Outbound Contact Deployment Guide

Apply to Record Actions

12/20/2025

Apply to Record Actions

Contents

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.

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.

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: 3.` The total number of times that the cycle through the chain occurs will be 3 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. CS 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 begin with the first treatment step. This cycle will repeat two more times; a total of three times, because the `Cycle Attempts` is set to 3 in our example.
2. For the next two cycles, the `Increments` between intervals will now take effect and they will increase between each cycle. On 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. On 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	RequiredSet the time interval until the next pass through the chain,	RequiredSet the time interval for the subsequent chain passes	Not Applicable

	and 1 = one pass; the next record in the chain is dialed immediately	after the last record in the chain has been dialed.		
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
`$treat_name`	Name of the current Treatment configuration object.	string	Update Treatments History SQL clause
`$treat_dbid`	DBID of the current Treatment configuration object.	integer	101
`$camp_group_dbid`	DBID of the Campaign Group configuration object to which the record being processed logically belongs.	integer	101
`$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`	DBID of the Group configuration object.	integer	101
`$camp_dbid`	DBID of the Campaign configuration object.	integer	101
`$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
`$list_dbid`	DBID of the Calling List configuration object.	integer	101
`$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` is defined.	string	Customer since 1980
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.