



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

Calling Lists

12/18/2025

# Calling Lists

## Contents

- **1 Calling Lists**
  - **1.1 Do Not Call**
  - **1.2 Dealing with a Large Do Not Call List**

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\_dontcall\_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](#).