



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

Filter Object

# Filter Object

## Contents

- [1 Filter Object](#)
  - [1.1 Creating a New Filter Object](#)
  - [1.2 Filter Object Configuration Tab Fields](#)
  - [1.3 Filter Object Options Tab Fields](#)
  - [1.4 Configuring the criteria Option](#)
  - [1.5 Configuring the order\\_by Option to Sort Records](#)
  - [1.6 Defining a Dialing Filter that Exceeds 255 Characters for Configuration Server 7.6.x and Earlier](#)
  - [1.7 Filters that Break a Chain of Records](#)
  - [1.8 Full Chain Update](#)

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`.