counts inbound calls that are at all stages of processing and not just those that are queued.
  - It allows outbound calls to be initiated, even if one or more calls remain in the dialing stage, rather than waiting until all dialed calls are completed.

Using the advanced_small_group value may result in busier agents, as the Busy Factor increases or the waiting time between calls reduces for agents for the same Overdial Rate.

This value can be set in conjunction with the time-to-ready-tolerance option.

- time_optimized_odr—OCS uses the time-optimized predictive algorithm. When using this value, OCS monitors the predictive_max_overdial_rate. If it is greater than zero but less than the current overdial rate, OCS switches from the Predictive to the Progressive dialing mode.

Note:	The predictive algorithm self-diagnostic mechanism considers the situation where the `predictive_max_overdial_rate` is greater than zero but less the the current overdial rate as an incorrect configuration and reacts with a proper log message. See time-optimized predictive algorithm for more information.

predictive_callback

- Default Value: `no/false`
- Valid Values: `yes/true`, `no/false`
- Configuration Level: Campaign Group, Application
- Logical Group: Call Processing

Specifies how Outbound Contact Server handles a Campaign Callback. If this option is set to yes or `true`:

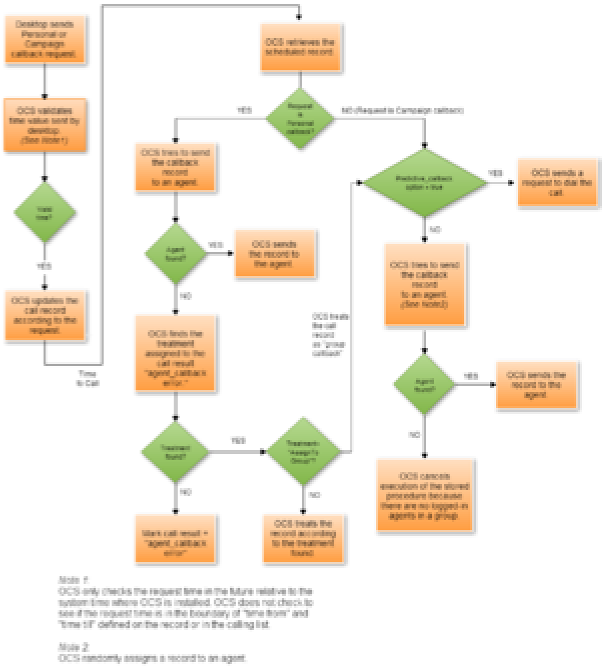
- In Predictive or Progressive modes, OCS automatically dials callbacks.
- In Preview mode, OCS puts records that are scheduled as Campaign Callback in the preview records list, and delivers them to the agent's desktop upon receipt of a `PreviewRecordRequest`.

If this option is set to `no or false`:

- Regardless of the dialing mode, OCS delivers records scheduled for callback directly to the agent's desktop through a `UserEvent`. The agent then decides whether or not to dial this record. If the desktop application is not capable of processing preview records and this record remains on the agent's desktop, OCS marks the call result of the record as `stale` when the timer set by the `stale_clean_timeout` option expires.
- When the Dialing Session is running, OCS retrieves Campaign callbacks from the database and delivers them to agent desktops.
- When the Dialing Session is active (loaded or stopped), OCS doesn’t retrieve Campaign callbacks from the database on a periodic basis except for Campaign callbacks that can be retrieved immediately after executing the Load or `Stop` command. OCS delivers Campaign callbacks that have already been retrieved, and also delivers Campaign callbacks that are produced from records with other types, if they are not returned to the database.

Note:	See CommunicationProtocols of the *Outbound Contact Reference Manual* for information about the events distributed when rescheduling records.

The following figure illustrates how Outbound Contact Server handles a Scheduled Call during a Predictive or Progressive campaign:


Desktop sends Personal or Campaign callback request.
OCS validates time value sent by desktop. (See Note1)
Valid time?
YES
OCS updates the call record according to the request.
Time to Call
OCS retrieves the scheduled record.
Request a Personal callback?
YES
NO (Request is Campaign callback)
OCS tries to send the callback record to an agent.
Agent found?
YES
OCS sends the record to the agent.
NO
OCS finds the treatment assigned to the call result "agent_callback error."
Treatment found?
YES
NO
Mark call result = "agent_callback error"
Treatment "Assign To Group"?
NO
OCS treats the record according to the treatment found.
OCS treats the call record as "group callback"
Predictive_callback option = true
YES
OCS sends a request to dial the call.
NO
OCS tries to send the callback record to an agent. (See Note2)
Agent found?
YES
OCS sends the record to the agent.
NO
OCS cancels execution of the stored procedure because there are no logged-in agents in a group.
Note 1
OCS only checks the request time in the future relative to the system time where OCS is installed. OCS does not check to see if the request time is in the boundary of "time from" and "time till" defined on the record or in the calling list.
Note 2
OCS randomly assigns a record to an agent.

Scheduled Call During a Predictive or Progressive Campaign

predictive_hit_ratio

- Default Value: 75
- Valid Values: `1-100`
- Configuration Level: Campaign Group, Application
- Logical Group: Predictive Algorithm

Specifies the Starting Value for Hit Ratio, as a percentage (%).

predictive_hot_start

- Default Value: `false`
- Valid Values: `yes/true`, `no/false`
- Configuration Level: Campaign Group, Application
- Logical Groups: Predictive Algorithm, Predictive Hot Start

Specifies how OCS uses predefined statistical data in certain circumstances.

If this option is set to `yes` or `true`, OCS uses the predefined statistical data to start predictive dialing, without first using the preliminary dialing in `Progressive` mode. This predefined statistical data will be taken from the following options: **predictive_hit_ratio, predictive_outbound_call_duration, predictive_inbound_rate, and predictive_inbound_call_duration**.

Note:	OCS always uses Hot Start in the Predictive GVP dialing mode. If the options for the predefined statistical data (predictive_patience_time, predictive_hit_ratio, predictive_outbound_call_duration, predictive_inbound_rate, and predictive_inbound_call_duration) are not configured, OCS applies their default values, which may not be efficient.

predictive_inbound_call_duration

- Default Value: `300`
- Valid Values: 1--N
- Configuration Level: Campaign Group, Application
- Logical Groups: Predictive Algorithm, Predictive Hot Start

Specifies the starting value for the average inbound call duration, in seconds.

predictive_inbound_rate

- Default Value: `0`
- Valid Values: `0 - N`
- Configuration Level: Campaign Group, Application
- Logical Groups: Predictive Algorithm, Predictive Hot Start

Specifies the starting value for the average number of inbound calls, per hour.

predictive-longcalls-truncation

- Default Value: `false`
- Valid Values: `false/no` or `true/yes`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Predictive Algorithm

Instructs the predictive algorithm on how to handle so called "long" outbound calls, (that is, if a few outbound calls last significantly longer than the majority of the outbound calls for the given Campaign Group). Although these "long" calls happen rarely, they are taken in the account by the predictive algorithm when average call duration is calculated. As a result, these long calls will increase the average call duration, negatively affecting the dialing efficiency. OCS uses the 95th percentile calculation to identify these types of calls, meaning that the top 5% of calls in the statistical distribution of call duration are excluded from the call duration calculation.

When this option is set to `true`, the predictive algorithm truncates the length of time associated with these calls when calculating the average call duration, which may positively effect the predictive dialing efficiency.

Note:	When configured, this option does not affect OCS's use of the outbound_agent_outlier_limit option. This option applies only to outbound calls.

predictive_max_overdial_rate

- Default Value: `0`
- Valid Values: Number between `0 and 100 in decimal format (for example, 3.55)`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Predictive Algorithm, Dialing Regulations

Specifies the maximum allowed overdial rate for dialing in `Predictive` mode. If the real overdial rate of a running dialing session/campaign, calculated from the time when the dialing session/campaign is started, exceeds this value, OCS switches the dialing mode from `Predictive` to `Progressive`. If the real overdial rate falls below or equal to this value and the `predictive_min_overdial_rate` option is not configured, OCS switches the dialing mode from `Progressive` to `Predictive`.

This checking of the limit is performed separately from all predictive optimization parameters; that is, overdial rate, busy factor and average waiting time.

If this option is set to `0` (zero) or `100`, OCS does not perform separate control of the overdial rate.

predictive_min_overdial_rate

- Default Value: No default value
- Valid Values: Number between `0 and 100 in decimal format (for example, 1.5)`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Predictive Algorithm, Dialing Regulations

Specifies the lower threshold of the overdial rate at which OCS switches the dialing mode from `Progressive` back to `Predictive`. If the real overdial rate later rises above this value, the dialing mode remains `Predictive` until the rate passes the level set by predictive_max_overdial_rate.

This option can be used only after the predictive_max_overdial_rate option has switched the dialing mode from `Predictive` to `Progressive`.

If this option is not set, is set to `0` (zero), or is set to a value greater than the value of predictive_max_overdial_rate, OCS ignores this option.

Warning

Using the **predictive_min_overdial_rate** option might significantly reduce the Busy Factor because it delays switching the dialing mode from Progressive back to Predictive.

predictive_outbound_call_duration

- Default Value: `120`
- Valid Values: `1-N`
- Configuration Level: Campaign Group, Application
- Logical Groups: Predictive Algorithm, Predictive Hot Start

Specifies the starting value for the average outbound call duration, in seconds.

predictive_patience_time

- Default Value: `180`
- Valid Values: Any non-negative integer
- Configuration Levels: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Predictive Algorithm, GVP, Outbound Contact VoIP Dialing Modes

Specifies the starting value of the average patience time (in seconds) for the Predictive GVP dialing mode.

Note:	If this option value is set to `0` and the Agent Busy Factor, Overdial Rate, or Average Waiting Time optimization method is used, OCS uses the Erlang-B model with 0 (zero) waiting time in the queue.

preview_release_nocontact_action

- Default Value: `soft_previous`
- Valid Values: `hard_ready, hard_not_ready, soft_previous`
- Configuration Level: Switch, Application
- Changes Take Effect: Immediately
- Logical Group: Agent Desktop

Determines the agent's place state after an agent releases an unsuccessful outbound call that the agent had placed manually in the Preview dialing mode.

- When set to `hard_ready`, OCS sends a request to T-Server to force the teleset to the Ready state.
- When set to `hard_not_ready`, OCS sends a request to T-Server to force the teleset to the `Not Ready` state.
- When set to `soft_previous`, OCS does not send any requests to T-Server.

Note:	When using this option, OCS takes the value of the hard_request_to_login_dn option into account. In other words, if **hard_request_to_login_dn** is set to `true`, OCS sends `RequestAgentReady` to the login DN instead of the DN where the call was answered.

progressive_blending_reserved_agents

- Default Value: `0`
- Valid Values: Any non-negative integer
- Configuration Level: Campaign Group, Application
- Logical Group: Predictive Algorithm

Enables OCS to keep a specified number of agents who are in the `Progressive` dialing mode available for inbound traffic or some other non-outbound activity.

Outbound Contact initiates the next outbound call when both of the following are true:

- There is at least one ready agent.
- The total number of ready or busy non-outbound agents exceeds the value of the option.

The agent is treated as busy outbound when he or she is in the `BusyOutbound`, `BusyPaper`, or `BusyRinging` state. All other states except the Ready state are treated as non-outbound.

The following is an example of how this option works.

An agent group has a total of four agents:

- One agent is in the `BusyInbound` state.
- One agent is in the `BusyOutbound` state.
- Two agents are in the Ready state.

At this time, the **progressive_blending_reserved_agents** option is set to 2. As a result, OCS sends one outbound call.

progressive_blending_reserved_status

- Default Value: `all_seized_inbound`
- Valid Values: `all_seized_inbound, ready`

- Configuration Level: Campaign Group. Application
- Logical Group: Predictive Algorithm

Specifies how agents are reserved for outbound calls.

- If **progressive_blending_reserved_status** is set to `all_seized_inbound`, OCS reserves all agents that do not have any of the following outbound statuses: `BusyOutbound, BusyPaper, NotReady`, and `BusyRinging`. This value is similar to the **progressive_blending_reserved_agents** option functionality in OCS 7.x
- If **progressive_blending_reserved_status** is set to `ready`, OCS reserves only those agents who are in Ready status. This value is similar to the **progressive_blending_reserved_agents** option functionality in OCS 6.5, which kept a specified number of reserved agents who were in progressive dialing mode available for inbound traffic. In this scenario, OCS waited until more than the specified number of reserved agents were in the Ready status before initiating the next outbound call.

protocol.add-record.uri

- Default Value: An empty string
- Valid Values: A valid URI
- Configuration Level: Application
- Changes Take Effect: Immediately
- Logical Group: Record Processing, CX Contact

Specifies the URI of the CX Contact (List Builder) endpoint for sending requests. Only the **http:** schema is supported.
For example: `http://cxc-int:8888/listbuilder/v2/contact-lists/$list_dbid/contacts`

All macro expressions are expanded in the same manner as for the Execute SQL Statement treatment. Refer to Execute SQL Statement Action Type.

public_network_access_code

- Default Value: An empty string
- Valid Values: Any character string
- Configuration Level: Calling List, Campaign Group, Switch, Application
- Logical Group: Dialing

Specifies the string that is added as a prefix to each phone number that OCS sends to a specific switch. The phone numbers in the call `UserData` remain unchanged. This number specifies the PSTN access code for the switch to which T-Server is connected. When an access code is added as a value, the system always places the access code in front of the phone number that is dialed. For example, if you set the value 9 into the value field, then the prefix 9 is always dialed before each phone number. If you are using the Outbound Contact Wizard, it prompts you to enter a value for this option. To accept the default value for this option, click the `Cancel` button on the wizard screen. You can add Dialogic dialing control parameters as a prefix to the dialed numbers from a calling list. When combined with the string defined for the **public_network_access_code** option, Dialogic dialing control symbols (prefix) determine the dialing time.

The following Dialogic symbols are available:

- L—Wait for the local dial tone before dialing
- I—Wait for the international dial tone before dialing
- X—Wait for the special dial tone before dialing
- , (comma)—Pause 2 seconds

For example, suppose that the `public_network_access_code` is defined as L9,. In this case, the dialer will wait for a dial tone, dial 9, pause for 2 seconds, then dial the number from the calling list.

Note:	Dialing control parameters work only if the CPD Server dials the calls through the Dialogic card (`tscall=false`). **Warning:** This option can be set at three levels: `Switch/Application` (which has the lowest priority), `Campaign Group` (which has a medium priority), and `Calling List` (which has the highest priority).

recall-on-unload

- Default Value: always
- Valid Values: One of the following:

always	OCS always recalls the interactions.
never	OCS never recalls the interactions.
only-if-item	OCS recall the interactions only if the Campaign Group is executed as an item of a Sequence.
only-if-standalone	OCS recall the interactions only if the Campaign Group is executed independently and not as an item of Sequence.

- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Interactions Processing in Push Preview Mode

Specifies if OCS should recall interactions submitted to Interaction Server but not yet processed when the Campaign Group is unloaded.

record-count-use-timeframe

- Default Value: `false`/no
- Valid Values: `false`/no and `true`/yes
- Configuration Level: Campaign, Application
- Changes Take Effect: When Campaign is activated

- Logical Group: Real-Time Reporting

Specifies whether OCS considers the `time_from/time_till` boundaries when the number of ready records and ready chains in the calling list is calculated.

record_processed

- Default Value: `true`
- Valid Values: `yes/true, no/false`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Agent Desktop

Specifies whether OCS waits for the agent's signal to finalize processing of a record.

- If **record_processed** is set to `true` or `yes`, Outbound Contact waits for a `RecordProcessed` request from the agent handling the calling list record. A `RecordProcessed` request is mandatory from the desktop application to Outbound Contact Server. This request informs OCS that the agent has finished with this call record and that OCS can update the database with the final information about the call record.

Note:	If an agent changes his or her state (such as `EventAgentReady` or `EventAgentLogout`), OCS treats the agent state as `ReleaseNextCall` or `LoggedOut`. The change in agent state informs OCS that the agent has finished with this call record, and that OCS can update the database with the final information about the call record. OCS can change an internal agent state after it receives `EventReleased` from the T-Server.

- If **record_processed** is set to `false` or `no`, Outbound Contact will assume that the record is processed when the agent or customer releases the call; that is, when OCS receives an `EventReleased` message for this call, or when it receives an `EventLogout` message for the agent who handled the call. Usually `false` is used when agents in an Outbound campaign do not have desktop phone applications and therefore cannot send the `RecordProcessed` event.

This option is used only in `Predictive` and `Progressive` dialing modes. In `Preview` dialing mode, Outbound Contact Server updates a record only after receiving a `RecordProcessed` event from a desktop, regardless of whether the **record_processed** option is set to `true` or `false`.

Note:	The desktop can send multiple `UpdateCallCompletionStats` requests to Outbound Contact Server to update the record before sending the final `RecordProcessed` request. `UpdateCallCompletionStats` requests are stored in OCS memory and are not recorded into the Calling List Record database until the `RecordProcessed` request is received.

For more information, see the Updating Genesys Mandatory Fields and Custom Fields section in the

Outbound Contact Reference Manual.

record_save_intermediate_results

- Default Value: `no/false`
- Valid Values: `yes/true, no/false`
- Configuration Level: Campaign Group, Application
- Logical Group: Record Processing

Specifies whether the calling list should be updated with intermediate dialing results during the lifetime of a calling list record.

- If the value is set to `yes` or `true`, the Calling List table is updated with intermediate dial results for each calling list record after each dial attempt. OCS writes the following information to the database: call result, scheduled time, number of attempts, and other data that is pertinent for treatment redials and for calls that are rescheduled by an agent.
- If the value is set to `no` or `false (default)`, OCS updates the record in the Calling List table with the final result only. Only the information about the last treatment application result is recorded for each record before the record is deleted from OCS memory.

remote_release_action

- Default Value: `error`
- Valid Values: `error, no_contact, abandoned`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Call Processing

Specifies how OCS handles calls with the call result of `RemoteRelease`.

- If set to `error (default)`, OCS processes the call as if OCS has received an `EventError`. The call result is `RemoteRelease`. The dialing rate does not change.
- If set to `no_contact`, OCS processes the call as if no contact has been made, as with call results `Busy` and `NoAnswer`. OCS decreases the Hit Ratio. The call result is `RemoteRelease`.
- If set to `abandoned`, OCS processes the call as if the customer has hung up. OCS increases the abandoned rate when the call result is `Abandoned`.

report-procedure-body

- Default Value: Empty
- Valid Values: Body of the custom procedure in binary format; for the Application level is a string format
- Configuration Levels: Calling List (higher priority), Campaign Group, Application
- Changes Take Effect: Immediately

- Logical Group: Real-Time Reporting

Specifies the body SQL code of the reporting stored procedure. The body of the stored procedure is stored in binary format in Configuration Server. Genesys Administrator provides a user interface to edit and store the value of this option.

report-procedure-location

- Default Value: `instead`
- Valid Values: `before, after, instead`
- Configuration Levels: Calling List (higher priority), Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Real-Time Reporting

Specifies the location of custom code within a standard reporting stored procedure (for the values `before` and `after`), or specifies that the whole body of the procedure is customized (for the value `instead`).

right_person

- Default Value: `""`
- Valid Values: Any value that can be stored in the field for which this option is configured
- Configuration Level: Field
- Logical Group: Historical Reporting

Marks the field that indicates that the answered call was with the proper contact. If it is updated with a value equal to what is configured in this option's value, it will be recognized by Genesys Info Mart as a right person contact indicator.

Note:	This option should be specified for only one field within a calling list.

scxmli.persistence.max_active

- Section: `scxml` in OCS application/Annex
- Default Value: `10000`
- Configuration Level: Application
- Changes Take Effect: During startup. Changes to this option are not applied dynamically.
- Logical Group: SCXML-based Treatments

This option specifies the maximum number of active SCXML treatments. When it is set to `0`, it uses the default value (`10000`). When a value is less then `500` (except `0`), it uses the value `500`. No max value is defined.

schedule-states-uri

- Default Value: An empty string
- Valid Values: Any valid URI
- Configuration Level: Application
- Changes Take Effect: At the next activation of the schedule item
- Logical Groups: CX Contact, Outbound Schedule

Specifies the URI of the external system (Campaign Manager) to which OCS sends Outbound Schedule states.

scxmli.transcoder.default

- Section: `scxml` in OCS application/Annex
- Default Value: UTF-8
- Valid Values: Any valid converter name supported by the ICU Library.
- Configuration Level: Application
- Changes Take Effect: During startup. Changes to this option are not applied dynamically.
- Logical Group: SCXML-based Treatments

This option defines the source encoding that OCS will use to encode all input String data into Unicode before it is passed into the SCXML engine. The input String data includes SCXML and JavaScript String literals, variables, and properties of String type.

Note:	This option is valid only on Windows and Linux platforms.

send_attribute

- Default Value: `""`
- Valid Values: Any name that Outbound Contact Server can use as a key in a key-value pair in `UserData`.
- Configuration Level: Field
- Logical Group: Record Processing

OCS sends the data attached to a call (`UserData`) as a key-value pair. The value of this option defines the key in the pair. The value of the field is the value of the pair. For example, a user-defined field `LastName` may have the **send_attribute** option with `name` in the `Option Value` field. OCS attaches the key value pair `name = <LastName>` to the `UserData`. Suppose the `<LastName>` in a record is Smith. When OCS processes this record, it attaches the `UserData` `name = Smith` to a call. The value of this option should not be an empty string (string of length 0). Also, all fields should have different values for this option in order to distinguish them from `UserData`.

For additional information on attaching field values using this option, see the Attaching Record

Information to Desktop and OCS User Events section in the *Outbound Contact Reference Manual*.

send-campaigngroup-states

- Default Value: `false`
- Valid Values: `true`, `false`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Next attempt to send notification to Web or Application server
- Logical Group: HTTP-based Notifications, CX Contact

Defines whether OCS delivers Campaign Group information to a Web or Application server. For more information, refer to Campaign Group Status Reporting.

send-history

- Default Value: `false`
- Valid Values: `true`, `false`
- Configuration Level: Calling List, Campaign, Application
- Changes Take Effect: Next dial attempt, or next attempt to send the Call Report Record to Web or Application Server, if the previous attempt failed
- Logical Group: Record Processing, HTTP-based Notifications, CX Contact

Specifies whether OCS sends call report records to Web or Application Server.

Related Documentation: Submitting Call Report Records

send-schedule-states

- Default Value: `false`
- Valid Values: `true, false`
- Configuration Level: Application
- Changes Take Effect: At the next activation of the schedule item
- Logical Groups: CX Contact, Outbound Schedule

Specifies whether OCS sends POST requests with Schedule state changes.

sip-cluster-asm-dial-node

- Default Value: `0` (not defined)
- Valid Values: The valid DBID of the primary SIP Server Application object
- Configuration Level: Agent Group object

- Changes Take Effect: After Campaign Group restart
- Logical Groups: SIP Cluster
- Introduced: 8.1.522.05

Specifies the DBID of the SIP Server Application object that a Campaign Group is using to dial outbound and engaging calls in Progressive with Seizing and Predictive with Seizing dialing modes. The same SIP Server application must be present on the OCS Application Connections tab. See OCS Load Balancing in SIP Cluster for details.

sip-cluster-lb-mode

- Default Value: `none`
- Valid Values: `none`, `round-robin`
- Configuration Level: Application
- Changes Take Effect: After OCS restart
- Logical Groups: SIP Cluster
- Introduced: 8.1.521.01

If set to none (the default), OCS does not apply load balancing between SIP Cluster nodes. If set to `round-robin`, OCS applies load balancing between SIP Cluster nodes in a standard round-robin fashion case if SIP Cluster Load Balancing is fully configured. See OCS Load Balancing in SIP Cluster for details.

sip-cluster-prime-node

- Default Value: `0` (not defined)
- Valid Values: The valid DBID of the primary SIP Server Application object
- Configuration Level: Application
- Changes Take Effect: After OCS restart
- Logical Groups: SIP Cluster
- Introduced: 8.1.521.01

Specifies the DBID of the primary SIP Server Application object in one of the SIP Cluster nodes, which OCS will use as the prime SIP Cluster node to register and monitor DNs when working in SIP Cluster Load Balancing mode. The same SIP Server application must be present on the OCS Application Connections tab. See OCS Load Balancing in SIP Cluster for details.

small_group_size

- Default Value: 7
- Valid Values: Any non-negative integer (`0-n`)
- Configuration Level: Campaign Group, Application

- Logical Group: Predictive Algorithm

Specifies the maximum number of available agents in a group that can be treated by OCS as a small group, and therefore have a special type of predictive algorithm applied.

If the current number of available agents in the group is less than or equal to this option's value, OCS uses a predictive algorithm specially optimized for small groups of agents for campaigns that use this group. This algorithm may provide better optimization results than the general predictive algorithm in some campaigns if the number of active agents is relatively low.

Note:	OCS considers this option's setting only when all of the following are true: The dialing session/campaign is running in Predictive, Predictive with seizing, or Predictive GVP dialing modes. The Overdial Rate is used as optimization parameter. The predictive_algorithm option is set to the `small_group` or the `advanced_small_group` value.

snapshot_interval

- Default Value: 600 (seconds)
- Valid Values: Any positive integer
- Configuration Level: Application
- Changes Take Effect: Immediately
- Logical Group: Historical Reporting

Defines the time interval, in seconds, between delivering snapshot statistics to Interaction Concentrator (ICON).

stale_clean_timeout

- Default Value: `30`
- Valid Values: Any integer from 1 to *n*`/60`, where *n* is the maximum integer value on the platform on which OCS is running.
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Groups: Agent Desktop, Record Processing

Specifies a timeout (in minutes) before OCS marks as `Stale` any records that were sent for processing (for example, `RequestMakePredictiveCall` is issued or `preview record` is sent to agent desktop), but were not yet processed.

stat-hit-ratio-count

- Default Value: 33

- Valid Values: One or more call results (integer values per GctiCallState enum, separated by a comma), or `all`
- Changes Take Effect: Immediately
- Configuration Level: Campaign Group, OCS Application
- Logical Groups: Predictive Algorithm

Specifies the call results for which OCS counts all established calls as answered calls when calculating and submitting the Hit Ratio of Campaign Groups and Calling Lists to the external Web/Application Server. If set to `all`, OCS counts all successful calls (such as Answer, Answering machine, Fax, Silence). When the option is changed for a running Campaign Group, OCS resets the counters and starts calculating the new Hit Ratio.

Examples: **stat-hit-ratio-count** = `all`, **stat-hit-ratio-count** = `33,32`

stat-server-ha

- Default Value: `false`
- Valid Values: `yes/true, no/false`
- Configuration Levels: Application
- Changes Take Effect: At the next RunModeInfo message arrival from Stat Server
- Logical Group: Real-Time Reporting
- Introduced: 8.1.527.14

Specifies how OCS handles `SEventRunModeInfo` notifications from Stat Server:

- If this option is set to `no` or `false`, OCS ignores `SEventRunModeInfo` notifications.
- If this option is set to `yes` or `true` and an `SEventRunModeInfo` notification contains `LongValue=0` (Backup Mode), OCS closes the connection to that Stat Server and attempts connecting to a backup Stat Server. This OCS behavior is recommended in configurations where Stat Server is configured to keep the listener port opened in backup mode.

time-to-ready-tolerance

- Default Value: `2 sec`
- Valid Values: Any positive integer of `2 sec` to the maximum integer supported by your operating system
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Predictive Algorithm

Specifies the allowed variance, in seconds, on the time estimate for when an agent will become Ready. OCS uses the value for this option in its calculations to determine whether an agent delivers correct (or "trusted") estimations on when the agent will complete the processing of the given call and become Ready. The calculations are based on information provided by the agent desktop to OCS about each agent. For more information on trusted versus distrusted agents, see Agent Feedback for the Predictive Dialing Modes.

Note:	OCS uses this information to further improve the dialing efficiency in Predictive modes; for example, to increase the `Busy Factor` for a given value of the `Abandon Rate`.

transfer_to_unknown_dn

- Default Value: `no/false`
- Valid Values: `yes/true, no/false`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Call Processing

Specifies OCS's behavior when an outbound call is transferred to an unknown DN or a DN without an agent logged into it.

This is important in multi-site group deployments, because OCS cannot properly resolve the destination DN. This option prevents OCS from treating such transfers as transfers to an unknown DN.

- If set to no or `false`, immediately after the transfer is completed, OCS updates the record with the `Stale` call result and the `Agent Error` record status.
- If set to yes or `true`, OCS does not update the record, and it continues to monitor the outbound call.

Note:	If a call is transferred to a DN that OCS has not registered, the record will not be updated in the calling list upon call release and its status will remain `Retrieved` until the timer set by the **stale_clean_timeout** option expires. **Warning:** In release 7.5 and higher, an "unknown DN" is a DN that is included in a `Place` object, but this `Place` object does not have an associated agent. An agent is associated to a `Place` object when a Campaign Group that is configured with this agent is activated within OCS.

treatment-action-uri

- Default Value: No default value
- Valid Value: The valid URI
- Configuration Level: Treatment
- Changes Take Effect: At the next treatment application
- Logical Group: HTTP-based Notifications

Speifies the URI for sending a request. All macro expressions are expanded in the same manner as for the Execute SQL Statement treatment. Only the `http:` schema is supported. See Send HTTP

Request Action Type for details.

treatment-holidays-table

- Default Value: No default value
- Valid Value: The name of the Statistical Table configuration object for holidays
- Configuration Level: Campaign Group, Application
- Changes Take Effect: At the next treatment application
- Logical Group: SCXML-based Treatments

Defines the name of the Statistical Table configuration object that OCS uses to determine the dates and time ranges for holidays.

treatment-preferred-contact-field

- Default Value: `none`
- Valid Value: The name of the field in the Calling List table
- Configuration Level: Calling List, Campaign, Application
- Changes Take Effect: At the next treatment application
- Logical Group: SCXML-based Treatments

Defines the field name in the Calling List table that OCS uses for the given record to determine if this record in a chain should be used for the first chain dial attempt. When the specified field exists and is not an empty string (for char/varchar field data types) or not zero (for integer field data types), OCS assumes that the associated record has a priority within the chain. If multiple records within the same chain satisfy the criteria for priority, OCS uses the first record found.

Note:	You can also configure this option at the Calling List Level. If configured at the Calling List Level and the Campaign Level, the Calling List Level takes precedence.

Dependency: If the CX Contact option `custom-format` is configured, OCS does not support the **treatment-preferred-contact-field** option for all custom CX Contact fields (`Other21+`) and fields `c_other1,..,c_other20`.

treatment_sched_threshold

- Default Value: `30` (minutes)
- Valid Values: Any non-negative integer (`0` and higher)
- Configuration Level: Calling List, Application
- Logical Group: Record Processing

Determines the threshold for OCS to either:

- Keep rescheduled chains in its memory.
- Remove rescheduled chains from its memory and return them to the database in the Ready status. OCS will store the treatment application history information so that it can continue to processing the chains at the scheduled time.

If the difference between the current time and the scheduled time exceeds the specified value, OCS will:

- Remove the chain from memory.
- Update it in the database as Ready.
- Store the treatment's application history info in a calling list's `treatments` field for each record in this chain.

OCS retrieves this chain back into memory shortly before the scheduled time of the next attempt and continues processing the chain of records in treatment sequence.

If the value is `0`, this functionality is disabled and prevents OCS from removing rescheduled chains from its memory until the dialing session/campaign group is stopped and unloaded.

treatment-uri

- Default Value: none
- Valid Values: String representing URI of SCXML treatment script resource
- Changes Take Effect: At the next treatment application
- Configuration Level: Calling List, Campaign Group, Application
- Changes Take Effect: At the next treatment application
- Logical Group: SCXML-based Treatments

Defines the URI to the SCXML treatment script resource on the Application Server, which specifies the treatment used for this Campaign Group. The URI can contain treatment parameters and values that are applied to the instances of the treatment script created when running a dialing session for a campaign. The parameters can include any of the following:

- Contact type to use when starting the dialing session/campaign (`start_contact_type`).
- Maximum number of dialing attempts (`max_dial_attempts`).
- User-defined variables, as identified in the script.

Separate all parameters and their values using the and (&) symbol. For example: **treatment-uri**=`http://server/ocs/treatment1?start_contact_type=HOME&max_dial_attempts=8`

Notes:	• This option can also be configured at the Calling List Level, which has a higher priority than when it is configured at this Campaign Group Level.

	• If this option is not defined here or at the Calling List Level, OCS uses the treatment configuration described in Call Handling Treatments. • If the value for this option contains parameters/ values that are defined in the script and the script values differ, the values in the **treatment-uri** option take precedence. • OCS does not process URI parameters. Parameters from the URI are supposed to be processed by a Web Server active component (JSP, PHP, and so on), so that OCS receives a dynamically-formed SCXML script with proper parameters already substituted for their values.

treatment-weekdays-table

- Default Value: No default value
- Valid Values: The name of the Statistical Table configuration object for weekdays
- Configuration Level: Campaign Group, Application
- Changes Take Effect: At the next treatment application
- Logical Group: SCXML-based Treatments

Defines the name of the Statistical Table configuration object that OCS uses to determine the time ranges for business weekdays.

update_all_records

- Default Value: `false`
- Valid Values: `true/yes, false/no`
- Configuration Level: Calling List, Application
- Changes Take Effect: Immediately
- Logical Group: Record Processing

Specifies if a dialing filter determines the set of records that OCS updates after the chain processing is completed.

- If set to `true` or `yes`, OCS updates all records in the chain, regardless of the dialing filter applied. All records in the chain that are outside the parameters of the dialing filter are updated with the same status that the chain receives after OCS has finished processing it.
- If set to `false` or `no`, records in the chain that were filtered out by the dialing filter are marked as `record_status = Ready`.

user_data_section_name

- Default Value: None
- Valid Values: Any valid name for a configuration section
- Configuration Level: Application
- Logical Group: Dialing

Enables you to attach additional information as a permanent key-value pair to the `UserData` attribute for dialing requests, preview records, or interactions. The value of this option identifies the name of a section that you created on the `Options` tab of the following configuration objects:

- Calling List
- Campaign Group
- Campaign
- Agent Group or Place Group
- Switch
- OCS application

OCS searches these configuration objects for a section with this value. When found, OCS attaches any options that are specified within the section as key-value pairs to the `UserData` attribute for all dialing requests, preview records, or interactions that are associated with the configuration object. The configuration object that you choose depends on the desired level of distribution:

- When you specify an option value that is the section name for a `Calling List` object, the specified data is attached only to calls for that calling list.
- When you configure this option for a `Campaign Group` object, the specified data is attached to calls for all calling lists in the Campaign Group.

You can explicitly identify the type of data to be attached by adding a specifier as a prefix to the data (that is, the option's value) for the options listed under the section name identified by the **user_data_section_name** option. When OCS finds these specifiers, it converts the data to the specified type. The 4-character specifier, however, is removed from the resulting data. The type-specifier descriptions are as follows:

- `str:`—All characters following this specifier are attached as a string. The value can consist of any characters. (Example: By specifying `str:String example 123`, the string value `String example 123` is attached.)
- `int:`—OCS converts the value that follows this specifier to an integer. The value should consist of digits (`0` through `9`) only and not exceed the maximum integer supported by the operating system where OCS is running. (Example: By specifying `int:2563`, the integer value 2563 is attached.)
- `bin:`—OCS converts the value that follows this specifier to a binary type. The value should consist of pairs of hexadecimal numbers separated by spaces. (Example: By specifying `bin:0A 0D 22 13 33 FF`, the binary value `0A 0D 22 13 33 FF` is attached.)

Note:	If no type specifier is included in the data value, the user data pair is attached as a string.

Configuration Example

The following example clarifies the relationship between the **user_data_section_name** option and configuration object sections associated with attaching User Data. If you do the following:

- Configure a section named **userdatalist** on the `Options` tab of the Calling List object.
- Within that **userdatalist** section, configure:
  - a key-value pair: `name=str:premier`
  - a key-value pair: `acct=int:1234`
- Configure the **user_data_section_name** option with a value of `userdatalist` in the `Options` tab of the OCS application object.

When OCS starts processing a specific chain, it searches for the `userdatalist` section among the configuration objects, and when it finds this section name, it attaches the two key-value pairs to the User Data for all dialing requests, preview records, and interactions associated with this Calling List object. The pair with the `acct` key is attached with the integer value of 1234. The pair with the `name` key is attached with a string value of `premier`.

validation-keys

- Default Values: Empty string (all keys are sent for pre-dial validation)
- Valid Values: One or more key names, comma-separated
- Configuration Level: Calling List, Application
- Changes Take Effect: Immediately
- Logical Group: Pre-dial Validation

Limits the keys included with the JSON body of the HTTP request for pre-dial validation to only those that are specified.

Note:	For this option, specify the list of key names (as configured by the **send_attribute** option for user-defined fields), not the calling list field names.

validation-propagate-user-data

- Default Value: `false`
- Valid Values: `yes/true`, `no/false`
- Configuration Level: Campaign Group, Application
- Changes Take Effect: Immediately
- Logical Group: Pre-dial Validation

Specifies if OCS propagates user fields modifications made by the pre-dial validation server and returned to OCS in a positive pre-dial validation response, for Email and SMS campaigns. When set to `true`, the user fields modifications are propogated.

validation-race-condition-call-result

- Default Value: 3 (General error)
- Valid Values: Positive Integer
- Configuration Level: Calling List, Campaign, Application
- Changes Take Effect: Next dial attempt
- Logical Groups: Record Processing, HTTP-based Notifications

Specifies the call result to be applied to pre-dial validation if a race condition is detected.

Related Documentation: Submitting Call Report Records

validation-timeout-call-result

- Default Values: 3 (General error)
- Valid Values: Call result enumeration value (as defined in the Call Result Types table in the *Outbound Contact Reference Manual*).
- Configuration Level: Calling List, Campaign, Application
- Changes Take Effect: Next dial attempt
- Logical Group: Pre-dial Validation

Specifies the call result value that is assigned to the dial attempt if pre-dial validation is turned on and OCS does not receive HTTP response for the validation request within the specific timeout (defined via the http-response-timeout option). In case of the timeout, OCS does not dial the record and attempts to apply the treatment to call result as defined by this option.

This option also applies when there is a failure contacting a Web Application Server during pre-dial validation. When OCS tries to establish a connection to a web application server and it is unavailable, OCS assigns the call result specified in this option to the record that is being validated.

validation-uri

- Default Value: An empty string ("")
- Valid Values: String representing URI of the Web or Application Server, in the following format: `http[s]://<host>:<port>/<path to pre-dial validation resource>`
- Configuration Level: Calling List, Campaign, Application
- Changes Take Effect: Next dial attempt
- Logical Group: Pre-dial Validation, CX Contact

Defines the URI of the pre-dial validation processing engine. Supported schemes are HTTP and HTTPS. The Port part of the configuration is optional and defaults to 80 for HTTP and 443 for HTTPS schemes.

Note:	For pre-dial validation to take place, this option must be set to a non-empty string (valid URI).

validation-wait-history

- Default Value: No default value
- Valid Values: `never, phone, clientid, always`
- Configuration Level: Calling List (first priority), Campaign (second priority)
- Changes Take Effect: Next dial attempt
- Logical Group: Pre-dial Validation

Specifies the OCS behavior for pre-dial validation. Valid values:

- `never`—OCS does not detect race conditions.
- `phone`—OCS detects race conditions by a phone number only.
- `clientid`—OCS detects race conditions by a customer ID only.
- `always`—OCS detects race conditions by a phone number and a customer ID.

This option is used for pre-dial validation only. It is not used for pre-desktop validation, because of on pre-desktop validation there is some time gap before calling and this time gap is used for other attempts to validate a record. So the race-condition detection for pre-desktop validation is always performed by a phone number and a customer ID.

> **Warning**
>
> Setting the option value to any value except `always` will impact the OCS compliance with regulatory requirements and may cause double dialing.

vtd-override

- Default Value: No default value
- Valid Values: Any string that represents a valid DN name
- Configuration Level: Campaign Group
- Logical Group: ASM Dialing

Defines the name of the Voice Transfer Destination (VTD) DN that OCS sends to CPD Server in the extensions of `RequestSeizeAgent` and `RequestMakePredictiveCall`.When CPD Server receives it, that number will be used as the destination number to which an engaging call is delivered and for the T-Server registration as the Voice Transfer Destination (VTD) DN. If this option is not configured, OCS does not add the corresponding key-value pair in to `RequestSeizeAgent` and `RequestMakePredictiveCall` extensions and CPD Server takes the destination number from the Voice Transfer Destination DN configured for the Campaign Group.

CPD Server Options

The CPD Server options are located in section folders on the `Options` tab of the CPD Server application object in `Genesys Administrator > Provisioning > Outbound Contact.` These sections include new options that enable you to configure CPD Server to meet the needs of your contact center.

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

- General Section
- HMP Section
- ISDN Section
- Line-Side Section
- Tones Section
- AM-beep-tones Section
- License Section

Note:	Ignore messages about dynamic option changes when the `keep-channels-open` option is set to `yes,` or if Dialogic DM3 hardware or HMP software is used. CPD Server does not read options that are not supported for a particular line type. The `pre-connect-cpd-priority` and `post-connect-cpd-priority` CPD Server options are ignored when the `line-type` option contains the `hmp` value.

General Section

am-beep-detect

- Type: Optional
- Default Value: `no/false`
- Valid Values: `yes/true` or `no/false`
- Enables or disables detection of the answering machine final beep tone.

am-beep-detect-timeout

- Type: Optional
- Default Value: `10000`
- Valid Values: `1000`- `60000`
- Specifies the maximum time interval (in ms) that CPD Server waits for the AM final beep tone. If the timeout expires, CPD Server stops AM beep tone detection and performs further call processing.

continuous-no-signal

- Type: Optional
- Default Value: `20`
- Valid Values: `unsigned integer`
- Specifies the time, in seconds, of continuous silence (no ring back timeout).

Note:	You can configure CPD Server to create two voice files for each outbound call that it dials on a Dialogic port. The next two options define the prefixes in the file names for two types of voice recording files.

cpd-recording

- Type: Optional
- Default Value: `no/false`
- Valid Values: `yes/true` or `no/false`
- Enables recording of call progress detection.

conversation-file-name-prefix

- Type: Required if the `cpd-recording` option is enabled
- Default Value: `conv_`
- Valid Values: Any ASCII string
- Specifies the prefix for the file name for the conversation recording file.

cpd-file-name-prefix

- Type: Required if the `cpd-recording` option is enabled
- Default Value: `cpd_`
- Valid Values: Any ASCII string
- Specifies the prefix for the file name for the call result recording file.

cpd-if-established

- Type: Optional
- Default Value: `false`
- Valid Values: `true/yes, false/no`

Part of the OCS support for uncommon CPD/CTI event flows. It controls the way the preconnect Call Progress Analysis (CPA) is performed. This option defines when Dialogic voice resource should be engaged for CPA.

- If set to `true` or `yes`, Dialogic CPA is engaged after the `EventEstablished` message has been received.
- If set to `false` or `no`, Dialogic CPA is engaged immediately at the beginning of the call start.

For increased flexibility in CPA, two additional options, `pre-connect-cpd-priority` and `post-connect-cpd-priority`, work together to control the outcome of conflicting call results that come separately from T-Server (CTI) and Dialogic (CPD).

Note:	The `cpd-if-established` option is ignored when CPD Server is used with Dialogic DM3 hardware. It is also ignored for Springware hardware with an isdn line-type. When you are using the `line-side-dm3, isdn-dm3, cas-dm3`, or `hmp` line types, the value for this option automatically adjusts to `false`.

destination-busy-timeout

- Type: Optional
- Default Value: `2000 (milliseconds)`

- Valid Values: <A numeric value representing milliseconds>
- CPD Server normally waits for both of the following indicators that a dialed number is busy before it terminates call progress detection.
- Busy call result from the Dialogic board
- EventDestinationBusy from T-Server
- This option prevents CPD Server from waiting indefinitely for a Busy call result by specifying the length of time (in milliseconds) that CPD Server waits for confirmation of the call result from the second source after the first has arrived. When the timeout expires, CPD Server accepts the Busy call result as correct.

keep-channels-open

- Type: Optional
- Default Value: no
- Valid Values: yes, no
- Changes Take Effect: Immediately
- Determines how CPD Server uses Dialogic channels.
- When set to yes, CPD Server keeps all Dialogic channels open and reuses them for the next call. CPD Server is unable to process dynamic option changes when this value is specified.
- When set to no, CPD Server closes the Dialogic channels after use and reopens them for the next call.

Note:	Please ignore messages about dynamic option changes when the keep-channels-open option contains a value of yes.

line-type

- Type: Required
- Default Value: None
- Valid Values: analogue, line-side, line-side-dm3, isdn, isdn-dm3, cas-dm3, sip-hmp, sip-hmp-asm
- Specifies the type of line (Dialogic card) that is being used. It also specifies the type of the protocol used to connect the Dialogic board to the switch or PSTN; for example: dm3, analogue, line-side, or isdn.
- The analogue value represents analog Dialogic boards that are supported in transfer mode only because answer supervision is not supported by analog lines.
- The line-side-dm3 and sip-hmp values are for transfer mode only.
- The isdn-dm3, cas-dm3,and sip-hmp-asm values are for ASM-mode only.

Note:	The sip-hmp and sip-hmp-asm values prompt CPD Server to process the sip-proxy option. See the HMP Section for more information about this option.

	Ignore messages about dynamic option changes when Dialogic DM3 hardware is used.

location

- Type: Required
- Default Value: None
- Valid Values: Any string
- Specifies the name of the folder under the `DN` section of the `Switch` object of the `T-Server` application that CPD Server uses when dialing.

max-number-ports-to-record

- Type: Optional
- Default Value: `0`
- Valid Values: `0, any integer higher than 1`

Enables CPD Server to record files in a plain VOX format (PCM/8000Hz/8bps). It creates the following two voice files for each outbound call:

- File 1 contains the line recording for the call progress detection stage.
- File 2 records the conversation between an agent and the called party if the call result is `answer` (ASM mode only).

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

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

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

- The number of configured recording ports
- The hardware limit for the number of Dialogic ports per box

For example: switch DNs

-

'location_name'

-

regular

-

engaging

-

recording

-

dxxxB1

-

dxxxB1C1

-

dxxxB2C2 When a number of voice channels are reserved for recording, the same number of digital channels should be disabled so that CPD will report the correct number of available ports for dialing. This should be done if the following is true:

- CPD Server is working in transfer mode and the number of either digital or loop start channels is equal to the number of voice channels.
- CPD Server is working in ASM ISDN mode and the number of non-engaging channels is equal to the number of voice channels.
- CPD Server is working in ASM Melcas mode and the number of digital channels is equal to the number of voice channels.

You can upgrade the number of voice channels by adding more Dialogic voice only boards.

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

off-hook-delay

- Type: Optional
- Default Value: `-1`
- Valid Values: Any integer
- Specifies the delay, in seconds, between off-hook and TMakeCall:
- If set to a negative value,: `off-hook`, and then `tmakecall` is sent.
- If set to a positive value: `tmakecall`, and then `off-hook` is sent.

off-hook-timeout

- Type: Optional
- Default Value: 1

- Valid Values: Any positive integer
- Specifies the amount of time to wait, in seconds, that CPD Server waits before assuming that the channel is in the `off-hook` state.

on-hook-timeout

- Type: Optional
- Default Value: 1
- Valid Values: Any non-negative integer
- Specifies the amount of time, in seconds, that CPD Server waits before assuming that the channel is in an `on-hook` state.

out-of-service-attempts

- Type: Optional
- Default Value: 2
- Valid Values: Any non-negative integer; minimum 2
- Specifies the number of failed attempts to use a port (channel) before CPD Server marks it out of service.

out-of-service-timeout

- Type: Optional
- Default Value: 1
- Valid Values: Any non-negative integer
- Specifies the amount of time (in minutes) that CPD Server waits before trying to use an out-of-service port again. If the value is 0, CPD Server ignores out-of-service conditions and continues to attempt to use the port.

post-connect-cpd-priority

- Type: Optional
- Default Value: `tserver`
- Valid Values: `tserver, dialogic`
- Defines the CPD priority for post-connect CPA. This option works together with the `pre-connect-cpd-priority` option.

pre-connect-cpd-priority

- Type: Optional

- Default Value: `tserver`
- Valid Values: `tserver, dialogic`
- Defines the source of Call Progress Information that is used for preconnect CPA.

For increased flexibility in CPA for uncommon CPD/CTI event flow support, the `pre-connect-cpd-priority` and `post-connect-cpd-priority` options work together.

They control the outcome of conflicting call results that come separately from T-Server (CTI) and Dialogic (CPD). They also define the conclusive source of CPA. Each of these options controls some possible conflicting CPA outcomes for the pre- and post-connect CPA.

This option works together with the `post-connect-cpd-priority` option.

sit-detection

- Type: Optional
- Default Value: `yes/true`
- Valid Values: `yes/true` or `no/false`

Defines the SIT signal parameters in the Dialogic DX_CAP data structure to enable CPD Server to more easily identify a SIT signal.

Controls SIT detection under the following conditions: SIT is disabled, and the `tscall` option is enabled to determine, from the `DestinationBusy` event, whether a number is valid.

tscall

- Type: Optional
- Default Value: `no/false`
- Valid Values: `yes/true` or `no/false`
- Specifies whether CPD Server sends a request to T-Server (`tscall = true or yes`) or to Dialogic board (`tscall = false or no`) to make a call.

tsclear

- Type: Optional
- Default Value: `no/false`
- Valid Values: `yes/true` or `no/false`
- Controls the way in which a call is released. If this option is set to `yes`, a `RequestClearCall` (G3 specific) is issued to release an active call.
- This option is useful, but not required, if a two-step transfer fails on a G3 switch.

use-engaging-lines

- Type: Optional
- Default Value: no/false
- Valid Values: yes/true or no/false

Applicable only to ASM mode:

- If set to no or false, CPD Server engages agents by using lines from the regular lines pool--that is, from the regular folder.
- If set to yes or true, CPD Server engages agents by using lines from a special lines pool that is dedicated to engaging purposes--that is, from the *engaging* folder.

You can benefit from this option if, for example, you wants to use special trunks for agent engagement--for example, lines that connect directly to the switching office and deliver calls directly to agents, free of charge. When this value is set to yes, this option provides an alternative to using trunks that incur a charge for every call.

Note:	Changes to the engaging folder are not supported while CPD Server is running. All voice channels (dxxx) are used by the regular network channels (dti) first, and then are used by the engaging channels. If you do not have more voice channels than network channels in your regular folder, then there will be no voice channels remaining for use for the engaging channels. In this case, engaging calls cannot be made.

use-busy2-as-nu-tone

- Type: Optional
- Default Value: no/false
- Valid Values: yes/true or no/false
- Controls the way in which the Busy2 tone is detected. This option enables CPD Server to detect a NU (Number Unobtainable) tone call result by using the Busy2 tone.

When the value is set to yes, the NU tone call result is assigned to all Busy2 tone detected calls. You can set this option at the Application level.

use-fax2-as-am-tone

- Type: Optional
- Default Value: no/false
- Valid Values: yes/true or no/false
- Controls the way in which the Fax2 tone is detected. This option enables CPD Server to detect an Answering Machine call result by using the Fax2 tone.

When the value is set to yes, the Answering Machine call result is assigned to all Fax2 tone detected calls. You can set this option at the Application level.

wait-off-hook

- Type: Optional
- Default Value: no/false
- Valid Values: yes/true or no/false
- Specifies whether CPD Server waits (a value of yes or true) or does not wait (a value of no or false) for an off-hook event from T-Server.

Note:	This value is used when the tscall option has a value of true and the off-hook-delay option has a negative value.

CPD Proxy Server Options

The following options apply to CPD Proxy Server only.

max-dialing-channels

- Type: Optional
- Default Value: 0
- Valid Values: 0...n
- Changes Take Effect: Immediately
- Specifies the maximum number of dialing channels that can be used by the OCS client of CPD Proxy Server. If the value is 0, there is no restriction to the number of channels used.

max-engaging-channels

- Type: Optional
- Default Value: 0
- Valid Values: 0...n
- Changes Take Effect: Immediately
- Specifies the maximum number of engaging channels that can be used by the OCS client of CPD Proxy Server. If the value is 0, there is no restriction to the number of channels used.

HMP Section

The options in this section are enabled when the line-type option contains the sip-hmp or sip-hmp-asm values.

audio-codecs

- Type: Optional
- Default Value: PCMU
- Valid Values: PCMU, PCMA, or both separated by a comma
- Changes Take Effect: After restart
- Defines the codec(s) to use for HMP. If more than one codec is listed, the first codec has priority over the second.
- The PCMU value corresponds to the G.711 mu-law codec.
- The PCMA value corresponds to the G.711 a-law codec.

sip-local-address

- Type: Optional
- Default Value: No default value
- Valid Values: IPv4 IP address only (no domain names), for example 172.21.1.19
- Changes Take Effect: After application restart
- If configured, enforces HMP to use the specified IP address instead of the default IP address as defined in the Dialogic DCM. Only IP version 4 addresses are supported. If the value cannot be interpreted as a valid IP address at CPD Server startup, CPD Server will stop the start-up routine, generate the appropriate log message, and exit.

This option should be used on hosts which have several Network Interface Cards (NIC) on board, because it enables you to explicitly define the NIC to be used by HMP.

sip-proxy

- Type: Mandatory
- Default Value: ""
- Valid Values: <any valid host name or IP address>
- Defines the SIP Proxy host name or IP address.

Note:	Ignore messages about dynamic option changes when HMP software is used.

ISDN Section

The isdn section is ignored when CPD Server is use with Dialogic DM3 hardware.

Note:	If you are using the line-side-dm3, isdn-dm3, and cas-dm3 line types, all options in this section are ignored, except calling-party-number.

called-party-number-plan

- Type: Optional
- Default Value: isdn
- Valid Values: The string: unknown, isdn, telephony, or private
- Identifies the number plan of the party called.

called-party-number-type

- Type: Optional
- Default Value: national
- Valid Values: national, international, en-block, overlap
- Indicates the type of number for the party called.

calling-party-number-plan

- Type: Optional
- Default Value: isdn
- Valid Values: unknown, isdn, telephony, or private
- Identifies the number plan of the calling party.

calling-party-number-type

- Type: Optional
- Default Value: national
- Valid Values: national, international, en-block, or overlap
- Indicates the type of number for the calling party.

calling-party-number

- Type: Optional
- Default Value: None
- Valid Values: Any digit string not longer than 31 characters
- Specifies the number of the calling party.

Note:	The calling party number (`isdn-dm3`) is always sent and cannot be an empty string when CPD Server is used with Dialogic DM3 hardware. If you are using `isdn-dm3` and `cas-dm3` line types this option is mandatory, and its value cannot be an empty string.

engage-cpd-on-call-setup

- Type: Optional
- Default Value: `no/false`
- Valid Values: `yes/true` or `no/false`
- This option can also be located in the HMP section.
- Instructs the CPD Server to engage in Call Progress Analysis during the ISDN call setup. This option is especially useful in the following situations:
- Setup for ISDN to non-ISDN destination calls.
- In-band signaling is present before the connection is established.

Note:	For `isdn-dm3, sip-hmp,` and `sip-hmp-asm` line-types, CPD Server ignores a voice message and detects only tones during preconnect call progress analysis (CPA), even if the voice message is detected before the tone. To enable preconnect CPA (that is, the CPA is performed before the connection is established), set this option to yes in the `isdn` section for the `isdn-dm3` line-type or in the `hmp` section for all HMP line-types. If you are reconfiguring specific tones for Dialogic DM3 hardware, for `isdn-dm3, sip-hmp,` and `sip-hmp-asm` line-types, set this option to `yes`. For more information, see Reconfiguring Tones for Dialogic DM3 Hardware.

isdn-trace

- Type: Optional
- Default Value: `no/false`
- Valid Values: `yes/true` or `no/false`

Specifies whether CPD Server creates an ISDN board resource (d-channel) trace log (`value = yes` or `true`) or does not create this trace log (`value = no` or `false`). Genesys recommends that you use this `isdn-trace` option for log purposes only when additional call information is required. The default value (`no` or `false`) normally applies.

If the value is set to `yes` or `true`, CPD Server receives error messages (`Device busy` and/or `Invalid device`) after approximately 100,000 calls. CPD Server then stops Outbound calling due to the inability to dial through the Dialogic channels.

If this happens, complete the following procedure:

1. Stop the CPD Server.
2. Change the value of the `isdn-trace` option to `no` to turn off the ISDN board resource (d-channel) trace log.
3. Restart the Dialogic drivers.
4. Restart the CPD Server.

isdn-trace-file-name-prefix

- Type: Optional
- Default Value: `isdn_`
- Valid Values: Any string
- Specifies the prefix for the ISDN trace file.

presentation-indicator

- Type: Optional
- Default Value: `restricted`
- Valid Values: `allowed, restricted, not-available`
- Specifies whether a presentation indicator (Caller_ID) is allowed, restricted, or not available for an ISDN call-setup message.

use-channel-state

- Type: Optional
- Default Value: `no/false`
- Valid Values: `yes/true` or `no/false`
- Specifies whether CPD Server uses a channel state (value = `yes` or `true`) or does not use a channel state (value = `no` or `false`). If `yes` or `true`, an in-service message is sent to the switch through the d-channel (ISDN line). Available in North America only.

user-info-layer-1

- Type: Optional
- Default Value: `g711-ulaw;`
- Valid Values: `g711-ulaw; g711-alaw;`
- CPD Server reads the value of this option from the ISDN section of its application object and applies the value to the `user info layer 1` field in the ISDN call setup message.

Line-Side Section

The line-side section is ignored when CPD Server is used with Dialogic DM3 hardware. The /config/fcd/pcd Dialogic files must be used instead.

Note:	If you are using the line-side-dm3 and cas-dm3 line types, you cannot modify the bit mask parameters through the CPD Server configuration options. Instead, you use the Dialogic *.config, *.fcd, and *.pcd files. These files are located at $DIALOGICDIR\data. Consult the Dialogic documentation for more information on modifying these files.

off-hook-bit-mask

- Type: Optional
- Default Value: a-on;b-on;
- Valid Values: a-off;b-on; a-off;b-off; a-on;b-off; a-on;b-on;
- Specifies the mask for a T1/E1 line that is off-the-hook.

on-hook-bit-mask

- Type: Optional
- Default Value: a-off;b-on;
- Valid Values: a-off;b-on; a-off;b-off; a-on;b-off; a-on;b-on;
- Specifies the mask for a T1/E1 line that is on-the-hook.

rcv-idle-bit-mask

- Type: Optional
- Default Value: a-off;b-on;
- Valid Values: a-off;b-on; a-off;b-off; a-on;b-off; a-on;b-on;
- Specifies the mask that indicates a T1/E1 line in an idle state.

snd-idle-bit-mask

- Type: Optional
- Default Value: a-off;b-on;
- Valid Values: a-off;b-on; a-off;b-off; a-on;b-off; a-on;b-on;

- Specifies the mask that the Dialogic board sends to a switch to make a T1/E1 line idle.

Tones Section

Tones Section

The `tones` section does not configure the tone definition when CPD Server is used with Dialogic DM3 hardware, with the exception of the `use-fax2-as-am-tone` and `forth-tone` options. If you need to reconfigure specific tones for Dialogic DM3 hardware, you can do so by creating a `dm3tones` section (see Reconfiguring Tones for Dialogic DM3 Hardware) or through `/config/fcd/pcd` Dialogic files.

Note:	This section is not used when the `line-type` option is set to `sip-hmp`.

Any of the following can be represented by a tone:

- Busy signal
- Dial tone
- Fax machine
- Ring back
- Beep signal

Within each of these broad categories of tones, there are specific types of tones--for example a local dial tone or an international dial tone.

Note:	The `qual` templates are configured in the `/config/fcd/pcd` Dialogic files when CPD Server is use with Dialogic DM3 hardware. If you are using the `line-side-dm3, isdn-dm3`, or `cas-dm3` line types, you can either reconfigure specific tones using a separate `dm3tones` section (Reconfiguring Tones for Dialogic DM3 Hardware) or by changing the tone definition in the Dialogic `*.config, *.fcd`, and `*.pcd` files. These files are located at `$DIALOGICDIR\data`. For more information on modifying these files, consult the Dialogic documentation. For `isdn-dm3, sip-hmp`, and `sip-hmp-asm` line-types, CPD Server ignores a voice message and detects only tones during pre-connect call progress analysis (CPA), even if the voice message is detected before the tone. To enable preconnect CPA (that is, the CPA is performed before the connection is established), set the `engage-cpd-on-call-setup` option to yes in either the `isdn` section for the `isdn-dm3` line-type or in the `hmp` section for all HMP line-types.

Tone Options

There is a Dialogic option for each type of tone, that the Genesys configuration environment recognizes. These tone options are as follows:

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

Tone Parameters provides more details about each of these tone options.

Tone Parameters

For the tone options described in this section, a series of numbers separated by semicolons represents the following nine parameters for each tone:

- Frequency of first tone
- Frequency deviation for first tone
- Frequency of second tone
- Frequency deviation for second tone
- On duration
- Ontime deviation
- Off duration
- Offtime deviation
- Repetition count

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

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

Note:	In the following tone option descriptions, parameters are in 10 ms units.

busy-tone-1

- Type: Optional
- Default Value: `500;200;0;0;55;40;55;40;4;`
- Valid Values: Any string of numbers separated by semicolons and representing the nine tone parameters. (see Tone Parameters).
- Defines a template for the first busy tone.

busy-tone-2

- Type: Optional
- Default Value: `500;200;500;200;55;40;55;40;4;`
- Valid Values: Any string of numbers separated by semicolons and representing the nine tone parameters. (see Tone Parameters).
- Defines a template for the second busy tone.

disconnect-tone

- Type: Optional
- Default values: `500;200;500;200;55;40;55;40;10;`
- Valid Values: Any string of numbers separated by semicolons and representing the nine tone parameters.
- Defines a template for disconnect tone, also known as a "fast busy tone."

extra-dial-tone

- Type: Optional
- Default Value: `401;125;401;125;0;0;0;0;0;`
- Valid Values: Any string of numbers separated by semicolons and representing the nine tone parameters. (see Tone Parameters).
- Defines a template for an extra dial tone.

fax-tone-1

- Type: Optional
- Default Value: `2150;150;0;0;25;-25;0;0;0;`
- Valid Values: Any string of numbers separated by semicolons representing the nine tone parameters.

(see Tone Parameters).

- Defines a template for the first FAX tone.

fax-tone-2

- Type: Optional
- Default Value: `1100;50;0;0;25;-25;0;0;0;`
- Valid Values: Any string of numbers separated by semicolons representing the nine tone parameters. (see Tone Parameters).
- Defines a template for a second FAX tone.

forth-tone

- Type: Optional
- Default Value: `0;0;0;0;0`
- Valid Values: Any string of numbers separated by semicolons representing the following tone parameters: (see Tone Parameters).

Frequency of first tone:

- Frequency range: 200 Hz to 4000 Hz
- Frequency resolution: 1 Hz

Frequency of second tone:

- Frequency range: 200 Hz to 4000 Hz
- Frequency resolution: 1 Hz

Amplitude of first tone:

- (E-1) -40 dBm0 to +0 dBm per tone nominal
- (T-1) -43 dBm0 to -3 dBm per tone nominal

Amplitude of second tone:

- (E-1) -40 dBm0 to +0 dBm per tone nominal
- (T-1) -43 dBm0 to -3 dBm per tone nominal

Duration: 10 millisecond increments

The forth-tone option enables or disables a beep signal that alerts an agent immediately before a customer is connected to him or her. When an agent is in the *engaged* mode and waiting to be connected to a customer, this low-frequency tone notifies him or her that a connection is imminent. You can configure the tone frequency, duration, and amplitude of the signal. CPD Server supports this signal on DM3 hardware and SIP-HMP-ASM line-type.

intl-dial-tone

- Type: Optional
- Default Value: `402;125;402;125;0;0;0;0;0;`
- Valid Values: Any string of numbers separated by semicolons representing the nine tone parameters (see Tone Parameters).
- Defines a template for an international dial tone.

local-dial-tone

- Type: Optional
- Default Value: `400;125;400;125;0;0;0;0;0;`
- Valid Values: Any string of numbers separated by semicolons and representing the nine tone parameters (see Tone Parameters).
- Defines a template for a local dial tone.

ring-back-tone-1

- Type: Optional
- Default Value: `450;150;0;0;130;105;580;415;0;`
- Valid Values: Any string of numbers separated by semicolons and representing nine tone parameters (see Tone Parameters).
- This option defines a template for the first ring-back tone.

ring-back-tone-2

- Type: Optional
- Default Value: `450;150;450;150;130;105;580;415;0;`
- Valid Values: A string of numbers separated by semicolons representing nine tone parameters (see Tone Parameters).
- This option defines a template for a second ring-back tone.

sit-tone

- Type: Optional
- Default Value: `900;1000;5;50;0;0;0;0;0;0;0;0;`
- Valid Values: Any string of numbers separated by semicolons and representing the following tone parameters.
- Defines the SIT signal parameters in the Dialogic DX_CAP data structure.

A series of 12 numbers separated by semicolons representing these 12 parameters for the sit-tone:

- Lower Frequency: Lower bound for 1st tone in an SIT. (Call Analysis) Length: 2. Default Value: 900. Units: Hz.
- Upper Frequency: Upper bound for 1st tone in an SIT. (Call Analysis) Length: 2. Default Value: 1000. Units: Hz.
- Time Frequency: Minimum time for 1st tone in an SIT to remain in bounds. The minimum amount of time required for the audio signal to remain within the frequency detection range specified by upper frequency and lower frequency for it to be considered valid. (Call Analysis) Length: 1. Default Value: 5. Units: 10 ms
- Maximum Time Frequency: Maximum allowable time for 1st tone in an SIT to be present. Length: 1. Default Value: 0. Units: 10 ms.
- Lower Bound for 2nd Frequency: Lower bound for 2nd tone in an SIT. Length: 1. Default Value: 0. Units: Hz.
- Upper Bound for 2nd Frequency: Upper bound for 2nd tone in an SIT. Length: 1. Default Value: 0. Units: Hz.
- Time for 2nd Frequency: Minimum time for 2nd tone in an SIT to remain in bounds. Length: 1. Default Value: 0. Units: 10 ms.
- Maximum Time for 2nd Frequency: Maximum allowable time for 2nd tone in an SIT to be present. Length: 1. Default Value: 0. Units: 10 ms.
- Lower Bound for 3rd Frequency: Lower bound for 3rd tone in an SIT. Length: 1. Default Value: 0. Units: Hz.
- Upper Bound for 3rd Frequency: Upper bound for 3rd tone in an SIT. Length: 1. Default Value: 0. Units: Hz.
- Time for 3rd Frequency: Minimum time for 3rd tone in an SIT to remain in bounds. Length: 1. Default Value: 0. Units: 10 ms.
- Maximum Time for 3rd Frequency: Maximum allowable time for 3rd tone in an SIT to be present. Length: 1. Default Value: 0. Units: 10 ms.

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

Reconfiguring Tones for Dialogic DM3 Hardware

If you want to reconfigure specific tones for Dialogic DM3 hardware, you can do so by configuring a `dm3tones` section and associated options, instead of changing the tone definition in the Dialogic `*.config, *.fcd`, and `*.pcd` files. Ten additional user tones can be defined for DM3 hardware in the `dm3tones` section. Options names for user tones are from `user-tone-1` to `user-tone-10`.

The option values for user tones have the same format as other tones in the `dm3tones` section, but they also require the addition of the `tone-id` parameter after the `treat-as` parameter.

Format Example:

The user tone format is as follows: `treat-as=<tone-name2>;tone-id=<id_of_the_tone>; numofseg; tn_rep_cnt; (tn_dflag; tn1_min; tn1_max; tn2_min; tn2_max; tn_twinmin; tn_twinmax; tnon_min; tnon_max; tnoff_min; tnoff_max)` The `id_of_the_tone` parameter is an integer within the interval 268 - 282 (which corresponds to the `TID_CUSTOM_SIT1` to `TID_CUSTOM_SIT15` Dialogic tones). Specific hardware might have restrictions on which tone IDs can be used. Check the Dialogic documentation to find the tone IDs that can be used for user tones in your configuration.

Note:	The `treat-as` and `tone-id` parameter is mandatory for user-defined tones.For a description of other parameters, see the configuration of non-user tones in the dm3tones section.

Format Example:

You can define additional busy tones that will be treated as `busy-tone-1` by defining `user-tone-1` with the following value:

```
treat-as=busy-tone-1;tone-id=268;3;1;(0;950;1020;0;0;0;0;32;45;0;5)
(0;1310;1430;0;0;0;0;15;30;0;5)(0;1740;1850;0;0;0;0;0;0;0;0).
```

This means that the additional tone with ID 268 is defined for the board. When this tone is detected during call progress analysis (CPA), CPD Server behaves as if `busy-tone-1` is detected.

Configuring a dm3tones Section

Start

1. Configure a new `dm3tones` section.
2. In this new section, configure the following tones, as required:
    - `busy-tone-1`
    - `busy-tone-2`
    - `disconnect-tone`
    - `extra-dial-tone`
    - `fax-tone-1`
    - `fax-tone-2`
    - `intl-dial-tone`
    - `local-dial-tone`
    - `ring-back-tone-1`
    - `ring-back-tone-2`
    - `sit-no-circuit`
    - `sit-operator-intercept`

- sit-reorder
- sit-vacant-circuit

Note:	The format for these option differs from those in the tone section. See Option Value Format for dm3tones Section. For a description of these options, with the exception of the sit options, refer to their associated descriptions in the tones section at the page number indicated. For the sit options, the option is described in this section. The sit-no-circuit, sit-operator-intercept, sit-vacant-circuit, and sit-reorder tones are specific to DM3 hardware. As such, they do not appear in the tones section.

3. For isdn-dm3, sip-hmp, and sip-hmp-asm line-types, set the engage-cpd-on-call-setup option to yes.

End

Option Value Format for dm3tones Section

The format for the option values in this dm3tones section differs from those in the tones section, and is as follows:
treat-as=<tone-name2>; numofseg; tn_rep_cnt; (tn_dflag; tn1_min; tn1_max; tn2_min; tn2_max; tn_twinmin; tn_twinmax; tnon_min; tnon_max; tnoff_min; tnoff_max) Where:

- The treat-as=<tone-name2> parameter is used to map one tone to another and is optional. It is applicable for mapping SIT tones to any of the other tones.
- <tone-name2> can be the name of any of the other tones, whether or not its value is defined this section.

The text in the parentheses can be repeated any number of times, as specified by the numofseg parameter.

The format parameters include:

- numofseg--Specifies the number of segments (defined by the parameters in the parentheses) for a multi-segment tone.
- tn_dflag--Specifies whether the tone is a dual tone (a value of 1) or a single tone (a value of 0).
- tn_rep_cnt--Specifies the debounce repetition count.
- tn1_min--Specifies the minimum frequency in Hertz (Hz) for tone 1.
- tn1_max--Specifies the maximum frequency in Hz for tone 1.
- tn2_min--Specifies the minimum frequency in Hz for tone 2.
- tn2_max--Specifies the maximum frequency in Hz for tone 2.
- tn_twinmin--Specifies the minimum frequency in Hz of the single tone proxy for the dual tone.
- tn_twinmax--Specifies the maximum frequency in Hz of the single tone proxy for the dual tone.
- tnon_min--Specifies the debounce minimum ON time in 10 msec units.

- tnon_max--Specifies the debounce maximum ON time in 10 msec units.
- tnoff_min--Specifies the debounce minimum OFF time in 10 msec units.
- tnoff_max--Specifies the debounce maximum OFF time in 10 msec units.

Format Example:

You can redefine the sit-vacant-circuit tone to treat it as busy-tone-1, and configure the sit-vacant-circuit option with the following value: treat-as=busy-tone-1;3;1;(0;950;1020;0;0;0;0;32;45;0;5)(0;1310;1430;0;0;0;0;15;30;0;5)(0;1740;1850;0;0;0;0;0;0;0;0).

This means that the board's tone definition for sit-vacant-circuit will be redefined according to previously listed parameters and, when this tone is detected during call progress analysis (CPA), CPD Server will behave as if busy-tone-1 was detected.

Notes:	• Changes to the options in the dm3tones section take effect only after CPD Server is restarted. • Redefined tones remain active in the DM3 board until the Dialogic service is restarted. • Some boards may not support redefining every tone. • When redefining tones, make sure that their redefinition does not prevent a protocol from working properly. Consult your Dialogic manuals for more information. • There are no default values for the options in this dm3tones section. The Dialogic driver loads the default tones definition into the DM3 board during startup of the Dialogic system. The values of the tones definition may depend on Dialogic protocol and other Dialogic configuration. Consult your Dialogic manuals for more information

sit-no-circuit

- Type: Optional
- Default Value: none
- Valid Values: See Option Value Format for dm3tones Section
- Defines a template for a No Circuit SIT tone.

sit-operator-intercept

- Type: Optional

- Default Value: none
- Valid Values: See Option Value Format for dm3tones Section
- Defines a template for an Intercept SIT tone.

sit-reorder

- Type: Optional
- Default Value: none
- Valid Values: See Option Value Format for dm3tones Section
- Defines a template for a Reorder SIT tone.

sit-vacant-circuit

- Type: Optional
- Default Value: none
- Valid Values: See Option Value Format for dm3tones Section

Defines a template for a Vacant Circuit (Vacant Code) tone.

AM-beep-tones Section

The am-beep-tones section contains definitions of the answering machine final beep tone, as given by the option `am-beep-tone->n>`. This section can contain up to 10 different tone definitions, with each tone numbered 1 to 10 in the option name (for example, `am-beep-tone-1`). CPD Server only processes this section if the AM final beep detection is enabled with the option am-beep-detect set to `yes` or `true`.

am-beep-tone-<n>

- Type: Optional
- Default Value: No default value
- Valid Values: Valid definition of a beep tone, in the format described below.
- Changes Take Effect: After the application is restarted
- Specifies an AM final beep tone in the following format:

`<tone type>;<first frequency, Hz>;<first frequency deviation,Hz>;<second frequency, Hz>;<second frequency deviation, Hz>`
where:
`<tone type>`--1 (single-frequency tone) or 2 (dual-frequency tone).
`<first frequency, Hz>`--First frequency in Hz.
`<first frequency deviation, Hz>`--First frequency deviation in Hz.
`<second frequency, Hz>`--Second frequency in Hz; `0` (zero) if single-frequency tone.
`<second frequency, Hz>`--Second frequency deviation in Hz; `0` (zero) if single-frequency tone.

- For example, the following is a definition for a single-frequency tone:

```
am-beep-tone-1 = 1;1000;50;0;0
```

Note:	Only burst tones (no cadence) can be configured and detected as AM final beep tones. CPD Server detects the trailing edge of the tone burst.

License Section

The License section contains two options that support the licensing of CPD Server.

license-file

- Type: Optional
- Default Value: `None`
- Valid Values: Any string
- Changes Take Effect: After the application is restarted
- Specifies the license address in one of these formats:
- The host name and port of the license server, as specified in the SERVER line of the license file, in the port@host format--for example:
  `7260@ctiserver`
- The full path to and the exact name of the license file--for example:
  `/opt/mlink/license/license.dat`

num-occ-port-licenses

- Type: Optional
- Default Value: `0` (all available licenses)
- Valid Values: `max` or an integer from `0` to `9999`
- Specifies how many licenses CPD checks out initially.
- If the value of this option is increased, CPD will attempt to check out an additional amount of licenses. This change takes effect immediately.
- If the value of this option is decreased, CPD Server will check in the extra licenses and will not release the ports. CPD Server will not use the ports until the number of used ports falls below the number of licenses available for the CPD Server.

Correlation Between CPD Server Options and Dialogic Boards

This section provides information on the correlation between CPD Server options, Dialogic Board types, and Dialer modes.

In the tables, note the following:

- No indicates that the option is not used in the given Dialing Mode / Board Type / Line-Type.
- Yes indicates that the Option is used in the given Dialing Mode / Board Type / Line-Type.
- A dash (-) indicates that the Dialing Mode / Board Type / Line-Type combination is not possible.

ASM Dialer Mode

The section provides correlation information on the ASM (Active Switching Matrix) Dialer Mode for the following board types (see the accompanying tables):

- DMV
- JCT
- HMP

DMV Dialogic Board Type

Section	CPD Server Option	analogue line-type	line-side line-type	line-side-dm3 line-type	isdn line-type	isdn-dm3 line-type	cas-dm3 line-type	sip-hmp line type	sip-hmp-asm line-type
tones	busy-tone-1	—	—	—	—	No	No	—	—
	busy-tone-2	—	—	—	—	No	No	—	—
	extra-dial-tone	—	—	—	—	No	No	—	—
	fax-tone-1	—	—	—	—	No	No	—	—
	fax-tone-2	—	—	—	—	No	No	—	—
	intl-dial-tone	—	—	—	—	No	No	—	—
	local-dial-tone	—	—	—	—	No	No	—	—
	ring-	—	—	—	—	No	No	—	—

	back-tone-1								
	ring-back-tone-2	—	—	—	—	No	No	—	—
	disconnect-tone	—	—	—	—	No	No	—	—
	forth-tone	—	—	—	—	Yes	Yes	—	—
	sit-tone	—	—	—	—	No	No	—	—
AM-beep-tones	am-beep-tone-<n>	—	—	—	—	Yes	Yes	—	—
line-side	off-hook-bit-mask	—	—	—	—	No	No	—	—
	on-hook-bit-mask	—	—	—	—	No	No	—	—
	rcv-idle-bit-mask	—	—	—	—	No	No	—	—
	snd-idle-bit-mask	—	—	—	—	No	No	—	—
isdn	called-party-number-plan	—	—	—	—	No	No	—	—
	called-party-number-type	—	—	—	—	No	No	—	—
	calling-party-number	—	—	—	—	Yes	Yes	—	—
	calling-party-number-plan	—	—	—	—	No	No	—	—
	calling-party-number-type	—	—	—	—	No	No	—	—
	isdn-	—	—	—	—	No	No	—	—

	trace								
	isdn-trace-file-name-prefix	—	—	—	—	No	No	—	—
	presentation-indicator	—	—	—	—	No	No	—	—
	use-channel-state	—	—	—	—	Yes	Yes	—	—
	engage-cpd-on-call-setup	—	—	—	—	No	No	—	—
	user-info-layer-1	—	—	—	—	No	No	—	—
hmp	audio-codecs	—	—	—	—	No	No	—	—
	sip-local-address	—	—	—	—	No	No	—	—
	sip-proxy	—	—	—	—	No	No	—	—
general	am-beep-detect	—	—	—	—	Yes	Yes	—	—
	am-beep-detect-timeout	—	—	—	—	Yes	Yes	—	—
	continuous-no-signal	—	—	—	—	No	No	—	—
	conversation-file-name-prefix	—	—	—	—	Yes	Yes	—	—
	cpd-file-name-prefix	—	—	—	—	Yes	Yes	—	—
	cpd-recording	—	—	—	—	Yes	Yes	—	—
	cpd-if-established	—	—	—	—	No	No	—	—
	destination-busy-	—	—	—	—	Yes	Yes	—	—

	timeout								
	keep-channels-open	—	—	—	—	No	No	—	—
	line-type	—	—	—	—	Yes	Yes	—	—
	location	—	—	—	—	Yes	Yes	—	—
	max-number-ports-to-record	—	—	—	—	Yes	Yes	—	—
	off-hook-delay	—	—	—	—	No	No	—	—
	off-hook-timeout	—	—	—	—	No	No	—	—
	on-hook-timeout	—	—	—	—	No	No	—	—
	out-of-service-attempts	—	—	—	—	Yes	Yes	—	—
	out-of-service-timeout	—	—	—	—	Yes	Yes	—	—
	pre-connect-cpd-priority	—	—	—	—	No	No	—	—
	post-connect-cpd-priority	—	—	—	—	No	No	—	—
	sit-detection	—	—	—	—	Yes	Yes	—	—
	tscall	—	—	—	—	No	No	—	—
	tsclear	—	—	—	—	No	No	—	—
	use-busy2-as-nu-tone	—	—	—	—	No	No	—	—
	use-fax2-as-am-tone	—	—	—	—	No	No	—	—
	use-engaging-lines	—	—	—	—	Yes	Yes	—	—

	wait-off-hook	—	—	—	—	No	No	—	—

JCT Dialogic Board Type

Section	CPD Server Option	analogue line-type	line-side line-type	line-side-dm3 line-type	isdn line-type	isdn-dm3 line-type	cas-dm3 line-type	sip-hmp line type	sip-hmp-asm line-type
tones	busy-tone-1	—	—	—	Yes	—	—	—	—
	busy-tone-2	—	—	—	Yes	—	—	—	—
	extra-dial-tone	—	—	—	Yes	—	—	—	—
	fax-tone-1	—	—	—	Yes	—	—	—	—
	fax-tone-2	—	—	—	Yes	—	—	—	—
	intl-dial-tone	—	—	—	Yes	—	—	—	—
	local-dial-tone	—	—	—	Yes	—	—	—	—
	ring-back-tone-1	—	—	—	Yes	—	—	—	—
	ring-back-tone-2	—	—	—	Yes	—	—	—	—
	disconnect-tone	—	—	—	Yes	—	—	—	—
	forth-tone	—	—	—	Yes	—	—	—	—
	sit-tone	—	—	—	Yes	—	—	—	—
line-side	off-hook-bit-mask	—	—	—	No	—	—	—	—
	on-hook-bit-mask	—	—	—	No	—	—	—	—
	rcv-idle-bit-mask	—	—	—	No	—	—	—	—
	snd-idle-bit-mask	—	—	—	No	—	—	—	—

isdn	called-party-number-plan	—	—	—	Yes	—	—	—	—
	called-party-number-type	—	—	—	Yes	—	—	—	—
	calling-party-number	—	—	—	Yes	—	—	—	—
	calling-party-number-plan	—	—	—	Yes	—	—	—	—
	calling-party-number-type	—	—	—	Yes	—	—	—	—
	isdn-trace	—	—	—	Yes	—	—	—	—
	isdn-trace-file-name-prefix	—	—	—	Yes	—	—	—	—
	presentation-indicator	—	—	—	Yes	—	—	—	—
	use-channel-state	—	—	—	Yes	—	—	—	—
	engage-cpd-on-call-setup	—	—	—	Yes	—	—	—	—
	user-info-layer-1	—	—	—	Yes	—	—	—	—
hmp	audio-codecs	—	—	—	No	—	—	—	—
	sip-proxy	—	—	—	No	—	—	—	—
general	continuous-no-signal	—	—	—	Yes	—	—	—	—
	conversation-file-name-prefix	—	—	—	Yes	—	—	—	—

	cpd-file-name-prefix	—	—	—	Yes	—	—	—	—
	cpd-recording	—	—	—	Yes	—	—	—	—
	cpd-if-established	—	—	—	No	—	—	—	—
	destination-busy-timeout	—	—	—	Yes	—	—	—	—
	keep-channels-open	—	—	—	Yes	—	—	—	—
	line-type	—	—	—	Yes	—	—	—	—
	location	—	—	—	Yes	—	—	—	—
	max-number-ports-to-record	—	—	—	Yes	—	—	—	—
	off-hook-delay	—	—	—	No	—	—	—	—
	off-hook-timeout	—	—	—	No	—	—	—	—
	on-hook-timeout	—	—	—	No	—	—	—	—
	out-of-service-attempts	—	—	—	Yes	—	—	—	—
	out-of-service-timeout	—	—	—	Yes	—	—	—	—
	pre-connect-cpd-priority	—	—	—	Yes	—	—	—	—
	post-connect-cpd-priority	—	—	—	Yes	—	—	—	—
	sit-detection	—	—	—	Yes	—	—	—	—
	tscall	—	—	—	No	—	—	—	—
	tsclear	—	—	—	No	—	—	—	—

	use-busy2-as-nu-tone	—	—	—	No	—	—	—	—
	use-fax2-as-am-tone	—	—	—	No	—	—	—	—
	use-engaging-lines	—	—	—	Yes	—	—	—	—
	wait-off-hook	—	—	—	No	—	—	—	—

HMP Dialogic Board Type

Section	CPD Server Option	analogue line-type	line-side line-type	line-side-dm3 line-type	isdn line-type	isdn-dm3 line-type	cas-dm3 line-type	sip-hmp line type	sip-hmp-asm line-type
tones	busy-tone-1	—	—	—	—	—	—	—	No
	busy-tone-2	—	—	—	—	—	—	—	No
	extra-dial-tone	—	—	—	—	—	—	—	No
	fax-tone-1	—	—	—	—	—	—	—	No
	fax-tone-2	—	—	—	—	—	—	—	No
	intl-dial-tone	—	—	—	—	—	—	—	No
	local-dial-tone	—	—	—	—	—	—	—	No
	ring-back-tone-1	—	—	—	—	—	—	—	No
	ring-back-tone-2	—	—	—	—	—	—	—	No
	disconnect-tone	—	—	—	—	—	—	—	No
	forth-tone	—	—	—	—	—	—	—	Yes
	sit-tone	—	—	—	—	—	—	—	No
AM-beep-tones	am-beep-tone-<n>	—	—	—	—	—	—	—	Yes

line-side	off-hook-bit-mask	—	—	—	—	—	—	—	No
	on-hook-bit-mask	—	—	—	—	—	—	—	No
	rcv-idle-bit-mask	—	—	—	—	—	—	—	No
	snd-idle-bit-mask	—	—	—	—	—	—	—	No
isdn	called-party-number-plan	—	—	—	—	—	—	—	No
	called-party-number-type	—	—	—	—	—	—	—	No
	calling-party-number	—	—	—	—	—	—	—	Yes
	calling-party-number-plan	—	—	—	—	—	—	—	No
	calling-party-number-type	—	—	—	—	—	—	—	No
	isdn-trace	—	—	—	—	—	—	—	No
	isdn-trace-file-name-prefix	—	—	—	—	—	—	—	No
	presentation-indicator	—	—	—	—	—	—	—	No
	use-channel-state	—	—	—	—	—	—	—	No
	engage-cpd-on-call-setup	—	—	—	—	—	—	—	No
	user-	—	—	—	—	—	—	—	No

	info-layer-1								
hmp	audio-codecs	—	—	—	—	—	—	—	Yes
	sip-local-address	—	—	—	—	—	—	—	Yes
	sip-proxy	—	—	—	—	—	—	—	Yes
general	am-beep-detect	—	—	—	—	—	—	—	Yes
	am-beep-detect-timeout	—	—	—	—	—	—	—	Yes
	continuous-no-signal	—	—	—	—	—	—	—	Yes
	conversation-file-name-prefix	—	—	—	—	—	—	—	Yes
	cpd-file-name-prefix	—	—	—	—	—	—	—	Yes
	cpd-recording	—	—	—	—	—	—	—	Yes
	cpd-if-established	—	—	—	—	—	—	—	No
	destination-busy-timeout	—	—	—	—	—	—	—	Yes
	keep-channels-open	—	—	—	—	—	—	—	No
	line-type	—	—	—	—	—	—	—	Yes
	location	—	—	—	—	—	—	—	Yes
	max-number-ports-to-record	—	—	—	—	—	—	—	Yes
	off-hook-delay		—	—	—	—	—	—	No
	off-		—	—	—	—	—	—	No

	hook-timeout								
	on-hook-timeout	—	—	—	—	—	—	—	No
	out-of-service-attempts	—	—	—	—	—	—	—	Yes
	out-of-service-timeout	—	—	—	—	—	—	—	Yes
	pre-connect-cpd-priority	—	—	—	—	—	—	—	No
	post-connect-cpd-priority	—	—	—	—	—	—	—	No
	sit-detection	—	—	—	—	—	—	—	Yes
	tscall	—	—	—	—	—	—	—	No
	tsclear	—	—	—	—	—	—	—	No
	use-busy2-as-nu-tone	—	—	—	—	—	—	—	No
	use-fax2-as-am-tone	—	—	—	—	—	—	—	No
	use-engaging-lines	—	—	—	—	—	—	—	Yes
	wait-off-hook	—	—	—	—	—	—	—	No

Transfer Mode (TM) Dialer Mode

The section provides correlation information on the TM Dialer Mode for the following board types (see the accompanying tables):

- DMV
- JCT
- HMP

DMV Dialogic Board Type

Section	CPD	analogue	line-side	line-side-	isdn line-	isdn-dm3	cas-dm3	sip-hmp	sip-hmp-

	Server Option	line-type	line-type	dm3 line-type	type	line-type	line-type	line type	asm line-type
tones	busy-tone-1	—	—	No	—	—	—	—	—
	busy-tone-2	—	—	No	—	—	—	—	—
	extra-dial-tone	—	—	No	—	—	—	—	—
	fax-tone-1	—	—	No	—	—	—	—	—
	fax-tone-2	—	—	No	—	—	—	—	—
	intl-dial-tone	—	—	No	—	—	—	—	—
	local-dial-tone	—	—	No	—	—	—	—	—
	ring-back-tone-1	—	—	No	—	—	—	—	—
	ring-back-tone-2	—	—	No	—	—	—	—	—
	disconnect-tone	—	—	No	—	—	—	—	—
	forth-tone	—	—	No	—	—	—	—	—
	sit-tone	—	—	No	—	—	—	—	—
AM-beep-tones	am-beep-tone-<n>	—	—	Yes	—	—	—	—	—
line-side	off-hook-bit-mask	—	—	No	—	—	—	—	—
	on-hook-bit-mask	—	—	No	—	—	—	—	—
	rcv-idle-bit-mask	—	—	No	—	—	—	—	—
	snd-idle-bit-mask	—	—	No	—	—	—	—	—

isdn	called-party-number-plan	—	—	No	—	—	—	—	—
	called-party-number-type	—	—	No	—	—	—	—	—
	calling-party-number	—	—	No	—	—	—	—	—
	calling-party-number-plan	—	—	No	—	—	—	—	—
	calling-party-number-type	—	—	No	—	—	—	—	—
	isdn-trace	—	—	No	—	—	—	—	—
	isdn-trace-file-name-prefix	—	—	No	—	—	—	—	—
	presentation-indicator	—	—	No	—	—	—	—	—
	use-channel-state	—	—	No	—	—	—	—	—
	engage-cpd-on-call-setup	—	—	No	—	—	—	—	—
	user-info-layer-1	—	—	No	—	—	—	—	—
hmp	audio-codecs	—	—	No	—	—	—	—	—
	sip-local-address	—	—	No	—	—	—	—	—
	sip-proxy	—	—	No	—	—	—	—	—
general	am-beep-detect	—	—	Yes	—	—	—	—	—

	am-beep-detect-timeout	—	—	Yes	—	—	—	—	—
	continuous-no-signal	—	—	No	—	—	—	—	—
	conversation-file-name-prefix	—	—	No	—	—	—	—	—
	cpd-file-name-prefix	—	—	Yes	—	—	—	—	—
	cpd-recording	—	—	Yes	—	—	—	—	—
	cpd-if-established	—	—	No	—	—	—	—	—
	destination-busy-timeout	—	—	Yes	—	—	—	—	—
	keep-channels-open	—	—	No	—	—	—	—	—
	line-type	—	—	Yes	—	—	—	—	—
	location	—	—	Yes	—	—	—	—	—
	max-number-ports-to-record	—	—	No	—	—	—	—	—
	off-hook-delay	—	—	Yes	—	—	—	—	—
	off-hook-timeout	—	—	Yes	—	—	—	—	—
	on-hook-timeout	—	—	Yes	—	—	—	—	—
	out-of-service-attempts	—	—	Yes	—	—	—	—	—
	out-of-service-timeout	—	—	Yes	—	—	—	—	—
	pre-connect-	—	—	No	—	—	—	—	—

	cpd-priority								
	post-connect-cpd-priority	—	—	No	—	—	—	—	—
	sit-detection	—	—	No	—	—	—	—	—
	tscall	—	—	Yes	—	—	—	—	—
	tsclear	—	—	Yes	—	—	—	—	—
	use-busy2-as-nu-tone	—	—	Yes	—	—	—	—	—
	use-fax2-as-am-tone	—	—	Yes	—	—	—	—	—
	use-engaging-lines	—	—	No	—	—	—	—	—
	wait-off-hook	—	—	Yes	—	—	—	—	—

JCT Dialogic Board Type

Section	CPD Server Option	analogue line-type	line-side line-type	line-side-dm3 line-type	isdn line-type	isdn-dm3 line-type	cas-dm3 line-type	sip-hmp line type	sip-hmp-asm line-type
tones	busy-tone-1	Yes	Yes	—	—	—	—	—	—
	busy-tone-2	Yes	Yes	—	—	—	—	—	—
	extra-dial-tone	Yes	Yes	—	—	—	—	—	—
	fax-tone-1	Yes	Yes	—	—	—	—	—	—
	fax-tone-2	Yes	Yes	—	—	—	—	—	—
	intl-dial-tone	Yes	Yes	—	—	—	—	—	—
	local-dial-tone	Yes	Yes	—	—	—	—	—	—
	ring-back-tone-1	Yes	Yes	—	—	—	—	—	—
	ring-back-tone-2	Yes	Yes	—	—	—	—	—	—

	disconnect-tone	Yes	Yes	—	—	—	—	—	—
	forth-tone	Yes	No	—	—	—	—	—	—
	sit-tone	Yes	Yes	—	—	—	—	—	—
line-side	off-hook-bit-mask	No	Yes	—	—	—	—	—	—
	on-hook-bit-mask	No	Yes	—	—	—	—	—	—
	rcv-idle-bit-mask	No	Yes	—	—	—	—	—	—
	snd-idle-bit-mask	No	Yes	—	—	—	—	—	—
isdn	called-party-number-plan	No	No	—	—	—	—	—	—
	called-party-number-type	No	No	—	—	—	—	—	—
	calling-party-number	No	No	—	—	—	—	—	—
	calling-party-number-plan	No	No	—	—	—	—	—	—
	calling-party-number-type	No	No	—	—	—	—	—	—
	isdn-trace	No	No	—	—	—	—	—	—
	isdn-trace-file-name-prefix	No	No	—	—	—	—	—	—
	presentation-indicator	No	No	—	—	—	—	—	—
	use-channel-	No	No	—	—	—	—	—	—

	state								
	engage-cpd-on-call-setup	No	No	—	—	—	—	—	—
	user-info-layer-1	No	No	—	—	—	—	—	—
hmp	audio-codecs	No	No	—	—	—	—	—	—
	sip-proxy	No	No	—	—	—	—	—	—
general	continuous-no-signal	Yes	Yes	—	—	—	—	—	—
	conversation-file-name-prefix	No	No	—	—	—	—	—	—
	cpd-file-name-prefix	Yes	Yes	—	—	—	—	—	—
	cpd-recording	Yes	Yes	—	—	—	—	—	—
	cpd-if-established	Yes	Yes	—	—	—	—	—	—
	destination-busy-timeout	Yes	Yes	—	—	—	—	—	—
	keep-channels-open	Yes	Yes	—	—	—	—	—	—
	line-type	Yes	Yes	—	—	—	—	—	—
	location	Yes	Yes	—	—	—	—	—	—
	max-number-ports-to-record	No	No	—	—	—	—	—	—
	off-hook-delay	Yes	Yes	—	—	—	—	—	—
	off-hook-timeout	Yes	Yes	—	—	—	—	—	—
	on-hook-	Yes	Yes	—	—	—	—	—	—

	timeout								
	out-of-service-attempts	Yes	Yes	—	—	—	—	—	—
	out-of-service-timeout	Yes	Yes	—	—	—	—	—	—
	pre-connect-cpd-priority	Yes	Yes	—	—	—	—	—	—
	post-connect-cpd-priority	Yes	Yes	—	—	—	—	—	—
	sit-detection	Yes	Yes	—	—	—	—	—	—
	tscall	Yes	Yes	—	—	—	—	—	—
	tsclear	Yes	Yes	—	—	—	—	—	—
	use-busy2-as-nu-tone	Yes	Yes	—	—	—	—	—	—
	use-fax2-as-am-tone	Yes	Yes	—	—	—	—	—	—
	use-engaging-lines	No	No	—	—	—	—	—	—
	wait-off-hook	Yes	Yes	—	—	—	—	—	—

HMP Dialogic Board Type

Section	CPD Server Option	analogue line-type	line-side line-type	line-side-dm3 line-type	isdn line-type	isdn-dm3 line-type	cas-dm3 line-type	sip-hmp line type	sip-hmp-asm line-type
tones	busy-tone-1	—	—	—	—	—	—	No	
	busy-tone-2	—	—	—	—	—	—	No	
	extra-dial-tone	—	—	—	—	—	—	No	
	fax-tone-1	—	—	—	—	—	—	No	
	fax-tone-2	—	—	—	—	—	—	No	
	intl-dial-tone	—	—	—	—	—	—	No	

	local-dial-tone	—	—	—	—	—	—	No	
	ring-back-tone-1	—	—	—	—	—	—	No	
	ring-back-tone-2	—	—	—	—	—	—	No	
	disconnect-tone	—	—	—	—	—	—	No	
	forth-tone	—	—	—	—	—	—	No	
	sit-tone	—	—	—	—	—	—	No	
AM-beep-tones	am-beep-tone-<n>	—	—	—	—	—	—	Yes	
line-side	off-hook-bit-mask	—	—	—	—	—	—	No	
	on-hook-bit-mask	—	—	—	—	—	—	No	
	rcv-idle-bit-mask	—	—	—	—	—	—	No	
	snd-idle-bit-mask	—	—	—	—	—	—	No	
isdn	called-party-number-plan	—	—	—	—	—	—	No	
	called-party-number-type	—	—	—	—	—	—	No	
	calling-party-number	—	—	—	—	—	—	No	
	calling-party-number-plan	—	—	—	—	—	—	No	—
	calling-	—	—	—	—	—	—	No	—

	party-number-type								
	isdn-trace	—	—	—	—	—	—	No	—
	isdn-trace-file-name-prefix	—	—	—	—	—	—	No	—
	presentation-indicator	—	—	—	—	—	—	No	—
	use-channel-state	—	—	—	—	—	—	No	—
	engage-cpd-on-call-setup	—	—	—	—	—	—	No	—
	user-info-layer-1	—	—	—	—	—	—	No	—
hmp	audio-codecs	—	—	—	—	—	—	Yes	—
	sip-local-address	—	—	—	—	—	—	Yes	—
	sip-proxy	—	—	—	—	—	—	Yes	—
general	am-beep-detect	—	—	—	—	—	—	Yes	—
	am-beep-detect-timeout	—	—	—	—	—	—	Yes	—
	continuous-no-signal	—	—	—	—	—	—	Yes	—
	conversation-file-name-prefix	—	—	—	—	—	—	No	—
	cpd-file-name-prefix	—	—	—	—	—	—	Yes	—
	cpd-recording	—	—	—	—	—	—	Yes	—

	cpd-if-established	—	—	—	—	—	—	No	—
	destination-busy-timeout	—	—	—	—	—	—	Yes	—
	keep-channels-open	—	—	—	—	—	—	No	—
	line-type	—	—	—	—	—	—	Yes	—
	location	—	—	—	—	—	—	Yes	—
	max-number-ports-to-record	—	—	—	—	—	—	No	—
	off-hook-delay	—	—	—	—	—	—	No	—
	off-hook-timeout	—	—	—	—	—	—	No	—
	on-hook-timeout	—	—	—	—	—	—	No	—
	out-of-service-attempts	—	—	—	—	—	—	Yes	—
	out-of-service-timeout	—	—	—	—	—	—	Yes	—
	pre-connect-cpd-priority	—	—	—	—	—	—	No	—
	post-connect-cpd-priority	—	—	—	—	—	—	No	—
	sit-detection	—	—	—	—	—	—	Yes	—
	tscall	—	—	—	—	—	—	No	—
	tsclear	—	—	—	—	—	—	No	—
	use-busy2-as-nu-tone	—	—	—	—	—	—	No	—
	use-fax2-as-am-tone	—	—	—	—	—	—	No	—

	use-engaging-lines	—	—	—	—	—	—	No	—
	wait-off-hook	—	—	—	—	—	—	No	—

Log Files

These topics discuss general logging and three types of log files:

- General Logging
- Record History Logging
- Extended Audit Trail Log
- gsw request log

General Logging

The applications that are part of the Outbound Contact Solution conform to Genesys logging standards. See the *Framework Management Layer User's Guide* for more information about enabling and modifying the output log.

Outbound Contact Server generates a log file after its first successful start. This log file contains configuration settings printout for OCS, as well as a detailed description of the actions that have occurred.

If you configure the Framework `keep-startup-file` option to `true`, a second log is also generated, which contains initial information about connecting to Configuration Server. For additional information on configuring this option, see the description of the `keep-startup-file` option in the *Framework Configuration Options Reference Manual*. The level of output is determined by the `verbose` option in the `log` section.

Record History Logging

Record History logging provides you with additional reporting options for calling lists. This logging process does not use database access to write logs; instead it uses flat (text) files that the customer defines. With flat files you can selectively turn logging on or off for different calling lists. Flat files must be imported into a database in order to generate reports (see your database administrator for assistance).

Prestart Information

Before you create a Record History Log for your calling list, make sure that you have:

- Properly configured the destination for your log (see Log Options Defined).
- Provided a sufficient amount of free disk space on the target drive.

Configuration Options

Two options configure the Record History Logs; they are `dial_log_destination` and `dial_log_delimiter`.

Log Options Defined

This topic describes the Record History Log options `dial_log_destination` and `dial_log_delimiter`. You must configure the option `dial_log_destination` to use Record History logging. Then, if the option `dial_log_delimiter` is omitted, a tab delimiter is used in the log file.

The `dial_log_destination` option is used to provide the path where a log for the calling list is stored. This option can be placed in the OCS application object or in a specific calling list object. First, OCS looks for this option in the `OCServer` section in the `Options` tab of a list; if does not find the option there, it looks on the `Options` tab (`OCServer` section) in the application object; if it still does not find this option, then no logging will be performed for the list.

The `dial_log_delimiter` option is used to create delimiters between the fields of the log for the calling list. Because the files for this option are flat files, delimiters must be added to the log. If the `dial_log_destination` option is configured, but the `dial_log_delimiter` is omitted, then tabs are used as the delimiters. By default, if the value of this option is not set, it uses a tab `(\t)` to delimit fields.

The `dial_log_delimiter` option can be placed in the `OCS Application` object or in a specific list. First OCS looks for this option in the `OCServer` section in the `Options` tab of a list. If OCS does not find this option there, it looks in the `Options` tab (`OCServer` section) in the application object. If it still does not find this option, then OCS will use a <tab> character for the delimiter. For more information about these options, see Outbound Contact Configuration Options.

File Structure

The file structure of a Record History Log is shown in columns that include:

- Tracking information.
- Information from Genesys mandatory fields.
- Information from user defined fields.

Information from only some of the Genesys mandatory fields are included when generating the `Record History` Log. These fields include:

- `record_id`
- `contact_info`
- `contact_info_type`
- `record_type`
- `record_status`
- `call_result`
- `attempt`
- `dial_sched_time`
- `call_time`
- `daily_from`
- `daiy_till`
- `tz_dbid`
- `agent_id`
- `chain_id`
- `chain_n`

This information is followed by rows of the Actions and Events for a record. The following table shows an example of the `Record History` log file structure.

Note:	The column name that is listed in the history log is not always the same name as it is in the calling list. For example, the `contact_info` field in the calling list corresponds to the phone field in the Record History Log).

Log File Structure

Example											
time	action	record_handle	list_id	campaign_id	group_id	ocs_app_id	tenant_id	connection_id	dn_id	Genesys Mandatory	User Defined

For more information about how to configure the user-defined fields so that the data from the user-defined fields will be written into the flat file, see the "Attaching Record Information to Desktop and OCS User Events" section in the *Outbound Contact Reference Manual* for more information.

Tracking Information

The first ten columns in the log represent the tracking information for each record. The following table shows the tracking fields and their descriptions. These fields are followed by Genesys mandatory fields.

Tracking Information Fields

Field	Description
#time	The time logging for the record began.
#action	The actions and events logged for the record.
#record_handle	The record_handle of the record.
#list_id	The DBID of the list.
#campaign_id	The DBID of the campaign.
#group_id	The DBID of the group.
#ocs_app_id	OCS's application DBID.
#tenant_id	The tenant ID.
#connection_id	The Connection_id of the call.
#dn	The dn used for this campaign.

Log Data

The subsequent rows in the Record History Log represent the event data for the record. The following table shows the actions and events that may display in the Record History Log. OCS will log numeric data using numeric values.

The fields in the Record History Log are separated by a dialing_log_delimiter, which is defined with the `dial_log_delimiter` option.

The data for each record in the log breaks when:

- The dialing session/campaign group for a campaign stops or starts.
- The format of the list changes (for example, a send attribute was added or changed).

Actions and Events in Record History Log

Action/Event	Value
DA_CALL_DIALED_OUTBOUND	1
DA_CALL_DIALED_PREVIEW	2
DA_CALL_DIALED_CALLBACK	3
DA_CALL_DIALED_CHAIN	4
DA_RECORD_APPLY_TREATMENT	5

DA_CALL_QUEUED	6
DA_CALL_ESTABLISHED	7
DA_CALL_RELEASED	8
DA_RECORD_RESCHEDULED	9
DA_RECORD_UPDATED (UpdateCallCompletionStats, means that user data has changed)	10
DA_RECORD_PROCESSEDUsually is associated with a `RecordProcessed` request from the Desktop or an `EventAgentReady` when the record is updated in a calling list. DA_RECORD_PROCESSED action can be also triggered by other events--for instance, call abandoned, call released with an unsuccessful call result, record returned to a Calling List database table while the dialing session/campaign group is unloading or a call filter is changing.	11
DA_CALL_COMPLETED (to have an agent's timing statistics, not related to a record N.B. abandoned, cancel, do not call will be logged as DA_RECORD_PROCESSED with the corresponding call result)	12
DA_CALL_TRANSFERRED	13
DA_RECORD_PROCESSED_EVENT (A desktop `RecordProcesssed` event. In the log file, this event is always followed by DA_RECORD_PROCESSED.)	14

Log File Naming Conventions

OCS creates a name for each Record History Log. Every log file is named according to the following rule:

<ListName>_<ListDBID>_<CampaignDBID>_<GroupDBID>_<OCSServerApplicationDBID>_<DateTime>

The angle brackets indicate variables. Substitute the actual values for the type of data named in brackets.

The <DateTime> field uses this format: `mmddyy_hhmmss`

Where:
`mm` is replaced by a two-digit representation of the month.
dd is replaced by a two-digit representation of the day.
yy is replaced by a two-digit representation of the year.
hh is replaced by a two-digit representation of the hour.
`mm` is replaced by a two-digit representation of the minutes.
`ss` is replaced by a two-digit representation of the seconds.

For example:
`112305_193805`

Generating Record History Log Reports

The flat files generated in the Record History Log file must be imported into a database in order to generate reports. Genesys recommends that you check with your database administrator for assistance with this process.

Extended Audit Trail Log

Outbound Contact provides an audit trail for each outbound call dialed in preview, progressive, or predictive mode. A new OCS `log_call_stats` option creates a separate logging subsystem for Audit Logging to capture additional statistics on telephony events. OCS does not overwrite the existing data in this log or replace it with new data. The audit logging function adds data to a cumulative log.

The extended Audit Trail Log is discussed in detail in the sections below.

log_call_stats Configuration Option

You define the `log_call_stats` option in the OCS application object. If you set this option to `yes`, OCS creates the separate Audit Log. If you set the option to no or if the option is not present, the audit log function is not enabled.

Configuration of Audit Logging

The following table summarizes the configuration of Audit Logging in OCS release 8.1.0 and earlier, and in release 8.1.1. In Outbound Contact Server 8.1.0 and earlier, the configuration of Audit Logging is hard-coded in OCS. In Outbound Contact Server 8.1.1, the configuration of Audit Logging is user-defined using OCS Application-level configuration options. For more information about these options, refer to Section log call stats.

Configuration of Audit Logging

Audit Log File Attribute	**OCS 8.1.0 and Earlier Not Configurable**	**OCS 8.1.1 Configurable** [a]

File name	`CallStats.<MMDDYY_HHMMSS_MS>.log`
	For example:
	`CallStats.072303_182754_216.log`
		Configuration option: `all`
		This option accepts a full path and file name for the audit log.

File location	OCS Application folder	
Segment Size	`10 MB`	Configuration option: `segment`
Segment expiration	None	Configuration option: `expire`

a. These options are set in the OCS application configuration section `log_call_stats`, which is not to be confused with the configuration option of the same name (see Section log call stats).

Audit Log Record Format

OCS produces log records in a comma-delimited format, which can be easily imported into third-party applications, such as MS Excel or SQL Server's BCP utilities, for further processing. These are the comma-delimited fields in the log record:

- Date in this format: `'MM/DD/YYYY'`
- Tenant name in single quotation marks
- Tenant DBID
- Campaign name in single quotation marks
- Campaign DBID
- Phone number in single quotation marks
- Call result (name of CallState from `GctiCallState` enumerable set, obtained from Configuration Server) in single quotation marks
- Time when dialing starts (`'HH:MM:SS.MS'`)
- Approximate time when ringing starts (`'HH:MM:SS.MS'`)
- Time when an unsuccessful call (such as "no answer") is released (`'HH:MM:SS.MS'`)
- Time when called party picks up the phone (`'HH:MM:SS.MS'`)
- Time when call progress detection is completed and when CPD Server initiates the call transfer to a queue (`'HH:MM:SS.MS'`)
- Time when call is placed in a queue (`'HH:MM:SS.MS'`)
- Time when ringing starts on an agent's DN (`'HH:MM:SS.MS'`)
- Time when call is established on the agent's DN (`'HH:MM:SS.MS'`)
- Time when call is diverted to an auto-answering device (`'HH:MM:SS.MS'`) (for example, calls that are not connected to an agent in two seconds might be redirected to a pre-recorded message)
- Name of the file with CPD recording in single quotation marks (only for dialing with CPD Server)
- Name of the voice file that contains the recorded conversation in single quotation marks
- Approximate time difference between OCS and CPD Server computers (in milliseconds)

This field cannot be empty. The field value is zero (0) if the data is not available.

- Time when call is released on the agent's DN (`'HH:MM:SS.MS'`)

Note:	The timestamps (above) are precise to the millisecond and enclosed in single quotation marks. For example, `'15:20:50.245'`

To support Audit Logging, the CPD Server conveys its timestamp values to OCS. For example, CPD Server passes to OCS the values that indicate when CPD Server has completed call progress detection and when it has initiated the call transfer to a queue.

Fields cannot be omitted. Two commas with no value between them indicate that there is no data for that field. That type of placeholder follows, for example, the time when an unsuccessful call is released. The reason is that the field represented by this particular placeholder is reserved for the time when a called party picks up the phone, which does not occur once the call is released. There is no field value in this case, thus nothing between the commas.

Timed Events in PBX and CPD Server

Which telephony events are timed and when the timestamps are issued depend on the type of hardware (PBX or Dialogic Board) and software (OCS, CPD Server) that you use for placing and monitoring calls. The next three tables provide details on timed telephony events for various hardware and software. The following table shows telephony events that trigger the timing mechanisms in a PBX board/OCS.

Timed Events for PBX

Timestamp For:	Application
Dialing started	OCS
Ringing started	OCS
Unsuccessful call released	OCS
Call answered	N/A
CPD completed / Transfer initiated	N/A
Call placed in Queue	OCS
Agent DN ringing	OCS
Connected to an agent	OCS
Diverted for auto-answer	OCS
Call released on agent's DN	OCS

The following table shows telephony events that trigger the timing mechanisms in a CPD Server in transfer mode.

Timed Events for CPD Server (Transfer Mode)

Timestamp For:	Application
Dialing started	CPD Server
Ringing started	CPD Server
Unsuccessful call released	CPD Server
Call answered	CPD Server
CPD completed / Transfer initiated	CPD Server
Call placed in Queue	CPD Server or OCS
Agent DN ringing	CPD Server or OCS
Connected to an agent	CPD Server or OCS
Diverted for auto-answer	OCS
Call released on agent's DN	OCS

The following table shows telephony events that trigger the timing mechanisms in a CPD Server in ASM mode.

Timed Events for CPD Server (ASM Mode)

Timestamp For:	Application
Dialing started	CPD Server
Ringing started	CPD Server
Unsuccessful call released	CPD Server
Call answered	CPD Server
CPD completed / Transfer initiated	CPD Server
Call placed in Queue	N/A
Agent DN ringing	N/A
Connected to an agent	CPD Server
Diverted for auto-answer	CPD Server
Call released on agent's DN	OCS

Timing Data Availability in Preview Mode

In Preview dialing mode, OCS attempts to obtain the data required for audit logging from the telephony events `(EventDialing, EventRinging)` on the agent's DNs. These events should have outbound-specific pairs in the user data. Thus, the desktop application has to apply `TMakeCall` to all the user data received in Preview Record.

gsw_request_log

A gsw_request_log table stores outbound dialing activity for reporting, auditing, and other purposes. During installation, the Wizard prompts the user to create a gsw_request_log Table Access object, which allows OCS to access the log table. If the Wizard is not used, than the Table Access object for gsw_request_log table can be created manually by creating a new Table Access object under the tenant that will contain campaign information. This Table Access object must be a table of type, Log table and have the name gsw_request_log. It must also contain a reference to a valid database access point. The actual name of the database table is user-defined.

Note:	In multi-tenant environments, each tenant can have a separate gsw_request_log table.

Outbound Contact Manager and Genesys Administrator checks the configuration of this log table every time it starts. If the gsw_request_log Table Access object is configured, but the actual table does not exist in the database, the GUI will create the table. Outbound Contact Server opens the gsw_request_log table for a particular tenant when the dialing session for first campaign/campaign group for this tenant is loaded. Outbound Contact tracks the following call request messages in the gsw_request_log table:

Request Type	Enumerated Value (stored in request_type field)
PreviewRecordRequest	1
ChainedRecordRequest	2
AddRecord	3
PreviewDialingModeStart	4
PreviewDialingModeOver	5
UpdateCallCompletionStats	9
RecordProcessed	10
RecordReject	11
RequestRecordCancel	12
RecordReschedule	13
ScheduledRecordReschedule	14
DoNotCall	15
ContinueProcessing	16
Logout	18
ReadyTime	19
CampaignStatus	20

Outbound Contact includes an expanded list of call request messages in a gsw_request_log table. Because the log tracks call request messages for agents, a gsw_request_log table can be used for auditing purposes. The log does not provide any statistical analysis. To view a gsw_request_log table, use the server's DMBS SQL features. For more details, see the "Communication Protocol" chapter in the *Outbound Contact Reference Manual*.

Pre-Dial Validation

This topic provides an overview of the pre-dial validation feature that is introduced in Outbound Contact 8.0.001. It describes how to provision and implement pre-dial validation of the dialing records over secure and non-secure HTTP connections.

Pre-Dial Validation Over Non-Secure Connection

Pre-dial validation is an optional first step in the processing of the record. With pre-dial validation in place, OCS connects to the specific Web or Application Server using the HTTP or HTTPS and delivers a specifically formed HTTP `POST` request for each record before dialing this record. The body of the HTTP `POST` request contains properties of the individual record, just as an outbound call contains record-specific attached data. Based on the information received in HTTP `POST` request, Web or Application Server makes a decision whether or not the record should be dialed, and replies to OCS with either an HTTP `200 OK` or a `409 Conflict` response. As a next step in processing, OCS dials the records that were validated successfully (`200 OK` response) and attempts to apply treatments (without dialing) to those records that were validated negatively (`409 Conflict` response). Only those records that have successfully passed the validation are dialed by OCS.

Provisioning

Pre-dial validation is controlled by a number of OCS options, which can be set at various configuration levels. Unless the options are set correctly, pre-dial validation will not take place. By default, OCS behaves consistently with the previous release where pre-dial validation is not possible.

Configure the pre-dial validation feature by using the following options:

- pre-dial-validation
- validation-uri
- validation-timeout-call-result
- http-response-timeout
- http-connection-pool-size

Recommendations for Configuration

The pool size and HTTP request handling timeout (**http-connection-pool-size** and **http-response-timeout** options, respectively) must be set in accordance with the anticipated load on Web or Application Server. OCS will reuse open connections (although request pipelining is not supported by OCS) and will never exceed the connection pool size. After a connection has been opened, OCS will not attempt to close it and will reuse it as long as it's available (no request which has not yet been responded to is currently submitted for this connection) and valid (Web or Application Server has not closed this connection). In case an HTTP request needs to be sent out and no available (spare) connection is found in the pool and no new connection can be opened, OCS will queue the request internally until either the spare connection appears or until the handling timeout for the given request expires. Whenever the request is queued internally, log message `93100` is produced (see also Specific Log Messages for log messages description). When the request completes due to the timeout, log message `93202` is generated (see also Specific Log Messages for log messages description). The presence of `93100` and/or `93202` messages indicates that the setting is too low for the timeout, a slow responsiveness on Web or Application Server side, and/or insufficient connection pool size.

OCS creates a separate connection pool for each `host:port` pair it needs to maintain a connection to. This means that HTTP and HTTPS types of the connections will have separate connection pools, even if the host name is the same for both (this is due to different port numbers, `80` and `443` [defaults]). Short and fully qualified domain names for the host are also qualified as separate hosts by OCS; for example `host1`and `host1.subdomain1.domain1.com` are treated as different hosts by OCS and will therefore be assigned separate connection pools.

Specific Log Messages

OCS logs the following standard messages when processing connections to Web or Application Server:

`2102`	(data send error)
`4500`	(connecting)
`4501`	(server contacted)
`4502`	(cannot connect)
`4503`	(connected)
`4504`	(connection lost)
`4541`	(message received)

In addition to the standard messages, OCS also logs the result of the pre-dial validation for each dial attempt (all pre-dial validation results messages are of Trace Level):

`93200`	Pre-dial check completed with positive result for phone <phone number>
`93201`	Pre-dial check completed with negative result for phone <phone number>
`93202`	Pre-dial check aborted (timeout elapsed) for phone <phone number>

Whenever the connection pool limit is reached the following Standard Level message is logged:

`93100`	Maximum connections limit <number of connection> reached for server <Chost:port>

Pre-Dial Validation Protocol Description

This section provides a description of the pre-dial validation HTTP requests and responses.

Requests for Pre-Dial Validation

OCS delivers POST request to Web or Application Server that contains BODY in the application/json format (JavaScript Object Notation). This BODY holds all key-value pairs of the record subject to validation in the same fashion that the outbound call produced by OCS contains key-value pairs in its attached data. Similarly to the outbound call, OCS packs in the JSON BODY some mandatory key-value pairs (those whose keys are prefixed with GSW_) and all user-defined fields configured for delivery using the **send_attribute** field-level option.

Example: Validation Request with BODY

The following example is a request with the BODY (mandatory fields begin with GSW_ and user-defined fields begin with USR_):

```
POST /validation/validate.php HTTP/1.1
Accept: */*
User-Agent: OCS/8.0.001
Host: host1.domain1:80
ETag: 390
Content-type: application/json
Content-length:720

        {
                "GSW_TZ_OFFSET":0,
                "GSW_PHONE":"01282663420",
                "GSW_CALLING_LIST":"PFR_CL_01",
                "GSW_CAMPAIGN_NAME":"Campaign One",
                ...
                "GSW_RECORD_HANDLE":390,
                ...
                "GSW_CALL_ATTEMPT_GUID":"00S0VQKQK0DHT2O518838SDAES000098",
                "USR_FIELD1": "John Doe",
                "USR_FIELD2": "501-12-4312",
                ...
                "USR_FIELDT": "2010-05-06 13:25:10.003",
                "USR_FIELDN": 1970
        }
```

Notice that the ETag header of the request is always present and holds the value of record handle of the record being populated.

Processing the Pre-Dial Validation Request

Web or Application Server needs to make a decision about whether or not the record that has been delivered to it in the POST request by OCS is allowed to be dialed. This decision is made based on the individual record properties passed in the JSON BODY of the POST request, as required by the business logic. Web or Application Server should reply with `200 OK` for a positive validation result and with `409 Conflict` for a negative validation result.

Both positive and negative validation results may contain a BODY that must also be in the application/json format. This BODY can contain mandatory and user-defined fields that are to be updated. For example, a negative validation response might include a call result value that will be assigned to record or a positive validation response might contain the timestamp of the validation attempt.

Positive Response to the Validation Request

The following example is a positive validation response. Notice that the ETag header must be present in the response and must contain the value of the record handle of the record being validated.

```
HTTP/1.1 200 OK
Date: Fri, 30 Apr 2010 19:40:58 GMT
Server: Apache/2.2.14 (Win32) mod_ssl/2.2.14 OpenSSL/0.9.8k PHP/5.2.8
X-Powered-By: PHP/5.2.8
ETag: 390
Content-Length: 112
Content-Type: application/json
        {
                "USR_FIELDT": "2010-05-06 15:00:00.000"
        }
```

As a result of this response received by OCS, the record with a record handle 390 will be dialed. The user-defined field of the record with the **send_attribute** option set to USR_FIELDT will be updated in the Calling List table with the new string value "2010-05-06 15:00:00.000". Notice that the BODY part of the positive response is optional and should be provided only if some fields of the record require an update.

Negative Response to the Validation Request

The following is an example of a negative validation response. Notice that the ETag header must be present in the response and must contain the value of the record handle of the record being validated.

```
HTTP/1.1 409 Conflict
Date: Fri, 30 Apr 2010 19:40:58 GMT
Server: Apache/2.2.14 (Win32) mod_ssl/2.2.14 OpenSSL/0.9.8k PHP/5.2.8
X-Powered-By: PHP/5.2.8
ETag: 390
Content-Length: 62
Content-Type: application/json
        {
                "GSW_CALL_RESULT":53,
                "USR_FIELDT": "2010-05-06 15:00:00.000"
        }
```

As a result of this response received by OCS, the record with a record handle 390 will be marked with call result 53 (Wrong Number) and will not be dialed; the user-defined field of the record with the **send_attribute** option set to USR_FIELDT will be updated in the Calling List table with the new string value "2010-05-06 15:00:00.000". After assigning call result 53 to the record, OCS attempts to apply the treatment to specified call result (if such treatment is configured). Notice, that the BODY part of the response is also optional. If no call result is explicitly specified, call result 52 (Cancel Record) is applied.

Timeout While Processing the Validation Request

It is possible that the Web or Application Server will not be able to handle a validation request during the time period that is specified by using the **http-response-timeout** option. In this situation, the validation request will be aborted, and OCS will treat the timeout situation as a negative validation outcome (no dialing will take place). OCS will apply the call result that is specified in the **validation-timeout-call-result** option (default, 3 [General Error]) and will attempt to apply the treatment to that call result.

Pre-Dial Validation Over Secure Connection

This section describes pre-dial validation over a secure connection, including information about provisioning and secure connection-specific log messages.

Preface

For pre-dial validation, 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*.

On Windows platforms, support for SSL/TLS is integrated into the operating system.

On UNIX, Genesys Security Pack must be installed on the OCS host. See the Installing Genesys Security Pack in the *Genesys Security Deployment Guide*.

There are two different ways to configure a secure connection:

- Simple TLS connection when only a server certificate is validated.
- Mutual TLS connection when both server and client certificates are validated.

Provisioning for Simple TLS Connection

1. The protocol (scheme) part of the URI that is specified in the **validation-uri** option must have a value of `https://`. In accordance with HTTPS definitions, OCS will connect to port 443 (instead of 80) if a port number is not explicitly specified in the URI.
2. The user must create a Host configuration object with the same **Name** property as the host name that is specified in the **validation-uri** option.
3. The issuer certificate chain (the root CA and intermediate CA if there was any) that was used to sign the server-side certificate must be accessible for OCS:
   - (OCS on UNIX) The path to the issuer certificate chain file (see Supplying Multiple Certificate PEM Files in the *Genesys Security Deployment Guide* for details) must be specified in the **Trusted CA** field of the Network Security section of a Host configuration object mentioned above. An issuer certificate chain file must be Privacy Enhanced Mail (PEM)-encoded, accessible, and have correct access rights.
   - (OCS on Windows) The issuer certificate chain must be installed into the **Trusted Root Certificate Authorities** directory under the **Local Machine** storage on the OCS host.
4. The **Common Name** property of the server-side certificate must match the corresponding host name.

Provisioning for Mutual TLS Connection

1. The protocol (scheme) part of the URI that is specified in the **validation-uri** option must have a value of `https://`. In accordance with HTTPS definitions, OCS will connect to port 443 (instead of 80) if a port number is not explicitly specified in the URI.

2. The user must create a Host configuration object with the same **Name** property as the host name that is specified in the **validation-uri** option. The new Host object must have the following configuration option set in the Annex section: **security/tls-mutual**=1.
3. The issuer certificate chains (the root CA and intermediate CA if there was any) that were used to sign both OCS instances and the pre-dial validation service's certificates must be accessible for OCS:
    - (OCS on UNIX) The path to the issuer certificate chain file (see Supplying Multiple Certificate PEM Files in the *Genesys Security Deployment Guide* for details) must be specified in the **Trusted CA** field of the Network Security section of a Host configuration object mentioned above. An issuer certificate chain file must be Privacy Enhanced Mail (PEM)-encoded, accessible, and have correct access rights.
    - (OCS on Windows) The issuer certificate chain must be installed into the **Trusted Root Certificate Authorities** directory under the **Local Machine** storage on the OCS host.
    - An issuer certificate chain file must contain issuer certificate chains for both server and client certificates.
4. The **Common Name** property of the server-side certificate must match the corresponding host name.
5. The client certificate with its private key must be accessible for OCS:
    - (OCS on UNIX) The path to the OCS certificate file must be specified in the **Certificate** field. The path to the OCS certificate private key must be specified in the **Certificate Key** field. The path to the pre-dial validation service CA certificate file, if this CA is not public, must be specified in the **Trusted CA** field in the **Network Security** section of the Host configuration object mentioned above.
    - (OCS on Windows) The OCS certificate with its private key must be installed into the Personal directory under the **Local Machine** storage on the OCS host. The **Certificate** field must be populated with the fingerprint of the OCS certificate.

For more information about configuring `Host` and `Application` objects, see Securing Connections Using TLS in the *Genesys Security Deployment Guide*.

Examples of Mutual TLS Connection Configurations

EXAMPLE 1. OCS configured on the Linux host **ocs-host** connects to the pre-dial validation service configured on the **pdvs-host** host and port **8080** through the mutual TLS using the public CAs for signing both certificates.

1. Set the **default/validation-uri**=`https://pdvs-host:8080` configuration option in the OCS application.
2. Create a Host configuration object with the **pdvs-host** name and the **security/tls-mutual**=1 option configured in Annex.
3. Because both OCS and the pre-dial validation service use public CAs, there is nothing to configure in this step.
4. Ensure that the pre-dial validation services certificate is issued for the **pdvs-host** host.
5. Populate the **Certificate** and **Certificate Key** fields in the OCS application with the paths to corresponding PEM files. For example: Certificate=`/genesys/ocs-certs/cert.pem`, Certificate Key=`/genesys/ocs-certs/cert-priv-key.pem`

EXAMPLE 2. OCS configured on the Windows host **ocs-host** connects to the pre-dial validation service configured on the **pdvs-host** host and port **8080** through the mutual TLS using the public CAs for signing both certificates.

1. Set the **default/validation-uri**=`https://pdvs-host:8080` configuration option in the OCS application.
2. Create a Host configuration object with the **pdvs-host** name and the **security/tls-mutual**=1 option configured in Annex.
3. Because both OCS and the pre-dial validation service use public CAs, there is nothing to configure in this step.
4. Ensure that the pre-dial validation service certificate is issued for the **pdvs-host** host.
5. Install the OCS certificate in the Personal directory under the **Local Machine** storage on the **ocs-host** host. Populate the **Certificate** field in the OCS application with the OCS certificate fingerprint—for example, `5f124fcc249d4b17947aa75083fa8606d5c4da60`.

EXAMPLE 3. OCS configured on the Linux host **ocs-host** connects to the pre-dial validation service configured on the **pdvs-host** host and port **8080** through the mutual TLS using custom CAs for signing both certificates.

1. Set the **default/validation-uri**=`https://pdvs-host:8080` configuration option in the OCS application.
2. Create a Host configuration object with the **pdvs-host** name and the **security/tls-mutual**=1 option configured in Annex.
3. Populate the **Trusted CA** field in the OCS application with the path to a PEM file containing the issuer certificate chains for both OCS and the pre-dial validation service certificates—for example, `/genesys/ocs-certs/full-ca-chain.pem` .
4. Ensure that the pre-dial validation service certificate is issued for the **pdvs-host** host.
5. Populate the **Certificate** and **Certificate Key** fields in the OCS application with the paths to corresponding PEM files—for example, Certificate=`/genesys/ocs-certs/cert.pem`, Certificate Key=`/genesys/ocs-certs/cert-priv-key.pem`

EXAMPLE 4. OCS configured on the Windows host **ocs-host** connects to the pre-dial validation service configured on the **pdvs-host** host and port **8080** through the mutual TLS using custom CAs for signing both certificates.

1. Set the **default/validation-uri**=`https://pdvs-host:8080` configuration option in the OCS application.
2. Create a Host configuration object with the **pdvs-host** name and the **security/tls-mutual**=1 option configured in Annex.
3. Install the issuer certificate chains for both OCS and the pre-dial validation service certificates in the **Trusted Root Certificate Authorities** directory under the **Local Machine** storage on the **ocs-host** host.
4. Ensure that the pre-dial validation services certificate is issued for the **pdvs-host** host.
5. Install the OCS certificate in the Personal directory under the **Local Machine** storage on the **ocs-host** host. Populate the **Certificate** field in the OCS application with the OCS certificate fingerprint—for example, `5f124fcc249d4b17947aa75083fa8606d5c4da60` .

Secure Connection-Specific Log Messages

When a secure connection to the Web or Application Server is established, OCS prints a specific trace-level message into the log output. As shown in the following example, this message contains the properties of used certificates:

```
8103 Secure connection is established. type 'client', info '1600-135.225.58.18:443',
issuer 'C=US, S=California, L=Daly City, O=Genesys, OU=Outbound, CN=Outbound
Certificate Authority.
```

If a secure connection cannot be established, the following standard-level messages are logged:

8100	Certificate is expired
8101	Certificate is not valid
8102	Secure connection error

Troubleshooting HTTPS Connectivity

HTTPS connection with server-side certificate verification

OCS successfully opens an HTTPS connection (server-side certificate verification only) if the conditions described below are satisfied.

To confirm the HTTPS connection, in the OCS application, set the **x-conn-debug-all** option in the **log** section to 1 and look for the following string (with the respective host and port specified) in OCS logs:

```
tls: ("host"="<host name>","port"=<port>,"protocol"="tcp","transport"=("tls"="1"))
```

If there is no such string, then some of the conditions are not satisfied.

To troubleshoot the HTTPS connection, ensure the following:

1. The target resource URI must contain the **https://** prefix.
2. There must be a Host object specified in the Configuration Settings that corresponds to the host specified in the target resource URI:
   - The name of the Host object must be equal to the host name of the target resource URI.
   - The Annex tab (Options tab in GAX) of this Host object must contain the **client-auth** option set to `0` in the **security** section.
   - (Linux only) The **Network Security/Trusted CA** configuration field of this Host must contain the path to the CA PEM certificate of the target resource on the OCS host, if the root CA of the target resource is not a public CA (DigiCert, and so on).
   - The account under which OCS is running must have sufficient access rights to access this Host object.

TLS connection to HTTPS resource

OCS successfully opens a TLS connection to an HTTPS resource if the conditions described below are satisfied. To confirm the secure connection, look for the following string in the OCS logs:

```
Secure connection is established
```

If there is no such string, then some of the conditions are not satisfied.

To troubleshoot the secure connection, run the following command on the OCS host and validate the output (specify **-CAfile** if the root CA of the target resource is not a public CA, and use the same value as specified in the **Network Security/Trusted CA** option of the target resource Host object):

```
openssl s_client -connect <host>:<port> [-CAfile <target resource root CA certificate file
path>]
```

Ensure the following:

- The specified host is accessible and the specified port is opened.
- There is a network connection between the OCS host and the target host.
- OCS should be able to validate the entire certificate chain related to the target resource certificate, which is obtained from the target resource during a TLS handshake.
- The certificate chain obtained from the target resource during the TLS handshake must not contain self-signed certificates, including the certificate of the target resource itself.

Note: When reporting an issue related to the OCS HTTPS connectivity to an external HTTPS resource, provide the entire output of the **openssl s_client** command that is run on the OCS host to Genesys Customer Care.

Example of successful "openssl s_client" output

This is an example of the successful **openssl s_client** output for a resource that provides a certificate signed with a public CA:

```
$ openssl s_client -connect predialvalidation.domain.com:443
CONNECTED(00000184)
---
Certificate chain
 0 s:CN = predialvalidation.domain.com
   i:C = US, O = Let's Encrypt, CN = Let's Encrypt Authority X3
 1 s:C = US, O = Let's Encrypt, CN = Let's Encrypt Authority X3
   i:O = Digital Signature Trust Co., CN = DST Root CA X3
---
Server certificate
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
subject=CN = predialvalidation.domain.com
issuer=C = US, O = Let's Encrypt, CN = Let's Encrypt Authority X3
---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 3158 bytes and written 422 bytes
Verification: OK
...
```

Example of unsuccessful "openssl s_client" output

This is an example of the unsuccessful **openssl s_client** output for a resource that provides a certificate signed with a non-public CA while the root CA certificate is not provided in both the server certificate chain and in the **-CAfile openssl** option:

```
$ openssl s_client -connect predialvalidation.domain.com:443
CONNECTED(0000016C)
---
Certificate chain
```

```
 0 s:CN = predialvalidation.domain.com
   i:CN = Example Test Root CA
---
Server certificate
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
subject=CN = predialvalidation.domain.com
issuer=CN = Example Test Root CA
---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 2425 bytes and written 405 bytes
Verification error: unable to verify the first certificate
...
```

TLS SNI Configuration

If a TLS connection is not established and the following pattern is found in the logs, it indicates that the remote server has a certificate with multiple names and a TLS SNI configuration is required from OCS:

```
@15:13:41.5756 error:14094458:SSL routines:ssl3_read_bytes:tlsv1 unrecognized name
@15:13:41.5756 Additional data: SSL alert number 112
```

To configure TLS SNI, add the **security/tls-target-name=hostname** configuration option to the Annex section of the host object related to the target pre-dial validation host, where **hostname** is the name of a specific target host.

SCXML-Based Treatments

Outbound Contact Server (OCS) enables you to create SCXML (State Chart Extensible Markup Language)-based treatments, instead of configuring standard treatments (see Call Handling Treatments).

Notes:	• As a general rule, you cannot use both SCXML treatments and standard treatments for the same calling list. • If you are using Outbound Contact release 8.0.1 or higher, the following exception exists: • If you use the `set_exec_flag` custom action with the `PostProcessing` flag and the value set to `Switch,` then when this custom action executes, OCS switches from SCXML treatments to standard treatments. However, you cannot switch in the other direction, from standard treatments to SCXML treatments.

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

- Overview
- Architecture
- Design and Configuration Task Summary
- Customer Contact According to Daytime Intervals
- Record-Processing Order
- Setting Options for Individual Records or Chain of Records
- Other Usages of SCXML Treatments
- Configuration Tasks

Overview

Through the use of SCXML treatment scripts, you can configure custom treatments to handle multiple contact methods and scenarios, providing greater flexibility for reaching a customer. Some of the scenarios that custom treatments can address include:

- Selecting the preferred way to reach a customer, according to the time of day or the day of week (see Customer Contact According to Daytime Intervals).
- Specifying the order in which to contact a customer, according to the contact type (see Record-Processing Order).
- Specifying the first contact type to use when contacting a customer (see First contact type).
- Specifying the maximum number of attempts to dial records (see Number of dial attempts).
- Specifying a treatment action according to the call result (see Call result).

What is SCXML?

SCXML is an event-based state machine language that accommodates various call control/customer service states and the transitions between them. While relatively new as a notation/language, SCXML is well-proven for building state-based models and facilitates the process of orchestrating customer-service solutions. The core SCXML provides state chart functionality, while Outbound Contact—specific instructions are specified in the executable content of SCXML in form of SCXML extensions (action elements) and/or ECMA script extensions (properties of special ECMA script objects).

Schema Updates for Release 8.1

Starting in release 8.1, Genesys uses an updated SCXML standard that contains a number of changes from the prior standard. Note the following changes:

- The syntax for Event Data has changed from `_eventdata` to `_event.data`.
- To assign value to a variable, X, defined in the data section, use the syntax `_data.X` in location expressions such as the following:
  - Previous Syntax: `<assign location="X" expr = "1" />`
  - New Syntax: `<assign location="_data.X" expr = "1" />`
- Mnemonic operands such as the following must be expressed differently, as shown in the following examples:

Operation	Previous Syntax	New Syntax
=	eq	==
<	lt	<

>	gt	>
<=	le	<=
>=	ge	>=
&&	and	&&
¦¦	or	¦¦
!=	ne	!=

- When writing a file URL, use "`:`" instead of "`|`":
  - Previous Syntax: `file:///C|/inetpub/wwwroot/sample02.scxml`
  - New Syntax: `file:///C:/inetpub/wwwroot/sample02.scxml`

For more information about SCXML or ECMA scripts, go to the following websites:

- State Chart XML (SCXML), State Machine Notation for Control Abstraction (http://`www.w3.org/TR/scxml/`)
- ECMA and ECMAScript Language (http://`www.ecma-international.org`)

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.

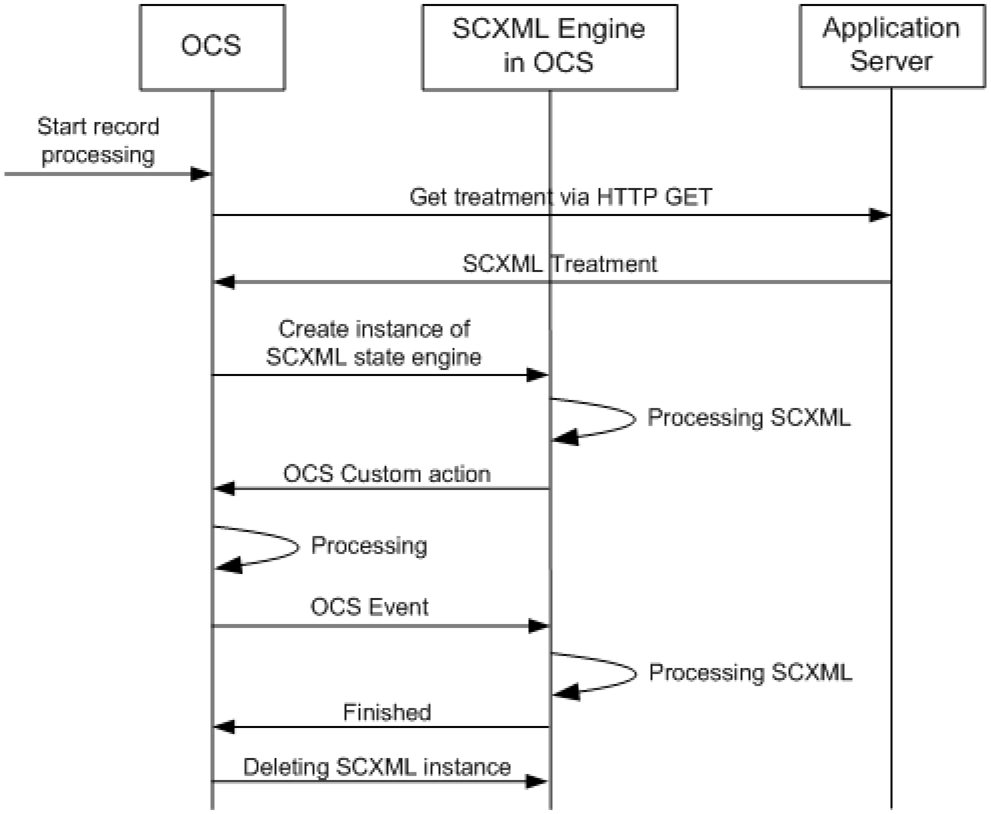
Note:	You specify the SCXML document that describes the treatment used by OCS in the `treatment-uri` option in the Campaign Group or Calling List object. For more information on configuring treatments, see Design and Configuration Task Summary.

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 successful`Error.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 does not apply treatments to `Unknown` and `OK` call results. The only exception is the Power mode (SMSs and E-Mails) for which OCS applies a treatment to the `OK` 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`. • `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. • `gsw_list_name`: (string) The name of the Calling List to which the record belongs. • 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.

Design and Configuration Task Summary

Designing and Configuring SCXML Treatments

Task	Related Procedures and Actions
1. Configure the Statistical Days and Statistical Tables.	See the procedure Configuring Statistical Days and Statistical Tables for Time Ranges.
2. Design the treatment scripts(s).	Create the `.xml` or `.scxml` files according to your treatment requirements. Some requirements may include the following: • Customer Contact According to Daytime Intervals • Record-Processing Order • Other Usages of SCXML Treatments
3. Configure the treatment options.	See Configuring SCXML Treatment Options.
4. Deploy the treatment script(s).	See the procedure Deploying a Treatment Script to an Application Server (IIS).

Customer Contact According to Daytime Intervals

You can design a treatment script that takes into account the time of day when determining how to contact a customer. Using predefined daytime intervals and configured Statistical Days and Statistical Tables with a SCXML treatment, OCS can also adjust how a customer is contacted if a daytime switch occurs. For example, the treatment may request that OCS dial a customer's business phone during business hours, but dial the customer's mobile phone after business hours.

The predefined daytime intervals include:

- `BH`: Business hours on a weekday, from its start time to its end time.
- `ABH`: After business hours on a weekday, from the end time of a weekday until midnight.
- `BBH`: Before business hours on a weekday, from midnight to the start time of the weekday.
- `WEND`: Weekend. Days that are identified by those days not configured as weekdays in the Weekday Statistical Days Table.
- `HDAY`: Holiday. Days configured as holidays.

When a dialing session/campaign group loads, OCS reads the time intervals, Statistical Days, and Statistical Tables configuration to determine the current time interval. OCS looks for the Statistical Tables in the configuration database in the following order:

- Under the Campaign Group Tenant.
- Under the Environment Tenant, if it is not found under the Campaign Group Tenant.

Note:	If these Statistical Tables are not found in either location or a Statistical Day is not configured, OCS considers the time range as a holiday (`HDAY`).

At startup, OCS determines the time and identifies the daytime interval according to the configured Statistical Days and Statistical Tables (see the procedure Configuring Statistical Days and Statistical Tables for Time Ranges). OCS dynamically processes configuration time interval changes (for example, from `BH` to `ABH`), while a Campaign Group is active. When the daytime interval changes, according to the contact's time zone, OCS informs the various instances of the state machine by sending the `ocs.daytime_change` event (see the Events from OCS table). This triggers the state machine to proceed to the new daytime interval.

In addition, when the SCXML engine downloads a script, the state machine requests the current daytime interval through the `get_daytime` action (see the Events from OCS table).

Script Design

When designing your SCXML treatment to handle daytime intervals and switching, include the following:

- Each daytime interval and transition associated with changing from one interval to another.
- For each interval, design the logic for making a call, and for handling call states, number of attempts, delays, and so on.
- Define variables (data model) to represent different parameters/values (for example, contact type, delay, number of attempts, and so on) that will be used through the script. The value for these variables can also be specified in the `treatment-uri` option.

You can also configure the script to delay switching of time intervals if the daytime switch occurs while a particular substate is still being processed (for example, the state machine is waiting for the `ocs.callresult` event). To accomplish this, configure the script to store the daytime interval in a variable and have the script check that variable when it is in a state for processing a daytime change.

To simplify the script design, you might create one high level script and multiple individual scripts to handle each daytime interval. When the daytime switching occurs, the current time interval script will be stopped and the new daytime interval script started. Then in the high level script, include a reference in the script to an external file located at another URL using the `src` attribute. For example: `<state id="ABH" src="someURL#ABH">`

When OCS requests a download of the script, the SCXML engine downloads the script and all referenced scripts into one combined state machine. As usual, an instance of this is created for each record/chain processed.

Note:	For an example of a script which handles daytime intervals, see the sample treatment script, `sample03.scxml`, that installs with OCS.

Record-Processing Order

You can design scripts handle chains of records according to the contact information type. You can further refine this ordering by designing the script to specify parameters such as the number of retries or delay associated with each contact info type or call result.

Script Design

When designing your SCXML treatment to specify record processing order, it is recommended that the script include the following:

- Data model variables for each contact type; for example, `Type1`, `Type2`, and so on. You may or may not specify the values for each of these types.
- A subsection that defines how to handle that contact type.

Note:	For an example script, see the SCXML script examples that are distributed with OCS. These example scripts are installed when you install OCS. They are located in the `scxml` folder.

Setting Options for Individual Records or Chain of Records

The following options can be set at the individual record or a chain of records basis:

- assured-connect--Enables and disables guaranteed connection for the outbound call
- CPNDigits--Defines the Caller ID for the outbound call
- CPNDisplayName--Defines the value of the DisplayName attribute used by SIP Server or CPD Server in HMP Transfer Mode
- pre-dial-validation--Controls whether or not pre-dial validation is in effect

The ability to set these options for individual records enables you to fine-tune record processing. Unlike the call_answer_type_recognition option which can be set at individual record level, based on the specific fields values of the record defined by using the Business Attributes configuration objects, these two options settings are defined by using flexible SCXML treatments. This allows much greater flexibility in the calculation of the option value and allows the definition of complicated business logic based on which these calculations are made. For example, any property of the record, or the time of day, the day of the week, or any other parameter that might be calculated inside the SCXML treatment executable content (JavaScript) can now affect the value of the option. For ease of use purposes, the call_answer_type_recognition option can now also be set by using an SCXML treatment.

Custom Actions

Custom actions for SCXML treatments, which enable you to set options, update user data, or pass on Attribute Extensions for individual records or a chain of records, are included in OCS 8.1. See the Custom Actions table for a description of the set_flex_attr and set_exec_flag custom actions.

Using Custom Actions

Both the set_flex_attr and the set_exec_flag custom actions can be used together, which enables you to set options, update user data, or pass on Attribute Extensions for records or a chain of records, and then continue with the regular processing of the chain (without the chain being processing by the SCXML Treatment after the initial option setting).

Custom actions can also be used separately. For example, you can set options, update user data, or pass on Attribute Extensions for records or a chain of records, and then the chain can be processed by SCXML Treatment logic.

Alternatively, at some point, the SCXML Treatment execution can be completed by using script logic and the control over the chain can be passed to traditional treatments. For a complete SCXML treatment that demonstrates the usage of both of these custom actions, see the sample06.scxml sample script, which is found in the OCS installation folder, in the scxml_samples sub folder.

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.

Configuration Tasks

Configuring Statistical Days and Statistical Tables for Time Ranges

- To configure Statistical Days and Statistical Tables in Genesys Administrator.

Start

1. In `Genesys Administrator > Provisioning tab > Routing`, in the Statistical Tables view, create an OCS folder (for example, `OCServer`).
2. Create a Statistical table with the Special Day type for weekdays.
3. Create Statistical Days for each weekday.
4. For each day, configure the start and end time for the workday.

Note:	A workdays list might not be limited to Monday through Friday if you configure Saturday and/or Sunday as workdays.

5. Create a Statistical table with the Special Day type for holidays.
6. Create a Statistical Day for each holiday.
7. For each holiday, configure its start and end time.

Note:	When a Dialing Session for a Campaign Group is loaded, OCS searches for the current date in the holiday table. If the timerange for this date is not found, OCS looks in the weekdays table. If the time range for this date is still not found, OCS considers it a weekend if one or both of the holiday and weekday tables are configured. If neither table is configured and a time range for this date is not found, OCS considers the time range as a holiday (HDAY). Warning: For more information about configuring Statistical tables and days, see *Framework Genesys Administrator Help.*

End

Configuring SCXML Treatment Options

Configure the following options at either the Campaign Group or Calling List level.

- `treatment-weekdays-table` -- Defines the name of the Statistical Table configuration object for the business weekdays, as created in the procedure Configuring Statistical Days and Statistical Tables for Time Ranges.
- `treatment-holidays-table` -- Defines the name of the Statistical Table configuration object for holidays, as created in the procedure Configuring Statistical Days and Statistical Tables for Time Ranges.
- `treatment-uri` -- Specifies the URI that defines the resource location of the SCXML script on the

Application Server. It can include call parameters/values that you want applied to the treatment script, allowing you to use the same treatment for different Calling Lists but specify unique parameters/values for each Calling List.

- treatment-preferred-contact-field -- Specifies the field name in the Calling List table which OCS uses for the given record to determine if this record in a chain should be used for the first chain dial attempt.

Note:	This value of this option identifies the contact type of the next record for which to search in a chain when the ocs.next_record event includes the gsw_preferred_flag equal to yes (see the Events from OCS table).

Deploying a Treatment Script to an Application Server (IIS)

- To deploy the treatments to an Application Server.

Start

1. Navigate to the C:\Inetpub\wwwroot folder.
2. Copy the treatment script files with the .xml or .scxml extensions to that folder.

End

Version Checking

If the script is updated while is it currently being used by a campaign that is running, OCS applies the new treatment to all chains processed after the update.

Error Handling

Error handling is addressed using the Error.Attribute and Error.Configuration events, as described in the Events from OCS table.

Call Handling and Treatments

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

- How Treatments Work
- Treatment Properties and Rules for Call Results
- Apply to Record Actions
- Treatment Sequences
- Apply to Call Outcome
- Treatments based on Disposition Codes
- Outbound Record Sharing Between Multiple Desktops
- Answering Machine Call Handling
- Using the Opt-Out Feature With CPD Server

How Treatments Work

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

A *Treatment object* tells Outbound Contact Server (OCS) how to respond to an unsuccessful call result (a call that does not reach the intended party) or to a disposition code.

A *treatment sequence* (sometimes called a linked sequence) is a general term describing a series of treatment actions, each assigned a unique sequence number and applied to the same unsuccessful call result or disposition code.

After Treatment objects are created and then applied to Calling List objects in Genesys Administrator, a treatment processing method is implemented by the `treatment_sched_threshold` option.

Outbound Contact Server stores the information that is required to restore the treatment application sequence properly in a dedicated `treatments` field for each chain. This information is stored as a string in the following format:
`<CfgTreatment DBID>@lgt;Cycle attempt number>@<Chain number>@<Treatment sequence number>`
where:

- `<CfgTreatment DBID>` is the DBID of the last treatment applied.
- `<Cycle attempt number>` is the number of cycle attempts within the current treatment starting from 1.
- `<chain number>` is the chain number value of the record which is to be dialled when the chain is retrieved by OCS.
- `<treatment sequence number>` is the number of cycle through the chain of records (for more details see Repeat the Treatment Cycle through the Chain of Records).

OCS retrieves and processes treatments history from the database only for records of types `CampaignRescheduled`, `PersonalRescheduled`, `CampaignCallback`, or `Personal Callback`. For other types of records (for example, `General`), the treatments history field is ignored. When the chain is selected, Outbound Contact Server retrieves information about the last applied treatment from the field `treatments` and use it to determine the next treatment action that should be applied.

Treatment Properties and Rules for Call Results

Properties are assigned to Treatment objects in the `Provisioning > Outbound Contact` tab in Genesys Administrator. The following table defines the primary treatment properties.

Primary Treatment Properties

Treatment Property	Definition
Call Result	Type of response received after dialing a calling list record's telephone number. An unsuccessful call result (for example, `Busy` or `No Answer`) is assigned to a call that does not reach the intended party.
Disposition Code	Used when the treatment action is based on disposition codes. See Treatments based on Disposition Codes for more information.
Apply to Record	Treatment action applied to the next dialing attempt. This value identifies the next action (for example, `Redial`) that OCS will take when an unsuccessful call result is received.
Apply to Call	Connection required. Determine if the `Answering Machine Detected` or `Fax Detected` call result should be transferred or dropped.
Number in Sequence	A unique number assigned to each treatment in a series of treatments linked to the same call result. This number sorts treatment sequences in the order they are to be applied to consecutive instances of the same call result. Creating multiple treatments for the same call result links the treatments.
Cycle Attempt	The number of times to apply a treatment action if the first dialing attempt has an unsuccessful call result.
Interval (minutes)	The number of minutes between cycle attempts if the first cycle attempt has an unsuccessful call result.
Increment (minutes)	The increment is added to the interval after the second cycle attempt of an unsuccessful call result. See Example of the Chain of Records Treatment Cycle Repetition.

Rules for Call Results

Four call result values have special treatment rules: Answer, Answering Machine Detected, Fax

Detected, and Silence. A complete list of call results with descriptions is provided in the *Outbound Contact Reference Manual*.

Note:	Silence is a possible call result when dialing via SIP Server.

Treatments are normally applied to a record for unsuccessful calls; however, under certain circumstances `Answer` (successful call) does require treatment.

Rules for Answer

Answer by a *live* person is a successful call result and usually does not require a treatment. Special circumstances, such as the following, do warrant a treatment for a call result of `Answer`:

- If a call originates at a Route Point and the call is answered, then the logical treatment is to direct the call to a queue.
- If an answered call is transferred (connected), then a treatment is required.

Rules for Answering Machine Detected or Fax Detected

These rules apply when a call result is `Answering Machine Detected` or `Fax Detected`. For a call that is answered, but the call result is `Answering Machine Detected` or `Fax Detected`, the system administrator may assign the "Apply to Call" treatment and specify the Destination DN to which the call is transferred. Here the Destination DN is an ACD queue for an Interactive Voice Response (IVR), such as a recorded message to be left on an answering machine or a fax sent to a fax machine. The administrator specifies the Destination DN in Genesys Administrator or through the Outbound Configuration Wizard.

Adding a Destination DN

Start

1. Add all required IVR DNs and their logins in Genesys Administrator under the `Switching` view on the `Provisioning` tab.
2. In the `Switches` view, add the queue into which the IVRs log in.
3. Create a `Place` object for each IVR DN and a shortcut to that DN.
4. Create a `Person` object for each IVR DN.

Note:	In this step, the "person" is actually an IVR that emulates a person; that is, the IVR does the work of an agent. In Steps 5 and 6 that follow, the "person" (IVR) is configured as an Agent because OCS requires a LoginID to facilitate access to a DN.

5. Configure the `Person` as an Agent.
6. Create a shortcut to the default Place and LoginID for that IVR Person.
7. Under the Agent Group object, add an Origination DN to the queue to which your IVR DNs log in.

8. Create an Answering Machine or Fax treatment using the following parameters:
   - Call Result: Answering Machine Detected or Fax Detected (depending on the desired treatment)
   - Apply to Record action: No Treatment
   - Apply to Call action: Transfer or Connect

Notes:	• In this context, `Transfer` and `Connect` are synonymous. Each transfers the call to the ACD queue for the IVR DNs. • For the Answering Machine Detected and Fax Detected call results, you can specify either `No Treatment` or `Update all records in chain` as the Apply to Record action. For information on configuring the `No Treatment` or `Update all records in chain` actions, see Apply to Record: No Treatment or the procedure Apply to Record, Updating All Records in a Chain respectively. • For the Answering Machine Detected Call Result, you can combine the Apply to Record actions `Next-In-Chain` or `Next-In-Chain-After` actions or Redial with the Apply to Call outcome, `Transfer` or `Connect`. For more information, see the procedure Combining the Apply to Call Outcome of Transfer or Connect Within the Apply to Record Action of Next-in-Chain or Next-in-Chain-After in One Treatment, below.

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

End

Combining the Apply to Call Outcome of Transfer or Connect Within the Apply to Record Action of Next-in-Chain or Next-in-Chain-After in One Treatment

Start On the `Configuration` tab of the `Treatment` object, do the following:

1. Verify that the `Call Result` type is set to `Answering Machine Detected.`
2. From the `Apply to Call` drop-down list, select `Transfer` or `Connect`.
3. Verify that the `Apply to Record` action is set to `No Treatment`.
4. In the `Destination DN` drop-down list, select the destination DN.
5. Return to the `Apply to Record` drop-down list and select `Next-In-Chain` or `Next-In-Chain-After`.
6. In the `Number in sequence` text box, specify a value of 1.
   With this configuration, when the Answering Machine Detected call result is detected, the call is transferred to the specified destination and the next record in the chain is dialed.

End

Apply to Record: No Treatment

OCS updates the record at the conclusion of the call; that is, when the IVR hangs up.

Apply to Record, Updating All Records in a Chain

With `Update all records in chain`, OCS updates all records in a chain as soon as OCS detects an answering machine or Fax machine and transfers the call to a queue for an IVR response.

Start

If the administrator chooses to `Update all records in chain` for `Apply to Record`, follow this procedure:

1. In `Genesys Administrator > Provisioning > Accounts Agent Group > Configuration tab >` Advanced section, add an Origination DN to the queue to which the IVR DNs log in.
2. Create an Answering Machine treatment using the following parameters:
   - Call Result: Answering Machine Detected
   - Apply to Record: `Update All Records in Chain`
   - Apply to Call: `Transfer` or `Connect` (Recall that a call can be transferred or connected to a queue for an IVR. The two terms are synonymous here.)
3. Specify the Destination DN: The ACD queue to which the IVR DNs log in. (See note below.)

Note:	Recall that the administrator specifies the Destination DN in the Genesys Administrator or through the Outbound Configuration Wizard.

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

End

Rules for Silence in a SIP Server Environment

SIP Server 8.1 and higher supports the Silence call result. On a call-by-call basis, you can configure whether to drop a call with a Silence call result, or have it connected. The details of how to configure OCS to handle the Silence call result from SIP Server depend on whether you are using SIP Server in a traditional telephony environment or a VoIP environment (see Configuring the Silence Call Result or VoIP-Only Functionality: Transferring a Call with the Silence Result to an Alternative DN, respectively, on Configuring the Silence Call Result).

Note:	If no treatment is configured for the Silence call result, and SIP Server is configured to connect such call, OCS treats the call in the same way as it would treat a call with Answer call result.

Configuring the Silence Call Result

You can configure OCS either to drop a call with the result Silence or connect it as if the result was Answer. To do so, configure a Treatment object for the appropriate Calling List for the call result Silence. The values for the Apply to Call action are either `drop` or `connect` (or `transfer`). If `connect` is specified, the Target DN must also be specified.

Notes:	• SIP Server can only connect such a call to the Voice Transfer Destination. • Configuration Server 8.1 is required to define Silence Call result in a treatment.

If a call with the result Silence is released by SIP Server, OCS attempts to apply a treatment to the chain of records in the same way it would for any other unsuccessful call result, such as Busy or No Answer. OCS then dispositions the Silence call result in the calling list in the same way as for any other call result.

Note:	For detailed instruction on configuring Treatment objects, see the Provisioning Your Environment > Outbound Contact Object Types > Treatments in the *Framework Genesys Administrator Help*.

VoIP-Only Functionality: Transferring a Call with the Silence Result to an Alternative DN

This additional functionality is available only in a VoIP environment when dialing on behalf of the Trunk Group DN. In all modes (Transfer, ASM, and GVP IP) OCS can transfer a call that is established on the Trunk Group DN, and that has the call result Silence, to an alternative destination.

To have OCS perform a transfer to an alternative destination, configure a Treatment with the action Transfer/Connect/Route for the calling list that originates the calling record OCS then performs the transfer in the same way as it does for the Fax and Answering Machine call results, by using single-step transfer (for details, see Rules for Answering Machine Detected or Fax Detected).

Apply to Record Actions

When creating a `Treatment` object in Genesys Administrator, you must assign an `Apply to Record` action to unsuccessful call results—for example, busy or no answer. For configuration instructions and `Apply to Record` definitions and configuration instructions, see Treatment Object.

Note:	Outbound Contact will update the `call_result` field of a record as `Answered` in the following scenario: • The `Apply to Record` action is `Assign To Group.` • An agent scheduled a personal `RecordReschedule` and then logs out. • At the scheduled call time, another agent receives and processes the record without specifiying a all result. **Warning:** Previously in this scenario, Outbound Contact would update the record as `Agent CallBack Error`.

Only the following `Apply to Record` actions can be configured for the "Agent CallBack Error" call result:

- Assign to Group
- Mark as Agent Error
- Redial
- Retry at specified date
- Retry in

Rules for Next-in-Chain Actions

The three Next-in-Chain actions have special rules. All records in a calling list are assigned a chain ID and chain number, even if there is only one record in a chain. These are unlinked records containing a unique chain ID and a chain number -- represented by a positive number. These unlinked records are not considered to be chained records.

The term "*chained records*" refers to two or more records linked to each other and assigned the same chain ID. Each chained record has a unique chain number within its chain.

In the example of chained records in the following Table, the Chain # Column shows the order of calling. The first number in the chain to be called starts with 1, the second number in the chain to be called starts with 2, and continues to increase.

The chain numbers can be in any order, but, by default, they are processed in ascending order. You can change the order by using a filter for a call list in Genesys Administrator, by specifying a list of the record fields followed by ASC or DESC and separated by commas.

Example of Chained Records

Phone Type	**Chain ID**	**Chain #**
James Smith (Home)	19	1
James Smith (Work)	19	3
James Smith (Mobile)	19	2
Janet Green (Work)	20	3
Janet Green (Home)	20	1
Janet Green (Pager)	20	2

The `Next-in-Chain` actions are as follows:

- `Next-in-Chain` (immediately dialed)
- `Next-in-Chain after` (dialed after a specified interval)
- `Next-in-Chain at specified date` (dialed at a date set in the treatment configuration)

The following guidelines apply:

1. To use any of the next-in-chain actions in a treatment, a calling list must have `chain_id` and `chain_n` populated correctly.
2. When creating a treatment sequence, use the next-in-chain action as the last member in the treatment sequence.
3. When defining treatments for the chained records, treatments of the type `Next-In-Chain` are used more often than the `Cycle` and `Redial` type of treatments.

Generally, if an unsuccessful call result is received, you do not want to redial that number if you have another number to dial for the same contact. However, the final decision on what type of treatment to apply depends upon the goal and strategy of the specific outbound campaign.

Repeat the Treatment Cycle Through the Chain of Records

When applying treatments, Outbound Contact has the capability of cycling through the chain of records more than once; when the chain has ended, OCS can start dialing the chain again from the first record. Both of the following must be true:

- The last record in the chain is dialed and receives a negative call result.
- The current treatment to be applied is either `Next-in-Chain`, `Next-in-Chain After`, or `Next-in-Chain at Specified Date`.

If both of these are true, OCS can cycle from the last record in the chain to the first record in the chain, and then start dialing the chain again. The behavior to cycle through the chain more than once is optional. To enable this chain cycling, you must properly set the timing properties for any of the three `Next-in-Chain` treatments, as follows:

- `Cycle Attempt` -- Determines the total number of times that a chain will jump from the last to the first record in the chain. The default value for this property is `0`, which means one cycle. The value 1 also means one cycle. Setting the value to be greater than 1 means that the cycling will occur for that specific number of times. The initial dialing of the chain is counted as the first cycle attempt.
- `Interval` -- Determines the time period to wait before beginning the cycle again. The interval measures the time between completing the last record in the chain and then jumping to begin again with the first record in the chain.
- `Increment` -- Determines the additional amount of minutes to add to the next interval, beyond the length of the previous interval.

Notes:	• The default behavior is one cycle. When a chain ends, it is updated in the `Calling List` and OCS stops processing that chain unless you have configured the system to repeat the cycle through the chain again. • The `Interval` and `Increment` properties are not applicable to the `Next-in-Chain at Specified Date` treatment. • For the `Next-in-Chain After` treatment, the `Interval` property specifies both the amount of time to wait before dialing the next record, and also, the time to wait before beginning the cycle again. The `Increment` property applies to the time interval before beginning the cycle again, but not to the time interval between records. • When subsequent call results are different, OCS applies the number of cycle attempts from a previous treatment to a current treatment only if both treatments are either `Next-in-Chain`, `Next-in-Chain After`, or `Next-in-Chain at Specified Date`. Otherwise, OCS sets this number to the first cycle attempt.

For more information about the timing properties for these treatments, see the Using Timing Properties table.

Example of the Chain of Records Treatment Cycle Repetition

The following provides an example of the cycle behavior for the chain of records for a `Next-in-Chain` treatment. In this example, the calling list has a `Next-in-Chain` treatment for a `No Answer` call result, with the following configuration:

- `Cycle Attempts: 4.` The total number of times that the cycle through the chain occurs will be 4 times.
- `Interval: 30.` The time to wait before beginning the cycle again is set for 30 minutes. So, the amount of time between ending the last record in the chain and jumping to the first record in the chain will be

30 minutes.

- Increment: 20. The time period to add to the interval for each subsequent cycling will be 20 minutes.

The configurations in the example above will result in the following behavior:

1. OCS retrieves a chain consisting of three records and dials the first record in the chain. There is no answer. The treatment is: if there is no answer, dial the next record in the chain.
2. OCS dials the next record in the chain. There is no answer. The treatment is: if there is no answer, dial the next record in the chain.
3. OCS dials the third record in the chain. There is no answer. The treatment is: if there is no answer, dial the next record in the chain.

The chain has ended. The number of Cycle Attempts in our example is set to greater than 1, so OCS will wait for the specified time of 30 minutes (Interval value in our example), and then proceed.

1. After waiting 30 minutes; OCS will now jump to the first record in the chain, and repeat treatment steps for records in the chain if the call result is not changed.
2. For the next two cycles, the Increments between intervals will now take effect and they will increase between each cycle. After the second cycle, OCS will wait 50 minutes (Interval of 30 minutes + Increment of 20 minutes in our example) before jumping to the first record in the chain. After the third cycle, OCS will wait 70 minutes (Interval of 30 minutes + Increment of 20 minutes + Increment of 20 minutes) before jumping to the first record in the chain.

Note:	The Next-in-Chain treatment must be the last treatment in the sequence. If the Next-in-Chain treatment is not the last treatment in the sequence, all treatments after the Next-in-Chain will be ignored.

Rules for Update All Records in Chain

The Update all records in chain action can be used even if there are no chained records in a calling list. It does not have the same restrictions as the next-in-chain actions for the following reasons:

1. This action does not require that timing properties be set.
2. All records contain a chain ID and number even when they are not chained.

Note:	The No Treatment action can be used for the call result Answering Machine Detected in a Connect/Transfer treatment only if AM calls are connected to agents that belong to the OCS Campaign Group. Otherwise, the treatment Update all records in chain must be configured for the Apply to Record action. The Update all records in chain action does not work for Answer call results in the Connect/Transfer to treatment.

Timing Properties Used with Apply to Record Actions

Understanding timing properties is essential when creating treatments and applying them to Calling List objects. The following table shows which timing properties are required for each treatment action.

Using Timing Properties

Apply to Record Action	Cycle Attempt	Interval	Increment(minutes)	DateTime
Assign To Group	Not Applicable	Not Applicable	Not Applicable	Not Applicable
Mark As Agent Error	Not Applicable	Not Applicable	Not Applicable	Not Applicable
Next-in-Chain	RequiredSet the maximum number of passes through the chain: values 0 and 1 = one pass; the next record in the chain is dialed immediately	RequiredSet the time interval until the next pass through the chain, after the last record in the chain has been dialed.	RequiredSet the time interval for the subsequent chain passes	Not Applicable
Next-in-Chain after	RequiredSet the maximum number of passes through the chain: values 0 and 1 = one pass	RequiredSet the time interval until the next record in chain is dialed, and until the next pass through the chain, after the last record in the chain has been dialed.	RequiredSet the time interval for the subsequent chain passes	Not Applicable
Next-in-Chain at specified date	RequiredSet the maximum number of passes through the chain: values 0 and 1 = one pass	Not Applicable	Not Applicable	RequiredSet the date/time to dial the next record in chain
No Treatment	Not Applicable	Not Applicable	Not Applicable	Not Applicable
Redial	RequiredSet the maximum number of retry attempts	RequiredSet the time interval until the first cycle attempt.	RequiredSet the time interval for subsequent cycle attempts.	Not Applicable
Reschedule	Not Applicable	Not Applicable	Not Applicable	Not Applicable
Retry at specified date	Not Applicable	Not Applicable	Not Applicable	RequiredSet the date/time for the next attempt
Retry in	Not Applicable	RequiredSet the time interval until the next (only one) attempt	Not Applicable	Not Applicable
Update all records in chain	Not Applicable	Not Applicable	Not Applicable	Not Applicable

When OCS calculates the time for the next dial according to the treatment configuration, it is possible that the new calculated time is out of the "daily from" -- "daily till" boundaries of the record. If this happens, OCS adds an appropriate amount of time to the calculated time so that it falls within the boundaries.

Execute SQL Statement Action Type

The `Execute SQL Statement` action allows executing a specific pre-configured SQL statement when a certain call result is received. There are no limitations for the type or complexity of SQL statements. OCS uses its connection to the Calling List database table, from which the current dialing record originates, to execute the SQL statement. OCS also allows additional flexibility by supporting macro expressions that can be used to form SQL statements, for example, the Calling List name and fields of the record that is being processed can be replaced with macro expressions. OCS expands those macro expressions to the actual values before requesting the DBMS to execute the SQL statement.

Unlike other treatments that affect a chain of records (for example, if it is rescheduled for a later time or the treatment causes the next record of the chain to be dialed), this treatment does not affect the record or chain in any way. Instead, when the treatment is completed, it passes control to the next treatment (if one is configured). This enables you to configure the sequence of treatments, similar to `Execute SQL Statement and Reschedule Chain to later time`, for the same dialing attempt.

Provisioning

Treatment is configured by selecting the `Execute SQL statement` action from the list of `Apply to Record` actions in the Treatment configuration object. As for any other type of treatment, `Call Result` and `Number in sequence` treatment properties must be defined. The actual SQL statement to be executed upon receiving a specified call result is configured in the `Annex` tab of the Treatment configuration object. The section name where the SQL statement option is configured should be entered as `default`, `OCServer`, or configured with the specific name for the `OCS Application` object. The option itself is named `sql` and should contain the SQL statement to be executed by OCS. Similar to any other Treatment, this Treatment must be referenced in the `Calling List` configuration object to take effect for the specific calling list.

Execution

When OCS completes the execution of this treatment, it immediately passes control to the following treatment, if it is available. The following treatment is a treatment that is configured for the same `Call Result` value and has a property `Number in sequence` equal to `(N + 1)` where `N` is the `Number in sequence` property of the `Execute SQL statement` treatment.

Macro Expressions

The SQL statement that OCS executes for this treatment may contain macro expressions. Macro expressions are pre-defined words that start with the '$' sign. OCS automatically expands these macro expressions to their values before the SQL statement is executed.

The following table summarizes macro expressions supported by OCS:

OCS Supported Macro Expressions

Macro	Expanded to	Type	Example
`$treat_name`	The name of the current Treatment configuration object.	string	Update Treatments History SQL clause
`$treat_dbid`	The DBID of the current Treatment configuration object.	integer	101
`$camp_group_dbid`	The DBID of the Campaign Group configuration object to which the record being processed logically belongs.	integer	101
`$camp_group_name`	The name of the Campaign Group configuration object.	string	Campaign 1@Agent Group 1
`$camp_name`	The name of the Campaign configuration object to which the record logically belongs.	string	Campaign 1
`$group_dbid`	The DBID of the Group configuration object.	integer	101
`$camp_dbid`	The DBID of the Campaign configuration object.	integer	101
`$group_name`	The name of the Group configuration object.	string	Agent Group 1
`$list_tbl_name`	The name of the calling list database object (table name) from which the record was fetched.	string	`calling_list_tbl`
`$list_name`	The name of the Calling List configuration object to which the record logically belongs.	string	Calling List 1
`$list_dbid`	The DBID of the Calling List configuration object.	integer	101
`$dial_filter_where`	`where` clause of the dialing filter that is defined for the Calling List.	string	`customer_code` not in (1,2,6)
`$dial_filter_order_by`	`order by` clause of the dialing filter that is defined for the Calling List.	string	chain_id ASC, chain_n ASC
`$<send_attribute>`	The value of the field of the current record for which `send_attribute`	string	Customer since 1980

	is defined.		

Note:	Macro expressions of type `string` should be enclosed in single quotes if they are used as string constants in the SQL statement.

Examples

The treatment can be used for various purposes, for example to insert values into separate historical tables or to update Calling Lists with treatment-specific information. Consider the following examples of the SQL queries configured for execution by this treatment:

Example 1

- `insert into treatments_history (treatment_dbid, treatment_name, list_dbid, chain_id, chain_n, exec_time) values ($treat_dbid, '$treat_name', $list_dbid, $GSW_CHAIN_ID, $GSW_CHAIN_N,'15:00:45')`

After all macro expressions are expanded by OCS, this SQL statement will insert treatment application fact data into the separate `treatments_history` table in the same database where the current Calling List table resides. Notice, that the `treatments_history` table with proper structure must exist in the database or the execution will yield an error.

Example 2

- `update $list_tbl_name set treatment_appl = '$treat_name' where chain_id = $GSW_CHAIN_ID and chain_n = $GSW_CHAIN_N and ($dial_filter_where)`

This query updates the current calling record in the Calling List table from which it originated. It sets a user-defined field `treatment_appl` to the name of the current treatment.

Note:	Both examples use `$GSW_CHAIN_N` macro. For this macro to be expanded to the value of the chain_n field of the record, the `send_attribute` option with the name `GSW_CHAIN_N` must be defined for the `chain_n field` configuration object.

A Word of Warning

Special care must be taken when SQL statements for the treatment are being defined. These SQL statements should not consume execution-time or put a heavy load on the DBMS. Remember that OCS uses the same DBMS for dialing records retrieval and update purposes. Therefore, database performance is a key factor in the performance of the whole dialing engine.

Additionally, extra care must be taken if the active Calling List is being updated by the `Execute SQL statement treatment` application (see Example 2). As a general rule, mandatory fields of the retrieved records must not be updated, as this might cause table-level locking and/or erroneous, repetitive dialing of the records that have just been dialed. For example, setting a record that could be identified by the clause, "`where chain_id = $GSW_CHAIN_ID and chain_n = $GSW_CHAIN_N`" to the Ready state will cause this record to be dialed again which is usually not the desired behavior.

Send HTTP Request Action Type

Starting with release 8.1.519.01, OCS supports the ability to send an HTTP request with the JSON body to a remote Web/Application Server as an action of the Apply to Record treatment based on a Call Result or Disposition Code. For example, when a person has been contacted (Answer) and requested to not be contacted ever again, the person's phone number is added to the mandatory suppression list through, for example, the CX Contact interface.

You configure parameters of the HTTP request in the Annex tab of the Treatment object as described below. The Send HTTP Request action is configured using the *Delegate for processing* Apply to Record action type.

Send HTTP Request Parameters

Section	Option	Level	Values	Default	Comments	Example
OCServer	**treatment-action-uri**	Treatment object	The URI for sending a request	Empty string	All macro expressions are expanded in the same manner as for the Execute SQL Statement treatment. Only the `http:` schema is supported.	
OCServer	**http-method**	Treatment object	POST | PUT | GET | DELETE	POST	No support for macro expressions.	
OCServer	**http-headers**	Treatment object	Exactly as specified in http-custom-headers	Empty string	All macro expressions are expanded in the same manner as for the Execute SQL Statement treatment. Note that the value of this option should be combined with the Application-level option	

Section	Option	Level	Values	Default	Comments	Example
					http-custom-headers, in case if it's present.	

OCServer	**http-body**	Treatment object	JSON string (and support for macro expressions)	Empty string	All macro expressions are expanded in the same manner as for the Execute SQL Statement treatment, especially record-level macro parameters, such as $GSW_PHONE and $GSW_CUSTOMER_ID.	{
						"data": {
						"listid": 5711,
						"appendOnly": false,
						"fields": {
						"client_id": "$GSW_CUSTOMER_ID"
						}
						}
						}
						
						{
						"data": {
						"listid": 5711,
						"appendOnly": true,
						"fields": {
						"device": "$GSW_PHONE"
						}
						}
						}

Treatment Sequences

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

Note:	Campaign sequences are not the same as treatment sequences. See Campaigns for more information about campaign sequences.

Rules for Treatment Sequences

Apply these general rules to treatment sequences:

- Treatment sequences are compatible with next-in-chain Apply to Record actions. Next-in-Chain treatments should be the last one in a sequence. When OCS dials the next record in a chain and receives the same call result for this record as it did for the previous record, the whole treatment sequence will be repeated.
- Always assign consecutive numbers to treatment sequences and always begin with 1. If you use non-consecutive numbers -- such as `1, 2,` and 4 the treatment sequence stops at the first non-consecutive number, which in this example is 4. The first two treatments would apply, but not the fourth.
- OCS applies each treatment in sequential order until a successful call result is received, a dialing attempt generates a different unsuccessful call result, or the number of dialing attempts equals the Maximum Attempts value assigned in the calling list object.
- If a treatment sequence for one call result (for example, Busy) is interrupted with a different call result (No Answer), the sequence is broken and the Number in Sequence value for Busy resets to 1. If the `Busy` call result is received again, the treatment sequence restarts at the beginning.
- Sequences of treatments based on disposition codes and call results do not overlap. (See sequence examples with disposition codes.) If a treatment sequence for a call result is interrupted with a disposition code, the sequence is broken and the New in Sequence value resets to 1.

Note:	Treatments using the Apply to Call action cannot be sequenced. Set the Number in Sequence value to 1, as OCS always identifies this action with a sequence value of 1 even if another value is specified.

Treatment Object List

The following table shows Treatment objects that will be used in Examples 1 and 2, which follow.

Treatment Objects List

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

Example 1: Treatment Sequence Exhausted

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

Example 1: Treatment Sequence Exhausted

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

In Example 1, the call did not reach its intended party and, on the fifth dial attempt, the second number in the Answering Machine Detected treatment sequence is executed.

Example 2: Record Reaches Maximum Attempts Value

When the number of times the telephone number is dialed matches the Maximum Attempts value from the calling list, the final outcome of the last attempt is applied and logged in the database. In this example the calling list object's Maximum Attempts value is 8. Using the same Treatment objects from Example 1, a record is dialed eight times, with the results shown in the following table.

Example 2: Record ReachesMaximum Attempts Value

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

5	Answering Machine Detected	Retry in (60 min.)	No
6	Busy	Redial	No
7	Answering Machine Detected	Retry in (60 min.)	No
8	Busy	Redial	No

In Example 2, the call did not reach its intended party after eight attempts. Because the Maximum Attempts value is 8, dialing stops and OCS applies the final outcome.

Example 3: The sequence with Disposition Codes is started and completed

In this example, the sequence of treatments based on a Disposition Code is started and completed:

1. The first RecordProcessed request containing Disposition Code = 92 arrives.
2. OCS applies a treatment with the name of “CXRD:92:..” and the number in sequence = 1.
3. The second RecordProcessed request containing Disposition Code = 92 arrives.
4. OCS applies a treatment with the name of “CXRD:92:..” and the number in sequence = 2.
5. The third RecordProcessed request containing Disposition Code = 92 arrives.
6. There is no treatment with the name of “CXRD:92:..” and the number in sequence = 3.
7. OCS finalizes the chain.

Example 4: The sequence with Disposition Codes is not started

In this example, the sequence of treatments based on a Disposition Code is not started:

1. The first RecordProcessed request containing Disposition Code = 92 arrives.
2. There is no treatment with the name of “CXRD:92:..” and the number in sequence = 1.
3. OCS applies a treatment with a matching call result and the number in sequence = 1 if it is present.

Example 5: The sequence with different Disposition Codes

In this example, the sequence of treatments based on a Disposition Code switches between different Disposition Codes:

1. The first RecordProcessed request containing Disposition Code = 92 arrives.
2. OCS applies a treatment with the name of “CXRD:92:..” and the number in sequence = 1.
3. The second RecordProcessed request containing Disposition Code = 92 arrives.
4. OCS applies a treatment with the name of “CXRD:92:..” and the number in sequence = 2.
5. The third RecordProcessed request containing Disposition Code = 93 arrives.
6. OCS applies a treatment with the name of “CXRD:93:..” and the number in sequence = 1.

Example 6: The sequence with Disposition Codes and Call Results

In this example, the sequence of treatments based on a Disposition Code switches to a treatment based on a Call Result.

1. The first call is dialed and released.
2. OCS applies a regular treatment based on a call result with the number in sequence = 1.
3. A new call is dialed and delivered to an agent DN.
4. The RecordProcessed request containing Disposition Code = 92 arrives.
5. OCS applies a treatment with the name of “CXRD:92:..” and the number in sequence = 1.
6. A new call is dialed and delivered to an agent DN.
7. The RecordProcessed request containing Disposition Code = 92 arrives.
8. OCS applies a regular treatment with the name of “CXRD:92:..” and the number in sequence = 2.
9. A new call is dialed and released.
10. OCS applies a regular treatment based on a call result with the number in sequence = 1.
11. A new call is dialed and delivered to an agent DN.
12. The RecordProcessed request containing Disposition Code = 92 arrives.
13. OCS applies a treatment with the name of “CXRD:92:..” and the number in sequence = 1.

Apply to Call Outcome

When creating a Treatment object in Genesys Administrator, you can assign an Apply to Call outcome for three call results: Answering Machine Detected, Fax Detected, and with SIP Server 8.1 or later, Silence. See Rules for Call Results for additional information. Also see Manually Configure and Install Outbound Applications for configuration instructions.

Note:	The Apply to Call treatments work when in Transfer mode.

The Apply to Call treatments are:

- `Drop`
- `Connect or Transfer`

These treatments are described in the following sections.

Drop Treatment

This is the default treatment if no treatment is specified. OCS drops the call and logs the call result in the database as Answering Machine Detected, Fax Detected, or Silence.

Connect or Transfer Treatment

This treatment can be user-assigned only to Answering Machine Detected, Fax Detected, or Silence call results. OCS transfers calls to the specified DN.

Note:	Connect or Transfer of a Silence call result works only with SIP Server. In a VoIP environment, the call is connected or transferred to an alternative destination. In a non-VoIP environment, the call is connected or transferred to a Voice Transfer Destination.

Automatic transfer is possible only in the following cases:

- When dialing via a T-Server that supports automatic transfer to the alternative destination. Refer to the Deployment Guide for your particular T-Server to verify that the `TMakePredictiveCall` request supports alternate destinations in the request Extensions.
- When CPD Server is used as a dialer.19 CPD Server performs the transfer after the call is established on a CPD port.
- In VoIP dialing modes when dialing from a Trunk Group DN with SIP Server.19OCS performs the transfer after the call is established on a Trunk Group DN.

If T-Server does not support automatic transfer of calls with an Answering Machine Detected call result to an alternative destination, and if you are using URS to route calls from a Voice Transfer Destination, you can use a custom strategy to route calls with Answering Machine Detected or Fax

Detected call results to alternative destinations. For example, calls with the call result Answering Machine Detected in a custom Universal Routing Server strategy can be identified by having the following characteristics:

`Attribute CallState 9` (Answering Machine)
`Attribute UserData 'AnswerClass' 'AM'`

These calls can then be routed via the strategy to the desired alternative destination instead of routed to agents.

Determining if Call has Reached Answering Machine

For the Answering Machine Detected and Fax Detected call results, OCS checks both the `'AnswerClass'` data and the call state data to determine if a dialed call has reached an answering machine. This process eliminates problems with some switches. For more information, see Rules for Answering Machine Detected or Fax Detected. This treatment does not apply to the Silence call result.

Routing to an Alternate Destination

In a VoIP environment, when dialing on behalf of the Trunk Group DN, OCS can transfer a call with the Silent call result to an alternative destination. For more information, see VoIP-Only Functionality: Transferring a Call with the Silence Result to an Alternative DN.

Rules for Apply to Call

There is one rule for Apply to Call:

The default value is `[Unknown Action code]`. Do not change this default unless you are creating a treatment for Answering Machine Detected, Fax Detected, or Silence call results and have created and configured a Destination DN for this purpose.

Note:	Treatments using the Apply to Call action can not be sequenced. Set the Number in Sequence value to 1, as OCS always identifies this action with a sequence value of 1 even if another value is specified.

Treatments based on Disposition Codes

Starting with release 8.1.519.01, OCS supports the ability to apply treatments based on disposition codes, in addition to treatments based on call results. Disposition codes can be specified by the agent desktop or by a routing strategy.

OCS relies on the following capabilities of the agent desktop:

1. To read the disposition codes stored as business attribute values in the configuration environment and allow an agent to apply a disposition code to any chain that has reached the desktop (for example, Answer or Answering Machine call results).
2. To provide the disposition code value (not the Display Name) in the record finalization desktop request in the key-value pair of type string with the constant key name GSW_DISPOSITION_CODE.
3. To provide the GSW_TREATMENT flag in the record finalization desktop request, along with GSW_DISPOSITION_CODE, if a treatment application based on a disposition code is expected of OCS.

Disposition-level treatments have the same treatment properties as call results-based treatments.

Sequences of treatments based on disposition codes and call results do not overlap. See sequence examples. If a treatment sequence for a call result is interrupted with a disposition code, the sequence is broken and the New in Sequence value resets to 1. Any “Next in Chain ...” treatment finalizes the current sequence and, depending on a cycle attempt, can start a new sequence with the number in sequence equal 1.

OCS stores the disposition codes treatments history in the Calling List treatments field, in the same way it stores call results-based treatments history.

OCS prints the “DC Treatment Sequence” strings in its log when processing treatments based on disposition codes.

Configuration

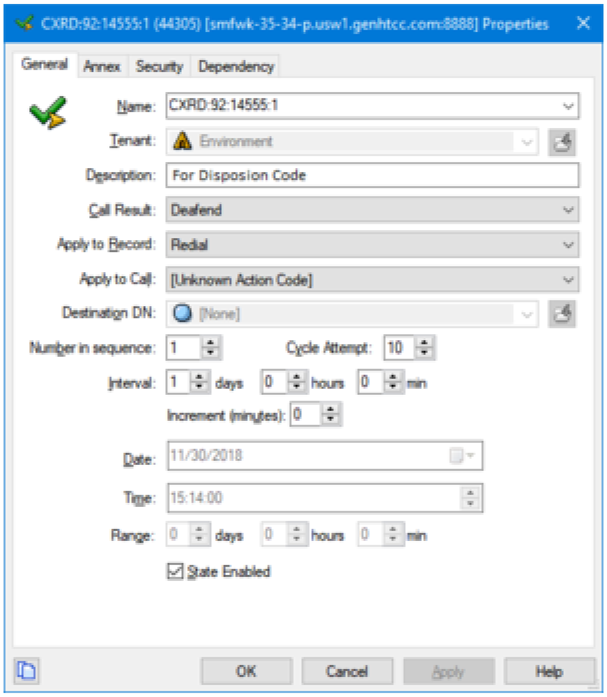
Treatments for disposition codes are configured at the Campaign Group level (as in the CX Contact) or at the individual Calling List level (traditional configuration). For information about configuring treatments based on disposition codes using the CX Contact, see Delivery and Retry Options in the CX Contact documentation.

Configuring Treatment Objects

To configure a Treatment object to use for a Disposition Code, specify the following properties:

- Name: `CXRD:<Disposition Code Value>:<Campaign Group DBID>:<Number in sequence>`
- Apply to Record: the desired record action
- Call Result: `Deafend` (OCS should not rely on a call result, it may or may not be Deafend in general case)

- Number in sequence: >= 1



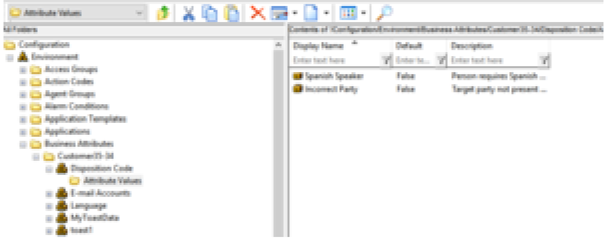
CXRD:92:14555:1 (44305) [smfwk-35-34-p.usw1.genhtcc.com:8888] Properties

Name:	CXRD:92:14555:1
Tenant:	Environment
Description:	For Disposion Code
Call Result:	Deafend
Apply to Record:	Redial
Apply to Call:	[Unknown Action Code]
Destination DN:	[None]
Number in sequence:	1
Cycle Attempt:	10
Interval:	1 days 0 hours 0 min
Increment (minutes):	0
Date:	11/30/2018
Time:	15:14:00
Range:	0 days 0 hours 0 min

☑ State Enabled

Configuring Disposition Codes

Disposition codes can be configured based on the customer needs and as supported by the agent desktop application. OCS does not deal with Disposition Codes configuration objects directly. The below illustrates one of the methods (default) of disposition codes configuration in the configuration environment.



Contents of \Configuration\Environment\Business Attributes\Customer35-34\Disposition Code\

Display Name	Default	Description
Spanish Speaker	False	Person requires Spanish ...
Incorrect Party	False	Target party not present ...

For information about assigning disposition codes in Workspace Desktop Edition, refer to Disposition Code in the Workspace documentation.

For more information about business attributes, refer to the Business Attributes page in the *Genesys Administrator Extension Help* guide.

Outbound Record Sharing Between Multiple Desktops

Outbound record sharing between multiple outbound-enabled desktops is possible when the outbound call is made for a certain record (that is, a call with the `GSW_RECORD_HANDLE` key and a valid record handle value attached to it) is transferred or conferenced between two or more agent places. For proper record sharing, OCS must monitor all of the DNs that participate in the call transfer or conference. In addition, all agents who intend to share the record must be logged in.

There are three possible scenarios for sharing records.

Scenario 1: Outbound Call Single Step Transfer or Mute (Blind) Transfer

Upon a successful transfer completion, all rights to the record and chained records access are revoked from the agent who initiated the transfer and are granted to the agent who received the transferred call.

Scenario 2: Outbound Call Two-step Transfer with Consultation Call

During the consultation phase of the transfer (or conference), when the consultation call is established on the agent's DN who should receive the transferred call (or be added to the conference), this agent has read-only permissions for the record. No desktop events will be accepted from this agent, except for a request for chained records (`ChainedRecordRequest`), which are also passed to the agent desktop in the read-only mode.

Scenario 3: Outbound Call Conference

All agents participating in the conference call have full access to the record (and chained records of the same chain) and the right to update the chain (that is, by sending desktop protocol events for specific record handling).

Note:	During the conference scenario, when more than one agent has full access to the record, record access is concurrent. This means, that if one agent sends the `RecordProcessed` event for the record and receives an `Acknowledgement` from OCS, the other agent(s) will not be able to update this record anymore.

Answering Machine Call Handling

You can handle calls that are detected with an Answering Machine (AM) call result on a per-record basis.

Per-Record Basis

OCS supports the ability to assign different types of AM-detection to each dialing request based on a custom-configured value of an arbitrary field in Calling List record.

To configure AM-detection on a per-record basis, OCS uses the `Business Attributes` configuration object. Use the following for guidelines when configuring it.

- Specify the name as `OC AM Detection`.
- Specify the type as `Custom`.
- Specify the display name and the description attributes as any value.

Note:	Any number of attribute values can be configured under this `Business Attribute` object. Each individual attribute value represents a field in the Calling List with a map of values for this field and the corresponding values for the desired AM-detection. You can specify only one map per calling list or OCS application. The name of the attribute value is arbitrary; however it should be meaningful so that it is easier to associate a calling list with this attribute value.

The actual map for field values for AM-detection settings translation is configured in the `Options` tab of the attribute value as follows:

- The section name must contain the name of the Calling List `Field` configuration object.
- The option name corresponds to the value of the `Field` name. The option name must contain the actual enumeration value of the field (the value of the field stored in the database). The supported option values correspond to the valid values for the `call_answer_type_recognition` option.
- The option value specifies the desired AM-detection setting.

To specify which Calling List uses which attribute value, the `am-detection-map` option can be used.

OCS uses the following approach to determine which AM-detection value to apply to a dialing request.

- OCS finds the name of the AM-detection map in the Annex tab of the Calling List that contains the record.
  - If it is not found, then the previous Campaign-level `call_answer_type_recognition` option is used

for AM-detection settings.

  - If it is found, OCS attempts to find the `Business Attribute` value with the specified name in the configuration. This attribute value should be present under the `Business Attribute` object with the `OC AM Detection` name in the same `Tenant` where the Calling List resides; if it is not found, it should be in the Environment section. If the value of the `am-detection-map` option is set to `default`, then OCS looks for the attribute value with the `Default` flag turned `ON`, as configured under the same `Business Attribute` object.

- When the map is found, OCS reads through the names of the sections in the `Options` tab for this map for the section name that matches a field name of the given calling list is used.
  - If the field is not found, or the value of this field in the current record does not have a representation in the option name, then the `call_answer_type_recognition` option is used.
- OCS assigns an AM-detection value to the Calling List record when the record is retrieved from the database if an AM-detection map is found. Otherwise, the `call_answer_type_recognition` option determines the AM-detection value when it is dialed. Any configuration changes in the AM-detection map will not affect any records that have been already retrieved by OCS.

If the attribute value is not found or disabled, then OCS uses the AM-detection method that is defined in the `call_answer_type_recognition` option.

Note:	When working with GVP, the `ANSWER_TYPE_RECOGNITION` attribute is added to the dialing attributes. This attribute holds the value for AM-detection for a given dialing request.

Starting with release 8.1.517.04, OCS provides the ability to handle an outbound call (drop or connect) on a particular device when the call is placed in any auto-dialing mode and AM is detected. The **am-drop-field** configuration option controls the AM handling logic.

Using the Opt-Out Feature With CPD Server

The OCS opt-out feature enables the call recipient to opt-out from any further outbound calls in ASM mode. This feature addresses legislative requirements and enables call recipients to opt-out by pressing certain buttons on the touch tone phone if there are no agents available to speak to them. A typical supported scenario is described in the section, Opting Out of Outbound Calls in ASM Mode.

The opt-out feature is supported with CPD Server (and optionally CPD Server Proxy) through Dual-Tone Multi-Frequency (DTMF) digit detection. This means that the call recipient must press a button(s) on the touch tone phone to be marked with a `DoNotCall` request. CPD Server uses Dialogic Application Programming Interfaces (API) to detect DTMF and supports this feature with the following Dialogic configurations

- Dialogic Springware boards (ISDN)
- Dialogic DM3 boards (ISDN)
- Dialogic HMP

OCS instructs CPD Server to play a message to the call recipient and optionally detect DTMF during or after the message if there are no agents available to speak to the call recipient. CPD Server performs the DTMF detection and passes the string of detected digits back to OCS for processing. OCS then processes the DTMF string and marks the recipient's number with a DoNotCall request if the DTMF string that was detected by CPD Server matches the pre-configured pattern of the opt-out selection. For example, if the call recipient presses the '9' button.

To enable this functionality, see the descriptions of the following options, as follows:

- `digits-detection`
- `digits-detection-pattern`
- `digits-detection-timeout`
- `digits-reaction`

OCS Feature Compliance with Regulatory Requirements

These topics describe OCS compliance with regulatory requirements:

- Ofcom Requirements
- FTC Requirements
- Submitting Call Report Records
- Contact-Processed Notifications
- OCS Lite for Regulated Environments

Ofcom Requirements

In 2010, the Office of Communications (known as Ofcom) in the United Kingdom published a statement concerning the treatment of abandoned and silent calls. The intent of the statement was to clarify and strengthen current policy and to reduce the consumer harm resulting from such calls.

To increase compliance with Ofcom, OCS 8.1.1 implements the following new functionality:

- Provide the outbound campaigns administrator with an automated way to reasonably estimate the AMD false positive rate.
- Ability to automatically include the AMD false positive rate in the Abandoned and Overdial rate calculations for Predictive campaigns.
- Ability to calculate the Abandoned rate over a fixed period of time (24 hours).
- Ability to dial outbound calls in automated dialing modes with guaranteed connection to an agent.

Estimating the AMD False Positive Rate

AMD false positive calls are calls that are answered by live individuals, but are mistakenly identified by Call Progress Detection (CPD) functionality as being answered by answering machines, and are disconnected (not connected to agents). AMD false negative calls are calls that are answered by answering machines, but are mistakenly identified by CPD functionality as being answered by live individuals. Both of these types of calls affect dialing of the campaign group. The rate at which these calls occur cannot be predicted by OCS, because it depends on a number of different factors and varies from one call center to another.

OCS has the capability to account for the AMD false positive rate when running Predictive (or Predictive with Seizing) campaigns. OCS is also capable of using the AMD false positive rate when running Progressive (or Progressive with Seizing) campaigns for reporting purposes.

If the false positive rate is determined, and the campaign administrator wants to account for it, the rate should be set using the option pa-amd-false-positive-rate and the predictive algorithm will account for it in dialing pace calculations for Predictive campaigns.

You can determine the actual value of AMD false positives and AMD false negatives in your call center using one of the following methods:

- The AMD false positive and AMD false negative rates for the campaigns are already known.
- Using percentage of calls nominated by the predictive algorithm to determine the AMD false positive and false negative rates.
- Using your own methodology to determine the AMD false positive and false negative rates.

Note:	The last two methods used for AMD false positive rate calculations can also be used in the call center to verify and fine-tune the AM detection facilities.

OCS can account for the AMD false positive rate when running Predictive (or Predictive with Seizing) campaigns in a VoIP environment. OCS can also use the AMD false positive rate when running Progressive (or Progressive with Seizing) campaigns for reporting purposes.

Note:	This feature is not available for deployments using CPD Server.

When AMD False Positive and False Negatives Rates are Known

If the AMD false positive rate for your campaigns are known at the beginning, OCS does not limit you to using its own method of determination. Just specify the rate using the `pa-amd-false-positive-rate` option and the predictive algorithm will account for it.

Using Percentage of Calls Nominated by Predictive Algorithm

You can use the predictive algorithm to nominate a percentage of calls for AMD false positive and false negative rate determination. To do this, configure the option `pa-amd-test-percentage` to the desired non-zero value in either:

- Each Campaign Group that you want to use for AMD false positive and false negative rate determination, or.
- The OCS application if you want to apply this configuration option for all campaign groups.

Once the value of this option is set to a non-zero value, the following occurs:

1. OCS randomly selects a portion of paced outbound calls to be nominated for AMD false positive and false negative rate determination. Because the calls are selected randomly, call results distribution in the nominated portion of calls will statistically be the same as between all outbound calls placed.
2. Each nominated outbound call dialed by OCS has a special key-value pair (`GSW_CALL_RESULT_FEEDBACK = 1`) attached to it.
3. OCS makes specific efforts to place each nominated call so that this call is delivered to an agent after (and if) an answering machine is detected for this call. This occurs regardless of treatments for AM calls by setting the `AttributeExtensions` in the request sent to the outbound dialer. You can turn AMD on by setting a required value of the `call_answer_type_recognition` option.
4. After a nominated call reaches an agent, it is the responsibility of the agent to detect actual call result (answered by a live individual or by an answering machine), and provide the actual result to OCS using a `RecordProcessed` or `UpdateCallCompletionStats` desktop protocol request. The call result should be delivered in the `GSW_CALL_RESULT` pair in accordance with desktop protocol. OCS can process either of these requests and will account for the first one that contains the call result.
5. After receiving the agent’s call result, OCS compares it to the call result received earlier from the dialer for this call.
6. OCS periodically reports the statistics for the campaign group for false positives and false negatives in 12-50145 and 12-50146 log messages, based on the input received from the agent desktops. Refer to the *Framework 8.1 Combined Log Events Help* for information about these log messages.
7. The campaign administrator should allow a campaign containing nominated calls to run for a period of time, then collect the accumulated data from OCS log messages and use it as a base for the calculation of the value of AMD false positives. This calculated value should then be used for `pa-amd-false-positive-rate` settings for this or other campaigns, as required.

This method allows you to determine the actual AMD false positive rate without interfering with the campaign run. You can set the value of the option pa-amd-test-percentage as required in your environment.

Note:	You can set this option to as high as 100% so that all AMD calls are delivered to agents for re-verification. However, the more AMD calls are passed to agents, the more performance of the outbound campaign is affected.

An agent’s desktop software can use the special key-value pair `GSW_CALL_RESULT_FEEDBACK=1` to ensure the agent provides the call result that was verified manually to OCS in the desktop protocol request. It is recommended that you set the OCS option `record_processed` to true for these campaigns. The key-value pair `GSW_CALL_RESULT_FEEDBACK=1` can also be used by other Genesys applications (such as Universal Routing Server) to ensure that a nominated call is delivered to an agent for call result verification if the dialer detected an answering machine for the call.

Note:	OCS provides the raw data for AMD false positive rate calculations. It is the responsibility of the campaign administrator to do the actual calculations and then set the rate for campaign groups using the pa-amd-false-positive-rate option. OCS does not calculate this percentage itself, nor does it account for it automatically for the campaign unless the rate is defined in the specified option.

Using Your Own Methodology

You are not required to use the predictive algorithm to nominate some outbound calls for AM call result verification. Instead, you can nominate your own portion of outbound calls (for example, using a Universal Routing Server strategy) to be delivered to agents for call result detection verifications. In that case, the option `pa-amd-test-percentage` should be set to `0` (zero) and the agent desktop must provide actual call result using `RecordProcessed` or `UpdateCallCompletionStats` desktop protocol requests. OCS will process these desktop requests for AMD false positive and AMD false negative statistics collection and report the resulting statistics via 12-50145 and 12-50146 log messages. These statistics can then be used for actual AMD false positive rate calculations as described earlier in this chapter.

Logging by OCS

To aid in monitoring and testing AM detection, log events 12-50145 and 12-50146 contain counts of answered and AMD calls since the last reset of the Abandoned Rate calculation using the `pa-odr-period-start-time` option. Refer to *Framework Combined Log Events Help* for the description of these log events.

Target Value of the Overdial Rate

If the goal is to optimize the Overdial Rate, and if the AMD false positive rate is configured, OCS can predict the dialing rate so that the Abandoned Rate, including AMD false positives, meets the Target Value of the Overdial Rate. OCS then calculates this reduced actual Target Value in comparison with the Target Value as defined by the Optimization parameter for the Campaign Group.

Note:	If the AMD false positive rate is not configured, OCS does not reduce the Target Value.

The reduced Target Value is calculated using the following formula:

$TR = (T - F_R) / (1 - F_R/100)$

where:

T_R = Reduced Target Value

T = Target Value specified for the Campaign Group

F_R = AMD false positive rate as a proportion of total calls answered by live individuals, expressed as a percentage

If the reduced Target Value (TR) is less than 0.5%, OCS uses 0.5% as the Target Value and periodically reports the situation.

Abandoned Rate Calculation

OCS calculates the Abandoned Rate as follows:

$AR = (A + F) / (A + C)$

where:

AR = Abandoned Rate

A = Total number of abandoned outbound calls. That is, the calls identified by CPA functionality as being answered by live individuals and not connected to agents (dropped by the dialer or abandoned by the customer), including AMD false negatives.

C = Total number of outbound calls connected to agents, including AMD false negatives.

F = Used to include AMD false positives in the Abandoned Rate in accordance with the DMA approach referenced by Ofcom, and is calculated as follows:

$F = C * F_R / 100$

with the following restriction:

$F = D$ if $(C * F_R / 100)$ is greater than D

where:

FR = AMD false positive rate as a proportion of total calls answered by live individuals, expressed as a percentage, and is set by the `pa-amd-false-positive-rate` option.

D = Total number of disconnected calls, identified by CPD functionality as being answered by answering machines, including AMD false positives.

OCS uses this Abandoned Rate for all types of reporting and logging, and for comparison with the `predictive_max_overdial_rate` and the `predictive_min_overdial_rate` (if configured) options to switch between Progressive and Predictive modes.

Time Interval for Abandoned Rate Calculation

As directed by Ofcom, OCS calculates the Abandoned Rate for each Campaign Group over a 24-hour period, starting from the time specified by the `pa-odr-period-start-time` option. At the end of the period, OCS reinitializes the calculation by setting the number of abandoned calls, calls connected to agents, and disconnected AMD calls to zero (0). If the option is absent, set to -1, or set to an invalid value, the calculation is not reinitialized and the Abandoned Rate remains unchanged.

Assured Connection

OCS is capable of dialing an outbound call in an automated dialing mode that is guaranteed to be connected to an agent. This may be desirable for calling some numbers as required by government regulations.

Outbound provides the ability to engage agents by using skills-based routing techniques with skill requirements that are defined by data in the mandatory and customer fields of a calling list record when using Progressive dialing mode and the Assured Connect functionality.

See Assured Connection for details about these functionalities.

Safe Dialing

The Safe Dialing feature protects outbound dialing from abnormal over-dialing which can be caused by incorrect configuration or routing. OCS automatically stops the outbound dialing (stops requesting new outbound calls) if abnormal over-dialing occurs.The Safe Dialing feature applies to the following dialing modes:

- Progressive
- Progressive with seizing
- Predictive
- Predictive with seizing
- Predictive GVP

Use the pa-safe-dialing configuration option to enable or disable this feature.

Note: OCS does not support Safe Dialing for IVR groups when the value of the `ivr_group` option is set to `true`, or when one or more Places that are associated with the Campaign Group contain Voice Treatment Port DNs.

FTC Requirements

Opting Out of Outbound Calls in ASM Mode

Outbound Contact supports the ability of the call recipient to opt out from any further outbound calls in ASM mode. This feature addresses legislative requirements and enables call recipients to opt-out by pressing certain buttons on the touchtone phone if there are no agents available to speak to them.

The following is a typical supported scenario:

1. An outbound call is dialed in any of the ASM modes and answered by a live person or an answering machine.
2. There is no available agent to handle the conversation.
3. A message is played to the call recipient, such as “To opt-out of any future outbound calls, press 9.”.
4. The call recipient presses 9 on the touchtone phone.
5. OCS is notified of the person's selection and marks their phone number with a DoNotCall request.
6. The call recipient does not get any more outbound calls.

OCS supports two types of ASM modes—one with CPD Server and one with SIP Server in a VoIP Environment. The opt-out feature for each type of ASM mode is implemented differently in OCS and is described in Using the Opt-Out Feature With SIP Server in VoIP Environments and Using the Opt-Out Feature With CPD Server.

End User Requirements

The end user is responsible for recording the proper drop messages that instructs the call recipient to press the touch tone buttons to opt-out of future outbound calls from the contact center. You can use a message, similar to the following example:

You have received a call from ABC Bank regarding your mortgage. Presently, there are no agents available to assist you. Please call us back anytime at 1-800-555-5555. To be excluded from receiving calls from us in the future, press 9 at any time during this message. We apologize for the inconvenience.

Submitting Call Report Records

When OCS attempts a call, it sends the call report record to Web or Application Server via an HTTP POST request. This record can be used to manage compliance with Contact Attempt rules.

Call report records are sent to an external system twice.

For each attempt to place a call (or interaction) made from a started outbound dialing campaign in any dialing mode, OCS sends two call report records. The first call record (an HTTP POST request with **GSW_HIST_SEQUENCE_NUM: 0**) to Web or Application Server is sent when a call attempt is initiated. When the call is finalized with any result (`Answer`, `Answering Machine`, `Abandoned`, `DoNotCall`, and so on), the second call record (an HTTP POST request with **GSW_HIST_SEQUENCE_NUM: 1**) containing the **GSW_CALL_RESULT** parameter is sent to Web or Application Server.

The **GSW_HIST_SEQUENCE_NUM: 1** record overwrites the **GSW_HIST_SEQUENCE_NUM: 0** record when dialing rules are applied by the compliance manager.

Feature Configuration

Add the following configuration options, as described on the OCS Option Descriptions page in this manual:

- `send-history`
- `history-uri`
- `history-wait-condition-threshold`
- `history-wait-recovery-timeout`
- `validation-race-condition-call-result`

The data posted to Web or Application Server can contain the following attributes. Refer to the Reference Manual for a detailed description of each attribute.

Field	Description	Type	Mandatory
GSW_CALL_ATTEMPT_GUID	The global unique identifier of the call processing attempt used for historical reporting.	String	Yes
GSW_HIST_SEQUENCE_NUM	The 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	Integer	Yes

Field	Description	Type	Mandatory
	report record submission for a given finalized call.		
GSW_TENANT_NAME	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.	String	Yes, with exception when the Annex of the Tenant is not configured properly
GSW_TENANT_CCID	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.	String	Yes, with exception when the Annex of the Tenant is not configured properly
GSW_PHONE	The customer’s phone number.	String	Yes
GSW_CHAIN_ID	The unique chain identifier.	Integer	Yes
GSW_CHAIN_N	The unique identifier of a record in a chain.	Integer	Yes
GSW_PHONE_TYPE	The customer's phone type.	Integer	Yes
GSW_CUSTOMER_ID	The customer identifier that is used for requests. The ID that is taken from the `c_client_id` field of the Calling List (this field is configured with the GSW_CUSTOMER_ID send_attribute). This field is optional but it is sent if it is defined in the Calling List table.	String	No
GSW_START_PROCESSING	The timestamp when OCS starts processing a given call attempt. If pre-dial validation is in place, this is the time when a pre-dial validation request is sent.	Unix UTC epoch dot milliseconds	Yes
GSW_CALL_TIME	The time when OCS	Unix UTC epoch dot	No. The field is available

Field	Description	Type	Mandatory
	sends a request to the dialer (for example, TMakePredictiveCall).	milliseconds	when a request was actually sent. The field is not available for negative pre-dial validation.
GSW_SCHEDULED_TIME	The time when the record is rescheduled. GSW_DATE_TIME in OCS desktop protocol terms.	Unix UTC epoch dot milliseconds	No. The field is available only with records of particular types.
GSW_TZ_OFFSET	The time difference, in seconds, between Universal Time Coordinated (UTC) and a particular time zone.	Integer	Yes
GSW_TZ_NAME	The Configuration Server time zone name (usually a standard three-letter abbreviation)--for example: PST, CTT, GMT.	String	Yes
GSW_CALL_RESULT	The call result of the primary record.	Integer	Yes
GSW_CAMPAIGN_NAME	The name of the Campaign.	String	Yes
GSW_CAMPAIGN_TEMPLATE_NAME	The name of the Campaign Template (the Script of type Outbound Campaign, referenced by the Campaign Group via the scriptDBID attribute).	String	No. OCS does not deliver it if the Campaign Group does not have a reference to the Template (possible in non-CX Contact configurations).
GSW_CALLING_LIST	The name of the Calling List.	String	Yes
GSW_CALLING_LIST_DBID	The database identifier of the calling list.	Integer	Yes
GSW_CAMPAIGN_GROUP_NAME	The name of the Campaign Group.	String	Yes
GSW_CAMPAIGN_GROUP_GUID	The global unique identifier of the Campaign Group, which is taken from uuid_generator::to_str(Session.pSessionGUID).	String	Yes
GSW_GROUP_NAME	The Agent or Place Group name.	String	Yes
GSW_QUEUE_NAME	The name of the Voice Transfer Destination DN.	String	Yes
GSW_DIAL_MODE	Current dialing mode of the Campaign Group.	Integer	Yes

Field	Description	Type	Mandatory
GSW_AGENT_ID	The Agent ID field value.	String	No. The field is available for successful calls only.
GSW_RECORD_TYPE	The record type of the primary record after the processing.	Integer	Yes
GSW_RECORD_STATUS	The record status.	Integer	Yes
GSW_OPTIMIZE_BY	The optimization method (enumeration).	Integer	No. The field is available for Predictive modes only.
GSW_OPTIMIZE_GOAL	The optimization goal.	Integer	No. The field is available for Predictive modes only.
GSW_DIALING	The time when call dialing is started. The timeDialing value that is taken from FTC timestamps.	Unix UTC epoch dot milliseconds	No
GSW_RINGING	The time when a call is ringing. The timeClientRinging value that is taken from FTC timestamps.	Unix UTC epoch dot milliseconds	No
GSW_RELEASED	The time when a call is released. The timeBadCallReleased value that is taken from FTC timestamps.	Unix UTC epoch dot milliseconds	No
GSW_ANSWERED	The time when a call is answered. The timeClientPickedUp value that is taken from FTC timestamps.	Unix UTC epoch dot milliseconds	No
GSW_CPD_COMPLETED	The time when CPD is completed. The timeCPDFinished value that is taken from FTC timestamps.	Unix UTC epoch dot milliseconds	No
GSW_QUEUED	The time when a call is placed in a queue. The timeQueued value that is taken from FTC timestamps.	Unix UTC epoch dot milliseconds	No
GSW_AGENT_RINGING	The timeAgentRinging value that is taken from FTC timestamps.	Unix UTC epoch dot milliseconds	No
GSW_AGENT_ANSWERED	The time when an agent answers a call. The timeAgentEstablished value that is taken from FTC timestamps.	Unix UTC epoch dot milliseconds	No

Field	Description	Type	Mandatory
GSW_DIVERTED	The time when a call is diverted. The timeAMDiverted value that is taken from FTC timestamps.	Unix UTC epoch dot milliseconds	No
GSW_ABANDONED	The time when a call is abandoned. The timeAbandoned value that is taken from FTC timestamps.	Unix UTC epoch dot milliseconds	No
GSW_AGENT_RELEASED	The time when an agent releases a call. The timeAgentCallReleased value that is taken from FTC timestamps.	Unix UTC epoch dot milliseconds	No
GSW_ACW_COMPLETED	The time when After Call Work is completed.	Unix UTC epoch dot milliseconds	No
GSW_COMPLETE_PROCESSING	The time when processing of a call attempt is completed. The time when an outbound record for the given call is considered completed and is removed from OCS active processing.	Unix UTC epoch dot milliseconds	No
GSW_POSTAL_CODE	The client's postal code.	String	No. The field is available when it's defined in the calling list.
GSW_COUNTRY_CODE	The client's country code.	String	No. The field is available when it's defined in the calling list.
GSW_TZ_JAVA_NAME	The client's time zone.	String	No. The field is available when it's defined in the calling list.
GSW_STATE_CODE	The client's region code. This is a two-letter abbreviation and not ISO 3166-2 (for example, CA and not US-CA).	String	No. The field is available when it's defined in the calling list.
GSW_D_AREA_CODE	The area code associated with the device.	String	No. The field is available when it's defined in the calling list.
GSW_D_COUNTRY_CODE	The country code associated with the device.	String	No. The field is available when it's defined in the calling list.
GSW_D_TZ_JAVA_NAME	The time zone associated with the device.	String	No. The field is available when it's defined in the calling list.

Field	Description	Type	Mandatory
GSW_D_STATE_CODE	The region code of a device. This is a two-letter abbreviation and not ISO 3166-2 (for example, CA and not US-CA).	String	No. The field is available when it's defined in the calling list.
GSW_DEVICE_MASK	The device mask.	Integer	No. The field is available when it's defined in the calling list.
GSW_BLOCKING_RULE	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.	String	No. The field is available only for negative pre-dial validation results.
GSW_BLOCKING_RULE_TYPE	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 validation response. In such case, OCS passes it in a call attempt record submission unchanged.	Integer	No. The field is available only for negative pre-dial validation results.
User-defined fields: <USER FIELD KEY NAME>	Any user-defined fields with the send attribute, where send attribute values does not start with **GSW_**.	String	No
GSW_CPN_DIGITS	The value of CPNDigits that is used in this call attempt. (The TMakePredictiveCall request parameter or attached to the preview record via the GSW_CPN_DIGITS key-value pair). May not be present if no CPNDigits has been configured at any level.	String	No
GSW_CPN_DIGITS_SET	The name of the Caller	String	No

Field	Description	Type	Mandatory
	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_DISPOSITION_CODE	The value of the Disposition Code that is provided to OCS by the agent desktop in a record finalization request or third-party validation server in a 409 Cancel negative response body in the GSW_DISPOSITION_CODE key-value pair.	String	No
GSW_DEVICE_INDEX	The device index value for a given record.	Integer	Yes

Responses

When OCS sends the HTTP POST request, Web or Application Server returns a response code:

- `200 OK` — success
- 4xx or 5xx — failure

Re-submitting Failed Records

If OCS receives any response other than `200 OK` from Web or Application Server after submitting the HTTP POST request, it treats the record as failed and re-submits the record.

OCS checks the value in the `history-wait-recovery-timeout`, which stores the time OCS should wait before re-submitting another HTTP POST request and will re-submit the HTTP POST request to Web or Application Server.

If OCS reaches the threshold specified in the `history-wait-condition-threshold` option, the wait condition is triggered, which means dialing is suspended until re-submission of all failed records is completed.

Important

With a wait condition in place, the campaign remains in a running state, but dialing is suspended.

If OCS reaches the threshold specified in the `history-wait-condition-threshold` option before receiving a `200 OK` response, it stops dialing.

Race Conditions

If OCS sends a pre-dial validation request to Web or Application Server for a contact record that Web or Application Server is currently processing but has not yet saved (for example, it has not yet sent a response or the record is marked for re-processing), a race condition occurs.

As a prevention measure, OCS checks for a race condition before submitting a pre-dial validation request. If it detects a race condition, it will not dial the record. Instead, OCS refers to the call result specified in the `validation-race-condition-call-result` option and attempts to apply that treatment to the call result.

In this scenario, the following log entry appears:

```
Predial check aborted due to race condition for phone %s
```

Contact-Processed Notifications

Starting with release 8.1.528.21, OCS can deliver contact-processed HTTP[S] notifications to an external Web or Application Server.

This feature can be used in an environment where feedback from OCS call processing results is required in real time and must be integrated with the customer's backend CRM or similar systems, to provide real-time synchronization of call results and outbound activity outside of the OCS/CX Contact solution. OCS sends notifications of call processing results through HTTP POST requests in a JSON body, delivered over HTTP and HTTPS (TLS 1.2) protocols to a Web or Application Server. HTTPS notifications can be sent via a tunneling HTTP Proxy.

In a typical scenario, a record is retrieved from a calling list and dialed; a call is considered successful and an agent dispositions the call. OCS sends a notification containing a call result to a third-party Web server, based on which the customer CRM system performs further actions.

OCS sends a *contact-processed notification* per contact (per chain) upon completion of the chain processing for contacts or chains in any dialing mode and for Campaign Groups handling voice, email, and SMS media types. Notifications contain only chain and primary record-specific information in the JSON body of the message.

Feature Configuration

Use the following OCS options to enable this feature:

- contact-processed-notification
- contact-processed-notification-uri
- http-retry-applies-to
- http-retry-max-attempts
- http-connection-pool-size
- http-custom-headers
- http-proxy
- http-response-timeout

Message Content Details

OCS generates a message for a chain when the chain reaches the *processed* state in its Finite State Machine. For example, a negative call result is received for a chain and a treatment (if present) is prepared to be applied, or an agent has finalized the chain processing from the desktop.

Message Headers

The message contains standard HTTP message headers (similar to pre-dial validation requests). The http-custom-headers option, defined at the Application or Host object level, specifies custom headers that can be added to the message. For example:

```
Accept: */*
User-Agent: OCS/8.1.528.15
Host: 172.21.83.91:80
ETag: <Value of GSW_MESSAGE_GUID>
Content-type: application/json
Content-length: 1307
```

Message Body

The following table describes attributes that can be included in the JSON body.

Field	Description	Type	Mandatory
GSW_MESSAGE	ContactProcessed	String	Yes
GSW_MESSAGE_GUID	The unique ID of the HTTP notification message being sent. It also is passed in the ETag header of the HTTP POST request. If a notification is re-sent because of the HTTP retry logic, GSW_MESSAGE_GUID remains the same.	String, GUID	Yes
GSW_TIMESTAMP	The UTC timestamp when the message is being generated. If it is resent because of the HTTP retry logic, the timestamp remains the same.	String, Unix UTC epoch dot milliseconds	Yes
GSW_CALL_ATTEMPT_GUID	The unique ID of the call processing attempt.	String, GUID	Yes
GSW_PHONE	The phone number (contact_info) field of the primary record in the chain.	String	Yes
GSW_PHONE_TYPE	The phone type (contact_info_type) field of the primary record in the chain.	Integer	Yes
GSW_CHAIN_ID	The value of the Chain ID field of the primary record.	Integer	Yes

Field	Description	Type	Mandatory
GSW_CHAIN_N	The value of the Chain N field of the primary record.	Integer	Yes
GSW_RECORD_HANDLE	The Record Handle of the primary record in the chain, for troubleshooting purposes.	Integer	Yes
GSW_REASON	The reason code for the event generation that is equal to the Cause of the GOChainProcessed event—for example, GOCauseChainProcessingDone.	Integer	Yes
GSW_CHAIN_LIFO_GUID	The GUID of the chain that is created by CX Contact if the Campaign Group is a CX Contact and LIFO Campaign Group. This field is CX Contact specific. It allows to identify LIFO-processed contacts vs. standard contacts.	String, GUID	No
GSW_CUSTOMER_ID	The ID that is taken from the `c_client_id` field of the Calling List (this field is configured with the GSW_CUSTOMER_ID send_attribute). This field is optional but it is sent if it is defined in the Calling List table.	String	No
GSW_CALL_RESULT	The call result field of the primary record.	Integer	Yes
GSW_DISPOSITION_CODE	The value of the Disposition Code that is provided to OCS by the agent desktop in a record finalization request or third-party validation server in a 409 Cancel negative response body in the GSW_DISPOSITION_CODE key-value pair. This field is important; it affects processing on the remote end.	String	No
GSW_CAMPAIGN_NAME	The name of the Campaign (not	String	Yes

Field	Description	Type	Mandatory
	Campaign Group).		
GSW_CAMPAIGN_TEMPLATE_NAME	The name of the Campaign Template (the Script of type Outbound Campaign, referenced by the Campaign Group via the scriptDBID attribute). This field is CX Contact specific.	String	No. It will not be delivered by OCS if the Campaign Group does not have a reference to the Template (possible in non-CX Contact configurations).
GSW_CONTACT_MEDIA_TYPE	As set for the Campaign Group, "voice" | "sms" | "email".	String	Yes
GSW_CALLING_LIST	The name of the Calling List.	String	Yes
GSW_CAMPAIGN_GROUP_NAME	The name of the Campaign Group.	String	Yes
GSW_CAMPAIGN_GROUP_GUID	The Campaign Group (Session) GUID, which is taken from uuid_generator::to_str(Session.pSessionGUID).	String	Yes
GSW_GROUP_NAME	The Agent or Place Group name.	String	Yes
GSW_QUEUE_NAME	The name of the Voice Transfer Destination DN.	String	Yes
GSW_DIAL_MODE	Current dialing mode of the Campaign Group.	Integer	Yes
GSW_AGENT_ID	The Agent ID field value.	String	No. The field is available for successful calls only.
GSW_RECORD_TYPE	The record type of the primary record after the processing.	Integer	Yes
GSW_RECORD_STATUS	The record status. For chains that are finalized into the calling list, the status is final (Updated), not intermediate (Retrieved). For chains that are kept processing in memory by a treatment, the status may be Retrieved.	Integer	Yes
GSW_OPTIMIZE_BY	The optimization method.	Integer	No. The field is available for Predictive modes only.
GSW_OPTIMIZE_GOAL	The optimization goal.	Integer	No. The field is available for Predictive modes only.
GSW_START_PROCESSING	The timestamp when OCS starts processing a	String, Unix UTC epoch dot milliseconds	Yes

Field	Description	Type	Mandatory
	given call attempt. If pre-dial validation is in place, this is the time when a pre-dial validation request is sent.		
GSW_CALL_TIME	The time when OCS sends a request to the dialer (for example, TMakePredictiveCall).	String, Unix UTC epoch dot milliseconds	No. The field is available when a request was actually sent. The field is not available for negative pre-dial validation.
GSW_SCHEDULED_TIME	The time when the record is rescheduled. GSW_DATE_TIME in OCS desktop protocol terms.	String, Unix UTC epoch dot milliseconds	No. The field is available only with records of particular types.
GSW_DIALING	The timeDialing value that is taken from FTC timestamps.	String, Unix UTC epoch dot milliseconds	No
GSW_RINGING	The timeClientRinging value that is taken from FTC timestamps.	String, Unix UTC epoch dot milliseconds	No
GSW_RELEASED	The timeBadCallReleased value that is taken from FTC timestamps.	String, Unix UTC epoch dot milliseconds	No
GSW_ANSWERED	The timeClientPickedUp value that is taken from FTC timestamps.	String, Unix UTC epoch dot milliseconds	No
GSW_CPD_COMPLETED	The timeCPDFinished value that is taken from FTC timestamps.	String, Unix UTC epoch dot milliseconds	No
GSW_QUEUED	The timeQueued value that is taken from FTC timestamps.	String, Unix UTC epoch dot milliseconds	No
GSW_AGENT_RINGING	The timeAgentRinging value that is taken from FTC timestamps.	String, Unix UTC epoch dot milliseconds	No
GSW_AGENT_ANSWERED	The timeAgentEstablished value that is taken from FTC timestamps.	String, Unix UTC epoch dot milliseconds	No
GSW_DIVERTED	The timeAMDiverted value that is taken from FTC timestamps.	String, Unix UTC epoch dot milliseconds	No
GSW_ABANDONED	The timeAbandoned value that is taken from FTC timestamps.	String, Unix UTC epoch dot milliseconds	No
GSW_AGENT_RELEASED	The	String, Unix UTC epoch	No

Field	Description	Type	Mandatory
	timeAgentCallReleased value that is taken from FTC timestamps.	dot milliseconds	
GSW_ACW_COMPLETED	The time when After Call Work is completed.	String, Unix UTC epoch dot milliseconds	No
GSW_COMPLETE_PROCESSING	The time when processing of a call attempt is completed. The time when an outbound record for the given call is considered completed and is removed from OCS active processing.	String, Unix UTC epoch dot milliseconds	No
User-defined fields: <USER FIELD KEY NAME 1> <USER FIELD KEY NAME 2> ... <USER FIELD KEY NAME X>	Any user-defined fields with the send_attribute defined, which have been modified during this round of chain processing—for example, Other1-OtherX fields as modified by the agent desktop.	String	No

Feature Limitations

- Authentication (except for providing pre-configured HTTP headers) is not supported. OAuth and similar authentication schemas with OCS acting as a client are not possible.
- OCS delivers a notification per chain, not per record. For scenarios involving chain processing (with multiple records from the chain participating in the processing), the message will contain information for a primary record in the chain only. In the scenario below, the primary record is record 1, so the notification is sent with user data updated for record 1. No notifications to a third-party system about records 2 and 3 user data are being updated.
  - A chain consists of 3 records.
  - A call is placed using record 1 (the primary record) and delivered to the agent desktop.
  - An agent requests all chains, record 2 and 3 are sent to the desktop.
  - The agent releases the call.
  - The agent updates user data in record 2 (by sending an UpdateCallCompletionStats desktop request).
  - The agent updates user data in record 3 (by sending an UpdateCallCompletionStats desktop request).
  - The agent updates user data in record 1 and finalizes the chain with a RecordProcessed desktop request.
- HTTP redirections are not supported.

OCS Lite for Regulated Environments

Outbound Contact Server (OCS) Lite is an outbound server, available starting with the 8.1.530.03 release, that is used to run outbound dialing campaigns in Telephony Consumers Protection Act (TCPA) and other legislation-regulated environments, where regulations require manual call dialing with preliminary explicit consent to contact the consumers.

OCS Lite enables you to run outbound campaigns in Preview, Push Preview, or Direct Push Preview dialing modes.

The executable name is **cm_server_lite.exe** on Windows and **cmserver_lite_64** on Linux.

OCS Lite Configuration

OCS Lite is configured using a dedicated Application of type `Outbound Contact Server`, the same as OCS. No specific options settings are required for OCS Lite. OCS Lite can run on its own or along with the regular OCS, serving as a proxy and delegate execution of outbound campaigns in auto-dialing modes or non-TCPA campaigns to the regular OCS.

If required for your environment, you can add a primary OCS application to the Connections tab of OCS Lite. Connection can be optionally configured as ADDP with tracing. OCS Lite attempts to connect to the OCS application at startup if it is present in connections. OCS Lite must attempt to re-connect to the backup OCS instance if the connection to the primary OCS is lost or cannot be established.

Campaign Groups running on OCS Lite

Campaign Groups must be explicitly configured for regulated operations. CX Contact takes care of proper Campaign Groups configuration.

operation-mode

- Default Value: `normal`
- Valid Values: `normal, regulated`
- Configuration Level: Campaign Group
- Logical Group: Dialing Regulations
- Changes Take Effect: At next Campaign Group activation or when Campaign Group is loaded

When set to `regulated`, this option designates a Campaign Group as a Regulated Campaign Group, which can only be processed by OCS Lite and only in Preview dialing modes.

Dialing Modes

Regulated Campaign Groups can be executed only in Preview dialing modes, which could be one of the following:

- Preview, either in standard or LIFO mode
- Direct Push Preview, either in standard or LIFO mode
- Push Preview with Interaction Server, either in standard or LIFO mode

Starting and Stopping OCS Lite

You start and stop OCS Lite the same was as a regular OCS. See Starting and Stopping Procedures for details.

Deploying Do Not Call Functionality

This topic discusses the deployment of Do Not Call functionality.

Do Not Call Table Structure

The `Do Not Call` table has a fixed structure, as shown in the following table. The `customer_id` field, like the phone field, is part of that established structure. The `Do Not Call` table does not require Format and Field configurations. Genesys Administrator generates this table if the `Table Access` object is present, but the physical table does not yet exist.

Note:	If you manually add entries to the `Do Not Call` table, you must restart Outbound Contact Server (OCS) so that OCS can read the new records into its memory. 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 this functionality, see Rereading of the Do-Not-Call List.

Do Not Call Table Structure

Field Name	Type	Nullable
phone	varchar (64)	yes
customer_id	varchar (64)	yes
dnc_message	varchar (255)	yes
tenant_dbid	decimal (18, 0)	yes
campaign_dbid	decimal (18, 0)	yes
list_dbid	decimal (18, 0)	yes
is_internal	int	yes
time_stamp	int	yes
expires_at	int	yes

Note:	The phone field in 6.5 was changed to the `contact_info` field in the 7.0 Calling List table. The phone field is still in the Do Not Call List table.

User-Defined Field for Do Not Call

The restriction on dialing a particular customer is an alternative to the Do Not Call restriction applied to a particular telephone number. The ability to apply a a Do Not Call request to a specific customer is helpful when the same phone number appears on more than one customer's records. For example, in a family or roommate situation, one member of the household might want to be on the Do Not Call list while another person at the same residence and with the same telephone number might not make that request.

The value of the `customer_id` option in the OCS Application object is the name of the field that the user designates for the customer identifier. At start-up, OCS reads all the records from the table referenced in the `gsw_donotcall_list` Table Access and populates two separate tables in memory with unique values from the phone field and from the `customer_id` field. Do Not Call requests from the agent desktop can also populate these tables in memory during runtime.

Configuration Procedure

Perform this procedure in Genesys Administrator to create a user-defined field to identify customers for a Do Not Call List. This new field will serve as the `customer_id` for Do Not Call requests and will be included in the `UserData` attached to T-Server events. As the value of the `customer_id` option in the OCS Application object, this field will correspond to the `customer_id` field in the Do Not Call (`gsw_donotcall_list`) table.

Creating a User-Defined Field to Identify Customers for the Do Not Call List

Start

1. Configure a Table Access object for the `gsw_donotcall_list`:
   - In Genesys Administrator, go to `<Tenant> > Provisioning > Outbound Contact > Do Not Call List`.
   - In the `Configuration > Database`, specify the following:
     - DB Access Point = `<user-specific name of the DB Access Point>`
     - Database Table = `<user-specific name of the database table>`

     Genesys Administrator automatically populates other required properties from the corresponding DB Access Point.
2. Create the database table for the Do Not Call list:
   - In Genesys Administrator, go to `<Tenant> > Operations > Calling Lists`.

     Genesys Administrator automatically creates the database table if it is not created when the Table Access object for the `gsw_donotcall_list` is configured.
3. Create a new user-defined Field object:
   - In Genesys Administrator, go to `<Tenant> > Provisioning tab > Outbound Contact > Fields`.
   - On the Configuration tab, specify the following:
     - Name = `<user-specific name>`
     - Data Type = `varchar`
     - Length = 64

    - Field Type = `User-Defined Field`
    - Select the `Nullable` and `State Enabled` checkboxes.
  - On the Options tab, create a new option, as follows:
    - Section = `default`
    - Name = `send_attribute`
    - Value = `GSW_CUSTOMER_ID`

4. Create a Format which includes all the mandatory fields from `Default_Outbound_70` and also the new user-defined Field; and then create a Calling List based on that, as described in Field Object, Format Object, and Calling List Object.
5. Designate the new user-defined Field as Customer ID in the OCS Application object:
   - In Genesys Administrator, go to `<Tenant> > Provisioning > Environment > Applications > Outbound`.
   - In the OCS application, in the `Options > OCServer` section, create and define the `customer_id` option, as follows:
     - Section = `OCServer`
     - Name = `customer_id`
     - Value = `<name of the new user-defined Field>`

End

Updating the DNC List

Through Genesys Administrator, you can update a current DNC list with data from an external source. Genesys Administrator first reads data from an ASCII file, which is in comma-delimited format. The user then maps this data to the appropriate columns in the DNC list (database table).

The process to update or import data from an external source is composed of three steps:

1. Selecting the data file and type.
2. Assigning names to data columns.
3. Specifying data import options.

For detailed instructions, see *Framework Genesys Administrator Help* (Provisioning Your Environment > Outbound Contact Object Types > Do Not Call List > Importing a Do Not Call List File).

OCS-Desktop Protocol Changes for DNC

A `Do Not Call` (DNC) request from an agent (`GSW_AGENT_REQ_TYPE = DoNotCall`) includes an attribute to specify the `customer_id: GSW_CUSTOMER_ID`. At least one attribute (`GSW_PHONE or GSW_CUSTOMER_ID`) must be present in the UserData of the request if the `GSW_RECORD_HANDLE` is not specified.

Do Not Call Duration

The Do Not Call Duration functionality enables you to set an expiration date for a DNC request. On the date of expiration, OCS removes or ignores the DNC entry.

Configuration

To configure DNC Duration, you must first manually add the `expires_at` field to the `Do Not Call` table.

Enter the Database Definition Language (DDL) statement that applies to your database type:

- **PostgreSQL**

```
ALTER TABLE <dnc_table_name> ADD COLUMN expires_at integer NULL
```

- **MS SQL**

```
ALTER TABLE <dnc_table_name> ADD expires_at int NULL
```

- **Oracle**

```
 ALTER TABLE <dnc_table_name> ADD expires_at int NULL
```

This field will store the expiration time in POSIX format (seconds since midnight of 01/01/1970). A value of `0` or `NULL` means the DNC entry never expires.

Next, configure the following options:

- `dnc-use-duration` - Used to enable DNC Duration. If the value is set to `false`, there is no change in the way OCS treats a DNC entry. If the value is set to `true`, OCS stores the `expires_at` value found in the `Do Not Call` table and later retrieves the value to determine if a given DNC entry (for new dials only) is active (not expired) or expired.
- `dnc-default-duration` - Used when the `GSW_DURATION` attribute is not specified in a DNC request that has a value specified in the `GSW_RECORD_HANDLE` attribute.
- `dnc-purge` - Used to remove expired DNC entries. The purge occurs after OCS reads the DNC table after restart or after OCS re-reads the DNC table in accordance with the `dnc-reread` option.

Finally, add the `GSW_DURATION` attribute to the Desktop communication protocol (for both transports, T-Events and HTTP). This attribute represents the DNC duration, in seconds. OCS adds this value to the current time to calculate the expiration date, which it then stores in the `Do Not Call` table.

Handling Requests

The following scenarios describe the process OCS uses to handle DNC duration requests.

Scenario 1:

- `GSW_DURATION` is not specified.
- `GSW_RECORD_HANDLE` is not specified.
- **Resolution**: OCS applies a value of `0`. It does not refer to the `dnc-default-duration` option, regardless of the settings.

Scenario 2:

- `GSW_DURATION` is not specified.
- `GSW_RECORD_HANDLE` is specified.
- **Resolution**: OCS retrieves the DNC duration from the `dnc-default-duration` option.

Scenario 3:

- `GSW_DURATION` is specified.
- `dnc-default-duration` is specified.
- **Resolution**: OCS uses the `GSW_DURATION` value.

Handling Multiple Requests

If OCS receives multiple DNC requests for the same customer, and all of those requests have different expiration dates, it applies the later expiration date.

For example, if one DNC request for a given customer specifies an expiration date of January 1, 2018 and another DNC request for that same customer specifies an expiration date of March 1, 2018, OCS recognizes March 1, 2018 as the expiration date.

Campaign Group Status Reporting

Starting with release 8.1.512.02, Outbound Contact Server can send data about Campaign Group state changes to a Web or Application server. 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.

Starting with release 8.1.517.04, OCS submits a set of real-time statistics related to the Campaign Group and its Calling Lists in the Campaign Group State POST HTTP notifications.

The following table describes statistics-related attributes that OCS passes to an external Web/Application server in the JSON body of the HTTP notification:

Attribute	Type	Level	Example
GSW_STAT_BUSY_FACTOR	Non-negative numeric or null	Campaign Group	• "GSW_STAT_BUSY_FACTOR": 0.85 // 85% Agent Busy Factor • "GSW_STAT_BUSY_FACTOR": null
GSW_STAT_HIT_RATIO	Non-negative numeric or null	• Campaign Group • Calling List(s)	• "GSW_STAT_HIT_RATIO": 0.3 // 30% Hit Ratio • "GSW_STAT_HIT_RATIO": null
GSW_STAT_OVERDIAL_RATE	Non-negative numeric or null	Campaign Group	• "GSW_STAT_OVERDIAL_RATE": 0.02 // 2% ODR • "GSW_STAT_OVERDIAL_RATE": null
GSW_STAT_TIME_TO_COMPLETE	Non-negative numeric or null	• Campaign Group • Calling List(s)	• "GSW_STAT_TIME_TO_COMPLETE": 600 // Complete in 10 minutes

Attribute	Type	Level	Example
			- "GSW_STAT_TIME_TO_COMPLETE": null

Feature Configuration

To configure this feature, use the following configuration options:

- Set send-campaigngroup-states to `true` to enable the feature.
- Set campaigngroup-states-uri to the URI of the Web or Application server.
- If you want to generate periodic messages about current Campaign Group activity that also contain Campaign Group statistics, set campaigngroup-states-interval to the frequency at which you want to send the messages (the default is 10 seconds).

OCS uses the following options to retrieve connection information about the configuration:

- http-response-timeout
- http-connection-pool-size

Log Messages

OCS uses the following Trace-level log messages to report that HTTP POST requests have been sent. No actions are required in response to these events.

ID	Level	Text	Attributes	Description
81000	TRACE	Sending HTTP status notification CampaignGroupState for Campaign Group %s to %s	%s Campaign Group name %s URI to which the notification has been sent.	An HTTP status notification has been sent to a Web or Application server that the status of a Campaign Group has changed. For information included in the JSON body of HTTP POST requests, see Fields in HTTP POST Requests to Web or Application Server, below.
81001	TRACE	Sending periodic HTTP status notification	%s Campaign Group name	A periodic HTTP status notification has been sent to a

ID	Level	Text	Attributes	Description
		CampaignGroupState for Campaign Group %s to %s	%s URI to which the notification has been sent	Web or Application server for an active Campaign Group. For information included in the JSON body of HTTP POST requests, see Fields in HTTP POST Requests to Web or Application Server, below.

Error Handling

There is no error-handling carried out by this feature. If the HTTP POST to the Web or Application server fails, no post-processing or retries are executed.

Fields in HTTP POST Requests to Web or Application Server

The fields in the table below are passed by OCS in the JSON body of HTTP POST requests to the Web or Application server. All mandatory fields are sent, but fields not relevant to the status change or periodic activity scenario are left unassigned or set to a default value:

Attribute	Type	Required	Description
GSW_USER_EVENT	String	Yes	Set to CampaignGroupState.
GSW_DATE_TIME	String	Yes	Timestamp when action occurred; UTC time as seconds, with msec precision.
GSW_TENANT_NAME	String	Yes	Defaults to Unknown if not set.
GSW_TENANT_CCID	String	Yes	Defaults to Unknown if not set.
GSW_CAMPAIGN_GROUP_NAME	String	Yes	
GSW_CAMPAIGN_GROUP_GUID	String	Yes	Unique UUID for Campaign Group
GSW_CAMPAIGN_GROUP_DBID	String	Yes	Unique DBID for Campaign Group
GSW_GROUP_NAME	String	Yes	
GSW_CAMPAIGN_NAME	String	Yes	
GSW_CAMPAIGN_TEMPLATE_NAME	String	Yes, if present in	

Attribute	Type	Required	Description
		configuration	
GSW_QUEUE_NAME	String	Yes	Name of the Voice Transfer Destination DN.
GSW_DIAL_MODE	Integer	No	Defined only for running Campaign Groups. Use Enumeration Values for GSW_CM_AttrDialMode.
GSW_OPTIMIZE_BY	Integer	No	Optimization method for the Campaign. Use Enumeration Values for GSW_CM_AttrOptimizeBy. Available for Predictive modes only.
GSW_OPTIMIZE_GOAL	Integer	No	Available for Predictive modes only.
GSW_CAMPAIGN_GROUP_STATE	Integer	Yes	Use Enumeration Values for GSW_CM_AttrGroupCampStatus.
GSW_CAMPAIGN_GROUP_ACTION	Integer	Yes	Identifies client request which has caused message generation. Use CM_ReqSetDialingMode when dial mode is changed. Use CMMessageType enum values CM_ReqLoadCampaign, CM_ReqUnloadCampaign, CM_ReqStartDialing, CM_ReqStopDialing, or CM_ReqSetDialingMode.
GSW_CAMPAIGN_GROUP_ACTOR	Integer	Yes	Distinguishes action caused by OCS Client or automatic action taken by OCS itself (either per Schedule or per internal Campaign Group condition). Valid values are: 0 (zero) -- Actor Type unknown 1 -- Actor Type OCS Client (actions invoked by direct OCS Client connected via CM Library protocol)

Campaign Group Auto Completion

OCS can automatically complete a Campaign Group under the following conditions:

- When all retrieved calling list records have been finalized.
- If no calling list records were retrieved when the last request was sent to the database.

Scenario

1. For a `LoadCampaign`, `StartDialing`, or `SetDialingMode` event, an HTTP request is sent to OCS.
2. The request contains the attribute `GSW_AUTO_COMPLETION` with a value of 1.
3. OCS has no records to dial, either because all have been finalized or because none were retrieved when the last request was sent to the database.
4. OCS stops the Campaign Group automatically via `Stop` and `Unload` actions.
5. Agents receive the following notifications: `CampaignStopped` and `CampaignUnloaded`.

Feature Configuration

The following OCS Communication Protocol - direct, T-Events, and HTTP - attribute is required.

Protocol Attribute (KVP Key)	Value Type	Description
GSW_AUTO_COMPLETION	Integer	• A value of `0` indicates that the function is disabled • A value of 1 indicates that the Campaign Group will automatically stop when the record depletion conditions, described above, are met.

Important

Remember - this attribute can only be added to the following events:

- `CampaignLoad`

- `StartDialing`
- `SetDialingMode`

Processing

In the event the attribute contains a value of 1, OCS stops the Campaign Group automatically via the `Stop` and `Unload` actions. As a result, agents receive `CampaignStopped` and `CampaignUnloaded` notifications.

If the attribute contains a value of `0` or if a `CampaignLoad`, `StartDialing`, or `SetDialingMode` event does not contain the `GSW_AUTO_COMPLETION` attribute, OCS ignores the functionality.

Example Requests and Responses

HTTP Request to GWS for direct connect to OCS

Request

```
POST http://127.0.0.1:8098/ocs/v3/servers/156/load-campaign
{
"data":
{"campaignId":101,
"requestProperties":[
{"key":"GSW_AUTO_COMPLETION", "value":1}
],
"groupId":732}
}
```

Response

```
23:57:57.891 OCMClient[9]::message received
GSW_CM_MessageType             7 [GSW_CM_ReqLoadCampaign]
GSW_CM_AttrReferenceID         10
GSW_CM_AttrCampaignID          101
GSW_CM_AttrGroupID             732
GSW_CM_AttrProperties
 'GSW_AUTO_COMPLETION' '****'
```

HTTP Request to OCS

Request

```
CmHttpEvent[1-1]::14:10:13.014 Processing HTTP request; uri: /campaigngroups/101?req=Load

{"GSW_AUTO_COMPLETION": 1}
14:10:13.014 Parsed OCS request:                            'GSW_AUTO_COMPLETION'        1
                                'GSW_CM_AttrSessionID'             101
                                'GSW_CM_MessageType'            7
```

Response

```
HTTPProxyClient[]::Desktop request received
                                'GSW_AUTO_COMPLETION'        1
                                'GSW_CM_AttrSessionID'             101
                                'GSW_CM_MessageType'            7
                                'GSW_APPLICATION_ID' 4114
```

Caller ID Sets

Starting with 8.1.524.04, Outbound Contact Server supports the use of unrestricted numbers, or sets, of Caller IDs on Outbound Campaigns. A set of Caller IDs is assigned to a Campaign Group, which will rotate through the next available Caller IDs for each record attempt so that the called party sees a different Caller ID each time a call is received from the Campaign Group. This occurs by attaching the next available Caller ID in the set of Caller IDs to CPN digits for the call. OCS selects either a sequential or random Caller ID when contacting a particular phone number, so the party receiving the call may see a different Caller ID number each time the call is received.

Configuration

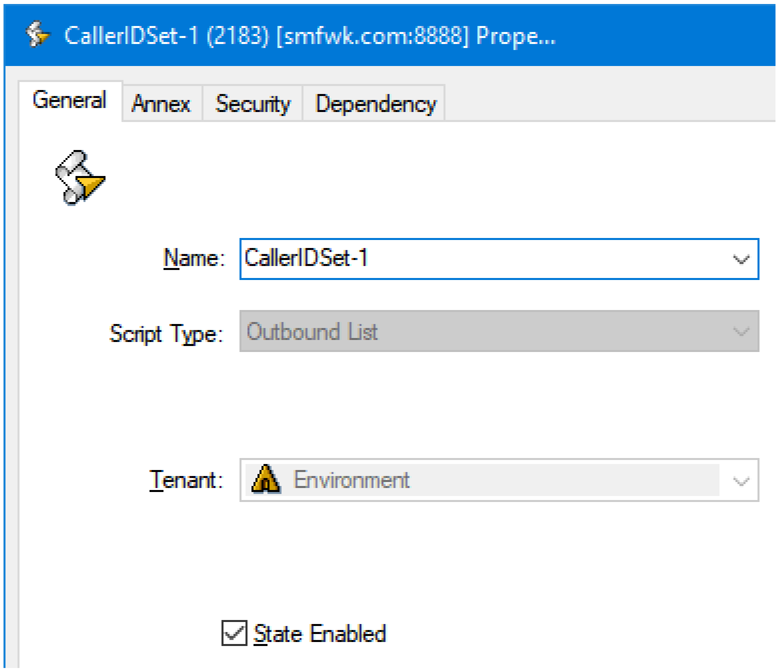
Each Caller ID set is represented by a configuration object with the Script Type **Outbound List**. Caller IDs are stored in the **OCServer** section on the Annex tab of this Outbound List object, and each Caller ID is represented by a String type option. The option name is a unique positive integer value, and the option value is the Caller ID to be used by OCS as CPNDigits.

If a Caller ID set is configured manually, any valid positive integer values are used as option names. Option values (Caller IDs) may repeat within a single Caller ID set.

Configure the Caller ID sets feature by using the following new options:

- caller-id-sets
- caller-id-rotation

Configuration Example



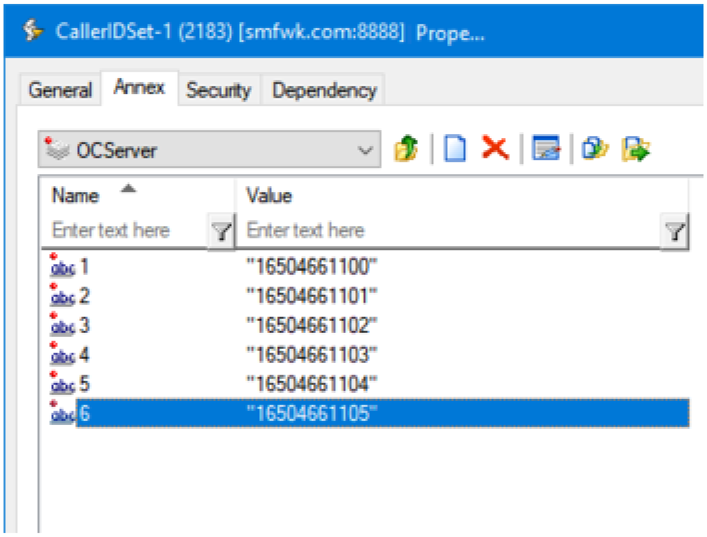
CallerIDSet-1 (2183) [smfwk.com:8888] Prope...

Name: CallerIDSet-1

Script Type: Outbound List

Tenant: Environment

☑ State Enabled



CallerIDSet-1 (2183) [smfwk.com:8888] Prope...

Name	Value
1	"16504661100"
2	"16504661101"
3	"16504661102"
4	"16504661103"
5	"16504661104"
6	"16504661105"

UTF-8 Encoding

Outbound Contact Server supports UTF-8 encoding of mixed calling lists on Linux and Windows platforms, for the following databases:

- PostgreSQL 10 (starting with 8.1.528.21)
- Microsoft SQL Server 2019 (starting with 8.1.528.21)
- Oracle 19c (starting with 8.1.529.07)

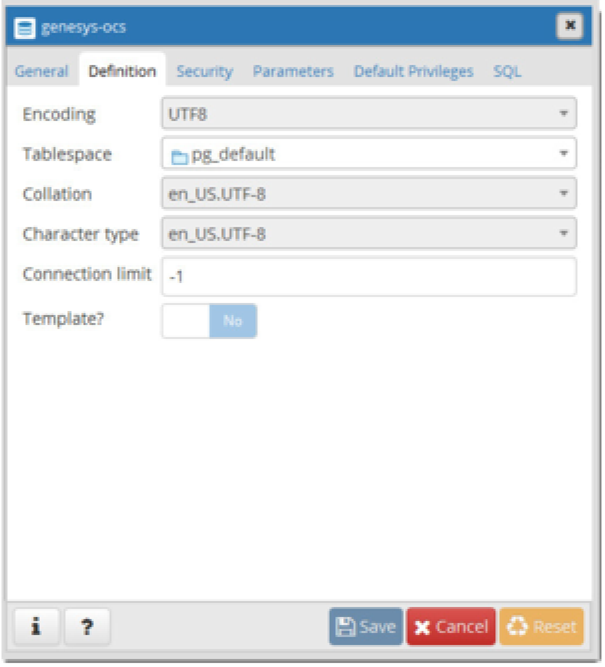
OCS Using PostgreSQL 10

Prerequisites

- PostgreSQL platform: Linux CentOS 7 or later
- PostgreSQL version: 10.14 or later
- DB Server version: 8.5.100.07 or later

Database Settings

- Database encoding: **UTF8**
- Database collation: **en_US.UTF-8**
- Database character type: **en_US.UTF-8**



Encoding: UTF8

Tablespace: pg_default

Collation: en_US.UTF-8

Character type: en_US.UTF-8

Connection limit: -1

Template?: No

OCS Using Microsoft SQL Server 2019

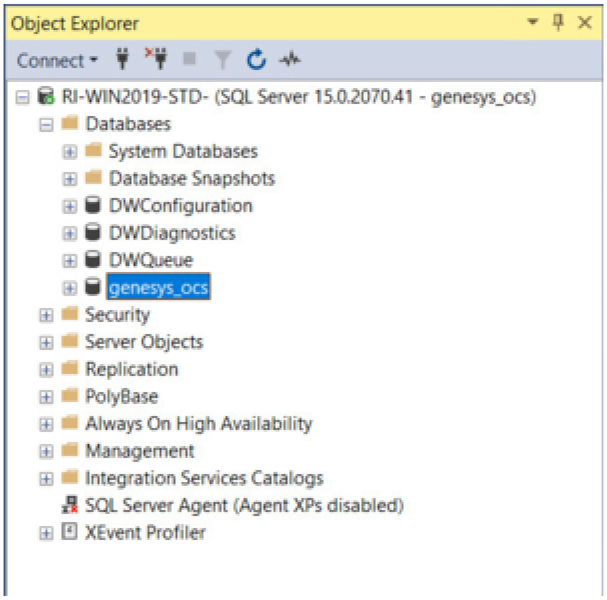
Prerequisites

- Microsoft SQL Server 2019 or later
- Microsoft ODBC Driver 17 for SQL Server or later
- DB Server version 8.5.100.07 or later

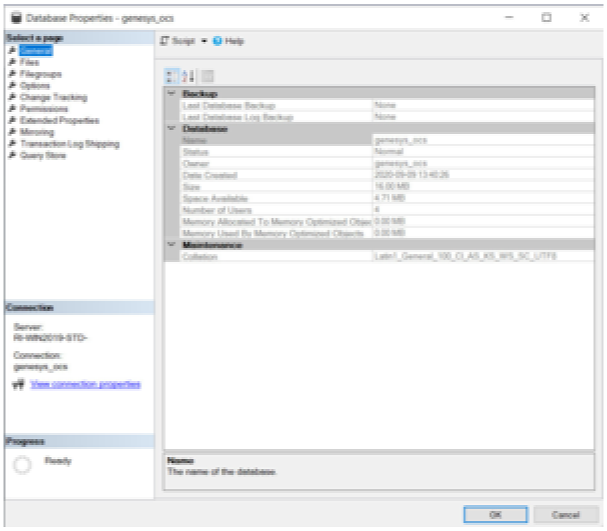
Database Settings

- Database collation: **Latin1_general_100_CI_AS_KS_WS_SC_UTF-8**
  Note: Ensure the **SC_UTF-8** suffix is added.

In these examples, **genesys_ocs** is the database name.



- RI-WIN2019-STD- (SQL Server 15.0.2070.41 - genesys_ocs)
  - Databases
    - System Databases
    - Database Snapshots
    - DWConfiguration
    - DWDiagnostics
    - DWQueue
    - genesys_ocs
  - Security
  - Server Objects
  - Replication
  - PolyBase
  - Always On High Availability
  - Management
  - Integration Services Catalogs
  - SQL Server Agent (Agent XPs disabled)
  - XEvent Profiler



Backup	
Last Database Backup	None
Last Database Log Backup	None
Database	
Name	genesys_ocs
Status	Normal
Owner	genesys_ocs
Date Created	2020-09-09 13:40:26
Size	16.00 MB
Space Available	4.71 MB
Number of Users	4
Memory Allocated To Memory Optimized Objec	0.00 MB
Memory Used By Memory Optimized Objects	0.00 MB
Maintenance	
Collation	Latin1_General_100_CI_AS_KS_WS_SC_UTF8

Connection

Server:
RI-WIN2019-STD-

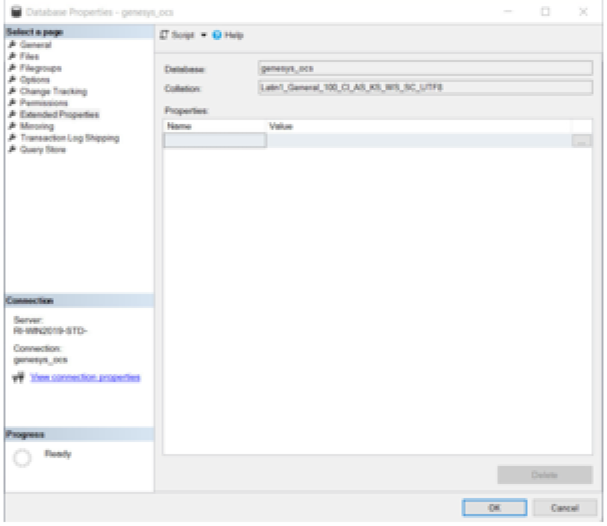
Connection:
genesys_ocs

View connection properties

Progress

Ready

Name
The name of the database.



Database: genesys_ocs

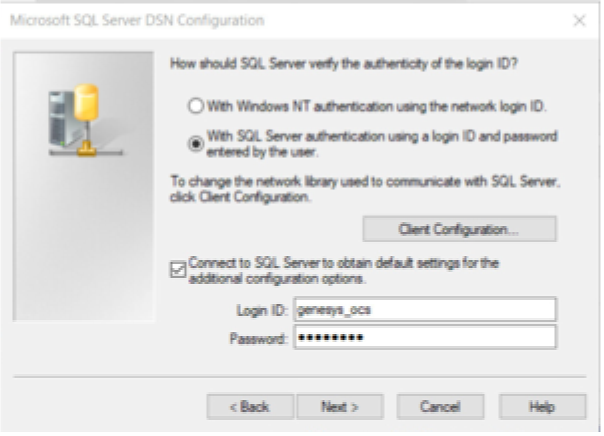
Collation: Latin1_General_100_CI_AS_KS_WS_SC_UTF8

Connection

Server: RI-WIN2019-STD-

Connection: genesys_ocs

ODBC Settings



How should SQL Server verify the authenticity of the login ID?

○ With Windows NT authentication using the network login ID.

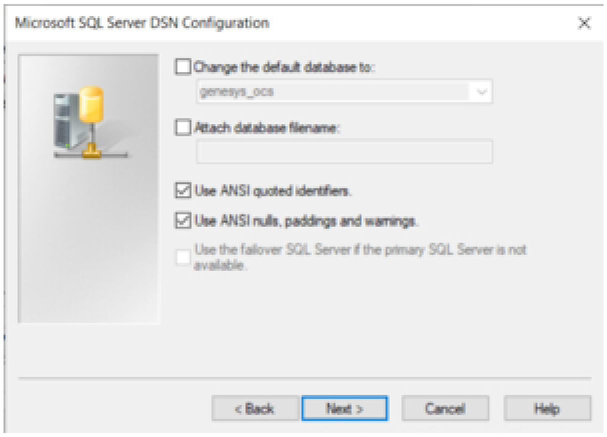
● With SQL Server authentication using a login ID and password entered by the user.

To change the network library used to communicate with SQL Server, click Client Configuration.

☑ Connect to SQL Server to obtain default settings for the additional configuration options.

Login ID: genesys_ocs

Password: •••••••••



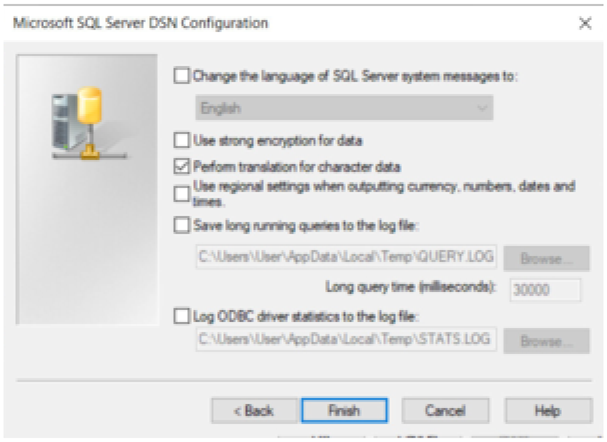
□ Change the default database to: genesys_ocs

□ Attach database filename:

☑ Use ANSI quoted identifiers.

☑ Use ANSI nulls, paddings and warnings.

□ Use the failover SQL Server if the primary SQL Server is not available.



Microsoft SQL Server DSN Configuration

□ Change the language of SQL Server system messages to:

English

□ Use strong encryption for data

☑ Perform translation for character data

□ Use regional settings when outputting currency, numbers, dates and times.

□ Save long running queries to the log file:

C:\Users\User\AppData\Local\Temp\QUERY.LOG

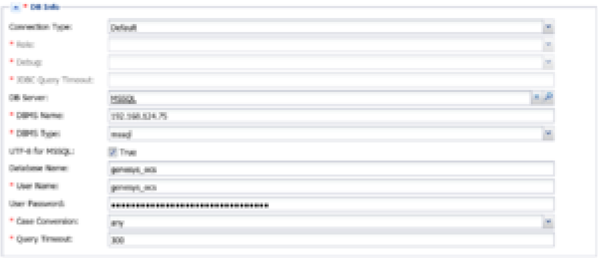
Long query time (milliseconds): 30000

□ Log ODBC driver statistics to the log file:

C:\Users\User\AppData\Local\Temp\STATS.LOG

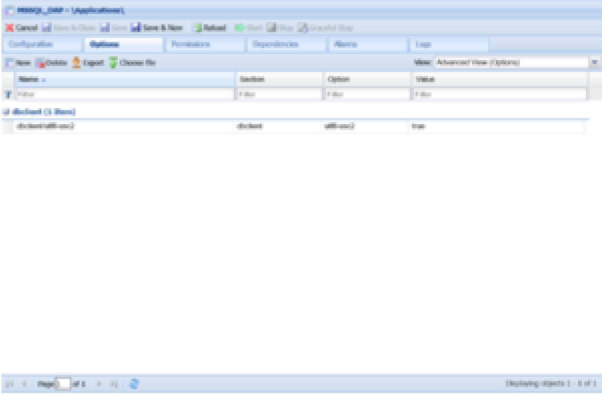
DAP Settings

- **UTF-8 for MSSQL**: select `True`.
- Set the **dbclient/utf8-ucs2** option to `true`.



DB Info

Connection Type:	Default
* Role:	
* Debug:	
* JDBC Query Timeout:	
DB Server:	MSSQL
* DBMS Name:	192.168.124.75
* DBMS Type:	mssql
UTF-8 for MSSQL:	☑ True
Database Name:	genesys_ocs
* User Name:	genesys_ocs
User Password:	••••••••••••••••••••••••••••••••
* Case Conversion:	any
* Query Timeout:	300



MSSQL_DAP - \Applications\

dbclient (1 Item)

Name	Section	Option	Value
dbclient/utf8-ucs2	dbclient	utf8-ucs2	true

OCS Using Oracle 19c

Prerequisites

- Oracle 19c platform
- Oracle instant client version: 12.2
- DB Server version: 8.5.100.07 or later
- OCS version: 8.1.529.07 or later
- **dbserver-less-mode** = `false`

Database Settings

- Database character set: **AL32UTF8** for UTF-8 encoding

Additional scripts to make DB Server working properly

- The following script files must be placed in the **/etc/profile.d** directory on the Linux host where DB Server resides:
  - **oracle_env.csh**
    [+] Example: oracle_env.csh

    ```
    NLS_LANG=.UTF8
    ORACLE_CLIENT_HOME=/usr/lib/oracle/12.2/client64
    ORACLE_HOME=$ORACLE_CLIENT_HOME
    LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_CLIENT_HOME/lib
    PATH=$PATH:$ORACLE_CLIENT_HOME/bin

    export NLS_LANG
    export ORACLE_CLIENT_HOME
    export ORACLE_HOME
    export LD_LIBRARY_PATH
    export PATH
    ```

  - **oracle_env.sh**
    [+] Example: oracle_env.sh

    ```
    NLS_LANG=.UTF8
    ORACLE_CLIENT_HOME=/usr/lib/oracle/12.2/client64
    ORACLE_HOME=$ORACLE_CLIENT_HOME
    LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_CLIENT_HOME/lib
    PATH=$PATH:$ORACLE_CLIENT_HOME/bin

    export NLS_LANG
    export ORACLE_CLIENT_HOME
    export ORACLE_HOME
    export LD_LIBRARY_PATH
    export PATH
    ```

  You must modify these scripts if you use a version of the Oracle instant client other than 12.2.

- The following script file must be placed in the directory where DB Server binaries reside:

multiserver.oracle.sh

[+] Example: multiserver.oracle.sh

```
. /etc/profile.d/oracle_env.sh
./multiserver "$@"
```

The **multiserver.oracle.sh** script must be used in all cases where the multiserver executable generally is used. This script accepts all the same command line parameters as the multiserver executable does.

Configuring Dialogic Boards

These topics contains information on how to configure dialogic boards with Outbound Contact:

- Introduction to Dialogic Boards
- Dialogic Board Setup
- Configuring Dialogic Boards
- Dialogic Board Configuration Structure
- HMP Software
- Dialogic Utilities

Introduction to Dialogic Boards

If you intend to install CPD Server, you must first set up the Dialogic boards. You need to install a hardware card and a software driver to implement Dialogic. Contact the Dialogic vendor for your company or visit the Dialogic support website for more information about Dialogic boards and supported software. Information about Genesys supported hardware and third-party software is available on the Genesys Technical Support website in the documents:

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

The following specific information applies to environments in which CPD Server is used with Dialogic DM3 hardware:

- All recordings are in a PCM/8000Hz/8bps (VOX) format. The `.wav` format is no longer supported.
- DM3 boards do not work with a NEC switch containing a line-side configuration and the `tscall` option value is set to `true` or on.
- DM3 boards are supported in line-side environments.

Note:	Starting with release 7.5, you can also use Dialogic's Host Media Processing (HMP) software instead of a physical hardware card. See HMP Software for more information.

Dialogic Board Setup

Setting Up Dialogic boards

Start

1. Make sure a supported version of Microsoft Windows operating system is installed.
2. Install Dialogic SR 6.
3. Select the required drivers, firmware, and configuration files from the custom package.
4. Configure the Dialogic boards. See Configuring Dialogic Boards for more information.
5. Start Dialogic Configuration Manager (DCM). DCM should automatically detect the board and find valid address/port/interrupt values for the particular computer.
6. Start the Dialogic drivers.
7. Use the Dialogic application provided with the Dialogic board to test each port for off-hook ability, to dial a test telephone number, and then to place the port on-hook.
8. Make sure that each port can complete this test before installing and configuring CPD Server.

Notes:	• When installing Dialogic boards, ensure that the board number set by the switch on the board is unique inside each board type. If you have two T1 boards, the first one must be 0, the next 1, and so on. Make sure the SCBus/CTBus cable connects the boards, if needed. • If there are error messages when starting the Dialogic drivers, the error messages display in the Windows Event Viewer. If there are errors, restart Dialogic Configuration Manager or reinstall the Dialogic boards. If these steps do not solve the problem, contact Dialogic support.

End

Active Switching Matrix Mode Call Flow

`Active Switching Matrix` (ASM) mode is a dialer mode that eliminates various problems of switches that do not support call progress detection or that produces unacceptable transfer delays or force-tone problems. You can configure ASM mode either in Genesys Administrator or in the Outbound Contact Wizard. The switch determines the correct dialing mode based on the presence or absence of a Dialogic board. CPD Server checks the ASM configuration (for ISDN or Melcas) set up in Genesys Administrator to determine what the dialing mode is. The key feature of the ASM-mode call flow is the Engage call, which OCS requests from the Dialogic board. This call engages the agent in a real, established call before the dialed party actually responds and is connected to the call. The sequence of events in this call flow is as follows:

1. The agent sets the phone set to Ready.
2. The switch notifies T-Server that this agent is ready, and T-Server informs OCS by distributing an `EventAgentReady` message.
3. As soon as OCS learns that one or more agents are available at a given queue, it sends an EngageAgent request to CPD Server.
4. CPD Server instructs the Dialogic board to send a special Engage call to the available agent's queue. This is a real call, but is created only on an internal segment.19from a Dialogic port to the ACD Queue.

Note:	The switch considers this an inbound call because the call is coming to an internal DN from a point outside the switch.

5. The Engage call is queued, which generates an `EventCallQueued` message. T-Server distributes this event to OCS, but OCS recognizes it as a special type of call and knows to ignore the event.
6. Since this queue has an available agent, the ACD diverts the Engage call to the agent's desktop. This generates an `EventDiverted` message, but OCS also ignores it, knowing that it refers to an Engage call.
7. The agent answers the Engage call, generating an `EventEstablished` message. At this point the agent is on a real call and hears a tone from the switch indicating that he or she is in an engaged state. The agent encounters a delay of some seconds while OCS and CPD Server generate the second segment of the call to a dialed number from the calling list.
8. In `Progressive` dialing mode, OCS waits until it knows the agent is Engaged before asking CPD Server to make an outbound call. In `Predictive` dialing mode, OCS is making outbound calls according to its predictive algorithm without regard to availability of agents.
9. CPD Server informs the Dialogic board to make a call to the customer number supplied by OCS.

Note:	This call is dialed outside the switch, so that neither the switch nor T-Server sees it. Therefore, no T-Server events are associated with this call.

10. If the call reaches a live customer, CPD Server attaches the account information or other data to the Engage call.

Note:	The first agents engaged are the first to be distributed.

11. T-Server delivers this data to the Engaged agent's desktop as a screen pop. Simultaneously, CPD Server connects the internal and external segments of the call, and the call is established between the agent and the customer. T-Server considers the call type for the call to be `Inbound`, whereas OCS sees the call type as `Outbound` because it was generated by CPD Server.
12. CPD Server informs OCS of the call result.

From this point, the call is handled according to the campaign requirements similar to `transfer`-mode.

Port Requirements for ASM Mode

CPD Server uses two ports in ASM-mode to connect the agent to the customer. CPD Server releases both ports at the end of the call.

When CPD Server is not in ASM-mode, it uses only one port to make a call and to transfer it to an agent. CPD Server releases the port when the call is transferred.

OCS reports the number of ports available. If for some reason, OCS does not track the number of ports available, CPD Server sends a `No Free Port` message to OCS.

Configuring Dialogic Boards

Outbound Contact Configuration Wizard puts the information about installed Dialogic boards and resources in CME. You can use Genesys Administrator to change the configurations, as needed. See Dialogic Board Configuration Structure.

Frequently Used Configurations

The following table lists frequently used configurations for Dialogic boards. The driver names are provided for a single board configuration. For multiple boards, the corresponding board numbers must be sequentially numbered.

Dialogic Board Configurations

Board	Type	Configuration
D/160JCT-LS	Virtual Board	4 LSI (dxxxB1...dxxxB4)
	Virtual Device	4 channels on each board (dxxxBxC1...dxxxBxC4)
	Voice Virtual Board	4 voice (dxxxB1...dxxxB4)
	Voice Virtual Device	4 channels on each board (dxxxBxC1...dxxxBxC4)
D/41JCT	Virtual Board	4 LSI (dxxxB1...dxxxB4)
	Virtual Device	4 channels on each board (dxxxBxC1...dxxxBxC4)
	Voice Virtual Board	4 voice (dxxxB1...dxxxB4)
	Voice Virtual Device	4 channels on each board (dxxxBxC1...dxxxBxC4)
D/480SC-2T1	Virtual Board	2 DTI boards (dtiB1, dtiB2)
	Virtual Device	24 channels (23 if) on each board (dtiBxT1...dtiBxT24)
	Voice Virtual Board	12 voice boards (dxxxB1...dxxxB12)
	Voice Virtual Device	4 channels on each board (dxxxBxC1...dxxxBxC4)
D/300JCT-2E1	Virtual Board	2 DTI boards (dtiB1, dtiB2)
	Virtual Device	30 channels on each (dtiBxT1...dtiBxT30)
	Voice Virtual Board	8 voice boards (dxxxB1...dxxxB8)
	Voice Virtual Device	4 channels on boards 1 to 7, 2 channels on board 8
DM/V300	DM3 board with CPD functionality	30 channels

DM/V600-2E1	DM3 board with CPD functionality	60 channels
DM/V600BTEPEQ	DM3 board with CPD functionality	60 channels
DM/V600BTEPW	DM3 board with CPD functionality	60 channels
DM/V960-4T1	DM3 board with CPD functionality	96 channels
DM/V1200-4E1	DM3 board with CPD functionality	120 channels
DM/V1200BTEPEQ	DM3 board with CPD functionality	120 channels
DM/V1200TEPW	DM3 board with CPD functionality	120 channels

Suggested DM/V Dialogic Board Configuration

When configuring a `DM/V`-type board, configure the Dialogic channels similar to the way in which you configure any CPD Server that is operating in `Transfer` mode and that interfaces with a lineside T1/E1. The following two sections provide more information about configuring specific models of this board type.

Note:	The `line-type` option must contain a value of `line-side-dm3`.

Configuring the DMV480A-2T1 Board

Start If you are using a `DMV480A-2T1` board, do the following:

1. Create a new file called `pdk.cfg` that contains the following statement:
   ```
   Board 0 fcdfile ml2_dsa_cas.fcd pcdfile ml2_dsa_cas.pcd variant pdk_sw_e1_ntmd_io.cdp
   ```
2. Move this file to `C:\Program Files\Dialogic\cfg`.
3. Open a command prompt and type in the following path:
   ```
   C:\Program Files\Dialogic\bin
   ```
4. Execute the following command:
   ```
   PDKManagerRegSetup add
   ```
   You should receive the following message:
   ```
   PDKManager key insertion succeeded.
   ```
5. Open Dialogic Configuration Manager and right-click your Dialogic board type.
6. Select `Configure device`.
7. On the `Misc` tab, select:
   - `ml2_dsa_cas.fcd` as the FCD file.
   - `ml2_dsa_cas.pcd` as the PCD file.
8. Restart the Dialogic System Service.
9. Start CPD Server.

Note:	If you are using a `DMV960A-4T1` board, replace `ml2_dsa_cas` with `ml2_qsa_cas`.

End

Configuring DMV480-4T1 Board

Start If you are using a `DMV480-4T1` board, do the following:

1. Create a new file called `pdk.cfg` that contains the following statement:
   `Board 0 fcdfile ml1_4x2_cas.fcd pcdfile ml1_4x2_cas.pcd variant pdk_sw_e1_ntmd_io.cdp`
2. Move this file to `C:\Program Files\Dialogic\cfg`
3. Open a command prompt and type in the following path:
   `C:\Program Files\Dialogic\bin`
4. Execute the following command:
   `PDKManagerRegSetup add`
   You should receive the following message:
   `PDKManager key insertion succeeded.`
5. Open Dialogic Configuration Manager and right-click your Dialogic board type.
6. Select `Configure device`.
7. On the `Misc` tab, select:
   - `ml1_4x2_cas.fcd` as the FCD file.
   - `ml1_4x2_cas.pcd` as the PCD file.
8. Restart the Dialogic System Service.
9. Start CPD Server.

End

Dialogic Board Configuration Structure

Note:	CPD Server must be stopped when making a change in the Dialogic Board configuration structure. To make a change in the structure, stop CPD Server, make the change, and restart CPD Server only after all changes have been completed.

In Outbound Contact, the configuration process for Dialogic boards has been simplified.

The structure of Dialogic board configuration is represented by a folder tree under the Switch/DN view in Genesys Administrator. The folder tree consists of the following folders:

<location>—This folder is taken from the location option (`default` section) and serves as the board root folder. When a board is deleted, Wizard deletes this folder and everything under it. The current Wizard implementation does not allow deletion of only part of a board; the whole board is deleted.

regular—This folder contains channel-DN configuration for dialing outbound calls.

recording — optional — folder contains channel-DN configuration for CPD recording on the regular DNs. Recording is used to tune up Voice/AM detection and should not be used in regular work since it takes half of the voice resources. This folder is created but not configured by the Wizard.

engaging—This folder (optional) contains channel-DN configuration assigned to engage agents when working in ASM mode.

dxxxB<n>—This root folder for an analog virtual (logical) board. It contains voice resources (channels). It may have one or more voice resources.

dxxxB<n>C<m>—Voice resource (channel). In the configuration it links a dialogic voice resource and a DN on the switch.

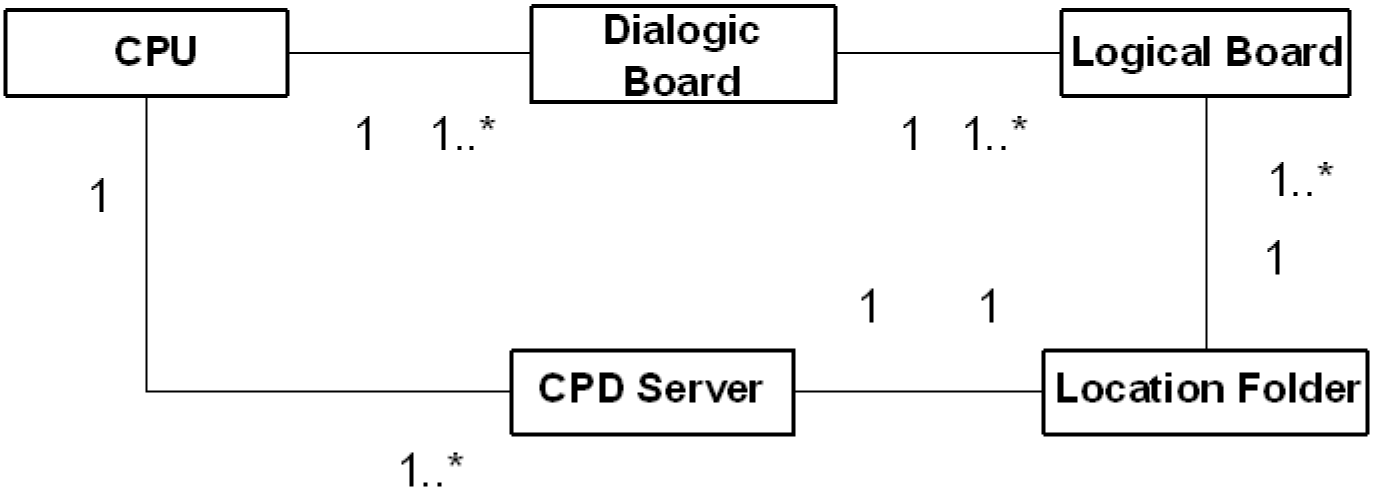
dtiB<n>—The root channel for a network (T1) virtual board. It contains one or more digital resources.

dtiB<n>T<m>—Digital (network) resource. In the configuration it links a Dialogic digital (network) resource and a DN on the switch.

Note:	Non-analog line type folders representing dxxx channels do not have to include real DNs. CPD Server works without DNs in dxxx folders and performs call progress detection on non-analog lines. Thus, multiple licenses are not required to allow for two DNs per channel and a subsequent increase in dialing ports.

Configuration Logical Structure

A logical structure describes a single configuration of many boards. Every logical board in a configuration is given a unique logical board number based on the order in which the boards are configured. See Configuring Dialogic Boards.



Logical Structure

Genesys recommends a 1-to-1 relationship between the CPU, CPD Server, and the `Location` folder.

Note:	The number 1 in the preceding diagram shows a 1-to-1 representation, and 1...* shows a 1-to-more-than-1 representation.

The following describes what each item in Configuring Dialogic Boards represents:

- *Box*—Represents a CPU (central processing unit of a computer running on a Windows operating system), where the Dialogic boards and CPD Servers are installed.
- *Dialogic Board*—Represents a hardware board installed into the CPU. The CPU may have more than one board installed.
- *Logical Board*—A logical part of the hardware board representing four analog or digital resources (for example, dxxxB1 and all the hierarchy belonging to it). One hardware board consists of several logical boards. One logical board belongs to only one hardware board.
- *CPD Server*—Represents a piece of software installed on the box. It can be one or more different CPD Server installations.
- *Location Folder*—Represents a set of logical boards, and one CPD Server. A Location Folder is allowed to have logical boards connected to different hardware boards (see example below). Genesys recommends that you set up a 1-to-1 association between CPD Server applications and the Locations folder.

Considerations

- The Location folder may have logical boards belonging to hardware boards installed in one, and only one, computer running on the Windows operating system.
- If you have two or more different Location folders, they must not have the same logical board configured.

- If you have more than one hardware board installed in the same computer, then the first board (minimal hardware ID) has 1—n logical boards numbering (dxxxB1—dxxxBn), the second board (next ID) has n+1—m (m > n+1) logical boards numbering (dxxxB<n+1>—dxxxBm), and so on. If a logical board is removed from a configuration, then all other logical boards in the configuration must be renumbered to preserve a logical numbering system.

CPD Server Dialogic Hardware Setup in Genesys Administrator

Genesys Administrator simplifies the configuration of the Dialogic hardware in the Configuration Database. When you configure the Dialogic hardware by going to `Provisioning > Outbound Contact > Dialogic Boards`, Genesys Administrator creates all of the folders and subfolders, based on the selected Dialogic Board and line type. For more information, see *Framework Genesys Administrator Help.*

You can also manually configure provides the Dialogic hardware, per the guidelines the follow.

Manually Setting Up Your Dialogic Hardware

Start

1. In either Genesys Administrator or Configuration Manager, create the `Location` folder in the Switch DN folder.
2. Create three subfolders that will contain board resource subfolders with the names:
   - `engaging` (should contain only DM3 (Melcas) resources of Dialogic board)
   - `recording` (should contain only voice resources)
   - `regular` (can contain any type of Dialogic resources)

Note:	These board resource subfolder names should be the same as the virtual board names used for the Dialogic hardware that is installed.

3. In each of these three subfolders, create another group of board resource subfolders named exactly as the individual channels of the corresponding virtual board.
4. In each channel subfolder, include the real DN (Extension) or virtual DN (Call Processing Port) that corresponds to the DN assigned to the board channel.
   - For voice-processing ports and ISDN or Melcas virtual channels, you must assign unique numbers to the `Call-Processing port` DN type.
   - For LSI and T1/E1 ports, you should assign the `Extension` DN type and real channel numbers that are recognizable by the switch.

Note:	The `ACD Position` DN type is not a valid value for CPD Server DNs.

5. The `Register` check box should be cleared (unchecked) for CPP and selected (checked) for ACD Extension DNs.

End

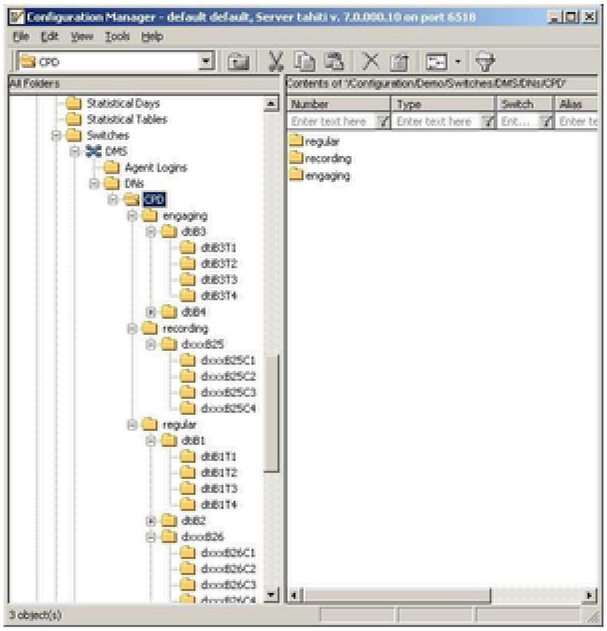
Dialogic Board Configuration

The following includes configuration information you can use to configure ASM (ISDN and Melcas), Analog, or T1/E1 line-side protocols.

Note:	The figures in this topic reflect the configuration as it would appear in Configuration Manager.

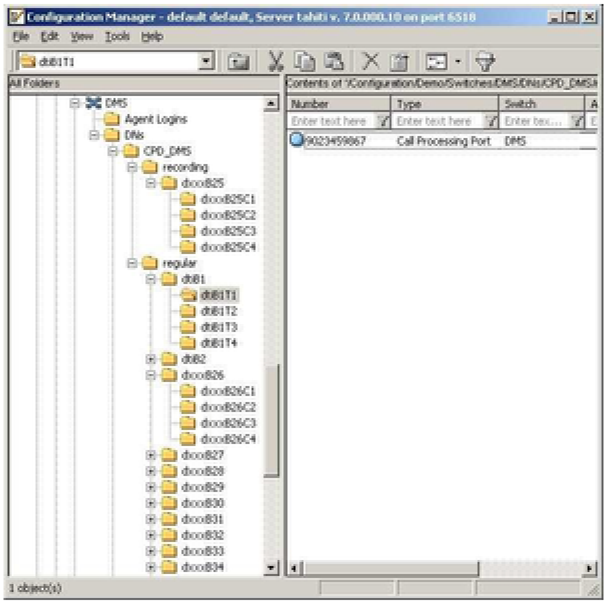
ASM Mode

An ASM configuration must have at least one digital board and one voice board in the regular folder. The following figure shows the directory structure for an ISDN or Melcas board.



ASM Mode Directory Structure

In the ISDN and Melcas configurations, network resources are associated with DNs of Call Processing Port type. The following figure shows a call processing port DN in an ASM configuration.



Contents of '/Configuration/Demo/Switches/DMS/DNs/CPD_DMS/

Number	Type	Switch
9023459867	Call Processing Port	DMS

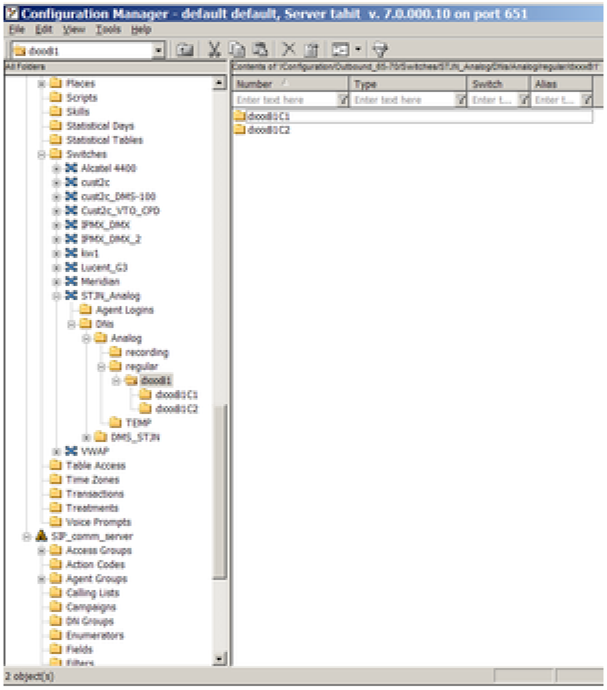
ASM Call Processing Port DN

Analog

An Analog board configuration must have the following folders and subfolders in this type of directory structure:

- A `regular` folder
  - At least one dxxxB<n> folder within the regular folder
    - At least one dxxxB<n>C<n> subfolder within the dxxxB<n> folder.

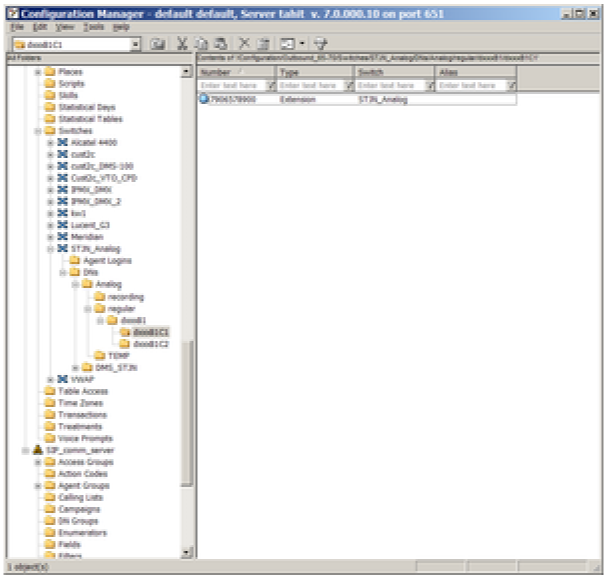
The following figure shows the directory structure for an Analog configuration.



Number	Type	Switch	Alias
dxxx81C1			
dxxx81C2			

Analog Directory Structure

The following figure shows an Extension DN in an Analog configuration.



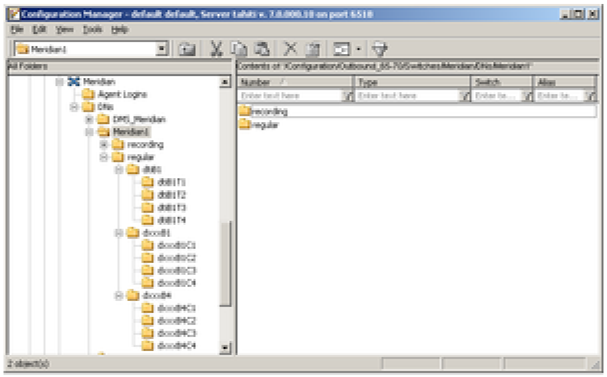
Number	Type	Switch	Alias
7906578900	Extension	STJN_Analog	

Extension DN in Analog Configuration

For more information, see CPD Server Dialogic Hardware Setup in Genesys Administrator.

T1/E1 Line-Side

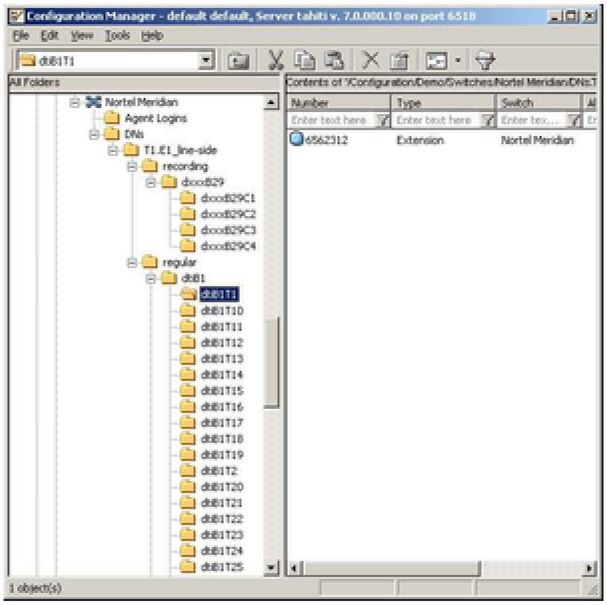
A T1/E1 line-side configuration must have at least one digital and one voice logical board configured in the `regular` folder. See the following figure.



Number	Type	Switch	Alias
recording			
regular			

T1/E1 Line-Side Directory Structure

In the line-side configuration, each network resource must have one, and only one, DN associated with it. Network resources are associated with DNs of type `Extension`. See the following figure.



Number	Type	Switch
6562312	Extension	Nortel Meridian

Extension DN in T1/E1 (Line Side) Configuration

Dialogic Board Configuration Examples

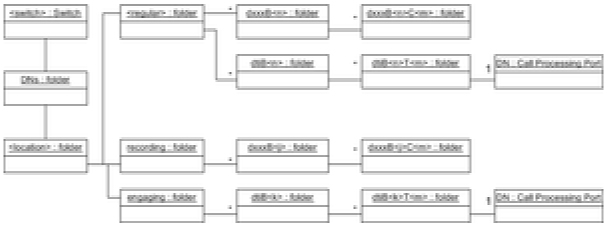
The following figures show sample configurations for ISDN or Melcas (in ASM), Analog, or T1/E1 protocols.

ASM Mode (ISDN or Melcas)

An ASM configuration must have at least one digital board and one voice board in the regular folder.

In the an ASM configuration, network resources are associated with DNs of Call Processing Port type.

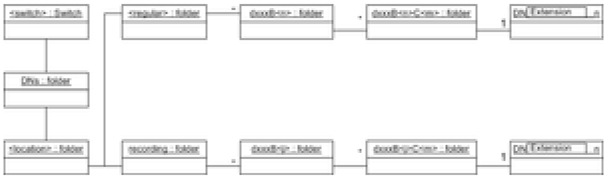
The following figure shows a configuration example for an ISDN or Melcas protocol.



Board Configuration Example for ISDN or Melcas Protocol

Analog Board Configuration

The following figure shows a configuration example for an Analog board.



Configuration Example for an Analog Board

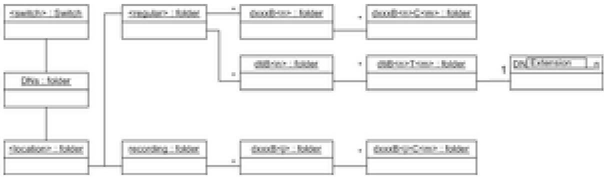
Note:	In the 7.x release, the only DN type for an Analog board or a T1/E1 board is `Extension`, not `ACD Position`.

T1/E1 Line-Side Configuration

A line-side configuration must have at least one digital and one voice logical board configured in the `regular` folder.

In the line-side configuration, network resources are associated with DNs of `Extension` type.

The following figure shows the configuration example for a T1/E1 Board.



Configuration Example for a T1/E1 Board

Note:	Each network resource may have one, and only one, DN associated with it.

DNs Inside Dialogic Channels Folders

For more information about the Dialogic boards and their channels, see the following documents available on the Genesys Technical Support website:

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

New Dialogic Board Configuration

Genesys Administrator simplifies the configuration of the Dialogic hardware in the Configuration Database. When you configure the Dialogic hardware by going to `Provisioning > Outbound Contact > Dialogic Boards`, Genesys Administrator creates all of the folders and subfolders, based on the selected Dialogic Board and line type. For more information, see *Framework Genesys Administrator Help.*

You can also configure a new Dialogic board manually, by following the procedure provided here.

Manually Configuring a New Dialogic Board

Start

1. In Genesys Administrator, create a `<location>` folder under the Switch object.
2. Under the `<location>` folder, create a folder named `recording`.
3. Again in the `<location>` folder, create a folder named `engaging`.
4. Create a folder named `regular`. For a description of this folder, see CPD Server Dialogic Hardware Setup in Genesys Administrator.
5. Configure the `regular` folder.

Note:	A DN can belong to only one channel across the switch.

End

HMP Software

CPD Server 7.6 and higher supports Dialogic HMP software in both `transfer`-mode and ASM-mode scenarios. HMP is enabled by the `line-type` option.

Note:	IP version 6 addressing schema (IPv6) is not supported by Dialogic HMP. CPD Server 8.1.2 will run only with Dialogic HMP software SU 328 and higher.

The following are the differences between standard and HMP deployment:

- HMP is simply another connection protocol used in CPD Server.
- An HMP board is represented in Genesys Administrator as an `iptB1` folder under the `regular` folder. There are no required changes for the names or locations of the voice resources.
- An HMP channel is represented as `iptB1TX` where X is a channel number. The DN type is `Call Processing Port`.

There is no restriction on the number of HMP channels for CPD Server. However, the maximum number of HMP ports that HMP can support on each host depends on various factors, including host performance. Dialogic provides a calculator that enables you to estimate the number of HMP ports that single host can support. For the recommended hardware requirements for a specific number of channels, refer to the Dialogic documentation.

- HMP `transfer`-mode requires the following amounts of resources:
  - The same amount of RTP/IPCC resources as standard `transfer`-mode if OCS requests one-step transfer (that is, the OCS `call_transfer_type` option is set to `one_step`.
  - The same amount of voice resources as the standard `transfer`-mode.
- HMP ASM-mode requires the same amount of resources as regular ASM-mode.
- Unlike standard `Transfer` mode, the `UserData` in HMP `Transfer` mode is attached when the call enters the agent ACD Queue or Routing Point. This enables CPD Server to run in "hybrid" mode when a standard T-Server target is used with a media gateway.
- SIP Server is supported without needing to use a media gateway. However, only the SIP protocol without registration is supported.
- Each CPD port will still use one Genesys license. However, three types of Dialogic HMP licenses are required for each successful outbound call:
  - One license for Voice (`dxxx`)
  - Two licenses for RTP_G_711
  - Two licenses for IP_Call_Control (`ipt`)

Contact your Dialogic representative for more information about HMP licensing.

Note:	It is recommended that you complete the media processing channel configuration in the Outbound Contact 7.5 Configuration Wizard or Genesys Administrator. Only one instance of CPD Server using HMP software can be executed on a host computer.

HMP and Multiple Network Interface Cards

If CPD Server with HMP is running on a host with multiple network interface cards (NIC), HMP software by default uses the first NIC from the list it gets from the operating system. Therefore, the IP address of the first NIC is used in SIP message headers (for example, the `Via` header). This is not always what is required. HMP does not allow the configuration of the IP address to be used. However, HMP does allow you to set the IP address to be used via an API call by the user agent (in this case, CPD Server).

CPD Server enables you to optionally assign a specific local IP address, to be used by HMP instead of the default IP address defined in the Dialogic DCM (as required for deployment on a host with multiple network interfaces). Use the configuration option `sip-local-address` to define the local IP address.

Configuration

To enable CPD Server to work with HMP software, you must complete the following configuration activity, starting with the procedure Configuring CPD Server for HMP Software.

Configuring CPD Server for HMP Software

Start In the CPD Server application in Genesys Administrator, configure the following two options:

1. Set the `line-type` option to `sip-hmp` for `Transfer` mode, or `sip-hmp-asm` for ASM mode.
2. Set the `sip-proxy` option in the `hmp` section. This option contains the IP address of the SIP Server that HMP will use for dialing.

End

Configuring the SIP Switch for HMP Software

The configuration of a Dialogic folder in Genesys Administrator is still required when running HMP software. The subdirectories are still named `regular`, `engaging`, and `recording`.

Note:	The `engaging` folder is for ASM mode only.

Genesys Administrator simplifies the configuration of the Dialogic hardware in the Configuration Database. When you configure the Dialogic hardware by going to `Provisioning > Outbound`

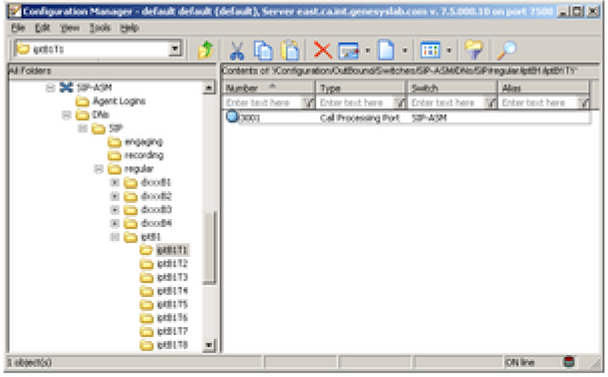
`Contact > Dialogic Boards`, Genesys Administrator creates all of the folders and subfolders, based on the selected Dialogic Board and line type. For more information, see *Framework Genesys Administrator Help.*

You can also manually configure the Dialogic hardware, per the procedure below.

Start

Configure the subfolders under the main `regular, engaging` and `recording` folders either manually or using the Configuration Wizard.

1. Configure the HMP board as an `iptB1` folder under either the `regular` or the `engaging` folder. No changes for the names or locations of the voice resources (dxxx) are required.
2. Configure the HMP channels as `iptB1TX` where `X` is a channel number. The DN type must be set to `Call Processing Port`.
   The maximum number of HMP ports that HMP can support on each host depends on various factors, including host performance. Dialogic provides a calculator that enables you to estimate the number of HMP ports that single host can support.
   The following figure provides an example of the configuration in Configuration Manager.



Configuration Manager - default default (default), Server east.ca.int.genesyslab.com v. 7.5.000.10 on port 7500
File Edit View Tools Help
iptB1T1
All Folders
SIP-ASM
Agent Logins
DNs
SIP
engaging
recording
regular
dxxxB1
dxxxB2
dxxxB3
dxxxB4
iptB1
iptB1T1
iptB1T2
iptB1T3
iptB1T4
iptB1T5
iptB1T6
iptB1T7
iptB1T8
Contents of '\Configuration\Outbound\Switches\SIP-ASM\DNs\SIP\regular\iptB1\iptB1T1'
1 object(s)
ON line

Number	Type	Switch	Alias
3001	Call Processing Port	SIP-ASM	

Configuring HMP Channels

3. Configure the `ipt` DNs as type `Call Processing Ports`. The `Options` tab of these ports must contain a `TServer` section with the following entries:
   - `contact`, with a value of `<DN@IP Address>.` This specifies the host running the HMP software.
   - `refer-enabled`, with a value of `false.`

   Refer to the *Framework SIP Server Deployment Guide* for more information about these options.

Note:	HMP transfer mode requires two times as many RTP/IPCC resources as hardware transfer mode, but the same number of voice resources as hardware transfer mode. HMP ASM mode requires the same number of resources as hardware ASM mode. Unlike hardware transfer mode, the `UserData` in HMP transfer mode is attached when the call enters the agent queue or route point. This allows CPD Server to run in environments where a conventional T-Server (not SIP Server) with a media gateway is used.

End

Configuring Host Media Processing (HMP) Software

Start Perform the following configuration instructions to use HMP software:

1. The fcd and pcd file names are the license file name with an fcd or pcd extension. They are stored in the ...\HMP\data directory on the host where HMP is installed. There are 4 license files with the following extensions:
   - lic, which is the actual license file provided by Dialogic.
   - pcd, which is the pcd file that will be used by HMP, as generated from the .lic file.
   - fcd, which is the fcd file that will be used by HMP, as generated from the .lic file.
   - config, which is the configuration file that will be used by HMP.
2. Generate these files by performing the following steps:
   1. Run the Dialogic NetStructure Host Media Processing (HMP) License Manager from the Windows Start menu.
   2. Browse to the desired license file.
   3. Click the Activate License file to generate the fcd, pcd and config files.
3. Configure the HMP software to use the newly generated files by performing the following steps:
   1. Start Dialogic Configuration Manager (DCM).
   2. Stop the HMP software (if not already stopped).
   3. Right-click the HMP entry in DCM and select Restore Device Defaults.
   4. Select the pcd file in the Assign Firmware File window that is associated with the license file created by Dialogic's License Manager.
   5. Start the system.

Note:	Only the SIP protocol without registration (registrar) is currently supported.

End

Configuring Caller ID on a Per-Record Basis using HMP in Transfer Mode

Starting in release 8.1.1, CPD Server can use the TMakeCall functionality of SIP Server to initiate dialing. This allows CPD Server to utilize standard TMakeCall functionality, including the ability to specify Caller ID on a per-call basis. Caller ID information for each outbound call is received by CPD Server from OCS. OCS needs to be specifically configured to provide this information to CPD Server. This functionality is available only if the CPD Server option tscall is set to true.

Start In Genesys Administrator, configure the following options in the appropriate configuration objects:

1. In the CPD Server application, in the general section, set the tscall option to true/yes.
2. In the OC Server application, at the Campaign Group, Application, or individual record level using SCXML treatments, set the CPNDigits option to the applicable set of digits.

3. In all Call Processing Port DNs in HMP channels, in the `TServer` section, set the `make-call-rfc3725-flow` option to 1, and set the `refer-enabled` option to `false`. These options are in addition to the required options for the HMP ports.

End

Dialogic Utilities

These utilities can help you to resolve problems that might occur when you are using Dialogic boards.

ISDIAG Dialogic Utility

The ISDIAG Dialogic Utility is part of the Dialogic software package. Use this utility to test lines, for example, when working with different protocols. You can also set call parameters using this utility.

The utility is a console application. To run the utility, follow the Help menu using the command prompts. If the system is not working, use this utility to make a call, if you can; then, if the system is still not working, check your hardware settings. If you cannot make a call using the utility, contact Genesys Technical Support.

Also, try comparing the Trace log against the Dialogic board Component log to check for possible problems.

PRI-Trace Utility

The PRI-Trace utility is part of the Dialogic software package. Use it to see all the parameters sent through the D-channel. The parameters display in binary log files; the PRI-Trace utility turns the logs into readable text.

Dialogic Configuration Manager

Dialogic Configuration Manager (DCM) is part of the Dialogic software package.

Make sure that you use this utility when you are configuring Dialogic boards for your system. To access this utility from the `Start` menu, select `Programs > Dialogic System Software > Dialogic Configuration Manager--DCM`.

Starting and Stopping Procedures

These topics explain how to start and stop Outbound Contact with the Solution Control Interface (SCI), and how to start and stop Outbound Contact Server and CPD Server manually:

- Prestart Information
- Using Solution Control Interface
- Starting Manually
- Stopping Manually
- Connection and Reconnection

Prestart Information

Before starting Outbound Contact, the following Framework components must be running:

- Configuration Server
- DB Server (for Configuration Server)

The next two components are optional; however, if you want to start Outbound Contact Server with the Solution Control Interface, these must be running:

- Solution Control Server
- Solution Control Interface

Note:	If you will be using the CPD Server, install all Dialogic components on the same computer as CPD Server before starting CPD Server. You can also start and stop Outbound Contact using Genesys Administrator. For more information, see *Framework Genesys Administrator Help*.

Reporting components can be started anytime after the Framework components are started.

Using Solution Control Interface

Outbound Contact components can be started and stopped through or Solution Control Interface (SCI) or Genesys Administrator according to the specified starting and stopping sequence in the Outbound Solution object.

Note:	Instructions in the section are based on starting using SCI.

Starting Outbound Contact

By starting the Outbound Contact solution, you start the components that make up that solution.

Starting Outbound Contact Solution/Components with SCI

Start

1. Start SCI either by clicking `Sci.exe` or selecting `Solution Control Interface` from the Start menu.
2. Go to the Solutions view.
3. Select the solution in the `List` pane.
4. Click `Start` or select `Start` from the shortcut menu. The solution's status changes from `Stopped` to `Started` after all solution components have reported that their status is `Running`.

Note:	Because many components are shared by several solutions, some Outbound Contact components may have the status `Started` before the solution is started.

End

Stopping Outbound Contact Solution/Components with SCI

There are three ways to stop all servers in Outbound Contact using SCI:

- In the `Solutions` folder, select the Outbound Contact to stop.
- Select `Stop` from the `Action` menu.
- From the toolbar, click the `Stop Solution/Application` button.

You can also stop components individually. See the next sections.

Note:	Stopping the solution does not stop OCM; You must

	stop OCM manually (see Starting and Logging In to Outbound Contact Manager). Warning: If you are using Genesys Administrator, because it is a web application, you log out and close the browser window.

Stopping CPD Server

Start To stop CPD Server using SCI:

1. Select the CPD Server component in the `Applications` folder.
2. Select `Stop` from the `Action` menu.

End

Stopping CPD Proxy Server

Start To stop CPD Proxy Server using SCI:

1. Select the CPD Proxy Server component in the `Applications` folder.
2. Select `Stop` from the `Action` menu.

End

Stopping OCS

Start To stop OCS in SCI:

1. Select the OCS component in the `Applications` folder.
2. Select `Stop` from the `Action` menu.

For more information, see *Framework Solution Control Interface Help*. To view the Help file, open SCI, and then launch the Help file.

End

Starting Manually

To start Outbound Contact manually, components must be started in a specific order. For information about starting the Framework components DB Server, Configuration Server, T-Server, and Stat Server, see the Framework Deployment Guide.

For information about starting the Reporting components ICON and CC Pulse+, see the *Interaction Concentrator Deployment Guide* and Reporting 8.0 Deployment Guide respectively. Start components in the following order:

1. DB Server for Configuration Server
2. Configuration Server
3. T-Server
4. All other DB Servers (for Calling Lists)
5. Stat Server
6. ICON components for historical reporting
7. CPD Server (if needed)/CPD Proxy Server (if used)
8. Outbound Contact Server (OCS)
9. CC Pulse+ (for real-time and historical reporting)
10. Outbound Contact Manager (OCM) or Genesys Administrator

Starting Outbound Contact Server

Start

You can start Outbound Contact Server (OCS) manually in three ways:

- From the `Start` menu.
- Open an MS-DOS command prompt and type the directory where Outbound Contact Server is installed, for example, `\Gcti\outbound contact\`, and use a command line. When starting OCS from a command line, use the following common command-line parameters:
  `-host` The name of the host on which Configuration Server is running
  `-port` The Configuration Server port
  -app The exact name of the Application as configured in the Configuration Database
  `-l` The full path to the license file or host and port of the License
  Manager in the <License Manager port>@<License Manager host> format.
  For example:
  `cm_server.exe -host ConfigServer1 -port 2020 -app OCServer1 -l C:\flexlm\license.dat`

Note:	If you are implementing the client-side port security feature, specify the additional arguments that are described in Client-Side Port Security Configuration.

- Run batch files, `startServer.bat`.
  The `startServer.bat` files are created automatically during installation and located at the same directories as executable files. Basically, these files include some comments and the same options as are used when starting the application from the command line.
  The following is an example of a batch file for OCS:

```
@echo off
rem ------------------------------------------------------------------
rem Copyright (C) 2009 Genesys Telecommunications Laboratories, Inc.
rem
rem startServer.bat file for Outbound Contact Server, version 8.1.000.05
rem ------------------------------------------------------------------
@TITLE Outbound Contact Server v. 8.1.000.05: Application OCServer
cm_server.exe -host lab1 -port 7070 -app OCServer -l 7260@lab1
```

End

Starting CPD Server

Start You can start CPD Server manually in three ways:

- From the `Start` menu.
- Open an MS-DOS command prompt and type the directory where CPD Server is installed--for example, `\Gcti\cpd_server\`—and use a command line.
  When starting CPD Server from a command line, use the following common command-line parameters:
  `-host` The name of the host on which Configuration Server is running
  `-port` The Configuration Server port
  `-app` The exact name of the application as configured in the Configuration Database
  `-l` The full path to the license file or host and port of the License
  Manager in the <License Manager port>@<License Manager host> format.
  For example:
  `cpdserver.exe -host ConfigServer1 -port 2020 -app CPDServer1 -l C:\flexlm\license.dat`.

Note:	If you are implementing the client-s—Client-Side Port Security Configuration]].

- Run batch files, `startServer.bat.`
  The `startServer.bat` files are created automatically during installation and located at the same directories as executable files. Basically, these files include some comments and the same options as are used when starting the application from the command line.
  The following is an example of a batch file for CPD Server:

```
@echo off
rem ------------------------------------------------------------------
rem Copyright (C) 2009 Genesys Telecommunications Laboratories, Inc.
rem
rem startServer.bat file for Call Progress Detection Server, version 8.1.000.04
rem ------------------------------------------------------------------
@TITLE Call Progress Detection Server v. 8.1.000.04: Application cpdserver

cpdserver.exe -host lab2 -port 7070 -app cpdserver -l 7260@lab2
```

End

Starting CPD Proxy Server

Start You can start CPD Proxy Server manually in three ways:

- From the `Start` menu.
- Open an MS-DOS command prompt and type the directory where CPD Proxy Server is installed—for example,
  `\Gcti\cpd_proxy\`—and use a command line. When starting CPD Proxy Server from a command line, use the following common command-line parameters:
  `-host` The name of the host on which Configuration Server is running
  `-port` The Configuration Server port
  `-app` The exact name of the application as configured in the Configuration Database
  For example:
  `cpdproxy.exe -host ConfigServer1 -port 2020 –app CpdProxy`

Note:	If you are implementing the client-s—Client-Side Port Security Configuration]].

- Run batch files, `startServer.bat.`
  The `startServer.bat` files are created automatically during installation and located at the same directories as executable files. Basically, these files include some comments and the same options as are used when starting the application from the command line.
  The following is an example of a batch file for CPD Proxy Server:

```
@echo off
rem ---------------------------------------------------------------
rem Copyright (C) 2009 Genesys Telecommunications Laboratories, Inc.
rem
rem startServer.bat file for Call Progress Detection Proxy, version 8.1.000.04
rem ---------------------------------------------------------------
@TITLE Call Progress Detection Proxy v. 8.1.000.04: Application cpdproxy

cpdproxy.exe -host lab3 -port 7070 -app cpdproxy
```

End

Starting and Logging In to Outbound Contact Manager

Note:	For instructions on how to start and log into Genesys Administrator, see the *Framework Genesys Administrator Deployment Guide.*

Start

1. To start Outbound Contact Manager, from the `Start` menu, select `All Programs > Genesys Solutions > Outbound Contact > Outbound Contact Manager` (OCM).

Note:	When using Windows Explorer, you can also locate the OCM executable in the destination folder of the OCM installation and double-click it.

2. In the Login dialog box, enter a user name that has permission to use this OCM. Change the user name to match Configuration Manager's user name.

Note:	Persons (non-agents) are added automatically to the `Admin` Access Group.

3. Click `Details`.
4. Enter the application name. The default application name is `blank`; this name must be changed to the instance of Outbound Contact Manager as it is registered in the Configuration Database.
5. Enter the host name of the computer running Configuration Server.
6. Enter the port number for the network port used by Configuration Server on the host computer.
7. Click `OK`.

Note:	Login will fail if Configuration Server is not running or if you do not have permission to log in.

8. In the `Select Tenant` window, select a tenant from the list and click `Next`. (Outbound Contact Manager can work with only one tenant at a time).

Note:	You only see all tenants if you are a Super Administrator.

9. In the next window, select the Outbound Contact Server to which you would like to connect and click `Finish`.

Campaigns, Calling Lists, Agent Groups, and Place Groups that have already been defined in the Configuration Database appear the first time OCM is run. If Outbound Contact Server is not running, Outbound Contact Manager still logs in to Configuration Server and the user can work with calling lists.

End

Stopping Manually

Stop components in the following order:

1. Outbound Contact Manager (OCM), if used instead of Genesys Administrator
2. CCPulse+
3. Outbound Contact Server (OCS)
4. CPD Server/CPD Proxy Server (if used)
5. ICON components for historical reporting
6. Stat Server
7. All other DB Servers (for Calling Lists)
8. T-Server
9. Configuration Server
10. DB Server for Configuration Server

Outbound Contact Manager (OCM)

Note:	Because Genesys Administrator is a web application, you log out of the application instead of stopping it. For more information about Genesys Administrator, see the *Framework Genesys Administrator Deployment Guide.*

Stopping OCM

Start

1. Select the `File` menu.
2. Select `Exit`.

End

Outbound Contact Server (OCS)

Stopping OCS Manually

Start

1. Find the `OCS` console window on the computer's desktop.
2. Click the `Close` button in the window's upper-right corner.

End

Stopping OCS as a Windows Service

Start

1. Select `Services` in the computer's Control Panel.
2. In the Windows services window, select the OCS to stop.
3. Select `Stop`.

End

CPD Server

Stopping CPD Server Manually

Start

1. Find the `CPD Server` console window on the computer's desktop.
2. Click the `Close` button in the window's upper-right corner.

End

Stopping CPD Server as a Windows Service

Start

1. Select `Services` in the computer's Control Panel.
2. In the Windows Services window, select the CPD Server to stop.
3. Select `Stop`.

End

CPD Proxy Server

Stopping CPD Proxy Server Manually

Start

1. Find the `CPD Proxy Server` console window on the computer's desktop.
2. Click the `Close` button in the window's upper-right corner.

End

Stopping CPD Proxy Server as a Windows Service

Start

1. Select `Services` in the computer's Control Panel.
2. In the Windows Services window, select the CPD Proxy Server to stop.
3. Select `Stop`.

End

Restarting Outbound Contact Components

To restart Outbound Contact components, follow the previously documented steps.

OCM does not keep long-term data, so it is safe to stop and restart/log in. OCS will provide OCM with the campaign status. The same applies to Genesys Administrator.

Connection and Reconnection

Outbound Contact components connect to servers to deliver and receive information about the contact center. OCS and CPD Server can be set up with the warm standby mode.

Note:	Currently, Outbound Contact components cannot be set up in cold standby or hot standby modes.

OCS has a warm standby mode to handle situations in which OCS stops working for any reason.

In Outbound Contact, the primary and backup OCS communicate through a direct TCP/IP connection. In the warm standby mode, a secondary OCS receives all of the same data about a dialing session/campaign group status (for example, `start, stop, load, unload`) as the primary OCS. If the primary OCS stops working for any reason, the secondary OCS takes over the operations of the primary OCS without interruption. All client applications are connected to the primary OCS.

For more information about the standby modes, refer to the Framework Deployment Guide.

OCS Connections

OCS has connections to these components:

- DB Server
- T-Server or SIP Server, see OCS and T-Server or SIP Server
- Stat Server, see OCS, Stat Server, and Reporting
- Configuration Server, see OCS and Configuration Server
- CPD Server, see OCS and CPD Server
- CPD Proxy Server (if used), see OCS and CPD Proxy Server
- Backup or primary OCS
- Interaction Server (if used), see OCS and Interaction Server
- OBN Manager (optional), see OCS and OBN Manager (Optional)

Using the `outbound_contact_server` option, you can set up special options for the Communication DN that OCS uses. The Communication DN communicates information about the status of dialing sessions/campaign groups (for example, `start, stop, load, unload`, and so on) between the backup and primary OCS as well as third-party applications, which can be used in place of OCM or Genesys Administrator.

OCS and DB Server

If the connection between OCS and DB Server is lost, OCS sends an error message. Depending on how long the loss of connection lasts, a message may display in real-time reporting and historical reporting to indicate that OCS is waiting for records.

Warm Standby

OCS supports DB Server in warm standby mode without manual intervention and without double dialing of records.

If the OCS connection to the primary DB Server is lost, the currently running dialing session for a campaign continues to dial calls until all records in the OCS buffer are depleted. OCS sends a `Waiting-for-Records` message to Stat Server, which causes the message to display also in CCPulse+. OCS then connects to the backup DB Server. The dialing session for a campaign continues to run after the switchover is complete. After switching to the backup DB Server, OCS retries the failed transactions from the transaction buffer. Double dials due to failed transactions are not induced by the switchover.

If the connection to the backup DB Server is lost, OCS will reconnect to the primary DB Server. OCS retries any failed transactions. During the switchover to the primary DB Server, OCS runs without manual intervention.

Because OCS periodically queries calling lists for the number of ready records and notifies Stat Server of the estimated time to complete the campaign calculation, this function may be interrupted during the loss of connection. However, OCS continues to send this information after the connection is reestablished.

No Standby

OCS tries to reconnect to DB Server without manual intervention and without double dialing of records. Failed transactions are retried.

OCS and T-Server or SIP Server

The OCS connection to T-Server or SIP Server is set up on the `Connections` list on the `Configuration` tab of the `OCS` Application object in Genesys Administrator.

Note:	Starting with release 7.5, the backup OCS application maintains a direct connection to the primary OCS application.

Hot Standby

T-Server can operate in hot standby mode; however, this mode is transparent to OCS. That is, although OCS is connected to T-Server, the switchover between primary and secondary T-Servers does not affect OCS call processing.

Warm Standby

OCS reconnects to T-Server without manual intervention and without double dialing of records.

OCS cannot send statistics during the switch to a backup server. OCS does not re-send the statistics after the reconnection; therefore, some campaign statistics might be lost.

Some calls are lost while OCS tries to reconnect to T-Server. According to an internal timer expiration, OCS is able to track the lost calls, and mark the call records with `stale` call result, which prevents double dialing of records.

If the OCS connection to the primary T-Server is lost, the currently running dialing session for a campaign pauses. OCS connects to the backup T-Server and then queries the backup T-Server for Queue, Agent, and Call Status. The dialing session for a campaign continues to run after the switchover is complete.

If the connection to the backup T-Server is lost, OCS reconnects to the primary T-Server.

ADDP

OCS supports Advanced Disconnect Detection Protocol (ADDP), which enables fast disconnect detection. Configure the connection protocols in the OCS application object (`Connections` section of the `Configuration` tab in Genesys Administrator).

The connection between OCS and T-Server is a passive connection; OCS receives information from T-Server and relies on TCP/IP timeouts to identify whether the disconnection might be too long. ADDP speeds up the detection of the disconnect.

If no warm standby mode is specified, OCS will keep trying to connect to the currently configured T-Server. The currently running dialing session for a campaign will be paused until connection is resumed.

OCS, Stat Server, and Reporting

OCS can now connect directly with Stat Server, and submit data required for statistical calculations for Stat Server extensions. OCS also communicates with Stat Server through a Communication DN. OCS can send statistics to multiple Stat Servers through one or more Communication DNs.

There is no warm standby or hot standby mode for Stat Server. Redundancy for Outbound Contact real-time reporting can be achieved by:

- OCS sending statistics to multiple Stat Servers running simultaneously.
- CCPulse+ reconnecting to a backup Stat Server when the primary is down.

If the OCS connection to Stat Server is lost, real-time reporting and data collection for historical reporting resume without manual intervention after Stat Server is running again. Note that real-time reporting and data collection for historical reporting is interrupted when Stat Server is down.

In a multi-site environment, OCS is able to identify the statistics for each T-Server and thus send statistics using the Communication DN registered with each T-Server. Redundancy for Outbound reporting is possible by setting up multiple Stat Servers running simultaneously.

OCS and Configuration Server

If the connection between OCS and Configuration Server is lost, OCS sends an error message to Management Layer and does the following:

- After waiting the length of time specified in the `Reconnect Timeout` parameter set in the application object, OCS tries to connect to a backup Configuration Server.

- If no backup Configuration Server exists, after the timeout it tries to reconnect to the primary Configuration Server.

OCS also supports Advanced Disconnect Detection Protocol (ADDP) to Configuration Server. To enable ADDP between this server and Configuration Server, add the Configuration Server Application (named `confserv`) to the `Connections` list on the `Configuration` tab of the OCS Application in Genesys Administrator, and specify the values in seconds for the connection protocol. For more information, refer to *Framework Genesys Administrator Help.*

OCS and CPD Server

OCS can connect to multiple CPD Servers. In the Outbound Contact Wizard, the association with CPD Server is set at the application level. To set it on the group level, use Genesys Administrator or Configuration Manager.

OCS also supports ADDP to CPD Server. To enable ADDP between OCS and CPD Server, add the CPD Server Application to the `Connections` list on the `Configuration` tab of the OCS Application in Genesys Administrator, and specify the values in seconds for the connection protocol. For more information, refer to Framework Genesys Administrator Help.

Warm Standby

OCS supports CPD Server in warm standby mode without manual intervention and without double dialing of records.

If OCS loses connection to CPD Server, OCS will send a message to Management Layer and a system error message to Stat Server for real-time and historical reporting.

If the OCS connection to the primary CPD Server is lost, the currently running dialing session for a campaign is paused. OCS then connects to the backup CPD Server and the dialing session for a campaign will continue to run after the switchover to the backup CPD Server is complete.

After switching over to the backup server, OCS does not retry the previous transactions handled by the disconnected CPD Server. Instead, call records in those transactions are marked with the `stale` call result when the OCS internal timer for those records expires. Double dialing of the same call record does not occur.

If the connection to a backup CPD Server is lost, OCS reconnects to the primary CPD Server without attempting to redial failed transactions.

No Backup

If there is no warm standby, OCS supports reconnection without manual intervention and without double dialing of records. OCS tries to connect to the currently configured CPD Server. The currently running dialing session for a campaign pauses until the connection is restored.

Some calls are lost while OCS tries to reconnect to CPD Server. OCS tracks lost calls according to the expiration of the internal timer for those records and marks the call records with the `stale` call result, which prevents double dialing of records.

OCS and CPD Proxy Server

OCS can connect to multiple CPD Proxy Servers. In the Outbound Contact Wizard, the association with CPD Proxy Server is set at the application level. To set it on the group level, use Genesys Administrator or Configuration Manager.

OCS also supports ADDP to CPD Proxy Server. CPD Proxy can also use ADDP to connect to CPD Servers.

Warm Standby

OCS supports CPD Proxy Server in warm standby mode without manual intervention and without double dialing of records.

If OCS loses connection to CPD Proxy Server, OCS will send a message to Management Layer and a system error message to Stat Server for real-time and historical reporting.

If the OCS connection to the primary CPD Server is lost, the currently running dialing session for a campaign is paused. OCS then connects to the backup CPD Proxy Server and the dialing session for a campaign will continue to run after the switchover to the backup CPD Server is complete.

After switching over to the backup server, OCS does not retry the previous transactions handled by the disconnected CPD Proxy Server. Instead, call records in those transactions are marked with the `stale` call result when the OCS internal timer for those records expires. Double dialing of the same call record does not occur.

If the connection to a backup CPD Proxy Server is lost, OCS reconnects to the primary CPD Proxy Server without attempting to redial failed transactions.

No Backup

If there is no warm standby, OCS supports reconnection without manual intervention and without double dialing of records. OCS tries to connect to the currently configured CPD Proxy Server. The currently running dialing session for a campaign pauses until the connection is restored.

Some calls are lost while OCS tries to reconnect to CPD Proxy Server. OCS tracks lost calls according to the expiration of the internal timer for those records and marks the call records with the `stale` call result, which prevents double dialing of records.

OCS and Interaction Server

For OCS to process requests and responses from Interaction Server, OCS must be a server for Interaction Server and must receive these responses on a special ESP-type port rather than its default listening port. This ESP PortID must be used instead of the default PortID, when configuring a connection to the OCS `Application` object on the `Configuration` tab of the Interaction Server `Application` object in Genesys Administrator. This makes OCS a server for Interaction Server and Interaction Server a client for OCS.

OCS and OBN Manager (Optional)

OCS uses the Outbound Notification Manager (OBN) application as a connector to GVP to request that GVP dial records. The OBN Manager GVP application should be created in Genesys Administrator or

Configuration Manager to provide its host and port information to OCS. This application must be specified on the Connections list on the Configuration tab of the Campaign Group configuration object in Genesys Administrator to enable GVP integration. This configuration will enable basic connection capabilities.

If full connection abilities are required, including Advanced Disconnect Detection Protocol (ADDP), then this application should also be present on the Connections list on the Configuration tab of the OCS application object in Genesys Administrator, where the ADDP connection protocol can be specified.

For more information about Outbound Contact and GVP integration, see Genesys Voice Platform. See also the *Genesys 8.0 Proactive Contact Solution Guide.*

OCS, Failover, Campaigns

If OCS is running dialing sessions for campaigns and then stops functioning due to software, hardware, or network issues, the campaigns and records associated with outbound calls are affected.

Note:	You can configure OCS to restart automatically if you have configured LCA, Solution Control Server, and Solution Control Interface accordingly.

Impact on Campaigns

When OCS stops functioning, dialing sessions/campaign groups will not be loaded and will not restart automatically even if OCS restarts.

If OCS is operating in the Warm Standby mode and running dialing sessions for campaigns, the failure of the primary OCS causes the backup OCS to continue running the dialing sessions for campaigns with a new set of available records, which must be retrieved from the database. The retrieval occurs immediately after the switchover to the backup OCS.

For Do Not Call records, when OCS switches from the backup server to the primary server, OCS rereads those records that were added after the Do Not Call list was initially read by OCS for all Tenants that have active/ running dialing sessions for campaigns and whose Do Not Call list(s) were imported. This synchronizes the Do Not Call list between the backup and primary OCS if the primary OCS updates the list after the backup reads it, due to the addition of new records to the Do Not Call list. No call requests will be created by OCS until the Do Not Call list table is completely read.

Note:	The primary and backup Outbound Contact Servers synchronize not only the state of the campaigns and sequences, but also the predictive algorithm information. The backup OCS is capable of taking over and continuing dialing as efficiently as the primary if a Predictive dialing mode is used. However, if you are using Campaign Sequencing, OCS does not support the synchronization of activating and deactivating parameters between the primary and backup servers. For example, if the primary OCS was using the StopAtDials deactivating parameter, the backup OCS will not receive the number of dials already made.

Impact on Records and Calls

When a failure and switchover occurs, be aware that:

- Records in use by the primary OCS remain in the `retrieved` state in the database. These must be manually updated if you want those records to be in a final state or if you want the associated numbers to be dialed again when you run the dialing session for the campaign another time.
- The internal state of the records and associated outbound calls stored in the memory of the primary OCS are lost when it stopped functioning.
- Any calls or interactions (if using the `Push Preview` dialing mode) being processed by the OCS at the time of the failure continue to be queued, routed, established with an agent DN, transferred, and so on.
- As these calls are processed, however, the agents handling them cannot update the associated records because the records were lost when the primary OCS stopped functioning.

Impact on Connections for Genesys Administrator

Genesys Administrator connects to OCS only when some action for a Campaign Group is required. It attempts to connect to a backup server if no primary server can be connected.

If no connection with OCS can be established, Genesys Administrator makes associated its controls unavailable.

Impact on Connections for Outbound Contact Manager

When Outbound Contact Manager loses its connection to the primary OCS, OCM displays a dialog box that allows the user to retry the connection to the primary server or connect to the backup server.

Note:	Starting with release 7.5, OCM supports a silent reconnection.

This reconnection issue does not apply to other servers or dialers to which OCS connects (for example, CPD Server, T-Server, DB Server, and so on), because connections to them should already be specified in the backup OCS `Application` object. Agent Desktops are also not affected as they do not have a direct connection to OCS. They only need the new OCS DBID, which will be delivered to the Desktop with a record's attached data.

OCM/Genesys Administrator Connections

OCM/Genesys Administrator have connections to the following components:

- OCS
- DB Server
- Configuration Server

OCM/Genesys Administrator and OCS

OCM

OCM connects to one OCS at a time by prompting the user during login. OCM supports dynamic reconfiguration of all OCS Application objects listed on OCM's `Connections` tab. Multiple instances of OCM can be connected to one OCS concurrently. If multiple OCS Application objects are defined on the OCM Application object's `Connections` tab, the list of OCS Application objects is displayed for the user to select.

If the connection between OCM and OCS is lost, OCM prompts the user to reconnect or to select a backup OCS (if available).

Genesys Administrator

Genesys Administrator allows the user to select OCS any time an operation for a Campaign Group is required.

OCM/Genesys Administrator and DB Server

OCM and Genesys Administrator establish an active connection to the DB Server that is associated with the calling list on which the user is working.

If the original connection fails, OCM prompts the user to reconnect to the DB Server or to select a backup DB Server (if available).

CPD Server Connections

A CPD Server connects to one T-Server (and one switch) only.

In a multi-site/multi-switch environment, Outbound Contact needs at least one CPD Server per switch.

CPD Server and T-Server

The CPD Server connection to T-Server is set up on the `Connections` list on the `Configuration` tab of the CPD Server Application object.

Warm Standby

CPD Server with a T-Server in Warm Standby mode supports reconnection without manual intervention and without double dialing of records.

If CPD Server's connection to the primary T-Server is lost, the currently running dialing session for the campaign pauses. CPD Server connects to the backup T-Server. The currently running dialing session for the campaign continues to run after the switchover to the new server. Some calls might be lost in warm standby mode. According to an internal timer expiration, OCS tracks the lost call and marks the calls record as `stale` to prevent double dialing.

If the connection to the backup T-Server is lost, CPD Server reconnects to the primary T-Server. After switching over to the primary T-Server, CPD Server runs without manual intervention.

No Backup

If CPD Server loses its connection to a T-Server that is not in a standby mode, CPD Server tries to reconnect to the same T-Server. It reconnects without manual intervention and without double dialing of records. The currently running dialing session for the campaign pauses until the connection is restored.

Some calls might be lost while CPD Server is trying to reconnect to T-Server. OCS tracks the lost calls and marks the call record as `stale` to prevent double dialing.

CPD Proxy Server Connections

A CPD Proxy Server connects to several CPD Servers, which must be connected to the same switch only.

CPD Proxy Server and CPD Server(s)

The CPD Proxy Server connections to CPD Servers are set up on the `Connections` list on the `Configuration` tab of the CPD Proxy Server Application object.

Warm Standby

CPD Proxy Server with a CPD Server in Warm Standby mode supports reconnection without manual intervention and without double dialing of records. If CPD Proxy Server's connection to the primary CPD Server is lost, the currently running dialing session for the campaign slows down due to a decreasing of the number of available dialing resources. CPD Proxy Server connects to the backup CPD Server. After the switchover to the backup CPD Server is completed the currently running dialing session for the campaign speeds up due to an increasing of the available dialing resources. Some calls might be lost during reconnection in warm standby mode on CPD Server side. CPD Proxy Server reports to OCS about such calls as an Error to OCS. OCS tracks the lost call and marks the calls record accordingly to prevent double dialing.

If the connection to the backup CPD Server is lost, CPD Proxy Server reconnects to the primary CPD Server. After switching over to the primary CPD Server, CPD Proxy Server runs without manual intervention.

No Backup

If CPD Proxy Server loses its connection to CPD Server that is not in a standby mode, CPD Proxy Server tries to reconnect to the same CPD Server. It reconnects without manual intervention and without double dialing of records. The currently running dialing session for the campaign slows down until the connection is restored.

Some calls might be lost while CPD Proxy Server is trying to reconnect to CPD Server.

CPD Proxy Server reports to OCS about such calls as an Error to OCS. OCS tracks the lost calls and marks the calls record accordingly to prevent double dialing.

Transport Layer Security Connections

Outbound Contact Server, CPD Server, and CPD Proxy Server support Transport Layer Security (TLS) with any Genesys TLS-supporting client/server.

Outbound Contact Manager and the Outbound Contact Configuration Wizard do not support TLS.

Note:	If you are configuring TLS between OCS and DB Server, add a Database Access Point (DAP) to DB Server on the `Connections` list on the `Configuration` tab of the OCS `Application` object in Genesys Administrator. Genesys Administrator also supports TLS.

See the TLS section of the *Genesys Security Deployment Guide* for more information.