Field Service

Message Scenario Configuration Guide

Field Service
Message Scenario Configuration Guide

F75119-07

Authors: The Field Service Information Development Team

Contents

Preface

This preface introduces information sources that can help you use the application and this guide.

Using Oracle Applications

To find guides for Oracle Applications, go to the *Oracle Help Center*.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the *Oracle Accessibility Program website*.

Videos included in this guide are provided as a media alternative for text-based topics also available in this guide.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we're working to remove insensitive terms from our products and documentation. We're also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Contacting Oracle

Access to Oracle Support

Customers can access electronic support through Oracle Support. For information, visit *My Oracle Support* or visit *Accessible Oracle Support* if you are hearing impaired.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides. Please take one of the following surveys:

- For web-based user guide, *Web-based User Guide Survey*
- For tutorial feedback, *Tutorial Survey*

1 Understanding Message Scenarios

About Message Scenarios

Oracle Field Service lets you send messages to your staff members, customers, and external systems containing the content you define and triggered by the events you specify. The application offers unlimited flexibility to configure message scenarios that meet all of your business needs.

When a field technician is running late, Oracle Field Service can notify your customers of the new estimated time of arrival. If a dispatcher reassigns activities, the application can notify the newly assigned technicians that their routes have changed. If an employee changes his home address, that information can be sent to the human resources management system. With the ability to configure notifications for these scenarios and unlimited numbers of others, Oracle Field Service helps you ensure up-to-date communication among people and systems.

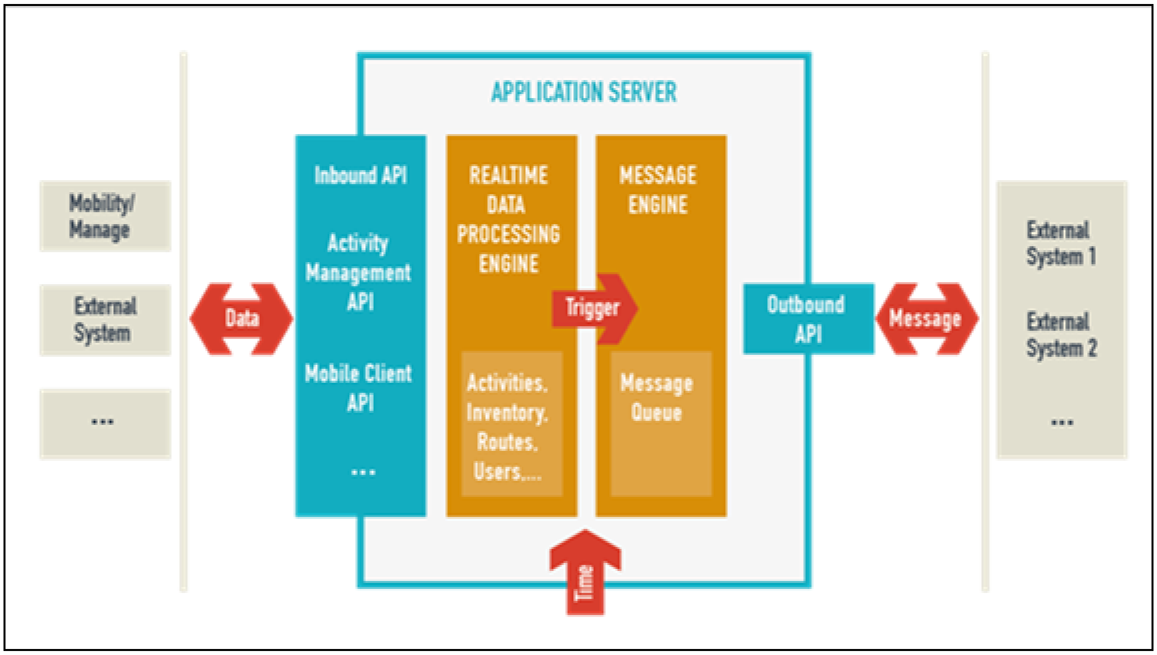
A set of APIs can receive data from employees and external systems, such as inventory management and human capital management systems. If the incoming data matches any of the conditions you have defined for message scenarios, the defined actions are initiated and a message is sent to the appropriate entity – customer, staff member, or external data system.

> **Note:** The recommended use of message scenarios is to deliver time-based notifications, typically using the Reminders, Alerts, or Visit selection in launch conditions. While it is possible to use message scenarios for all other launch conditions (such as Route, Activity, Inventory, and Service Requests), it is recommended that you use the Core API/Events REST API for integration.

A message scenario is a set of rules that specify how to process a message to a system or persons when a predefined launch condition occurs. A launch condition is an event within Oracle Field Service, for example, when a reminder notification should be sent to a customer 60 minutes prior to a technician’s estimated arrival time. When the launch condition occurs, the message scenario is initiated, running through the scenario steps as indicated and sending a notification though the specified delivery channel.

Message Scenario Notification Workflow

Messages are generated and sent based on pre-defined conditions.



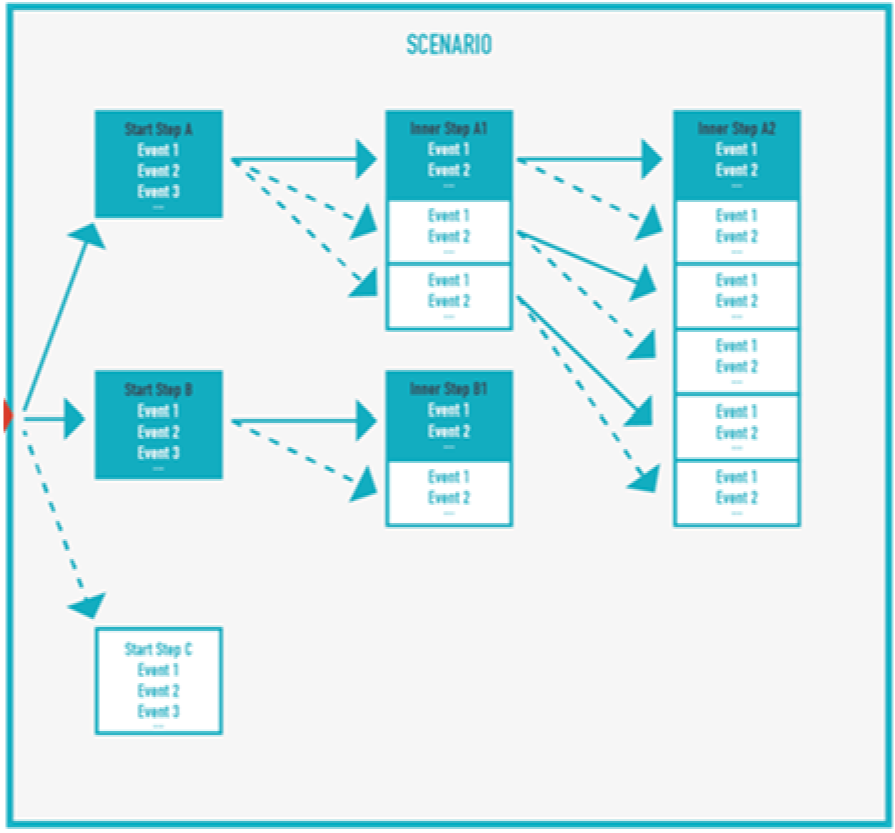
1. The system provides a set of APIs to receive data from different sources, agents and external systems like customer relationship management, inventory management, and human capital management systems.
2. If a certain condition is met when data is received, the relevant launch condition is activated.
3. The launch condition invokes the start of one or more steps in the message engine.
4. A message is generated and saved in the message queue at the end of every message scenario step.

5. Each message is processed until it reaches its final status.

   > **Note:** The following message statuses are possible:
   >
   > - **New**: Set by the message engine for successfully created messages that have not yet been sent to the message agent, which is the application that processes messages.
   > - **Falsemethod**: Set by the message engine for messages that cannot be created due to an invalid configuration or missing data.
   > - **Obsolete**: Set by the message engine for messages that are invalidated by an operation within the application, such as delete, move, or suspend, or by the creation of another message.
   > - **Sending**: Set by the message engine for messages that are sent to and initially processed by the message agent.
   > - **Sent**: Returned by the message agent when receipt of the message cannot be confirmed, for example, when a message was received by a voice mail system.
   > - **Delivered**: Returned by the message agent if the message was delivered successfully and the final recipient has received the message. This status is not typically used for e-mail since verifying that the message was read is not possible.
   > - **Failed**: Returned by the message agent for messages that cannot be delivered.

6. If a blocking condition is met, the message is either not generated or generated but not sent. The message is then set to its final status.
7. There may be inner steps included in the message scenario. An inner step is invoked in accordance with a final status of the message received as the result of the previous step processing.
8. When all pre-defined conditions have been met, the message is sent to the corresponding message agent via the Outbound API.
9. The scenario is complete when there are no more steps to be executed.

The following figure shows a possible scenario flow.



Overview of a Message Scenario

Once you have identified a situation for which you want to send a message, you create a message scenario.

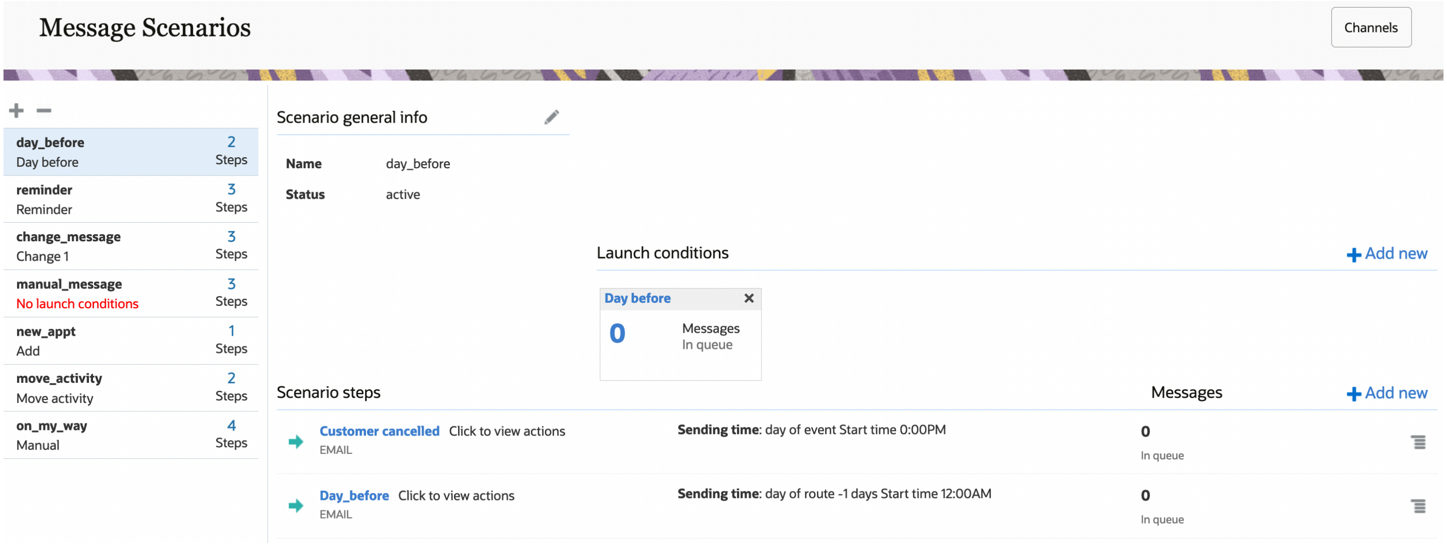
First, you'll create the message scenario by naming it and defining its active dates. To complete the message scenario, you must then define:

- Launch condition—The system event that initiates the message scenario.
- Scenario steps—The step or steps to be invoked once the launch condition occurs.
- Recipient—The persons to whom the message is sent.
- Delivery channel—The method by which the message is sent to the recipient.
- Notification details—The time the message will be sent, any delay or blocking conditions, and the number of delivery attempts.
- Message content—The body of the message, which can contain variables for specific events.

Message Scenarios Page

The **Message Scenarios** page lets you define and modify all elements of a message scenario.

You can open the **Message Scenarios** page by clicking **Configuration** and then selecting **Message Scenarios** under the Subsystems and Integrations heading. This screenshot shows the Message Scenarios page:



day_before Day before	2 Steps
reminder Reminder	3 Steps
change_message Change 1	3 Steps
manual_message No launch conditions	3 Steps
new_appt Add	1 Steps
move_activity Move activity	2 Steps
on_my_way Manual	4 Steps

Scenario general info

Name day_before

Status active

Launch conditions

Day before ✕

0 Messages In queue

Scenario steps

		Messages
Customer cancelled Click to view actions EMAIL	**Sending time**: day of event Start time 0:00PM	0 In queue
Day_before Click to view actions EMAIL	**Sending time**: day of route -1 days Start time 12:00AM	0 In queue

The left side of the page contains the message scenario list, and the right side includes three sections: the **Scenario general info** section that displays the information that was entered when the scenario was created, the **Launch conditions** section that triggers the scenario steps, and the **Scenario steps** section that defines the actions to take when the message scenario is invoked.

The list of message scenarios on the left panel of the Message Scenarios page shows all of the configured message scenarios, including the name, label, number of steps, and an error message, if applicable. The scenario you select is highlighted in the list. You can add message scenarios by clicking the plus sign at the top of the list, and remove them by clicking the minus sign.

The **Scenario general info** section includes the name of the message scenario, its label, and its status. The status is inactive if the start and end dates are both in the past, and no messages will be generated for this scenario. Otherwise the status is active. Three active conditions exist:

- Active—The start date is in the past, and no end date is set.
- Active from [start date]—The start date is in the future, and no end date is set or the end date is in the future.
- Active till [end date]—The start date is today or in the past, and the end date is in the future.

Message Scenario List Warning Messages

If a message scenario is configured improperly, a warning related to that scenario will appear in the list of message scenarios.

Message Scenario Warning Messages

Message	Description
No launch conditions	Appears when no launch condition has been defined for the scenario. New scenarios do not yet have launch conditions, nor do scenarios that have had their launch condition removed or assigned to another scenario. No message will be generated or sent for message scenarios that do not have launch conditions. Correct this error by adding a launch condition.
There is no start step in scenario	Appears when the scenario has no steps or when all the steps are defined as inner steps. Since there is no starting point, the scenario cannot be executed. Correct this error by adding a start step or designating one of the inner steps to be the start step.
Endless loop	Appears when the steps create a loop, potentially causing an endless loop situation. Correct this error by modifying the step sequence as needed to prevent a loop.

2 Configuring Message Scenarios

Create a Message Scenario

You must first create a message scenario before you can define the launch condition, scenario steps, message, and delivery channel..

1. Click **Configuration**.
2. Click **Message Scenarios** under **Subsystems and Integrations**.

   The **Message Scenarios** screen opens.
3. Click the + icon in the left panel.

   The **Add Message Scenario** dialog box opens.
4. Type the name users will see on the **Message Scenarios** screen in the **Name** field.
5. Type a unique identifier for the message scenario in the **Label** field.
6. Type the date the scenario becomes active in the **Start Date** field or click the calendar icon and select a date.
7. To set an end date at which the scenario becomes inactive, select the **End Date**check box and type or select a date in the field.

   > **Note:** If you do not select the **End Date** check box, the scenario will run indefinitely.
8. Click **OK**.

Results:

The **Add Message Scenario** window closes and the message scenario you created appears in the list of scenarios on the left of the **Message Scenarios** screen. The scenario will display an error message in the scenario list because you have not yet defined a launch condition for it.

What to do next

Now you can configure the message scenario.

Edit a Message Scenario

Each scenario consists of one or more start steps (and, in some cases, inner steps) that you can edit.

1. Click **Configuration**.
2. Click **Message Scenarios** under **Subsystems and Integrations**.

   The **Message Scenarios** page opens.
3. Select the appropriate message scenario from the left panel.

4. Do one of the following:
    - Click the title of the message scenario step you wish to edit.
    - Click ≡ next to the message scenario step, and select **Modify** from the drop-down list.

   The **Modify scenario step** dialog box opens. The dialog box contains four tabs from which you can specify the settings related to message generation:
    - **Settings** Use this tab to modify the basic parameters of the message generated at a certain scenario step, that is, how, when, and where it should be sent.
    - **Patterns** Use this tab to modify the content of the message to be sent.
    - **Conditions** Use this tab to modify the blocking conditions, that is, the conditions under which the message should not be generated at this step.
    - **Next steps** Use this tab to modify the conditions of the subsequent steps that you want to run.

Configure a Scenario Step

Configuring a scenario step is a process that offers maximum flexibility. The **Add scenario step** and **Modify scenario step** windows contain four tabs from which you can specify the settings related to message generation:

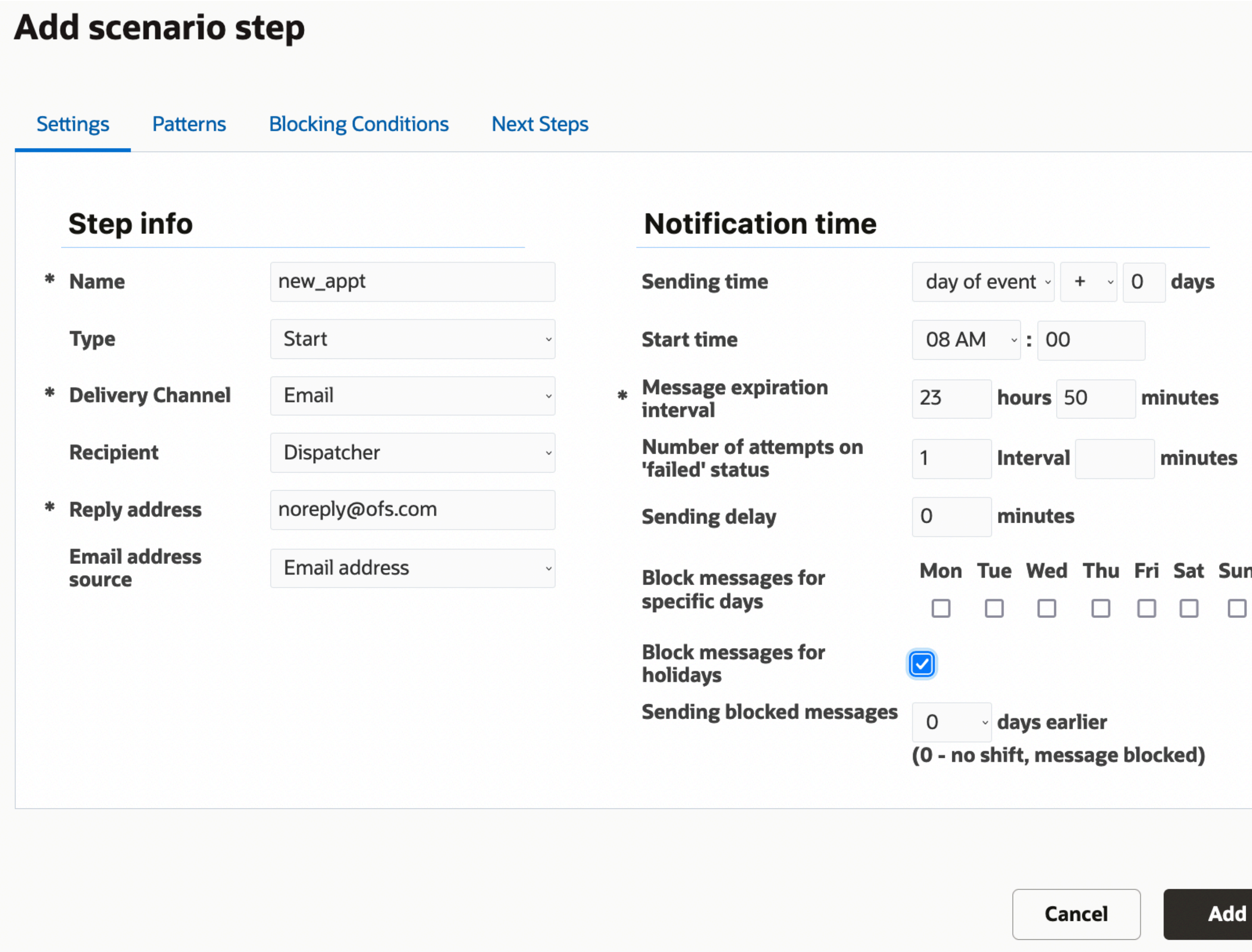
- The **Settings** tab contains the basic parameters of the message generated at a certain scenario step, that is, how, when, and where it should be sent.
- The **Patterns** tab defines the content of the message to be sent.
- The **Conditions** sets the blocking conditions, that is, the conditions under which the message should not be generated at this ste.p
- The **Next steps** tab defines the conditions of the subsequent steps run.

To configure a scenario step:

1. Click **Configuration**.
2. Click **Message Scenarios** under **Subsystems and Integrations**.

   The **Message Scenarios** screen opens.
3. Select the appropriate message scenario from the left panel.

4. Do one of the following:
    - Click the scenario step name or click the action button and select **Modify** to open the **Modify scenario step** window with the Settings tab on top.
    - Click **Add new** to open the **Add scenario step** dialog box with the Settings tab on top:



Add scenario step

Step info

* Name	new_appt
Type	Start
* Delivery Channel	Email
Recipient	Dispatcher
* Reply address	noreply@ofs.com
Email address source	Email address

Notification time

Sending time	day of event + 0 days
Start time	08 AM : 00
* Message expiration interval	23 hours 50 minutes
Number of attempts on 'failed' status	1 Interval minutes
Sending delay	0 minutes
Block messages for specific days	Mon □ Tue □ Wed □ Thu □ Fri □ Sat □ Sun □
Block messages for holidays	☑
Sending blocked messages	0 days earlier (0 - no shift, message blocked)

5. Refer to *How do I configure the Settings tab?* and complete the fields on the Settings tab.
6. Click the Patterns tab and the type content into the **Subject** and **Body** fields. For information about how the Patterns tab differs for each of the delivery channels, refer to *How do I configure the subject and body of a message in the Patterns tab?*.
7. Click the **Conditions** tab and complete the fields. For information about the Conditions, refer to *How do I define the blocking conditions for a message scenario?*.
8. Click the **Next steps** tab and complete the fields. Refer to *Add Next Steps*.

Working with Launch Conditions

The Message Scenarios page displays information about launch conditions for the selected scenario. Launch conditions are the system events that trigger a message scenario.

Message scenarios are not initiated unless and until something happens to trigger them. The event that sets the message scenario in motion is called a launch condition, and these events are predefined by Oracle Field Service. Launch conditions include events such as the delay of an estimated arrival time, the reassignment of an activity, or the installation of inventory. You define the launch condition that sends a specific message to a designated recipient using the delivery channel you identify at the time you want it to be sent.

You can specify multiple launch conditions for a single message scenario, and whenever one of those conditions is detected, the message scenario begins to run through the steps of the scenario. However, while message scenarios can have multiple launch conditions, each launch condition can be associated with only one message scenario because the system has to know which scenario to launch when the launch condition is detected.

The two types of launch conditions include action-driven and condition-driven. Action-driven launch conditions are those events that immediately launch the message scenario when they occur, for example, activating a route or completing an activity. Condition-driven launch conditions are determined by polling the system periodically to see if certain conditions have been fulfilled and, if they have, then launching the message scenario. For example, Oracle Field Service checks to see whether the current time is within a notification window for the start of a delivery window or in the time frame during which an escalation notice should be sent. So there is not a specific action that launches the message scenario for condition-driven launch conditions, but rather the passage of time that indicates the message should be sent now.

The **Launch conditions** section displays the scenario’s existing launch conditions and lets you add and remove conditions. A numeric indicator lets you know how many messages are in the queue for each of the launch conditions and lets you estimate the potential results of any changes you make to the scenario.

> **Note:** If the number of messages is greater than 999, the indicator displays the number in thousands, denoted by the letter k. For example, 5,000 messages will be indicated as 5k. If the number of messages is 1,000,000 or greater, the indicator displays the number in millions, denoted by the letter m. For example, 12,000,000 messages is displayed as 12m.

You can remove a launch condition by clicking the x in the upper right corner of the condition indicator.

Add a Launch Condition for a Message Scenario

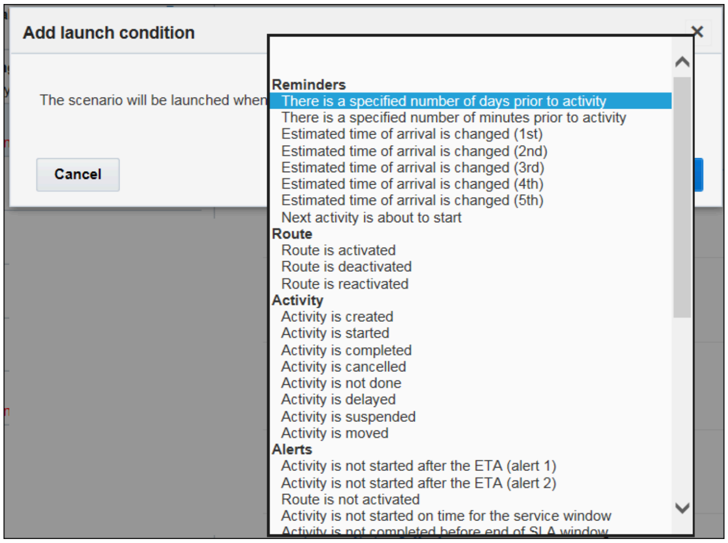
A launch condition is the event that triggers the message scenario.

Launch conditions are configured on the **Message Scenarios** screen.

1. Click **Configuration** and select **Message Scenarios** under the Subsystems and Integrations heading.
2. On the **Message Scenarios** screen, select the scenario for which you want to define a launch condition.

3. Click the **Add new** link for launch conditions.

   The **Add launch condition** window opens. The window contains a drop-down menu that, when clicked, lets you select the event that triggers the message scenario.


Add launch condition
The scenario will be launched when
Cancel
Reminders
There is a specified number of days prior to activity
There is a specified number of minutes prior to activity
Estimated time of arrival is changed (1st)
Estimated time of arrival is changed (2nd)
Estimated time of arrival is changed (3rd)
Estimated time of arrival is changed (4th)
Estimated time of arrival is changed (5th)
Next activity is about to start
Route
Route is activated
Route is deactivated
Route is reactivated
Activity
Activity is created
Activity is started
Activity is completed
Activity is cancelled
Activity is not done
Activity is delayed
Activity is suspended
Activity is moved
Alerts
Activity is not started after the ETA (alert 1)
Activity is not started after the ETA (alert 2)
Route is not activated
Activity is not started on time for the service window
Activity is not completed before end of SLA window

4. Select the launch condition for the message scenario, referring to the following table for a description of each condition.
5. Click OK.

Results:

Launch condition, associated scenarios, and description

Launch Condition	**Scenario to be associated to *OR* Scenario will be launched when**	**Description**
Reminders		For detailed information about how the reminder interval for message delivery is determined, refer to *Reminder and Change Notification Launch Conditions*
Day Before	There is a specified number of days prior to activity	Provides a proactive message to customers within the defined number of days before an activity is scheduled to start. The message is sent immediately at the specified time before a new or rescheduled activity. This launch condition is not invoked for non-scheduled, reopened, or pre-work activities. It also applies only to activities that have the **Enable 'day before' trigger** field selected when the activity type is created or modified. Messages invoked by this launch condition become obsolete after the following activity-related actions: • Cancel

Launch Condition	**Scenario to be associated to *OR* Scenario will be launched when**	**Description**
		• Delete • Start • Suspend • Reschedule No messages are generated when an activity is moved between resources on the same day.
Reminder	There is a specified number of minutes prior to activity	Provides a proactive message to customers within the defined number of minutes before an activity is scheduled to start. The message is sent immediately at the specified time before a new or rescheduled activity. This launch condition is not invoked for non-scheduled, reopened, or pre-work activities. Messages invoked by this launch condition become obsolete after the following activity-related actions: • Cancel • Delete • Start • Suspend • Reschedule No messages are generated when an activity is moved between resources on the same day. Additional information is required when you select this launch condition: • Reminder time in minutes (Specify more than one reminder time by separating the values with commas) • How the time is calculated: ○ Delivery window start ○ Service window start ○ ETA • Silent interval
Change 1 to Change 5	Estimated time of arrival is changed (1st) Estimated time of arrival is changed (2nd) Estimated time of arrival is changed (3rd)	Provides up to five proactive messages to customers when the estimated time of arrival for the activity has changed. Additional information is required when you select this launch condition: You will be asked to specify the difference between when the time is calculated (delivery window start, service window start, or ETA) and when the last notification was delivered to the customer. You

Launch Condition	Scenario to be associated to *OR* Scenario will be launched when	Description
	Estimated time of arrival is changed (4th) Estimated time of arrival is changed (5th)	will also define a window of time before the activity start for which the rule will apply. • How the time is calculated: ○ Delivery window start ○ Service window start ○ ETA • The number of minutes between the time and the last time a message was delivered to the customer, at which point this message will be delivered • The range of minutes prior to the activity start during which the launch condition applies These launch conditions are invoked only for pending ordered activities (regular or reopened) in an activated route on the current work day; they do not apply when a reminder is not sent and the current time is within the silent interval, when an incomplete reminder exists, or when a change message was already sent and the last change message was sent by the same launch condition. It also applies only to activities that have the **Enable 'reminder' and 'change' triggers** field selected when the activity type is created or modified. Messages generated by these launch conditions are removed from the message queue if one of the following events occurs after their generation and sending: • The activity status is changed. • The activity becomes non-ordered. • The activity is moved. Refer to *Reminder and Change Notification Launch Conditions*.
Call Ahead	Next activity is about to start	Provides a proactive message to customers when the next activity is about to start.
Route		
Activate	Route is activated	Invoked when a route is activated.
Deactivate	Route is deactivated	Invoked when a route is deactivated.
Reactivate	Route is reactivated	Invoked when a route is reactivated.

Launch Condition	Scenario to be associated to *OR* Scenario will be launched when	Description
Activity		
Add	Activity is created	Invoked when a new activity is created or an existing activity is moved to a different day or resource. This launch condition is invoked only for regular and reopened activities, but not for prework or instances of mass repeating activities, such as lunches or meetings.
Start	Activity is started	Invoked when an activity is started. This launch condition is invoked only for regular and reopened activities, but not for prework.
Complete	Activity is completed	Invoked when an activity is completed.
Cancel	Activity is cancelled	Invoked when an activity is cancelled.
Not done	Activity is not done	Invoked when the status of an activity is changed to **not done**.
Delay	Activity is delayed	Invoked when an activity is delayed beyond the number of minutes specified in the launch condition.
Suspend	Activity is suspended	Invoked when an activity is suspended. **Note:** If a started activity is suspended, a new suspended activity is created. This launch condition is then invoked for the new suspended activity. When this happens, both the pending and suspended activities have the same property values, and the suspended activity has no inventory.
Move Activity	Activity is moved	Invoked when an activity is moved to a different day or different resource. This launch condition is invoked only for regular and reopened activities, but not for prework. The messages that are generated with this launch condition refer to the origin resource. To retrieve information about the destination resource, use the destination_resource block. To retrieve information about the destination resource and date, use the following placeholders: • destination_resource_id • destination_resource_external_id • destination_resource_name

Launch Condition	Scenario to be associated to *OR* Scenario will be launched when	Description
		- destination_date For information about using blocks and placeholders, refer to *Message Content Overview* To control whether the activity can be moved to another resource or another day, use the **Resource changed?** or **Day changed?** blocking condition. For information about blocking conditions, refer to *About Blocking Conditions*.
Alerts		
Not Started 1 Not Started 2	Activity is not started after the ETA (alert 1) Activity is not started after the ETA (alert 2)	Invoked when an activity has not been started within the number of minutes specified when defining the launch conditions. The two launch conditions are independent and can be generated for the same activity at the same time. They can be invoked only for pending ordered activities (regular or reopened) in an activated route that belongs to the current working day.
Not Activated	Route is not activated	Invoked when a route has not been activated within the number of minutes specified when defining the launch condition. This launch condition applies only once per day per route, and messages are not regenerated if the calendar changes. Any existing not-activated messages become obsolete at the moment of route activation. If the resource is new, the messages generated by this launch condition are not generated until the day after the resource is created. Additionally, the resource must be associated with a resource type that has the **Enable 'Not activated in time' alert and trigger** field selected when the resource type is created or modified.
Service Window Warning	Activity is not started on time for the service window	Invoked when an activity has not started within the number of minutes before the end of the service window that is specified when defining the launch condition. It is also invoked when an activity is scheduled after the end of the service window. The launch condition can be invoked only for pending ordered activities (regular or reopened) in an activated route with a service window that belongs to the current working day, and it is invoked only once per activity. The activity must be associated with an activity type that has the **Enable 'SW Warning' trigger** field selected when the activity type is created or modified.
SLA Warning	Activity is not completed before the end of SLA window	Invoked when a pending activity has not been started within the defined number of minutes before the end of the SLA window or when a started activity is not completed within the defined number of minutes before the end of the SLA window. The launch condition is invoked only once per activity unless the **SLA window**

Launch Condition	Scenario to be associated to *OR* Scenario will be launched when	Description
		end changes after the generation of this alert, at which point it can be invoked again.
Service request		
Manual	Service request is created	Invoked when a service request is created. When you create the message, you can use placeholders related to the service request and its parent objects. For example, if the service request is for an activity, the content can contain placeholders related to the request, activity, route, and resource. Refer to *What activity message placeholders are available?*. Use this launch condition for the following situations: • Inventory tracking and hardware testing • Initiating SRO or sending any other form • Initiating support requests • When a transaction is initiated without being related to the activity • Other activity or inventory requests
Inventory		The inventory launch conditions are used to communicate inventory operations to an enterprise resource planning system and perform automated provisioning.
Install Inventory	Inventory is installed	Invoked when inventory is moved from the **resource** pool to the **install** pool or when a new **install** inventory record is created.
Deinstall Inventory	Inventory is deinstalled	Invoked when inventory is moved from the **customer** pool to the **deinstall** pool or when a new **deinstall** inventory record is created.
Exchange Inventory	Inventory is exchanged	Invoked when inventory exchange between the **resource** pool and the **customer** pool is performed.
Undo Install Inventory	Undo install inventory performed	Invoked when inventory is moved from the **install** pool to the **resource** pool.
Undo Deinstall Inventory	Undo deinstall inventory is performed	Invoked when inventory is moved from the **deinstall** pool to the **customer** pool.
Move Inventory	Inventory is moved	Invoked when inventory is moved between different resources. The launch condition applies when a user moves an inventory item belonging to his or her resource to another resource using Collaboration Service. Depending on visibility restrictions, the destination resource may be invisible to the user who originates the move.

Launch Condition	Scenario to be associated to *OR* Scenario will be launched when	Description
		The messages generated by this trigger refer to the **origin** resource. Use the **destination_resource** block to retrieve information about the **destination** resource. Use the following placeholders to retrieve information about the **destination** resource. • destination_resource_id • destination_resource_external_id • destination_resource_name Refer to *Message Content Overview* for details about using blocks and placeholders.
Visit		The visit launch conditions apply to groups of activities, called visits, which are a combination of several related activities for one customer. They are used to send proactive messages to customers for the entire visit, thus avoiding multiple messages for different activities within a single visit.
Visit Day Before	There is a specified number of days prior to visit	Provides a proactive message to customers within the defined number of days before a visit is scheduled to start. After a five-minute delay to allow the system to accept the other activities that comprise the visit, the message is sent at the specified time before a new or rescheduled visit. The delay also prevents generation of messages for temporary visits, which may be created with visit activities are moved between resources one by one.
Visit Reminder	There is a specified number of minutes prior to visit	Provides a proactive message to customers within the defined number of minutes before a visit is scheduled to start. The message is sent immediately at the specified time before a new or rescheduled activity. The launch condition is invoked only for pending visits and only once for the same visit. The first customer-related activity in the visit must be an ordered activity in an activated route for the current working day. Additional information is required when you select this launch condition: • The time prior to the visit in minutes • How the time is calculated: ◦ Delivery window start ◦ Service window start ◦ ETA • Silent interval

Launch Condition	Scenario to be associated to *OR* Scenario will be launched when	Description
		Refer to *Reminder and Change Notification Launch Conditions* to understand when the message is sent.
Visit Change 1 — Visit Change 5	(For visit) Estimated time of arrival is changed (1st) (For visit) Estimated time of arrival is changed (2nd) (For visit) Estimated time of arrival is changed (3rd) (For visit) Estimated time of arrival is changed (4th) (For visit) Estimated time of arrival is changed (5th)	Provides up to five proactive messages to customers when the estimated time of arrival for the visit has changed. Additional information is required when you select this launch condition. • How the time is calculated: ○ Delivery window start ○ Service window start ○ ETA • The number of minutes between the time and the last time a message was delivered to the customer, at which point this message will be delivered • The range of minutes prior to the activity start during which the launch condition applies These launch conditions apply only to pending visits, and the first customer-related activity in the visit must be an ordered activity in an activated route for the current day. They are not invoked in the following circumstances: • The **Visit reminder** is not sent and the current time is within the silent interval for the visit reminder. • An incomplete **Visit reminder** exists. • The **Visit change** message has already been sent and the **Visit change** message was sent by the same visit change launch condition. For more information about the when the message is sent, refer to *Reminder and Change Notification Launch Conditions*.
Visit Cancel	Visit is cancelled	Invoked when a visit is cancelled.
Visit Complete	Visit is completed	Invoked when a visit is completed.

Edit a Launch Condition

You can edit one or more launch conditions for a message scenario.

1. Click **Configuration**.

2. Click **Message Scenarios** under **Subsystems and Integrations**.

   The **Message Scenarios** screen opens.
3. In the left panel, select the message scenario that contains the launch condition you want to edit.
4. Click the launch condition name in the header of the launch condition box.

   The **Edit launch condition** window opens.
5. To replace the launch condition, click the drop-down menu and select a new launch condition.
6. To change the parameters of the launch condition—for example, how a time frame is calculated or the number of minutes before a message is launched—enter new values in the appropriate fields.
7. Click OK to save your changes and close the **Edit launch condition** window.

Delete a Launch Condition

You can delete one or more launch conditions for a message scenario.

1. Click **Configuration**.
2. Click **Message Scenarios** under **Subsystems and Integrations**.

   The **Message Scenarios** screen opens.
3. In the left panel, select the message scenario you want to edit.
4. On the right side of the screen, click the x in the upper right corner of each launch condition you want to delete.
5. Click OK in the confirmation dialog.

> **Note:** If you delete all launch conditions for a message scenario, a warning appears under the scenario name in the left column. No message will be generated or sent for message scenarios that do not have launch conditions.

Launch Condition Warnings and Notes

If a launch condition is configured improperly, a warning will appear when you try to save it.

Launch Condition Warnings and Descriptions

Type	Text	Description
Note	Note: Selected value **{PRIOR_TO_VALUE}** will be also applied for Reminder and other Change launch conditions.	Appears for **Change** launch conditions when the value of the **Notify30_60** parameter is changed.
Note	Note: Selected value **{PRIOR_TO_VALUE}** will be also applied for Change launch conditions.	Appears for the **Reminder** launch condition when the value of the **Notify30_60** parameter is changed.
Note	Note: Selected value **{PRIOR_TO_VALUE}** will be also applied for Visit reminder and other Visit change launch conditions.	Appears for **Visit change** launch conditions when the value of the **Visit notification base** parameter is changed.

Type	Text	Description
Note	Note: Selected value **{PRIOR_TO_VALUE}** will be also applied for Visit change launch conditions.	Appears for the **Visit reminder** launch condition when the value of the **Visit notification base** parameter is changed.
Warning	This launch condition already assigned to the **{MESSAGE_SCENARIO_NAME}** message scenario. It will be reassigned to the current message scenario.	Appears when you try to reassign a launch condition assigned to another message scenario.
Error	Start time has to be greater than end time	Appears when validation fails for start time and end time parameters.
Error	Incorrect parameter value	Appears when validation fails. The corresponding field will be highlighted.
Error	Entered time range [{EDITED_TRIGGER_START} – {EDITED_TRIGGER_END}] has intersection with other configuration: {EXISTED_TRIGGER_NAME} [{EXISTED_TRIGGER_START} – {EXISTED_TRIGGER_END}]	Appears when a time range for a Change (or Visit Change) event crosses the range of another already configured Change (or Visit Change) event.

Working with Scenario Steps

The message scenario steps indicate what should happen when a message scenario is triggered by a launch condition.

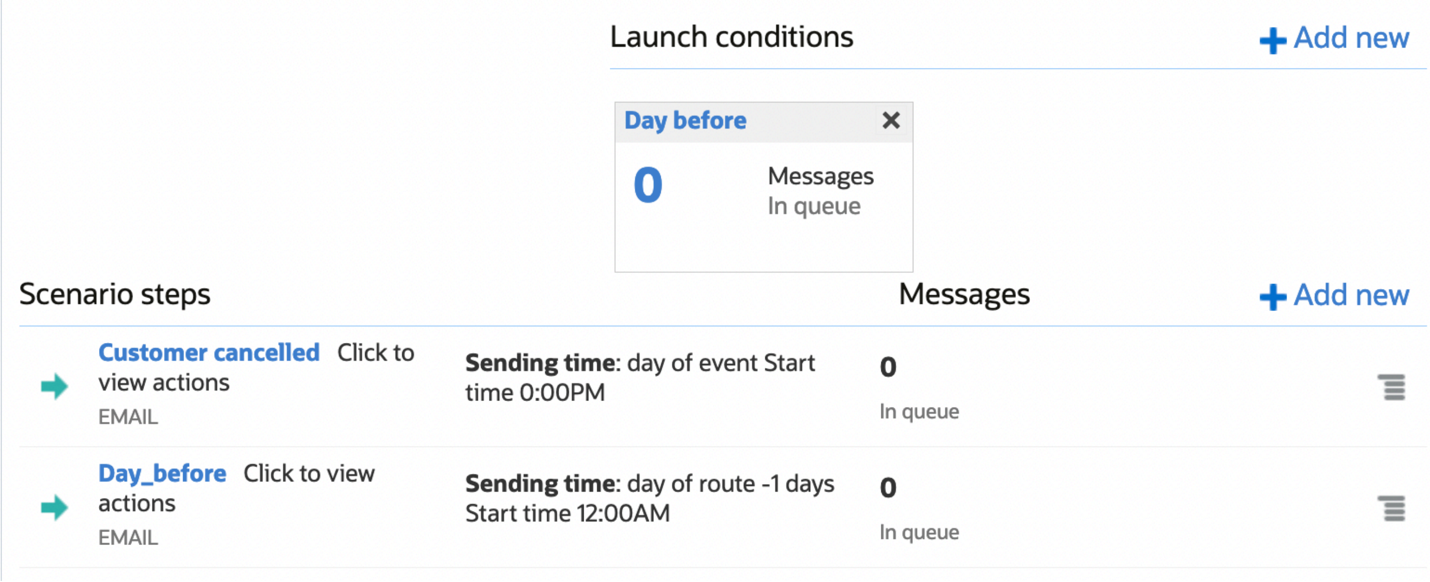
Every message scenario must include at least one start step. Without the start step, messages for the scenario will never be sent. Inner steps are those that are triggered by the results from the start step or a previous inner step. Each step defines the recipient and delivery channels as well as number of notification attempts, notification time, and message parameters. Additionally, you can define message patterns, blocking conditions, and next steps. Refer to *Configure a Scenario Step* for detailed information about setting up a message step.

For example, assume you want to send a message to customers when the estimated time of arrival changes. After selecting the **Estimated time of arrival is changed (1st)** launch condition, you can begin configuring the scenario steps. Your first step may be notifying the customer via an email message that you can configure with information specific to the customer. technician, and activity. You can then add a step that sends an email to the resource assigned to the activity if the customer cancels due to the delay, again using placeholders to provide activity-specific information.

Each step appears on its own line. For each step, the number of messages to be sent today and a graphic depiction of the number being sent appear on the line. The following color coding scheme is used:

- Green—No warnings and the channel is active
- Gray—No warnings and the channel is inactive.

- Red—Warnings exist or the number of message scenarios is greater than zero and the channel is inactive.



Customer cancelled Click to view actions
EMAIL
Sending time: day of event Start time 0:00PM
0
In queue

Day_before Click to view actions
EMAIL
Sending time: day of route -1 days Start time 12:00AM
0
In queue

The line for each step includes an action link that lets you modify or delete the step. You can also modify the step by clicking the step name to open it. Steps that have not been properly configured will display an error message in red.

> **Note:** If your display is not wide enough, you may not see all of the columns shown in the figure.

Add a Scenario Step

You must add at least one step to a message scenario in order for the message to be sent.

1. Click **Configuration**.
2. Click **Message Scenarios** under **Subsystems and Integrations**.
   The **Message Scenarios** screen opens.
3. In the left panel, select the scenario to which you want to add a scenario step.
4. Click the **Add new** button in the **Scenario steps section of screen**.
   The **Add scenario step** window opens.
5. Refer to *Configure a Scenario Step* for information about configuring all the settings for a message scenario step.

Edit a Message Scenario Step

Each scenario consists of one or more start steps (and, in many cases, inner steps) that you can edit.

1. Click **Configuration**.

2. Click **Message Scenarios** under **Subsystems and Integrations**.

   The **Message Scenarios** screen opens.
3. Select the appropriate message scenario from the left panel.
4. Do one of the following:
   - Click the title of the message scenario step you wish to edit.
   - Click the action button to the right of the message scenario step and select **Modify** from the drop-down menu.

   The **Modify scenario step** dialog box opens.
5. Refer to *Configure a Scenario Step* for information about configuring all the settings for a message scenario step.

Configure a Scenario Step

Configuring a scenario step is a process that offers maximum flexibility. The **Add scenario step** and **Modify scenario step** windows contain four tabs from which you can specify the settings related to message generation:

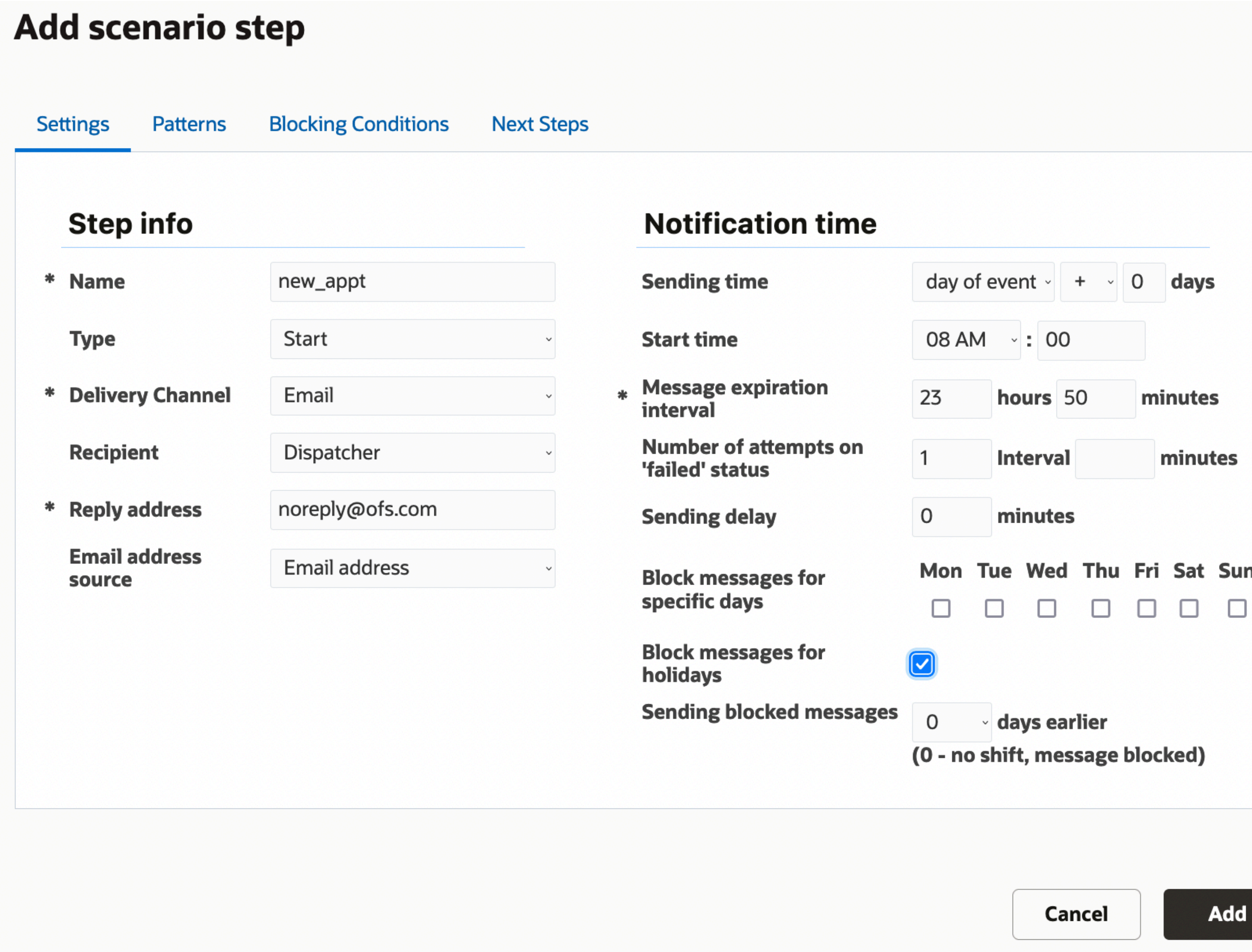
- The **Settings** tab contains the basic parameters of the message generated at a certain scenario step, that is, how, when, and where it should be sent.
- The **Patterns** tab defines the content of the message to be sent.
- The **Conditions** sets the blocking conditions, that is, the conditions under which the message should not be generated at this ste.p
- The **Next steps** tab defines the conditions of the subsequent steps run.

To configure a scenario step:

1. Click **Configuration**.
2. Click **Message Scenarios** under **Subsystems and Integrations**.

   The **Message Scenarios** screen opens.
3. Select the appropriate message scenario from the left panel.

4. Do one of the following:
   - Click the scenario step name or click the action button and select **Modify** to open the **Modify scenario step** window with the Settings tab on top.
   - Click **Add new** to open the **Add scenario step** dialog box with the Settings tab on top:



Step info

* Name: new_appt
Type: Start
* Delivery Channel: Email
Recipient: Dispatcher
* Reply address: noreply@ofs.com
Email address source: Email address

Notification time

Sending time: day of event + 0 days
Start time: 08 AM : 00
* Message expiration interval: 23 hours 50 minutes
Number of attempts on 'failed' status: 1 Interval minutes
Sending delay: 0 minutes
Block messages for specific days: Mon □ Tue □ Wed □ Thu □ Fri □ Sat □ Sun □
Block messages for holidays: ☑
Sending blocked messages: 0 days earlier
(0 - no shift, message blocked)

5. Refer to *How do I configure the Settings tab?* and complete the fields on the Settings tab.
6. Click the Patterns tab and the type content into the **Subject** and **Body** fields. For information about how the Patterns tab differs for each of the delivery channels, refer to *How do I configure the subject and body of a message in the Patterns tab?*.
7. Click the **Conditions** tab and complete the fields. For information about the Conditions, refer to *How do I define the blocking conditions for a message scenario?*.
8. Click the **Next steps** tab and complete the fields. Refer to *Add Next Steps*.

How do I configure the Settings tab?

Use the fields on the Settings tab to define the general information about a scenario step for a message scenario, including the recipient, delivery channel, and other message delivery parameters.

Fields in the Settings tab

Field Name	Description	Possible Values
Name	The name of the step	Name of the steps to a maximum of 64 characters.
Type	The type of step	Start, Inner An inner step is triggered by the results from the start step or a previous inner step. An inner step may or may not be performed depending on the result of the start step. Here's an example of an inner step: let's say you've configured a step to notify a customer by email. You can configure an inner step to notify the technician, if notifying the customer through email fails.
Recipient	The person or entity receiving the message.	Customer, Dispatcher, Resource, Use Static Address **Note:** When you select Customer, Resource, or Dispatcher, the recipient's address is obtained from the activity or resource fields. However, if you select Use Static Address, you must enter a static email address using the *notify@example.com*format. You can enter multiple email addresses, separating each by a comma or semicolon.
Recipients (Add new)	Visible only if you select Use static address in the Recipient field. Click Add new and select the email address of a helpdesk. These options are displayed: • Deliver to helpdesk: The message is broadcast to the helpdesk; it is available in the helpdesk as a system message. Any operator in the helpdesk can take this chat and it will be moved to the operator's active chat list where they can take action. Other operators will not see the message in the helpdesk queue, but they can view this message from the chats in progress menu to know who is working on this message. • Deliver to helpdesk operators: The message is delivered to all the operators within the helpdesk.	Resources or helpdesks
Delivery Channel	The method used to send the message **Note:** The methods used are company-specific and correspond to the list of delivery channels configured for the company.	Email, Set Property (used to set new property value for entities), Collaboration
Sending time	The time when the message is to be sent **Note:** The messages with **Day of route** option for non-scheduled activities get the status **falsemethod (ACTIVITY_IS_NONSCHEDULED)** as the route date is not defined.	Select one of following options: • Day of event • Time of event • Day of route For example, if you select Day of event, select +, and enter 2 in the Days field, then the messages are sent after two days from the Day of event.
from	The time when the scenario step can start	00:00 – 23:59 or 12:00AM – 11:59PM, depending on the time format settings

Field Name	Description	Possible Values
	Note: This option is not applicable when **Sending time** is **time of event**.	
Sending will time out in	The end of the time range during which the message can be delivered, measured in hours and minutes from the sending time	00:01 – 99:59
Sending delay	Time period in minutes between message creation and message sending	0 – 999
Block messages for specific days	The days of the week on which proactive customer messages should not be sent	Sun – Sat. When a day is selected, messages will not be sent on that day.
Block messages for holidays	Whether proactive customer messages can be sent on company holidays. **Note:** The list of holidays can be configured in **Company Settings** > **Holidays**.	When the check box is selected, messages will not be sent on company holidays.
Blocked messages sending	The number of days to shift proactive customer messages back in the calendar if messages are assigned for a day of the week for which a block is set or if they are assigned to fall on a company holiday when holidays are blocked. **Note:** If a message cannot be sent because it falls on a non-working day or a holiday and cannot be shifted to a working day, it will be blocked with the **falsemethod** status and the **NONWORKING_DAY** description.	0 – 10, when 0 means that the messages will be blocked, because there is no shift of days defined.
Number of attempts on 'failed' status	**Interval** is the maximum number of attempts (including the initial one) to resend a message if it is returned with a **Failed** notification status. The **minutes** field defines the number of minutes between attempts to resend the message. This functionality is available for all messages except **Set property** and **External launch condition**. The **Failed** attempts are ignored if: • Scenario processing has been stopped. (See the Scenarios in which messages are removed section later in this topic.) • The next attempt cannot be scheduled before the message expires. • Further attempts are pointless, for example, if the email address is invalid. **Note:** An agent can also stop further **Failed** attempts or change their number using the **fault_attempt** and **stop_further_attempts** fields in a **send_message** response or a **set_message_status** request.	1 – 999 for both the number of attempts and the minutes between attempts.
Number of attempts on 'sent'status	**Interval** is the maximum number of attempts (including the initial one) to resend a message if it is returned with	1 – 999 for both the number of attempts and the minutes between attempts.

Field Name	Description	Possible Values
	a **Sent** notification status. The **minutes** field defines the number of minutes between attempts to resend the message. This functionality is available only for **External system** messages. The **Sent** attempts are ignored if: • Scenario processing has been stopped. (See the Scenarios in which messages are removed section later in this topic.) • The next attempt cannot be scheduled before the message expires.	
Customer notification time	The time range to be communicated to the customer. If the final status for the message is **Sent** or **Delivered**, the **Customer notification time** is stored in the **time delivered start/end** activity fields. **Note:** This option is available only when the recipient is **Customer.**	Service Window, Delivery Window, ETA.
Reply address	The e-mail address (for example, notify@ofs.oracle.com) for sending notifications when you select Email as the Delivery channel and Customer, Dispatcher, or Resource as the Recipient. If you leave this field blank or enter an incorrect ID, Oracle Field Service uses the default reply address from Oracle (noreply@fs.ocs.oraclecloud.com). This address has the proper SPF and DKIM settings set up for the domain. However, if you use a custom reply address, you must enable SPF and DKIM on your email server. For more information on how to enable SPF and DKIM, see the Configure SPF topic in the Oracle Cloud Infrastructure Documentation.	Any valid email address.
Email address source	The field containing the email address to be used in the 'Email' notification method Not applicable for the 'use static address' recipient	Any field to be selected from the drop-down list of the email address sources available in the system. You can enter multiple email addresses, separating each by a comma or semicolon.

Scenarios in which messages are removed

There are a number of actions in the system which, under certain conditions, might remove the existing messages, if the messages have not yet been sent.

Activity Start action

Event/Action	Description
Removed Messages	`Reminder`, `Change`, `Day before`
Status	`obsolete`

Event/Action	Description
Description	`ACTIVITY_WAS_STARTED`

Activity Cancelation action

Event/Action	Description
Removed Messages	All
Status	`obsolete`
Description	`ACTIVITY_WAS_CANCELED`

Activity Notdone action

Event/Action	Description
Removed Messages	All
Status	`obsolete`
Description	`ACTIVITY_WAS_NOT_DONE`

Deletion of a Pending Activity action

Event/Action	Description
Removed Messages	All
Status	`obsolete`
Description	`ACTIVITY_WAS_DELETED`

Activity Suspend action

Event/Action	Description
Removed Messages	All except `SLA Warning`
Status	`obsolete`
Description	`ACTIVITY_WAS_SUSPENDED`

Activity Reschedule action

Event/Action	Description
Removed Messages	All except `SLA Warning`
Status	`obsolete`
Description	`ACTIVITY_WAS_RESCHEDULED`

Activity Move action

Event/Action	Description
Removed Messages	`Reminder`, `Change`, `Not started`, `Service window warning`, `Call ahead`, `Add`
Status	`obsolete`
Description	`ACTIVITY_WAS_MOVED`

Convert an activity to not ordered

Event/Action	Description
Removed Messages	`Reminder`, `Change`
Status	`obsolete`
Description	`ACTIVITY_IS_NOT_ORDERED`

Reminder message creation action

Event/Action	Description
Removed Messages	`Customer` messages: Recipient = `Customer`
Status	`obsolete`
Description	`NEW_CUSTOMER_MESSAGE_WAS_CREATED`
Notes	The `Reminder` launch condition is not invoked if the existing customer messages cannot be dropped using the `drop_message` call (if required).

Change message creation action

Event/Action	Description
Removed Messages	`Customer` messages: Recipient = `Customer`
Status	`obsolete`
Description	`NEW_CUSTOMER_MESSAGE_WAS_CREATED`
Notes	The `Change` launch condition is not invoked if an incomplete `Reminder` exists, or if the existing customer messages cannot be dropped using the `drop_message` call (if required).

Cancel visit action

Event/Action	Description
Removed Messages	All (visit related)
Status	`obsolete`
Description	`VISIT_WAS_CANCELED`

Delete visit action

Event/Action	Description
Removed Messages	All (visit related)
Status	`obsolete`
Description	`VISIT_WAS_DELETED`

Start visit action

Event/Action	Description
Removed Messages	`Visit reminder`, `Visit change #`, `Visit day before`
Status	`obsolete`
Description	`VISIT_WAS_STARTED`

Applying new visit formulas action

Event/Action	Description
Removed Messages	All (visit related)
Status	`obsolete`
Description	`VISIT_WAS_RECALCULATED`
Notes	The messages are only removed if the visit is removed as the result of applying formula changes.

Block/Shift messages action

Event/Action	Description
Removed Messages	N/A
Status	`false`
Method Description	`NONWORKING_DAY`
Notes	This removal is performed if message sending of is not allowed for a non-working day (or a holiday) and such message cannot be shifted to an appropriate working day.

How do I configure the Patterns tab?

The **Patterns** tab defines the content of the message that's sent for the scenario step, such as the subject and body. The options for the pattern depend on the selected delivery channel.

Every pattern has a subject and body. Some patterns can be defined for several languages, although English is the default language and it is used if the message step does not include a pattern in another language. The different pattern types are described below.

Patterns can use placeholders to represent actual values that will be inserted when the message is sent. For example, if you want to include the customer’s name in a message, you can use the **{activity_customer_name}** placeholder.

Placeholders

Use placeholders in messages when you want the message to include an actual value for an entity when the message is generated. Placeholders let you create a single message pattern that replaces the placeholders with information specific to each instance, such as the resource time and time of arrival. You can define the type of encoding to be applied to a placeholder value before printing. The encoding can be specified after a name of the placeholder. The | character is used as the delimiter in this case. The following encoding types are supported:

- none
- xml/html
- slashes/cslashes
- cescape
- url

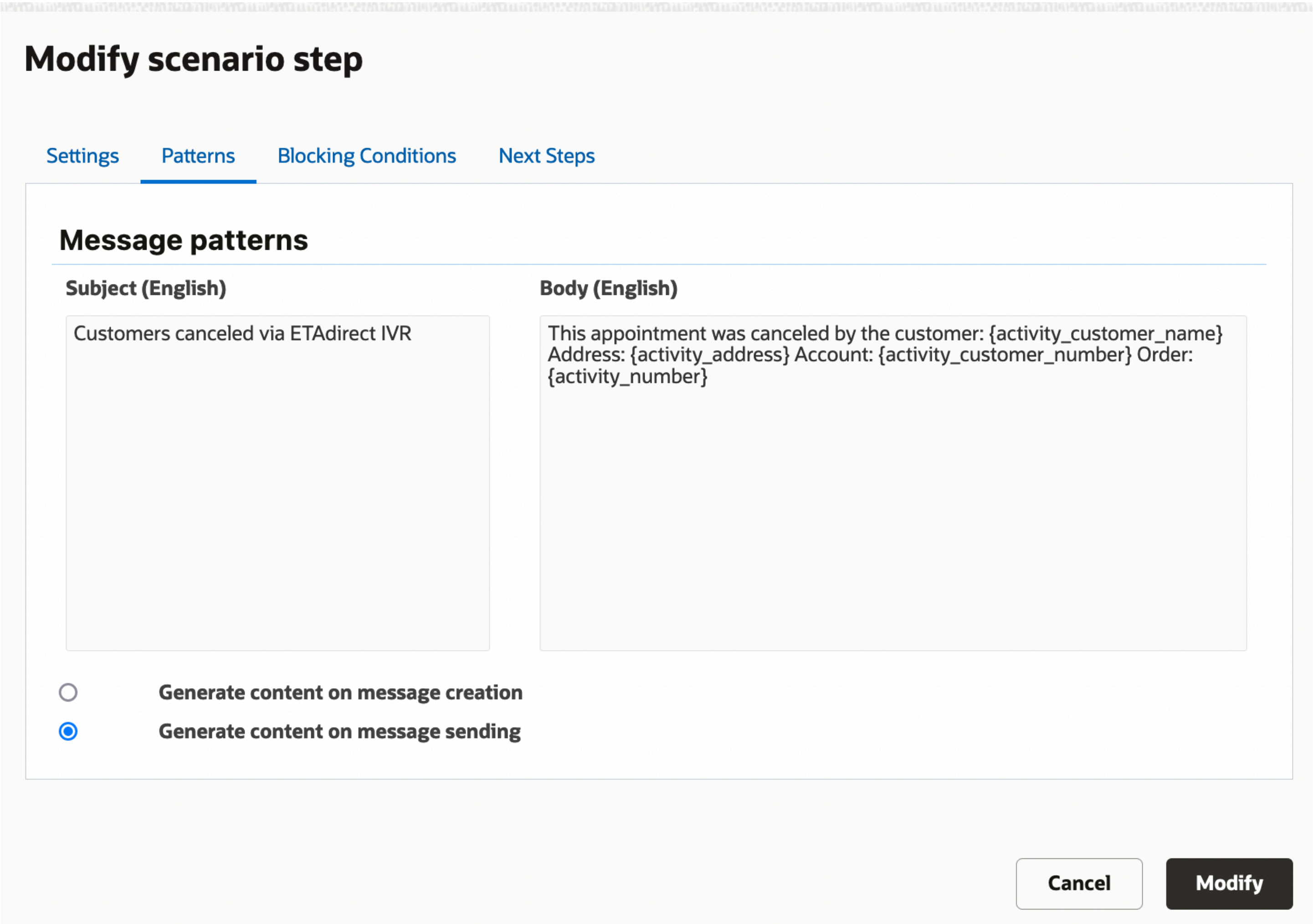
- cgi
- csv_item
- json
- sql_slashes

Here are a couple of examples for encoding placeholder values:

```
CSV: "{pr_address|csv_item}","{pr_comments|csv_item}","{pr_notes|csv_item}"
URL: address={pr_address|cgi}&comments={pr_comments|cgi}&notes={pr_notes|cgi}
```

Email notification pattern

When the delivery channel is email, you can define the subject and body for the message using placeholders to represent the actual value that will be inserted into the message.



Subject (English): Customers canceled via ETAdirect IVR

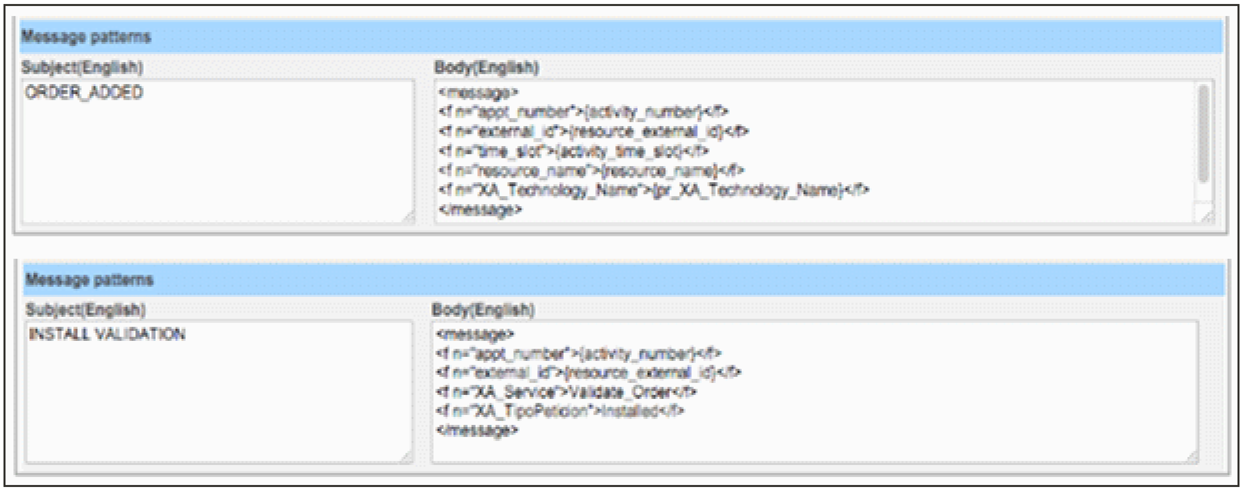
Body (English): This appointment was canceled by the customer: {activity_customer_name} Address: {activity_address} Account: {activity_customer_number} Order: {activity_number}

○ Generate content on message creation

● Generate content on message sending

External application notification pattern

External application notification patterns for the message body use XML, as shown in the figure.



Subject(English)

ORDER_ADDED

Body(English)

```
<message>
<f n="appt_number">{activity_number}</f>
<f n="external_id">{resource_external_id}</f>
<f n="time_slot">{activity_time_slot}</f>
<f n="resource_name">{resource_name}</f>
<f n="XA_Technology_Name">{pr_XA_Technology_Name}</f>
</message>
```

Subject(English)

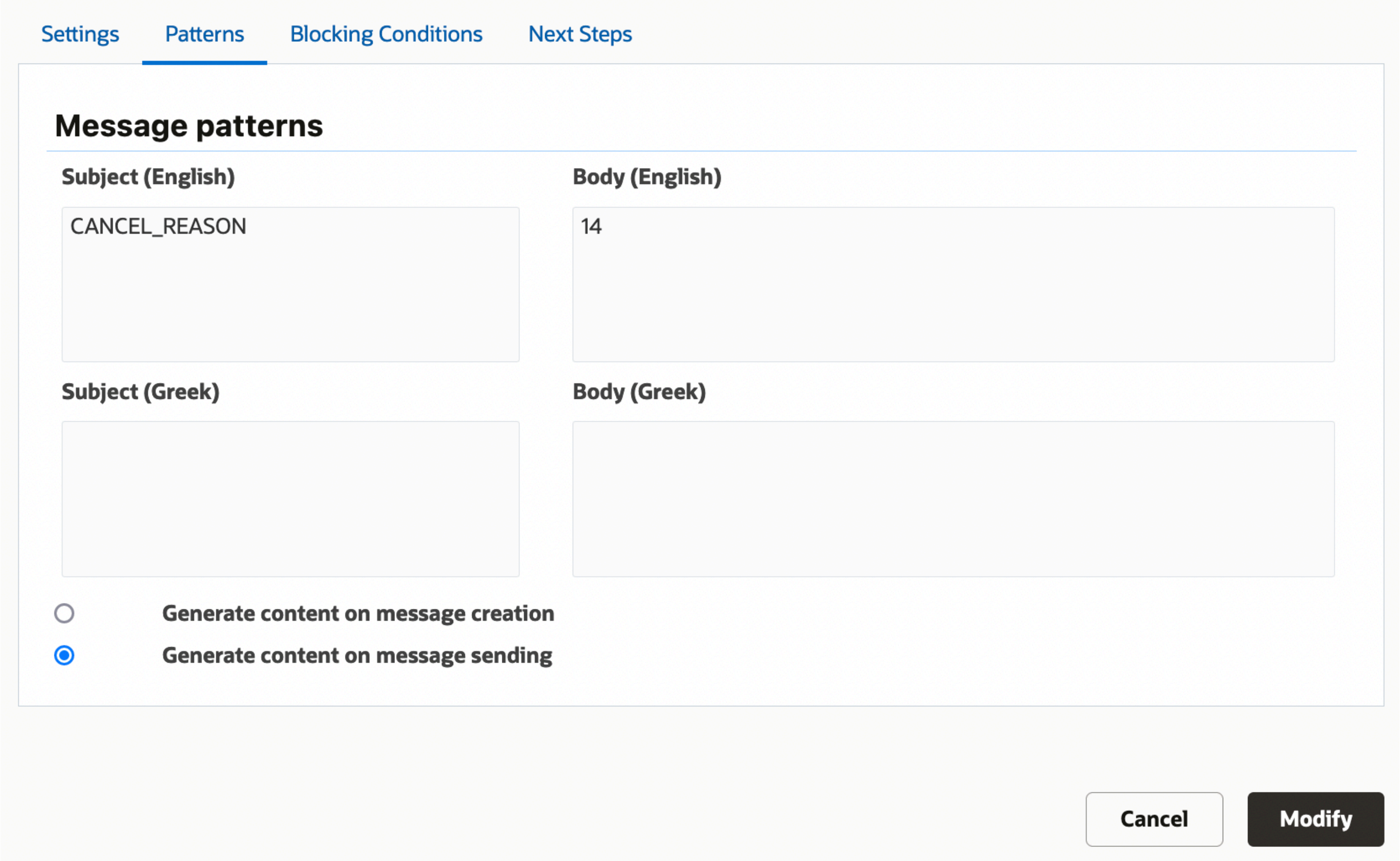
INSTALL VALIDATION

Body(English)

```
<message>
<f n="appt_number">{activity_number}</f>
<f n="external_id">{resource_external_id}</f>
<f n="XA_Service">Validate_Order</f>
<f n="XA_TipoPeticion">Installed</f>
</message>
```

Set property notification pattern

The following message for the **Set property** delivery channel sets the CANCEL_REASON property to the value with index14, indicating a customer request as defined in the property settings. – 'CUSTOMER REQUEST', as is defined in the property settings.



Message patterns

Subject (English): CANCEL_REASON

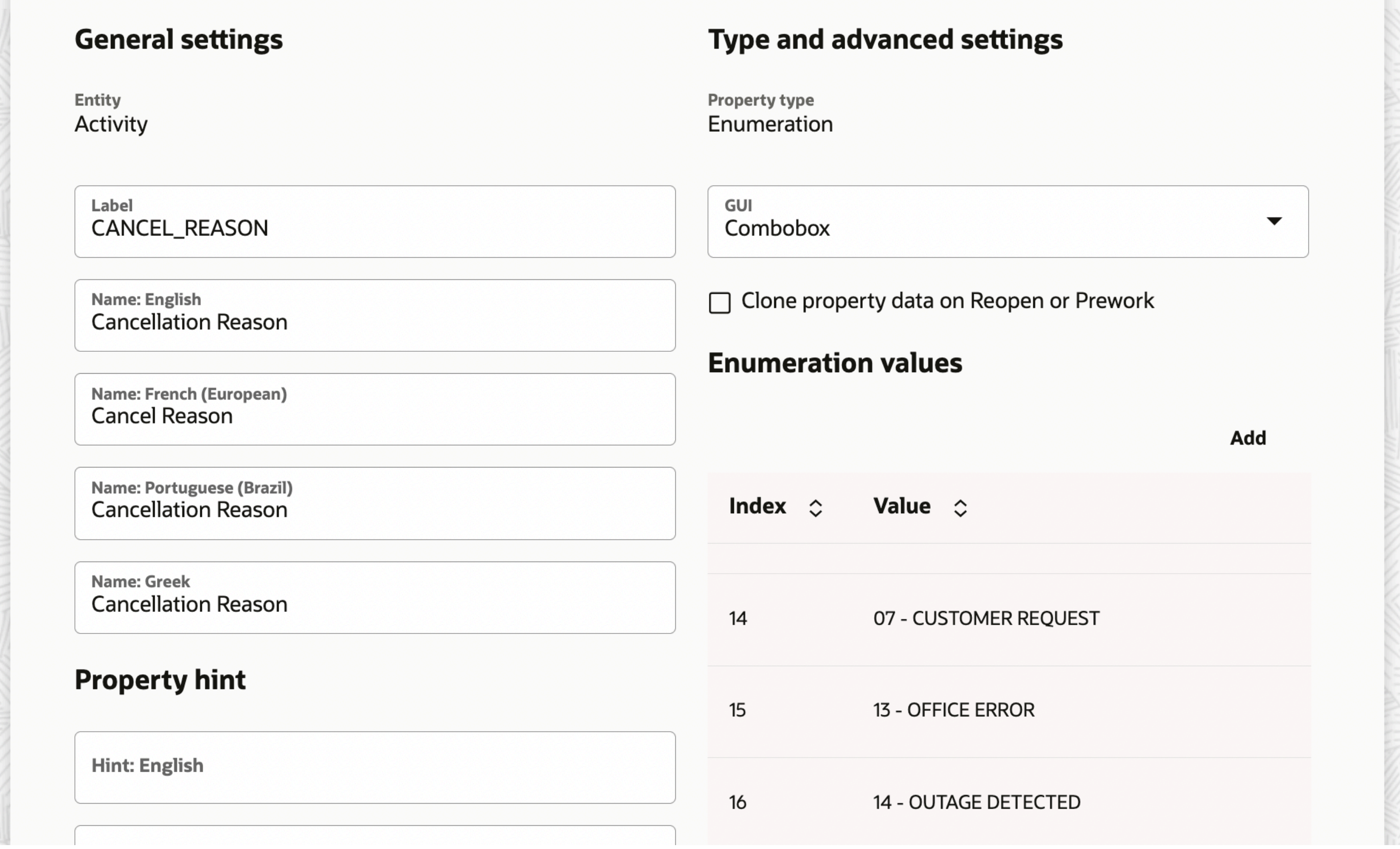
Body (English): 14

Subject (Greek):

Body (Greek):

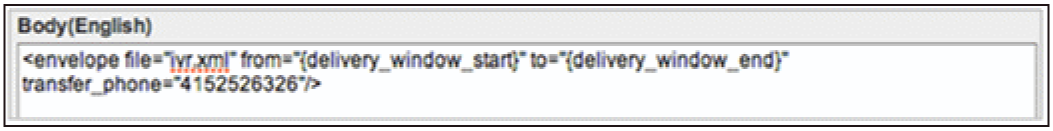
○ Generate content on message creation

● Generate content on message sending

> **Note:** Values for properties should be defined using an internal format. In particular, use **property_label** to define the subject value. For enumeration properties, use the **index** value in a message body rather than its corresponding translation. Translations for enumeration property values are shown in the following screenshot.
>
> 
>
> **General settings**
>
> Entity: Activity
>
> Label: CANCEL_REASON
>
> Name: English: Cancellation Reason
>
> Name: French (European): Cancel Reason
>
> Name: Portuguese (Brazil): Cancellation Reason
>
> Name: Greek: Cancellation Reason
>
> **Property hint**
>
> Hint: English
>
> **Type and advanced settings**
>
> Property type: Enumeration
>
> GUI: Combobox
>
> ☐ Clone property data on Reopen or Prework
>
> **Enumeration values**
>
> Add
>
> | Index | Value |
> |---|---|
> | 14 | 07 - CUSTOMER REQUEST |
> | 15 | 13 - OFFICE ERROR |
> | 16 | 14 - OUTAGE DETECTED |
>
> **Tip:** If you want to delete a custom property, add the property in the **Subject** field and leave the **Body** field blank.

External launch condition notification pattern

In external launch condition patterns, the body defines the activity information to be passed to an external application, as shown in the figure.



Body(English)

```
<envelope file="ivr.xml" from="{delivery_window_start}" to="{delivery_window_end}"
transfer_phone="4152526326"/>
```

Timing of message content generation

The **Patterns** tab also lets you define when the message content should be generated. The options are:

- **On message creation**—This option is intended for use when messages are related to synchronizing activity statuses and assignments. If multiple operations are performed for the same activity within a short time period, a separate message should be generated for each operation. Each message should contain activity details that are accurate at the moment of creation. For example, if several sequential move operations are performed, it might be necessary to include **from** and **to** values in all intermediate messages.
- **On message sending**–This option is recommended for messages that are generated in advance, such as **Day before** and other proactive customer notifications.

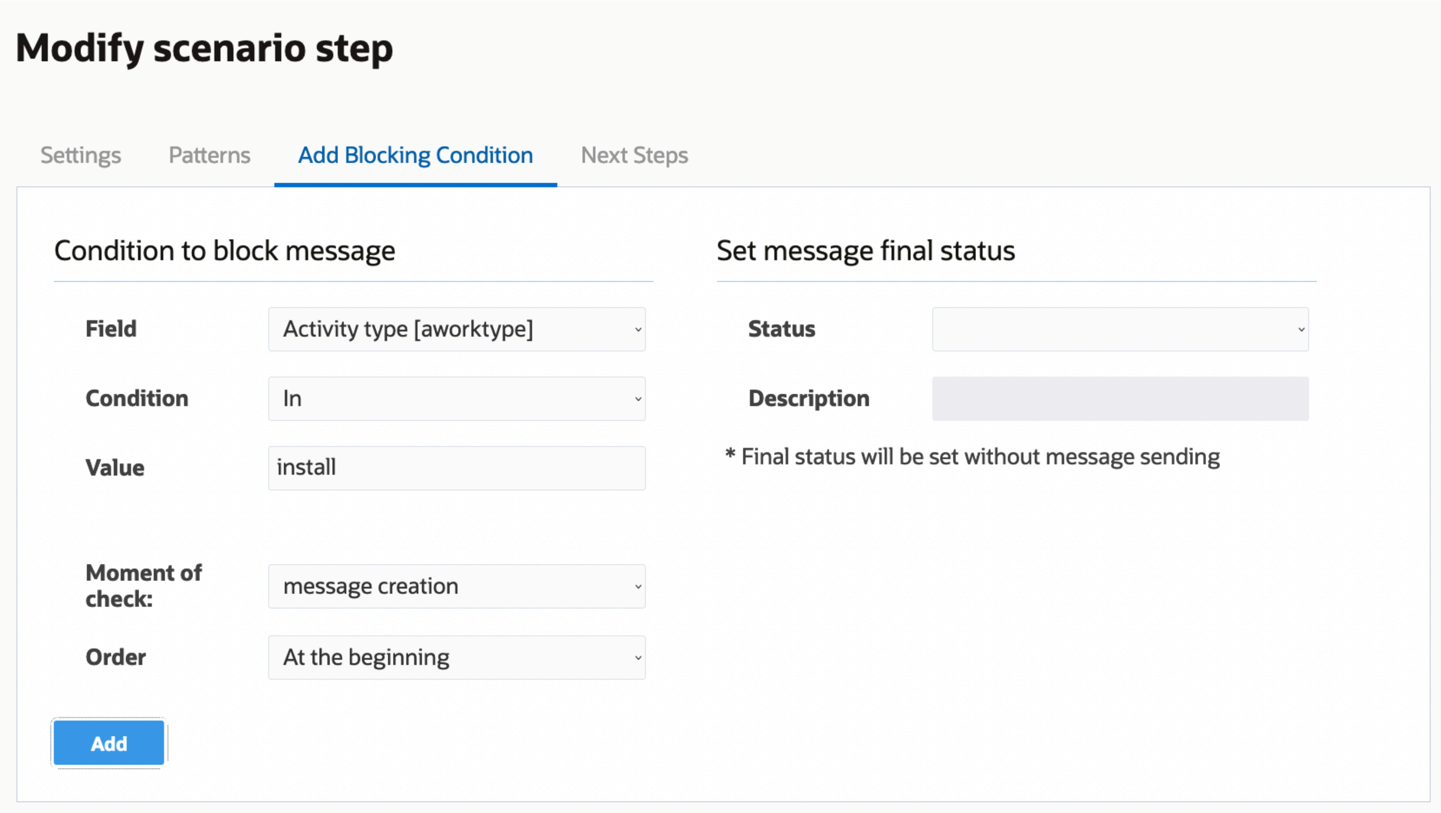
Complete the Conditions Tab

The **Conditions** tab is used to define the set of blocking conditions under which the message should not be sent.

To define blocking conditions:

1. Open the scenario step for which you want to define blocking conditions.
2. Click the **Conditions** tab.

   The **Add Blocking Condition** tab opens:



Modify scenario step

Condition to block message

Field: Activity type [aworktype]

Condition: In

Value: install

Moment of check: message creation

Order: At the beginning

Set message final status

Status:

Description:

* Final status will be set without message sending

3. Complete the fields as described in the following table.

Fields in the Add Blocking Condition tab

Field Name	Description
Condition to block message	
Field	Select the property from the menu that you want to use as the blocking condition. The menu includes all available properties related to activity, inventory, resource, service request, route, visit, user, and shifts and calendars. Additionally, several **other** fields, such as interface and day of the week, can be used for blocking conditions.
Check on	Choose one of the following options to define the time when the message's blocking conditions should be checked: ○ On creation—The conditions are checked at time of the message generation and not checked again. ○ On sending—The message is generated and stored in the message queue. The blocking conditions are checked at the time of message sending, which allows the system to account for any changes since the message was generated. ○ On creation and sending—The blocking conditions are checked at time of both message creation and sending.
Condition	Select the operator from the menu that relates the field you selected and the value you'll enter. For example, to block a message if the day hasn't changed, you'd select **Day changed?** for the field, enter **no** for the value, and select **in** from the **Condition** menu. The menu options include: In–The field value matches the entry in the **Value** field. Not in–The field value doesn't match the entry in the **Value** field. Contains—The field value contains the entry in the **Value** field. Does not contain—The field value doesn't contain the entry in the **Value** field. **Note:** Don't use the **Contains** or **Does not contain** option for enumeration properties. Is empty—The field value is null or undefined. Is not empty—The field value isn't null. > >= < <= These operators compare the field value to the entry in the **Value** field. They apply to integer, string, enumeration properties, and date/time fields. Does not start with—The field value doesn't start with the entry in the **Value** field. Starts with—The field value starts with the entry in the **Value** field.
Status	Select the final status that will be assigned to the message when the defined conditions are met. Setting the status lets the message be processed further in the scenario, even though the actual message creation is blocked by the defined conditions. The options include:

Field Name	Description
	○ Failed ○ Sent ○ Delivered ○ False Method ○ Obsolete **Note:** If you don't select an option from the **Result** menu, no message is created.
Set message final status	
Value	Type a value that will be compared to the field value with the following operators: **In**, **Not in**, **Contains**, **Does not contain**, **>**, **>=**, **<**, and **<=**. You can enter only one value for the **>**, **>=**, **<**, and **<=** operators. To enter multiple values for the other operators, separate the values with commas.
Description	Type a description that defines what happens when the message is blocked. **Note:** The **Description** field isn't enabled until you make a selection from the **Result** menu.

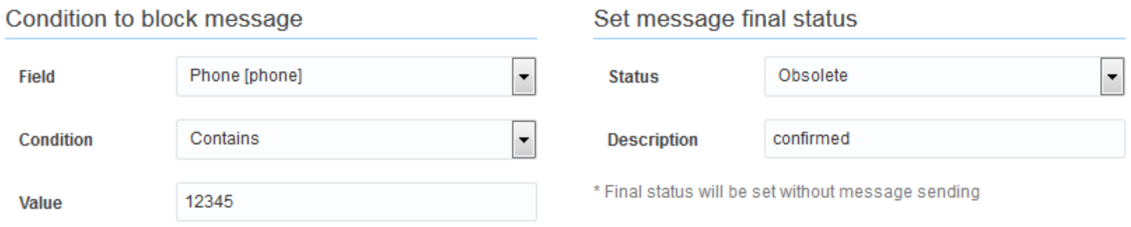
4. Click **Add** to add the blocking condition.

   The new blocking condition appears in the **Block Conditions** tab.
5. To add more blocking conditions, repeat steps 1 through 4.
6. To rearrange blocking conditions, select an option from the **Order** drop-down list in the **Add/Modify Blocking Condition** tab.

   The application processes the blocking conditions one by one in order of their appearance in the list in the right panel. Checking stops after the first blocking condition is met, so the order of the conditions may change the message-sending logic.
7. To delete a blocking condition, select it in the right panel and click **Delete**.

8. Click **Save** to save the blocking condition addition or changes to the scenario step.

   In this screenshot, the blocking condition has been set so that the customer with phone number 555–12345, who has confirmed the appointment earlier in another manner, will never receive any messages. Therefore, the message's final status will be set as **Obsolete** with the **Previously confirmed** description.



Condition to block message

Field: Phone [phone]

Condition: Contains

Value: 12345

Set message final status

Status: Obsolete

Description: confirmed

* Final status will be set without message sending

> **Note:** For troubleshooting message scenario configurations, ensure that you include properties used in Blocking Conditions and Message Scenario Steps to the **Monitored activity** and **Monitored inventory** fields. You can configure the properties on the **Configuration, Displays, Display, Activity history** page.

Add Next Steps

The **Next steps** tab defines what should happen when the scenario step is completed. You can specify different next steps that depend on the status of the message (for example, whether it was sent successfully or if it failed to send).

1. Open the scenario step for which you want to define the next steps.
2. Click the Next Steps tab.
3. Click the **Status** drop-down menu and select the status that will be the condition for triggering the next step.

   The available options include: Sent, Delivered, False Method, and Failed.
4. If you want the next step to depend on the description that was added on the **Conditions** tab, click the **Description** check box and type the description in the field.

   When the message step has the status you select and the description matches your entry in the **Description** field, the next step that you define will be initiated.
5. Click the **Next step** drop-down menu and select the step you want to initiate when the message step is completed and has the status you selected in step 3 and matches the optional description if you entered one in step 4.
6. Click **Add** to add the next step to the panel on the right side of the Next Steps tab.
7. To edit a next step, select it in the right panel, click **Edit**, and make the necessary changes to the **Status**, **Description**, and **Next step** fields.
8. To delete a next step, select is in the right panel and click **Delete**.
9. Click **Update** to save your changes on the **Next steps** tab
10. Click **Update** again to save the scenario step.

Delete a Scenario Step

You can remove scenario steps from a message scenario.

1. Click **Configuration**.
2. Click **Message Scenarios** under **Subsystems and Integrations**.

   The **Message Scenarios** screen opens.
3. Select the appropriate message scenario from the left panel.

4. Click the action link to the right of the message scenario step you want to delete, and select **Delete** from the drop-down menu.
5. Click **OK** in the confirmation dialog.

   The step is removed from the **Scenario steps** section of the page.

   > **Note:** If you delete the scenario's start step, an warning message appears under the scenario name in the left column. No message will be generated or sent for message scenarios that do not have a start step.

Scenario Step Warning Messages

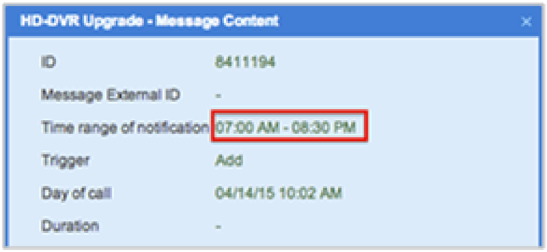
If a scenario step is configured improperly, a warning will appear when you try to save it.

Scenario Step Warning Messages

Message	Description
Message Step is not reachable	Appears when the message scenario step is not reachable, that is, when no other step triggers it as its next step. It can appear for a newly created inner step until it is configured as the next step of another step. **Note:** This message appears only for scenario steps that are designated as the inner type.
Endless loop	Appears when the steps create a loop, potentially causing an endless loop situation. Correct this error by modifying the step sequence as needed to prevent a loop.
Delivery Channel is missing	Appears when a step is configured with a delivery channel, but the system connectivity parameters are missing in the company configuration. Open the **Delivery channels** screen and configure the missing channel.
Delivery Channel has missing credentials	Appears when a delivery channel is misconfigured. Open the **Delivery channels** screen and configure the missing channel parameters.
Delivery Channel is inactive	Appears when an existing delivery channel is set as inactive. In this case, the step messages will not be sent to the external system. Activate the channel or select a different delivery channel.

Message Sending Interval

Each message has its own time interval within which it should be sent. The corresponding time boundaries are called 'Send from' and 'Send to'. They are displayed as **Time range of notification** in **Activity details** > **Messages tab** > **<activity_name>** > **Message Content**.



ID	8411194
Message External ID	-
Time range of notification	07:00 AM - 08:30 PM
Trigger	Add
Day of call	04/14/15 10:02 AM
Duration	-

The interval is initially determined on message creation and is automatically adjusted when the recipient's time zone is changed. Also, the 'Send from' value can be updated for a message that supports multiple 'Failed' or 'Sent' attempts. In this case, if the current attempt is 'Failed' (or 'Sent'), the 'Send from' is set to the time when the next attempt should performed.

The 'Send from' value is calculated by doing the following:

1. Determine the start time of the message sending interval without day shift (in the time zone of the recipient).

   One of these three options apply, depending on the value selected in the **Sending time** field:

   - Time of event: The current time in the recipient's time zone
   - Day of event: The current date in time zone of recipient + the 'from' value
   - Day of route: The route date + the 'from' value

2. Apply the day shift for the message (if needed) using the +/- operators in the **Sending time** field.

   a. Get a date part of the value determined at the previous step.
   b. Add day shift, which is the number of days before or after the corresponding event.
   c. Apply 'Block messages for specific days', 'Block messages for holidays', and 'Shift blocked messages' rules.

> **Note:** The 'Delay sending by' parameter is not involved in the 'Send from' calculation. It is checked on message sending.

The 'Send to' value is calculated using the following formula:

```
'Send to' = 'Send from' + the 'within' value
```

The 'New' messages are checked against the 'Send to' constraint. If the message has not been processed in time, the server updates it with the 'Failed' status and the 'MESSAGE_STEP_EXPIRED' description.

The system also checks the 'Send to' value for messages with the 'Sending' status. Maximum time to keep expired ('Send to' <= current time) messages in the 'Sending' state is 60 minutes. If the delay time has elapsed for a message, it is removed with the 'Failed' status and the 'MESSAGE_STEP_EXPIRED' description.

Also, the same 'Send to' logic is implemented in all notification methods. It is especially necessary for the messages that might remain in the 'Sending' state for a long time (e.g. customer notifications that require message delivery confirmation).

3 Understanding Delivery Channels

New Delivery Channels Screen

The Delivery Channels screen lets you configure all external system connectivity parameters on a single screen. To open the screen, click the **Channels** button on the Message Scenarios screen.

The Deliver Channels list uses color coding to indicate the number and status of message scenarios using each delivery channel. The following color codes are used:

- Green—No warnings and delivery channel is active.
- Gray—No warnings and delivery channel is inactive.
- Red—Warnings exist or the count is greater than zero and the channel is inactive.

The Advanced Settings section of the Delivery Channels screen is hidden if the **Connection** field is set to **Not encrypted**. Otherwise it is shown. The advanced settings include the following:

- x509 Trust File
- x509 CRL File
- Client Certificate
- Client Private Key

The format of the files in the Advanced Settings is PEM. You do not have to complete all fields. However, the **Client Certificate** and **Client Private Key** fields must both be either empty or completed.

Add a Delivery Channel

Add delivery channels for message scenarios on the **Delivery Channels** screen.

1. Click **Configuration**.
2. Click **Configuration** > **Message Scenarios**.
3. Click **Channels**.
   The **Delivery Channels** page opens and lists the existing delivery channels.
4. Click the plus sign.
   The page displays fields for entering general delivery channel information and end-point information.
5. Type the name of the delivery channel in the **Name** fields.
6. Select the required option from the **Status** drop-down list.

> **Note:** If notification scenarios contain at least one message step that uses an internal delivery channel (email or voice) then that channel is accessible in the list of channels. A user with appropriate permissions can select **Active** or **Inactive** to resume or stop the message delivery for any external or internal channel. For example, you can block a channel using the Inactive option in Test instances to disallow test messages to reach real customers. Messages that aren't delivered due to inactivated delivery channel get the status 'obsolete' with the description, EXTERNAL_NOTIFICATION_ARE_DISABLED. Note that the 'set property' messages don't have a delivery channel and can't be handled this way.

7. Type a value between 1 and 10,000 in the **Bulk Size** field to define the maximum number of messages per request.

   > **Note:** The default value is 10, which is also the recommended value.

8. Type a host field in the **Host/Port** field using the **example.com** format.
9. Type a port number in the section of the **Host/Port** field that appears after the colon.
10. Type a URL in the **URL Path** field.
11. Type a user name in the **User** field.
12. Type the user's password in the **Password** field.
13. Re-type the user's password in the **Confirm Password** field.
14. Select the **Allow basic access authentication** check box to implement HTTP basic authentication while integrating with external systems.

    When you select the check box, the outbound methods (such as send_message, drop_message, get_message_status methods) send the standard HTTP header "Authorization" with base64-encoded user credentials (standard basic access authentication). Also, the <user> SOAP structure is sent in the body of the request. The client application can either use the standard HTTP header "Authorization" or the <user> SOAP structure to send user credentials in the request.

    > **Note:** When the check box isn't selected, the standard HTTP header isn't used in the request and the client application can use the <user> SOAP structure for authentication. For more information, see the *Integrating with Outbound API Guide*.

    > **Note:** From the 19C release onward, the **Disable Weak Password** option is removed for delivery channels and all Outbound API integrations shall use the SHA256 algorithm for secured authentication. The delivery channels of the clients which used Weak Password Hashing (MD5) algorithm for Outbound API integrations, shall use the SHA256 algorithm for secured authentication.

15. Click the **Connection** menu and select an encryption method for the connection. The options include:
    - Not encrypted
    - Default encryption
    - SSL 3
    - TLS 1.0
    - TLS 1.1
    - TLS 1.2
16. If you selected any option except **Not encrypted**, the **Advanced settings** section opens to let you enter certificates and a client private key. Complete some or all the following fields:
    - x509 Trust File
    - x509 CRL File
    - Client Certificate
    - Client Private Key

    The format of the files in the Advanced Settings is PEM. You don't have to complete all fields. However, the **Client Certificate** and **Client Private Key** fields must both be either empty or completed.

    > **Note:** To configure and use mTLS connections for sending messages from Oracle Field Service to external systems, see *How do I configure an mTLS connection?*

17. Click **Save**.

    The newly created delivery channel appears.

Edit a Delivery Channel

You can edit any of the fields on an existing delivery channel.

1. Click **Configuration**.
2. Click **Message Scenarios** under **Subsystems and Integrations**.

   The **Message Scenarios** screen opens.
3. Click the **Channels** button in the upper right corner of the screen.

   The **Delivery Channels** screen opens, and existing delivery channels appear in the left column.
4. Select the delivery channel you want to edit.

   The right side of the screen displays information for the delivery channel.
5. Edit any of the fields for the delivery channel using the field descriptions found in *How do I add a delivery channel?*.
6. Click the **Save** button.

Delete a Delivery Channel

You can delete delivery channels that your organization no longer uses.

1. Click **Configuration**.
2. Click **Message Scenarios** under **Subsystems and Integrations**.

   The **Message Scenarios** screen opens.
3. Click the **Channels** button in the upper right corner of the screen.

   The **Delivery Channels** screen opens, and existing delivery channels appear in the left column.
4. Select the delivery channel you want to delete.

   A dialog opens, asking you to confirm the deletion.
5. Click **OK** in the Confirm dialog.

   The delivery channel no longer appears in the left column.

Delivery Channel Error Messages

If a delivery channel is configured improperly, an error message will appear when you try to save it.

The following error messages appear when fields are incomplete or wrong or when another error occurs:

- Unable to load Delivery Channel settings
- Unable to create a new Delivery Channel
- Unable to update Delivery Channel
- Unable to delete Delivery Channel

- Delivery Channel does not exist (Try to change or delete the channel.)
- Requested Delivery Channel is used by Message Scenarios and cannot be deleted
- Name of delivery channel already exists
- Bulk Size field must be in numeric symbols ranging from 1 to 10000
- Host field must have a corresponding format (http(s)://example.com)
- Port field must be in numeric symbols ranging from 1 to 65535
- User field is empty
- Password field is empty
- Confirm Password field is empty
- Password and Confirm Password do not match
- Client Certificate is empty
- Client Private Key is empty
- Mandatory field is empty
- {Field name} is wrong

4 Launch Conditions

Working with Launch Conditions

The Message Scenarios page displays information about launch conditions for the selected scenario. Launch conditions are the system events that trigger a message scenario.

Message scenarios are not initiated unless and until something happens to trigger them. The event that sets the message scenario in motion is called a launch condition, and these events are predefined by Oracle Field Service. Launch conditions include events such as the delay of an estimated arrival time, the reassignment of an activity, or the installation of inventory. You define the launch condition that sends a specific message to a designated recipient using the delivery channel you identify at the time you want it to be sent.

You can specify multiple launch conditions for a single message scenario, and whenever one of those conditions is detected, the message scenario begins to run through the steps of the scenario. However, while message scenarios can have multiple launch conditions, each launch condition can be associated with only one message scenario because the system has to know which scenario to launch when the launch condition is detected.

The two types of launch conditions include action-driven and condition-driven. Action-driven launch conditions are those events that immediately launch the message scenario when they occur, for example, activating a route or completing an activity. Condition-driven launch conditions are determined by polling the system periodically to see if certain conditions have been fulfilled and, if they have, then launching the message scenario. For example, Oracle Field Service checks to see whether the current time is within a notification window for the start of a delivery window or in the time frame during which an escalation notice should be sent. So there is not a specific action that launches the message scenario for condition-driven launch conditions, but rather the passage of time that indicates the message should be sent now.

The **Launch conditions** section displays the scenario’s existing launch conditions and lets you add and remove conditions. A numeric indicator lets you know how many messages are in the queue for each of the launch conditions and lets you estimate the potential results of any changes you make to the scenario.

> **Note:** If the number of messages is greater than 999, the indicator displays the number in thousands, denoted by the letter k. For example, 5,000 messages will be indicated as 5k. If the number of messages is 1,000,000 or greater, the indicator displays the number in millions, denoted by the letter m. For example, 12,000,000 messages is displayed as 12m.

You can remove a launch condition by clicking the x in the upper right corner of the condition indicator.

Day Before Launch Condition

The *day before* is an action-driven launch condition that generates messages to customers that should be sent before the activity within the predefined time period.

The *day before* launch condition is intended for use with PCC (predictive customer communications) messages. It is invoked when a new activity is created or when an existing activity is rescheduled. The launch condition is not invoked for non-scheduled, reopened, or prework activities. The launch condition generates messages only for the activities for which the **Enable 'day before' trigger** feature is enabled in **Company settings** > **Activity types**.

The day-before messages become obsolete after the following activity-related actions: `cancel`, `delete`, `start`, `suspend`, `reschedule`, as well as upon the creation of the `Reminder` and the `Change` messages with the `Customer` recipient. No day-before messages are generated on the activity move between resources within the same day.

> **Note:** For more information, see the *Add a launch condition for a message scenario* section.

Reminder and Change Notification Launch Conditions

Reminder and change are condition-driven launch conditions intended for use with PCC messages.

These launch conditions invoke message scenarios at the moment of time set in the launch condition configuration. The time is calculated in relation to the `Notification base` that is configured in the **Reminder and change notifications** selector. It can be one of the following activity fields:

- **ETA**
- **delivery window start**
- **service window start**

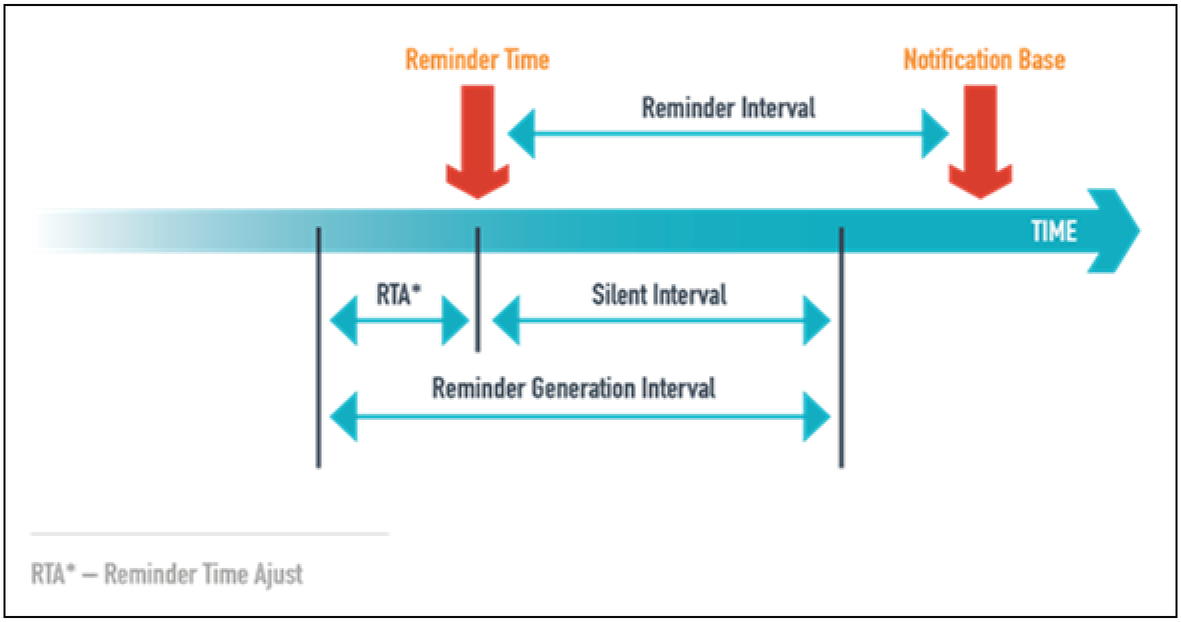
The Reminder launch condition is intended to generate a reminder message to a customer which is to be sent within a fixed time prior to an activity. The **allowed reminder time** values (for example, **45**, **60**, **90**) will be shown as the **Reminder** field values in the **Activity details** screen.

> **Note:** Subject to user visibility settings, the **Reminder** field may appear in read-only mode.

There are up to five Change launch conditions which can be used to send a message if the activity delivery time has changed.

In order for the Reminder and Change messages to be sent, the **Enable reminder and change triggers** feature should be enabled for the corresponding activity type. At the same time, both launch conditions can be invoked only for a pending ordered activity (regular or reopened) in an activated route on the current working day. The `Reminder` message is generated within the time period from `Reminder Time +`

`Reminder Time Adjust` to `Reminder Time – Reminder Silent Interval` before the activity Notification base.



The `Reminder` time is defined on the activity level using the **Reminder** field in the **Activity** details screen. When the **Reminder** time is set to 0, the `Reminder` launch condition is disabled for the selected activity. The **Reminder Time Adjust** is the system predefined time shift for the reminder, which is used to eliminate the delay in message processing. For example, if **Reminder Time Adjust** = 3 minutes and **Reminder Time** = 60 minutes, the reminder will be generated 63 minutes prior to the Notification base. The value of **Reminder Time – Reminder Silent Interval** defines the minimal time before the Notification base is to generate the Reminder.

The Reminder launch condition can be invoked only once for the same activity. Each of the Change launch conditions is defined by its own time interval, based on the time remaining to the Notification base, and the threshold value (in minutes). The Change message is initiated when the current time is within a specific Change launch condition time interval, and the difference between the current value of the Notification base and the time previously reported to the customer is equal or higher than the threshold.

The time intervals of different Change launch conditiontriggers should be configured in a way that they do not overlap. The Change launch condition is not invoked if:

- the Reminder is not sent and the current time is within the **Reminder Silent Interval**
- an incomplete Reminder exists
- a Change message has already been sent and the last Change message was sent by the same change trigger.

Before generating messages, the Reminder and Change launch conditions try to cancel all existing customer messages (if any). They use the `drop_message` SOAP function to cancel messages that are in the Sending status. If the corresponding agent is not accessible, or it returns a result indicating that the message is under processing and cannot be dropped, the generation of the Reminder and Change messages is stopped and will repeat during the next cycle.

Change and Reminder messages are removed from the message queue if one of the following events occurs after they have been generated and sent:

- the Activity status is changed
- the Activity becomes not ordered
- the Activity is moved

> **Note:** For more information, see the *Add a launch condition for a message scenario* section.

Launch Conditions Related to Route and Activity Actions

Most of these action-driven launch conditions are invoked by the resources manual action.

Following is the brief overview of the launch conditions:

- **Activate –** invoked on a route activation
- **Deactivate –** invoked on a route deactivation
- **Reactivate –** invoked on reactivation of a previously deactivated route
- **Add –** invoked when a new activity is added or an existing activity is moved to a different day or resource. This launch condition is not invoked for prework activities (regular and reopened only).
- **Start –** invoked on activity start. This launch condition is not invoked for prework activities (regular and reopened only)
- **Complete –** invoked on completion of a started activity
- **Cancel –** invoked when an activity is canceled
- **Not done –** invoked when the status of a started activity is changed to not done
- **Delay –** invoked on activity routing run delays if the activity duration after the `Delay` action exceeds the threshold value (in minutes)
- **Suspend –** invoked when a started or a pending activity is suspended. This launch condition is not invoked for pre-work activities

> **Note:** If the suspend action is performed for a started activity, a new `suspended` activity is created, and the launch condition is invoked for the newly-created activity (with the suspended status). In this case, both the pending and the suspended activities have the same property values, and the suspended activity has no inventory.

A *Move activity* is invoked when an existing activity is moved to another resource or day. This launch condition is not invoked for prework activities.

The messages generated by the Move activity launch condition refer to the origin resource. In order to retrieve information about the destination resource, use the `destination_resource` block

In order to retrieve information about the destination resource and date, use the following placeholders:

- *`destination_resource_id`*
- *`destination_resource_external_id`*
- *`destination_resource_name`*
- *`destination_date`*

The *`Resource changed?`* and the *`Day changed?`* message blocking conditions can be used to determine if the move is to be performed to another resource or to another day.

> **Note:** For more information, see the *Add a launch condition for a message scenario* section.

What escalation message launch conditions are available?

These condition-driven launch conditions provide notifications about activities that are not started on time.

Not started 1 – This launch condition is invoked when an activity has not been started within a certain time after ETA.

Not started 2 – This launch condition is similar to **Not started 1**. It allows setting the second message with a different delay time. The `Not started` launch condition can only be invoked for a pending ordered activity (regular or reopened) in an activated route which belongs to the current working day. The **Enable not started trigger** feature should be enabled for the corresponding activity type. The **Not started 1** and **Not started 2** messages are independent and can be generated for the same activity at the same time.

Not activated – This message is sent if the resource has not activated their route the defined number of minutes after the planned start of the working day, according to the calendar. It works only once a day for a specific route. The messages are not generated again if the calendar has been changed. If any not activated in time messages are present for the resource at the moment of route activation, these messages become obsolete. For a new resource the Not activated in time messages are only generated on the next day (in the company time zone) after its creation. This launch condition is only invoked for resources with the **Enable Not activated in time alert and trigger** feature enabled.

Service window warning - This message is intended to notify of a possibility to lose the service window. It is sent in the case when the Activity has not been started within the defined number of minutes before the service window end.

This trigger is only invoked for pending ordered activities (regular or reopened) in an activated route with a service window that belongs to the current working day. This trigger is invoked only once per activity. You should also activate the **Enable SW Warning trigger** feature for the corresponding activity type.

Configure this trigger using the threshold parameter near the **Service window warning** trigger selector on the **Notification Triggers** screen. It defines the number of minutes before the end of service window that is used in the condition.

SLA window warning – This message is intended to notify of a possibility to lose the SLA window. This trigger is only invoked for pending or started activities (regular or reopened). The warning is sent if the activity has not been started (for pending activities) or completed (for started activities) the defined number of minutes before the SLA window end. It is invoked only once per activity. But, if SLA window end has changed after the generation of the SLA window warning messages, the trigger can be invoked again.

Configure this trigger using the threshold parameter (**hours/minutes**) near the **SLA window warning trigger** selector on the **Notification Triggers** screen. It defines the number of minutes before the end of the SLA window that is used in the condition.

> **Note:** For more information, see the *Add a launch condition for a message scenario* section.

Manual Launch Condition

The `Manual` action-driven launch condition generates messages on the creation of a service request.

When creating a service request, a user should select the service request type and fill in the fields related to it. The message subject/body pattern for the `Manual` launch condition can contain placeholders that are related to the service request itself and to all its parent objects. For example, if the request is created on the activity level, the content can contain placeholders related to request, activity, route and resource.

The launch condition can be used in the following cases:

- inventory tracking and hardware testing
- I/initiating SRO or sending any other form
- initiating support requests
- manually generated cases when transaction is initiated by a person and is not related to the activity
- other activity or inventory requests

> **Note:** For more information, see the *Add a launch condition for a message scenario* section.

How does the Call Ahead launch condition work

This action-driven launch condition is initiated when a technician completes the previous activity or marks an activity as En route.

The launch condition is typically used to:

- Inform the customer.
- Initiate provisioning or hardware test while the resource is on the way.

When configured, the **Complete Activity** page has the mandatory Next Activity field. The resource is to select the next activity. When the resource submits the Complete Activity form, the application generates the `Call ahead` trigger for the next activity (selected by the resource).

Similarly, if the En route option is enabled and a technician sets the status of an activity as En route, the application generates the `Call ahead` trigger for the next activity in the route. You can use this trigger to notify your customers.

> **Note:** For more information, see the *Add a launch condition for a message scenario* section.

Inventory-Related Launch Conditions

These action-driven launch conditions are used to communicate inventory operations to an enterprise resource planning system and perform automated provisioning.

- **Install inventory**– invoked when the inventory is moved from the `resource` pool to the `install` pool or when a new `install` inventory record is created (other than those in the resource pool)
- **Deinstall inventory**– invoked when the inventory is moved from the customer pool to the `deinstall` pool or when a new `deinstall` inventory record is created (other than those in the customer pool)

- **Exchange inventory–** invoked when an inventory exchange between the resource and the customer pool is performed
- **Undo install inventory–** invoked when the inventory is moved from the `install` pool to the resource pool
- **Undo deinstall inventory–** invoked when the inventory is moved from the `Deinstalled` pool to the customers pool
- **Move inventory–** invoked when inventory is moved between different resources. It is invoked when an inventory item that belongs to a resource of a current user is moved by this user to another resource via SmartCollaboration. Note that the destination resource can be invisible to the current user due to visibility restrictions.

The messages generated by the `Move inventory` trigger refer to the `origin` resource. In order to retrieve information about the `destination` resource, use the `destination_resource` block.

The following placeholders can be used to retrieve information about the destination resource:

- `destination_resource_id`
- `destination_resource_external_id`
- `destination_resource_name`

> **Note:** For more information, see the *Add a launch condition for a message scenario* section.

Visit-Related Launch Conditions

The system can generate messages related to groups of activities called *visits*.

The following group of condition-driven launch conditions is intended to be used for PCC messages to prevent the same notifications sent to the same customer on different activities within the same visit.

- Visit Day Before – similar to the activity Day before launch condition but is applied to a visit. The only difference is that the message is created with a 5-minute delay after the visit creation to let the system accept other activities that compose the visit. The delay is also useful to prevent generation of messages for temporary visits, which may be created when visit activities are moved between resources one by one.
- Visit Reminder – similar to the activity Reminder launch condition, but is applied to a visit. This launch condition is intended to generate a reminder message to a customer which is to be sent a fixed time prior to a visit. It can be invoked only for pending visits and only once for the same visit. In addition, the first customer-related activity in the visit must be an ordered activity in an activated route for the current working day. The Visit reminder message is generated within the time period from [Visit Reminder Time + Reminder Time Adjust] to [Visit Reminder Time – Visit Reminder Silent Interval] before the Notification base.

Parameters:

Notification base – ETA/delivery window start/service window start from the first customer-related activity in the visit. A certain field selection is configured in the Notification Triggers screen using the Visit reminder and change notifications selector.

Visit Reminder Time is the time before the Notification base to send the reminder. As opposed to the regular launch conditions, this parameter is defined on the company level.

Reminder Time Adjust is the system predefined reminder time shift to be used to eliminate the messages processing delay.

For example: if Reminder Time Adjust = 3 minutes and Visit Reminder Time = 60 minutes, the reminder will be generated 63 minutes prior to the Notification base.

The value of (Visit Reminder Time – Visit Reminder Silent Interval) defines the minimal time before the Notification base to generate the reminder.

- `Visit change 1...Visit change 5` – similar to the activity Change launch conditions, but applied to a visit. There are up to five Visit change launch conditions which can be used to send a message if the visit delivery time has changed.

The Visit change launch condition can be invoked only for pending visits. In addition, thefirst customer-related activity in the visit must be an ordered activity in an activated route for the current working day.

The Visit change message is initiated when the current time is within the time interval of a specific Visit change launch condition, and the difference between the current value of the Notification base and the time previously reported to the customer is equal or higher than the threshold.

The time intervals of different Visit change launch conditions should be configured in a way that they do not overlap.

The Visit change launch condition is not invoked if:

- Visit reminder is not sent and the current time is within the Visit Reminder Silent Interval
- Incomplete Visit reminder exists
- Visit change message has already been sent and the last Visit change message was sent by the same visit change launch condition

Before generating messages, the Visit reminder and Visit change launch conditions try to cancel all existing customer messages (if any). They use the drop_message SOAP function to cancel messages that are in the Sending status. If the corresponding agent is not accessible, or it returns a result indicating that the message is under processing and cannot be dropped, the generation of the Visit reminder and Visit change messages is stopped and will repeat during the next cycle.

- `Visit cancel` – initiated when the last of the activities in a visit is canceled (none are pending, completed, or not done). This launch condition can be invoked several times for the same visit when new activities are added to an existing canceled visit.
- `Visit complete` – initiated when the last pending activity in a visit gets a final status, and at least one activity is completed or notdone. This launch condition initiates the PAS scenario and is invoked only once, when the visit becomes completed or notdone for the first time.

Visit launch conditions are designed to be used mostly for customer interactions reducing the number of day before calls, reminders, change messages and PAS messages to the same customer.

> **Note:** For more information, see the *Add a launch condition for a message scenario* section.

5 Setting Properties

What's the Set Property method?

One of the delivery channels for a scenario step is **Set Property**. When you select this method, the message scenario can update all company-defined properties, except file properties.

The `set_property` method can update the following entities:

- Activity (except mass/repeating)
- Inventory
- Resource
- User

The property values must use an internal format. In particular, use the index values for enumeration properties rather than their translations. In the **Subject** field, specify a label for the property you are assigning. Do not use a **pr_** or **prnum_** prefix for the label.

The `set_property` method supports the following predefined list of activity fields. Additional predefined fields are not available.

- appt_number
- customer_number
- customer_name
- cell (or “sms”)
- address
- city
- state
- zip
- email
- phone
- points
- time_slot
- service_window (hh:mm; hh:mm)
- sla_window_start
- sla_window_end
- action (Only the `cancel_activity` and `unschedule_activity` action are supported. See the following sections for more details.)

Cancel Activity Actions

The `set_property` method supports canceling a pending activity with the `cancel_activity` action. If the activity is non-scheduled, it is moved to the current resource's day and then canceled. Note that the message should contain the word "action" in the subject, and the name of the action (`cancel_activity` or `unschedule_activity`) should be stored in the message body. Also note that the `cancel_activity` action does not support the `notdone` and `deleted` actions depending on the activity and route status. After processing, the message is assigned one of the following statuses and its corresponding description.

Status 'Delivered'

Status	Description
Status	delivered
Description	
Notes	The action is performed successfully.

Status 'Failed'

Status	Description
Status	failed
Description	INVALID_RESOURCE_TYPE
Notes	This action is not supported for this type of resource. For example, the resource or its type is reconfigured, and activities can no longer be assigned to it.

Status 'Failed'

Status	Description
Status	failed
Description	NO_PERMISSION
Notes	This user has no access to the specified resource or activity.

Status 'Failed'

Status	Description
Status	failed
Description	RESOURCE_NOT_FOUND

Status	Description
Notes	The system is unable to find the resource the message refers to.

Status 'Failed'

Status	Description
Status	failed
Description	ACTIVITY_NOT_FOUND
Notes	The system is unable to find the activity the message refers to.

Status 'Failed'

Status	Description
Status	failed
Description	INACTIVE_RESOURCE
Notes	The resource is inactive.

Status 'Failed'

Status	Description
Status	failed
Description	NON_WORKING_DAY
Notes	The resource has a non-working day. For example, this error can appear for a not-scheduled activity that cannot be moved to the resource's current day because it is a non-working day for that resource.

Status 'Failed'

Status	Description
Status	failed
Description	ACTION_ON_PAST
Notes	The activity is located in the route for a day in the past that cannot be changed.

Status 'Failed'

Status	Description
Status	failed
Description	MISCONFIGURED_CALENDAR
Notes	The system is unable to determine if the resource is working because its calendar is not configured. For example, this error can appear for a not-scheduled activity.

Status 'Failed'

Status	Description
Status	failed
Description	MISCONFIGURED_TIME_ZONE
Notes	The system is unable to determine a current time for the resource because its time zone is not configured. For example, this error can appear for a not-scheduled activity.

Status 'Failed'

Status	Description
Status	failed
Description	INVALID_ACTIVITY_STATUS
Notes	This action is not supported for an activity with such status. Only `pending` activities can be canceled.

Status 'Failed'

Status	Description
Status	failed
Description	NON_MOVABLE_ACTIVITY
Notes	This type of activity cannot be moved to another resource or a day. For example, this error can appear for a not-scheduled activity.

Status 'Failed'

Status	Description
Status	failed
Description	RESCHEDULE_IS_NOT_ALLOWED

Status	Description
Notes	This type of activities cannot be rescheduled to another day. For example, this error can appear for a not-scheduled activity.

Status 'Failed'

Status	Description
Status	failed
Description	NOT_SCHEDULED_IS_NOT_ALLOWED
Notes	This type of activities cannot be not-scheduled. For example, this error can appear for a not-scheduled activity.

Status 'Failed'

Status	Description
Status	failed
Description	DEACTIVATED_ROUTE
Notes	The route is deactivated. For example, this error can appear for a not-scheduled activity if it cannot be moved to the resource's route for the current day because this route is deactivated.

Unschedule Activity Actions

The set_property method unschedules a pending regular and reopened activity with the unschedule_activity action. The message should contain "action" in the subject and the name of the action to be performed ('unschedule_activity') in the message body. Note that the unschedule_activity action is not supported for individual segments and segmentable activities. After processing, the message is assigned one of the following statuses with its corresponding description:

Status 'delivered'

Status	Description
Status	delivered
Description	
Notes	The action is performed successfully.

Status 'Failed'

Status	Description
Status	failed
Description	INVALID_RESOURCE_TYPE
Notes	This action is not supported for this type of resource. For example, the resource or its type are reconfigured and activities cannot be assigned to it anymore.

Status 'Failed'

Status	Description
Status	failed
Description	NO_PERMISSION
Notes	This user does not have access to the specified resource or activity.

Status 'Failed'

Status	Description
Status	failed
Description	RESOURCE_NOT_FOUND
Notes	The system is unable to find the resource the message refers to.

Status 'Failed'

Status	Description
Status	failed
Description	ACTIVITY_NOT_FOUND
Notes	The system is unable to find the activity the message refers to.

Status 'Failed'

Status	Description
Status	failed
Description	INACTIVE_RESOURCE
Notes	The resource is inactive.

Status 'Failed'

Status	Description
Status	failed
Description	ACTION_ON_PAST
Notes	The activity is located in a route for a day in the past that cannot be changed.

Status 'Failed'

Status	Description
Status	failed
Description	INVALID_ACTIVITY_STATUS
Notes	This action is not supported for an activity with such status. Only **`pending`** activities can be unscheduled.

Status 'Failed'

Status	Failed
Status	failed
Description	INVALID_ACTIVITY_STATUS
Notes	This action is not supported for an activity with such status. Only **`pending`** activities can be unscheduled.

Status 'Failed'

Status	Description
Status	failed
Description	INVALID_ACTIVITY_TYPE
Notes	This action is not supported for this type of activity. Only **`regular`** and **`reopened`** activities can be unscheduled.

Status 'Failed'

Status	Description
Status	failed
Description	NON_MOVABLE_ACTIVITY

Status	Description
Notes	This type of activity cannot be moved to another resource or day. For example, this error can appear for a not-scheduled activity.

Status 'Failed'

Status	Description
Status	failed
Description	RESCHEDULE_IS_NOT_ALLOWED
Notes	This type of activity cannot be rescheduled to another day. For example, this error can appear for a not-scheduled activity.

Status 'Failed'

Status	Description
Status	failed
Description	NOT_SCHEDULED_IS_NOT_ALLOWED
Notes	This type of activity cannot be not-scheduled. For example, this error can appear for a not-scheduled activity.

Setting Properties Using Agent Message Status

Using functionality similar to the `set property` method, agents can update properties and perform actions without additional scenario steps. All activity and inventory statuses and the predefined set of activity fields can be updated, and activities can be canceled and unscheduled.

This functionality allows:

- Updating properties based on processing the result of a PAS scenario message and controlling further scenario flow based on these properties
- Setting the `ICOMS` result property directly from the ICOMS agent without additional **Set Property** scenario steps
- Canceling an activity based on customer selection

The fields to be assigned and the corresponding values are passed in the `data` field. The `#params?` string is used as the delimiter between `data` itself and the passed parameters. The format of the parameter line is similar to URL. The `&` character is used as the delimiter between different parameters.

```
data = OK#params?cconfirmed=1&phone=12345678912
data = #params?icoms_eta_status=1
data = DONE#params?pas_requires_feedback=1
data = #params?action=cancel_activity&notes=Rescheduled
```

Names and values of the parameters are encoded as the URL parameters. So, if the URL property needs to be assigned to **http://localhost/v41/wwwroot/?param1=value1&param2=value2**, the resulting data value should be as follows:

```
data = OK#params?CANCEL_REASON=0&URL=http%3A%2F%2Flocalhost%2Fv41%2Fwwwroot%2F
%3Fparam1%3Dvalue1%26param2%3Dvalue2
```

The encoding requires that all non-ASCII and non-alphanumeric characters (with some exceptions) must be represented as hexadecimal values with leading '%'. Exceptions: the '.', ',', '-', '_' characters should be provided as is, and the ' ' (space) symbol is replaced with the '+' (plus).

The total length of the data field is restricted to 255 characters. If a submitted data value exceeds that limit, it can be processed correctly, but it will be truncated.

Status Descriptions for Set_Property Method

Messages processed by the set_property method are assigned a status.

Set Property action details for cancel_activity

In order to invoke an action in the set_property method, the subject field of the message should contain the action word. The name of the action to be performed should be in the body field.

The cancel_activity action cancels a pending activity. If the activity is not scheduled, it is moved to the current resource’s day and then canceled.

> **Note:** This action does not support the notdone and deleted actions depending on activity and route status. The Set Property message steps become OBSOLETE in case of Not Done.

After processing, the message is assigned one of the following statuses.

Status ‘Delivered’

Status	Description
Status	delivered
Description	ALREADY_SET
Data	
Comments	The property is not changed. It is already assigned to the given value.

Status ‘Delivered’

Status	Description
Status	delivered
Description	INSERTED

Status	Description
Data	
Comments	An empty property is assigned to the given value.

Status 'Delivered'

Status	Description
Status	delivered
Description	DELETED
Data	
Comments	An existing property is assigned to empty.

Status 'Delivered'

Status	Description
Status	delivered
Description	UPDATED
Data	
Comments	An existing property value is changed.

Status 'Failed'

Status	Description
Status	failed
Description	WRONG_CONFIGURATION
Data	Unknown label
Comments	The given label of a property/field is not found in the dictionary.

Status 'Failed'

Status	Description
Status	failed
Description	WRONG_CONFIGURATION
Data	Not supported entity

Status	Description
Comments	An attempt to update a property for an entity that is not supported.

Status 'Failed'

Status	Description
Status	failed
Description	WRONG_CONFIGURATION
Data	Not supported field
Comments	The given field cannot be updated using this method.

Status 'Failed'

Status	Description
Status	failed
Description	WRONG_CONFIGURATION
Data	Parent entity mismatch
Comments	The message has no relation to the requested entity.

Status 'Failed'

Status	Description
Status	failed
Description	WRONG_CONFIGURATION
Data	Unknown ID of the parent object
Comments	The system is unable to find an object by ID retrieved from the message.

Status 'Failed'

Status	Description
Status	failed
Description	WRONG_CONFIGURATION
Data	Invalid integer value
Comments	The given value of an integer property cannot be converted to `signed int32`.

Status 'Failed'

Status	Description
Status	failed
Description	WRONG_CONFIGURATION
Data	Invalid lookup value
Comments	The given value of a lookup property cannot be found in the list of items for this property.

Status 'Failed'

Status	Description
Status	failed
Description	WRONG_CONFIGURATION
Data	Invalid value

Status 'Failed'

Status	Description
Status	failed
Description	WRONG_CONFIGURATION
Data	Missing survey take

Status 'Failed'

Status	Description
Status	failed
Description	INTERNAL_ERROR
Data	Unknown result code
Comments	A result of the property update is not handled properly.

6 Message Content

Message Content Overview

When you are creating content for your messages, you can use placeholders, blocks, and formatting rules that are replaced with actual values when the message is created.

- Placeholder—The field name, that, when included in a message, is replaced with the actual field value for a particular message instance. You can use placeholders to create a single message pattern for different instances, and to create consistent, personalized messages for your customers. For example, in a message to a customer about a delayed start to a service activity, you may want to include the service ticket number, name of technician, and new estimated time of arrival. Refer to *Placeholders*.
- Block statements—The message engine tools used to iterate through a list of similar objects (visit orders, activity inventory) and print the same section of the pattern for every object from the list. Refer to *What are the types of message blocks available in Field Service?* and *Examples of message blocks*.
- Formatting rules—Rules that define date and time formatting patterns for use within a message. Refer to *Date and Time Formatting Patterns*.

Placeholders

Use placeholders in messages when you want the message to include an actual value for an entity when the message is generated. Placeholders let you create a single message pattern that replaces the placeholders with information specific to each instance, such as the resource time and time of arrival.

You can define the type of encoding to be applied to a placeholder value before printing. The encoding can be specified after a name of the placeholder. The | character is used as the delimiter in this case. The following encoding types are supported:

- none
- xml/html
- slashes/cslashes
- cescape
- url
- cgi
- csv_item
- json
- sql_slashes

Here are examples for encoding placeholder values:

```
CSV: "{pr_address|csv_item}","{pr_comments|csv_item}","{pr_notes|csv_item}"
URL: address={pr_address|cgi}&comments={pr_comments|cgi}&notes={pr_notes|cgi}
```

Click the link to see the available placeholders for each of the following entities:

- *What activity message placeholders are available?*
- *Route Message Placeholders*
- *Visit Message Placeholders*
- *Resource Message Placeholders*
- *Inventory Message Placeholders*
- *Required Inventory Message Placeholders*
- *Service Request Message Placeholders*
- *Message Type Message Placeholders*
- *User Message Placeholders*
- *Properties Message Placeholders*
- *Calendar Message Placeholders*

Encoding Placeholder Values

You can define the type of encoding to be applied to a placeholder value before printing.

The encoding can be specified after a name of the placeholder. The '|' character is used as the delimiter in this case.

The following encoding types are supported:

- `none`
- `xml/html`
- `slashes/cslashes`
- `cescape`
- `url`
- `cgi`
- `csv_item`
- `json`
- `sql_slashes`

```
CSV: "{pr_address|csv_item}","{pr_comments|csv_item}","{pr_notes|csv_item}"
URL: address={pr_address|cgi}&comments={pr_comments|cgi}&notes={pr_notes|cgi}
```

Suppressing Unassigned Variables

You can suppress the printing of empty variables. This functionality is mainly intended to decrease the size of the message body for internal messages.

The supported syntaxes are `{var:+text}` and `{var:-text}`. In the first case, the `text` will not be inserted into the result if `var` is empty. In the second case, text will be inserted into the result if `var` is empty.

```
{pr_ch_note:+<f n="pr_ch_note">{pr_ch_note}</f>
}{ch_drop_bury:+<f n="ch_drop_bury">{ch_drop_bury}</f>
}
```

The result for `pr_ch_note=""` and `ch_drop_bury="0"` is:

```
<f n="ch_drop_bury">0</f>
```

```
{pr_ch_w_o_notes:+<f n="notes">{pr_ch_w_o_notes}</f>
}{pr_ch_w_o_notes:-<f n="notes">{pr_ch_notes}</f>
}
```

The result for `pr_ch_w_o_notes="wo notes"` and `pr_ch_notes=""` is:

```
<f n="notes">wo notes</f>
```

The result for `pr_ch_w_o_notes=""` and `pr_ch_notes="notes"` is:

```
<f n="notes">notes</f>
```

Activity Message Placeholders

These are message placeholders related to activities.

Activity Number placeholder

Placeholder	Description
`activity_number`	ID of the corresponding work order in an external system

Activity Type placeholder

Placeholder	Description
`activity_type`	activity primary type (regular, reopened, prework, multiday_activity, multiday_activity_segment)

Activity Status placeholder

Placeholder	Description
`activity_status`	activity status

Activity Worktype placeholder

Placeholder	Description
`activity_worktype`	activity work type (name)

Activity Worktype ID placeholder

Placeholder	Description
`activity_worktype_id`	activity work type (id)

Activity Worktype Label placeholder

Placeholder	Description
`activity_worktype_label`	activity work type (label)

Activity Workzone placeholder

Placeholder	Description
`activity_workzone`	activity work zone (name)

Activity Worktype ID placeholder

Placeholder	Description
`activity_workzone_id`	activity work zone (id)

Activity Travel Area placeholder

Placeholder	Description
`activity_travel_area`	activity travel area (name)

Activity Travel Area ID placeholder

Placeholder	Description
`activity_travel_area_id`	activity travel area (id)

Activity Travel Area Label placeholder

Placeholder	Description
`activity_travel_area_label`	activity travel area (label)

Activity Workskill placeholder

Placeholder	Description
`activity_workskill`	activity work skill (name). can only be used in the 'activity_workskills' block.

Activity Workskill ID placeholder

Placeholder	Description
`activity_workskill_id`	activity work skill (id). can only be used in the 'activity_workskills' block.

Activity Workskill Label placeholder

Placeholder	Description
`activity_workskill_label`	activity work skill (label. can only be used in the 'activity_workskills' block.

Activity Workskill Required Level placeholder

Placeholder	Description
`activity_workskill_required_level`	required work skill level for an activity. can only be used in the 'activity_workskills' block.

Activity Workskill Preferable Level placeholder

Placeholder	Description
`activity_workskill_preferable_level`	preferable work skill level for an activity. can only be used in the 'activity_workskills' block.

Activity Start Time placeholder

Placeholder	Description
`activity_start_time`	estimated start time of the activity ("hh24:mi" format)

Activity End Time placeholder

Placeholder	Description
`activity_end_time`	estimated end time of the activity ("hh24:mi" format)

Activity duration placeholder

Placeholder	Description
`activity_duration`	estimated activity duration

Activity Service Window Start placeholder

Placeholder	Description
`activity_service_window_start`	service window start ("hh24:mi" format)

Activity Service Window End placeholder

Placeholder	Description
`activity_service_window_end`	service window end ("hh24:mi" format)

Activity SLA Window Start placeholder

Placeholder	Description
`activity_sla_window_start`	sla window start ("yyyy-dd-mm hh24:mi" format)

Activity SLA Window End placeholder

Placeholder	Description
`activity_sla_window_end`	sla window end ("yyyy-dd-mm hh24:mi" format)

Activity Delivery Window Start placeholder

Placeholder	Description
`activity_delivery_window_start`	delivery window start ("hh24:mi" format)

Activity Delivery Window End placeholder

Placeholder	Description
`activity_delivery_window_end`	delivery window end ("hh24:mi" format)

Activity Time Delivered Start placeholder

Placeholder	Description
`activity_time_delivered_start`	start of the time window delivered to customer ("hh24:mm" format)

Activity Time Delivered End placeholder

Placeholder	Description
`activity_time_delivered_end`	end of the time window delivered to customer ("hh24:mm" format)

Activity Traveling Time placeholder

Placeholder	Description
`activity_traveling_time`	estimated travel time from the previous activity (minutes)

Activity Time Slot placeholder

Placeholder	Description
`activity_time_slot`	activity time slot (name)

Activity Time Slot ID placeholder

Placeholder	Description
`activity_time_slot_id`	activity time slot (id)

Activity Time Slot Label placeholder

Placeholder	Description
`activity_time_slot_label`	activity time slot (label)

Activity Time Zone placeholder

Placeholder	Description
`activity_time_zone`	name of the time zone defined for the activity

Activity Time Zone ID placeholder

Placeholder	Description
`activity_time_zone_id`	id of the time zone defined for the activity

Activity Timezone Label placeholder

Placeholder	Description
`activity_time_zone_label`	label of the time zone defined for the activity

Activity Timezone Diff placeholder

Placeholder	Description
`activity_time_zone_diff`	time difference of the time zone defined for the activity

Activity Customer Name placeholder

Placeholder	Description
`activity_customer_name`	customer's name

Activity Customer Number placeholder

Placeholder	Description
`activity_customer_number`	id of the corresponding customer's account in an external system

Activity Phone placeholder

Placeholder	Description
`activity_phone`	activity/customer contact information

Activity Email placeholder

Placeholder	Description
`activity_email`	activity/customer contact information

Activity Cell placeholder

Placeholder	Description
`activity_cell`	activity/customer contact information

Activity Address placeholder

Placeholder	Description
`activity_address`	location of the activity

Activity City placeholder

Placeholder	Description
`activity_city`	location of the activity

Activity State placeholder

Placeholder	Description
`activity_state`	location of the activity

Activity Zip placeholder

Placeholder	Description
`activity_zip`	location of the activity

Activity Coord Status placeholder

Placeholder	Description
`activity_coord_status`	whether or not the activity coordinates were found

Activity Coordx placeholder

Placeholder	Description
`activity_coordx`	longitude of the activity (of customer's location)

Activity Coordy placeholder

Placeholder	Description
`activity_coordy`	latitude of the activity (of customer's location)

Activity Language placeholder

Placeholder	Description
`activity_language`	customer's messaging language (name)

Activity Language ID placeholder

Placeholder	Description
`activity_language_id`	customer's messaging language (id)

Activity Language Label placeholder

Placeholder	Description
`activity_language_label`	customer's messaging language (label)

Activity Reminder Time placeholder

Placeholder	Description
`activity_reminder_time`	customer's reminder notification time

Activity Position in Route placeholder

Placeholder	Description
`activity_position_in_route`	sequential position (starts from "1") of the corresponding activity in a route

Activity Time of Booking placeholder

Placeholder	Description
`activity_time_of_booking`	time when the activity was booked (customer activities) or created (internal activities). ("yyyy-dd-mm hh24:mi" format)

Activity Time of Assignment placeholder

Placeholder	Description
`activity_time_of_assignment`	For the activity currently assigned to a resource, the placeholder returns the time when the corresponding move/reschedule/create operation was performed. For the activity in a bucket, the placeholder's value is empty ("yyyy-dd-mm hh24:mi" format).

Activity Link Min Interval placeholder

Placeholder	Description
`activity_link_min_interval`	minimum duration of the time interval between linked activities. This placeholder can only be used in the 'linked_activities' block.

Activity Link Max Interval placeholder

Placeholder	Description
`activity_link_max_interval`	maximum duration of the time interval between linked activities. this placeholder can only be used in the 'linked_activities' block.

Route Message Placeholders

These are message placeholders related to routes.

Placeholders for route

Placeholder	Description
route_date	Route date ("YYYY-MM-DD" format)
route_start_time	Route activation time ("YYYY-MM-DD HH24:MI:SS" format)
route_end_time	Route deactivation time ("YYYY-MM-DD HH24:MI:SS" format)
route_reactivation_time	Route reactivation time ("YYYY-MM-DD HH24:MI:SS" format)

Visit Message Placeholders

These are message placeholders related to visits.

Placeholders for visit

Placeholder	Description
visit_id	ID of the visit
visit_status	Allows to retrieve the current status of the visit
visit_confirmed	Allows to retrieve ID of the visit
visit_time_delivered_start	Allow to retrieve the time delivered to a customer for companies that use visits
visit_time_delivered_end	Allow to retrieve the time delivered to a customer for companies that use visits

Resource Message Placeholders

These are message placeholders related to resources.

Placeholders for resource

Placeholder	Description
resource_external_id	Resource id
resource_language	Resource messaging language (name)
resource_language_id	Resource messaging language (id)
resource_name	Resource name
resource_phone	Resource phone number
resource_email	Resource email address
resource_type	Resource type (name)
resource_type_id	Resource type (id)
resource_type_label	Resource type (label)
resource_time_zone	Name of the resource's time zone
resource_time_zone_id	ID of resource's time zone
resource_time_zone_label	Label of resource's time zone

Placeholder	Description
resource_time_zone_diff	Time difference of the resource's time zone
parent_resource_external_id	Parent resource external id
parent_resource_name	Parent resource name
destination_resource_id	ID of the current resource for an activity (or inventory). Destination resource for the move activity/ inventory operations.
destination_resource_external_id	External ID of the current resource for an activity (or inventory). Destination resource for the move activity/inventory operations.
destination_resource_name	Name of the current resource for an activity (or inventory). Destination resource for the move activity/ inventory operations.
destination_date	Current date the activity is assigned to. Destination date for the reschedule activity operation.

Inventory Message Placeholders

These are message placeholders related to inventory.

Placeholders for inventory

Placeholder	Description
inventory_serial_number	Serial number of the inventory
inventory_type	Inventory type (name)
inventory_type_id	Inventory type (id)
inventory_type_label	Inventory type (label)
inventory_ex_serial_number	Serial number of the exchanged inventory. This placeholder is only available inside of the 'exchanged_inventory' block.
inventory_quantity	Quantity of the non-serialized inventory. **Note:** • This placeholder can also have "zero" or negative value. To obtain the non-serialized inventory quantity that was transferred using Collaboration, the Core API Collaboration Event = ‘transactionUpdated’ should be used. • In case of Move Activity action, the part that is recently transferred is also included in the updated on-hand quantity of the resource.
inventory_model_property_value	In case with serialized inventory, it always equals "1". Value of the 'Model' property.

Required Inventory Message Placeholders

These are message placeholders related to required inventory.

Placeholders for required inventory

Placeholder	Description
required_inventory_type_id	Required inventory type ID
required_inventory_type_name	Required inventory type name translated into message language
required_inventory_type_label	Required inventory type label
required_inventory_model	Required inventory model
required_inventory_quantity	Quantity of the required inventory

Service Request Message Placeholders

These are message placeholders related to service requests.

Placeholders for service request

Placeholder	Description
request_date	Service request date
request_type	Service request type (name)
request_type_id	Service request type (id
request_type_label	Service request type (label)
request_time_of_creation	Time when the service request was created. It is printed using the time zone of the message (based on the selected **`Recipient`**).

Message Type Message Placeholders

These are message placeholders related to messages.

Placeholders for message

Placeholder	Description
prev_status	Status of the previous message step
prev_desc	Description of the previous message step
prev_data	Data of the previous step
prev_subject	Subject of the previous step
prev_body	Body of the previous step
message_time_zone	Name of the time zone from the corresponding scenario step
message_time_zone_id	ID of the time zone from the corresponding scenario step
message_time_zone_label	Label of time zone from the corresponding scenario step
message_time_zone_diff	Time difference stored in the message
message_time_of_creation	Time when the message was created. In case of 'inner' steps, value might differ from the time of the event that resulted in the creation of this message. It is printed using the time zone of the message (based on the selected **`Recipient`**).
message_time_of_content_generation	Time when content was generated for this message. It is printed using the time zone of the message (based on the selected **`Recipient`**).

User Message Placeholders

These are message placeholders related to users.

Placeholders for user

Placeholder	Description
user_name	Name of the user who created the message
user_login	Login of the user who created the message

Properties Message Placeholders

These are message placeholders related to properties.

Placeholders for properties

Placeholder	Description
exprnum _label	Value of the property identified by the label. It works with **string**, **int**, **enum** and **file** properties. It does not support properties of the **field** type. In case of a **file** property the printed value contains a name of the file. By default the text representation of an **enum** property is printed. The **prnum_** prefix allows to retrieve ID of a lookup for an **enum** property. The **ex_** prefix is only applicable to the **inventory_exchange** launch condition or in the **exchanged_inventory** block. It provides access to properties of the deinstalled inventory in case of an exchange operation.

Calendar Message Placeholders

These are message placeholders related to calendars.

This placeholder group can be used in subject/body templates of messages that have reference to a route. Also, they work for the **manual** (service request) launch condition. The message engine retrieves the data from the **regular** calendar by default. But, it also allows to switch to the **on-call** calendar type using a special BLOCK statement. The calendar is retrieved for the date of the corresponding route (if applicable) or the service request.

Placeholders for calendars

Placeholder	Description
calendar_time_from	Start of resource's working day (**HH24:MI** format)
calendar_time_to	End of resource's working day (**HH24:MI** format
calendar_points	Points from the resource's calendar
calendar_comments	Comments from the resource's calendar
calendar_record_type	Type of the resource's calendar record, such as **shift**, **working**, **non-working**
calendar_shift	Resource's shift (name)
calendar_shift_id	Resource's shift (ID)
calendar_shift_label	Resource's shift (label)
calendar_work_schedule	Resource's work schedule (name)
calendar_work_schedule_id	Resource's work schedule (ID)
calendar_work_schedule_label	Resource's work schedule (label)
calendar_non_working_reason	Resource's non-working reason (name)

Placeholder	Description
calendar_non_working_reason_id	Resource's non-working reason (ID)
calendar_non_working_reason_label	Resource's non-working reason (label)

Date and Time Formatting Patterns

The message content generator can use the date and time formatting patterns listed here. Content for all the below listed patterns are generated using the locale en_us utf-8.

To use the pattern, write it after the placeholder, using the '|' sign as the delimiter. For example:

```
{date} = 2006-07-19
{ETA} = 16:00

{ETA|%R} = 16:00
{ETA|%H:%M} = 16:00

{ETA|%r} = 04:00:00 PM
{ETA|%I:%M %p} = 04:00 PM
{ETA|%I:%M:%S %p} = 04:00:00 PM

{date|%A, %d %b %G} = Wed, 19 Jul 2006
{ETA|%A, %d %b %G %I:%M} = Wednesday, 19 Jul 2006 04:00 PM
```

Format	Description
`%A`	Replaced with the full weekday name.
`%a`	Replaced with the abbreviated weekday name.
`%B`	Replaced with the full month name.
`%b`	Replaced with the abbreviated month name.
`%C`	Replaced with (year / 100) as decimal number; single digits are preceded by a zero.
`%c`	Replaced with the time and date.
`%D`	equivalent to %m/%d/%y.
`%d`	Replaced with the day of the month as a decimal number (01-31).
`%E, %O*`	POSIX locale extensions. The sequences `%Ec %EC %Ex %EX% Ey %EY %Od %Oe %OH %OI %Om %OM %OS %Ou %OU %OV %Ow %OW %Oy` are supposed to provide alternate representations. Additionally, `%OB` is implemented to represent alternative months names (used Standalone, without day mentioned).
`%e`	Replaced with the day of month as a decimal number (1-31); single digits are preceded by a blank.
`%F`	Equivalent to `%Y-%m-%d`.
`%G`	Replaced with a year as a decimal number with century. This year is the one that contains the greater part of the week (Monday as the first day of the week).

Format	Description
`%g`	Replaced with the same year as in %G, but as a decimal number without century (00-99).
`%H`	Replaced with the hour (24-hour clock) as a decimal number (00-23).
`%h`	Same as %b.
`%I`	Replaced with the hour (12-hour clock) as a decimal number (01-12).
`%j`	Replaced with the day of the year as a decimal number (001-366).
`%k`	Replaced with the hour (24-hour clock) as a decimal number (0-23); single digits are preceded by a blank.
`%l`	Replaced with the hour (12-hour clock) as a decimal number (1-12); single digits are preceded by a blank.
`%M`	Replaced with the minute as a decimal number (00-59).
`%m`	Replaced with the month as a decimal number (01-12).
`%n`	Replaced with a new line.
`%O*`	Same as `%E*`.
`%p`	Replaced with either "ante meridiem" or "post meridiem" as appropriate.
`%R`	Equivalent to `%H:%M`.
`%r`	Equivalent to `%I:%M:%S %p`.
`%S`	Replaced with the second as a decimal number (00-60).
`%s`	Replaced with the number of seconds since the Epoch, UTC (see mktime(3)).
`%T`	Equivalent to `%H:%M:%S`
`%t`	Replaced with a tab.
`%U`	Replaced with the week number of the year as a decimal number (00-53) with Sunday being the first day of the week.
`%u`	Replaced with the weekday as a decimal number (1-7) with Monday being the first day of the week.
`%V`	Replaced with the week number of the year as a decimal number (01-53) with Monday being the first day of the week. If the week containing January 1 has four or more days in the new year, then it is week 1; otherwise it is the last week of the previous year, and the next week is week 1.
`%v`	Equivalent to `%e-%b-%Y`
`%W`	Replaced with the week number of the year as a decimal number (00-53) with Monday being the first day of the week.
`%w`	Replaced with the weekday as a decimal number (0-6) with Sunday being the first day of the week.
`%X`	Replaced with the time.
`%x`	Replaced with the date.
`%Y`	Replaced with the year with century as a decimal number.
`%y`	Replaced with the year without century as a decimal number (00-99).

Format	Description
`%+`	Replaced with the date and time (the format is similar to that produced by date(1)).
`%-*`	Suppress padding when performing numerical outputs.
`%_*`	Explicitly specify space for padding.
`%0*`	Explicitly specify zero for padding.
`%%`	Replaced with `%`.

What are the types of message blocks available in Field Service?

These types of message blocks are available.

Block	Description
`installed_inventory`	This block retrieves the list of inventory from the **`install`** pool.
`deinstalled_inventory`	This block retrieves the list of inventory from the **`deinstall`** pool.
`customer_inventory`	This block retrieves the list of inventory from the **`customer`** pool.
`resource_inventory`	This block retrieves the list of inventory from the **`resource`** pool.
`exchanged_inventory`	This block retrieves the list of exchanged inventory.
`required_inventory`	This block retrieves the list of required inventory.
`visit`	This block can include information related to work orders that belong to a visit into a single message. This block can only be used in scenarios attached to the visit-related launch conditions.
`team-members`	This block retrieves the list of team-members for a given teamholder and a given day. It can be used in messages that are related to a route. It also it works for the **`Manual`** (service request) launch condition.
`regular_calendar/oncall_calendar`	These two blocks are intended for switching between the **`regular`** and **`on-call`** calendar types. The **`regular`** calendar is selected by default and it is not required to use the block in this case.
`activity_workskills`	Only the following placeholders can be used in this block: • Activity Workskill • Activity Workskill ID • Activity Workskill label • Activity Workskill Required level • Activity Workskill Preferable label See *What activity message placeholders are available?*

Block	Description
`destination_resource`	This block retrieves values of fields (resource_* placeholders) and custom properties for a destination resource in the following cases: • "move activity" trigger • "move inventory" trigger
`forbidden_resources/required_ resources/preferred_resources`	These blocks retrieve forbidden/required or preferred resources.
`linked_activities`	This block retrieves activities linked to the current one. The block has the following syntax: <BLOCK type="linked_activities: label of activity link type">. This block cannot be used without the label of activity link type. All related activity links are also handled as ordered even if the corresponding activity link type has two identical labels.
`multiday_activity`	This block temporarily switches the context to the related segmentable activity. It can be used in scenarios related to both segmentable activities and their segments. If this block is invoked for a single-day activity, it will not process any records and the result will contain no data.
`multiday_activity_segments`	This block prints information about all existing segments of a segmentable activity regardless of their statuses (except `deleted` segments). It can be used in scenarios related to both segmentable activities and their segments. If this block is invoked for a single-day activity, it will not process any records and the result will contain no data.

Examples of message blocks

These are the examples of message blocks.

Visit

```
<visit>
<BLOCK type="visit">
<work_order number="{activity_number}" status="{activity_status}" comments="{pr_comments}"/>
</BLOCK>
</visit>
```

Team

```
 <team>
 <BLOCK type="team-members">
 <team-member
 resource_external_id ="{resource_external_id}"
 resource_name = "{resource_name}"
 teamwork_start = "{activity_start_time:+{activity_start_time}}{activity_start_time:-
{activity_service_window_start}}"
 teamwork_end = "{activity_end_time:+{activity_end_time}}{activity_end_time:-{activity_service_window_end}}"
 teamwork_activity_number = "{activity_number}"
 teamwork_activity_worktype = "{activity_worktype}"
```

```
teamwork_activity_duration = "{activity_duration}"
teamwork_activity_time_slot = "{activity_time_slot}"
/>
</BLOCK>
</team>
```

Oncall_calendar

```
regular_shift = "{calendar_shift}"

<BLOCK type="oncall_calendar">
oncall_shift = "{calendar_shift}"
</BLOCK>
```

Activity_workskills

```
Pattern:

<desc><BLOCK type="activity_workskills">{activity_workskill},</BLOCK></desc>
<details>
<BLOCK type="activity_workskills">
<skill label="{activity_workskill_label}" required="{activity_workskill_required_level}"
preferable="{activity_workskill_preferable_level}"/>
</BLOCK>
</details>
```

Linked activities and preferred resources

```
Pattern:

<linked_activities>
<BLOCK type="linked_activities: start-before">
<activity link = "predecessor" min_delay = "{activity_link_min_interval}" max_delay =
"{activity_link_max_interval}" date = "{date}" resource = "{resource_name}" number = "{activity_number}"
worktype = "{activity_worktype}"/>

<forbidden_resources>
<BLOCK type="forbidden_resources">
<resource resource = "{resource_name}"/>
</BLOCK>
</forbidden_resources>

<required_resources>
<BLOCK type="required_resources">
<resource resource = "{resource_name}"/>
</BLOCK>
</required_resources>

<preferred_resources>
<BLOCK type="preferred_resources">
<resource resource = "{resource_name}"/>
</BLOCK>
</preferred_resources>

<resource_inventory>
<BLOCK type="resource_inventory">
<inventory serial = "{inventory_serial_number}" type="{inventory_type}"/>
</BLOCK>
</resource_inventory>

</activity>
</BLOCK>
<BLOCK type="linked_activities: start-after">
```

```
<activity link = "successor" min_delay = "{activity_link_min_interval}" max_delay =
"{activity_link_max_interval}" date = "{date}" resource = "{resource_name}" number = "{activity_number}"
worktype = "{activity_worktype}"/>
</BLOCK>
<BLOCK type="linked_activities: start-together">
<activity link = "simultaneous" date = "{date}" resource = "{resource_name}" number = "{activity_number}"
worktype = "{activity_worktype}"/>
</BLOCK>
</linked_activities>
```

Required inventory

```
 Pattern:
 <required_inventory>
 <BLOCK type="required_inventory">
 <item
 type="{required_inventory_type_label}"
 model="{required_inventory_model}"
 quantity="{required_inventory_quantity}"
 />
 </BLOCK>
 </required_inventory>

 </conbody>
</concept>
```

Message and inventory block examples

These are examples of messages and inventory blocks.

Inventory block examples

Deinstall inventory

```
<external_id>{resource_external_id}</external_id>
<cust_number>{activity_customer_number}</cust_number>
<appt_number>{activity_number}</appt_number>
<aid>{aid}</aid>
<astatus>{activity_status}</astatus>
<ETA>{activity_start_time}</ETA>
<aworktype>{activity_worktype_label}</aworktype>
<deinstalled_inventory>
<inventory>
<invsn>{inventory_serial_number}</invsn>
<invtype>{inventory_type}</invtype>
<eta_inventory_id>{invid}</eta_inventory_id>
<status>{invpool}</status>
</inventory>
</deinstalled_inventory>
```

Install inventory

```
<external_id>{resource_external_id}</external_id>
<cust_number>{activity_customer_number}</cust_number>
<appt_number>{activity_number}</appt_number>
<aid>{aid}</aid>
<astatus>{activity_status}</astatus>
<ETA>{activity_start_time}</ETA>
<aworktype>{activity_worktype_label}</aworktype>
<installed_inventory>
```

```
    <inventory>
    <invsn>{inventory_serial_number}</invsn>
    <invtype>{inventory_type}</invtype>
    <status>{invpool}</status>
    <resource_id>{inv_pid}</resource_id>
    <eta_inventory_id>{invid}</eta_inventory_id>
    </inventory>
    </installed_inventory>
```

Message examples

```
Start Activity
    <acoord_x>{activity_coordx}</acoord_x>
    <acoord_y>{activity_coordy}</acoord_y>
    <appt_number>{activity_number}</appt_number>
    <aid>{aid}</aid>
    <astatus>{activity_status}</astatus>
    <aworktype>{activity_worktype_label}</aworktype>
    <caddress>{activity_address}</caddress>
    <cphone>{activity_phone}</cphone>
    <ccity>{activity_city}</ccity>
    <cemail>{activity_email}</cemail>
    <date>{destination_date}</date>
    <eta>{activity_start_time|%F %H:%M}</eta>
    <external_id>{resource_external_id}</external_id>
    <activity_time_slot_label>{activity_time_slot_label}</activity_time_slot_label>
    <user_name>{user_name}</user_name>


Delay Activity


     <appt_number>{activity_number}</appt_number>
     <aid>{aid}</aid>
     <astatus>{activity_status}</astatus>
     <aworktype>{activity_worktype_label}</aworktype>
     <caddress>{activity_address}</caddress>
     <ccity>{activity_city}</ccity>
     <date>{destination_date}</date>
     <eta>{activity_start_time|%F %H:%M}</eta>
     <external_id>{resource_external_id}</external_id>
     <delay_comments>{pr_delay_comments}</delay_comments>
     <delay_reason>{pr_delay_reason}</delay_reason>
     <user_name>{user_name}</user_name>


Cancel Activity

    <appt_number>{activity_number}</appt_number>
     <aid>{aid}</aid>
     <astatus>{activity_status}</astatus>
     <aworktype>{activity_worktype_label}</aworktype>
     <caddress>{activity_address}</caddress>
     <ccity>{activity_city}</ccity>
     <cphone>{activity_phone}</cphone>
     <cemail>{activity_email}</cemail>
     <date>{destination_date}</date>
     <external_id>{resource_external_id}</external_id>
     <cancel_reason>{pr_cancel_reason}</cancel_reason>
     <cancel_notes>{pr_cancel_notes}</cancel_notes>
     <user_name>{user_name}</user_name>

Notdone Activity

    <appt_number>{activity_number}</appt_number>
```

```
     <aid>{aid}</aid>
     <astatus>{activity_status}</astatus>
     <aworktype>{activity_worktype_label}</aworktype>
     <cname>{activity_customer_name}</cname>
     <caddress>{activity_address}</caddress>
     <ccity>{activity_city}</ccity>
     <czip>{activity_zip}</czip>
     <cstate>{activity_state}</cstate>
     <cphone>{activity_phone}</cphone>
     <date>{destination_date}</date>
     <eta>{activity_start_time|%F %H:%M}</eta>
     <end_time>{activity_end_time|%F %H:%M}</end_time>
     <external_id>{resource_external_id}</external_id>
     <work_skills><BLOCK type="activity_workskills">{activity_workskill_label},</BLOCK></work_skills>
     <not_done_notes>{pr_not_done_notes}</not_done_notes>
     <not_done_reason>{pr_not_done_reason}</not_done_reason>
     <activity_time_slot_label>{activity_time_slot_label}</activity_time_slot_label>
     <user_name>{user_name}</user_name>
```

`Suspend Activity`

```
    <appt_number>{activity_number}</appt_number>
     <aid>{aid}</aid>
     <astatus>{activity_status}</astatus>
     <aworktype>{activity_worktype_label}</aworktype>
     <caddress>{activity_address}</caddress>
     <ccity>{activity_city}</ccity>
     <date>{destination_date}</date>
     <eta>{activity_start_time|%F %H:%M}</eta>
     <end_time>{activity_end_time|%F %H:%M}</end_time>
     <external_id>{resource_external_id}</external_id>
     <suspend_notes>{pr_suspend_notes}</suspend_notes>
     <suspend_reason>{prnum_suspend_reason}</suspend_reason>
     <activity_time_slot_label>{activity_time_slot_label}</activity_time_slot_label>
     <user_name>{user_name}</user_name>
```

`Complete Activity`

```
    <MessageItem>
     <ActivityId>{aid}</ActivityId>
     <LegacyServiceId>{activity_number}</LegacyServiceId>
     <CustomerId>{activity_customer_number}</CustomerId>
     <CustomerZip>{activity_zip}</CustomerZip>
     <ActivityStatus>{activity_status}</ActivityStatus>
     <ActivityType>{activity_worktype_label}</ActivityType>
     <ScheduleDate>{route_date}</ScheduleDate>
     <BeginTime>{activity_start_time|%F %H:%M}</BeginTime>
     <EndTime>{activity_end_time}</EndTime>
     <DurationInMinutes>{activity_duration}</DurationInMinutes>
     <PositionInRoute>{activity_position_in_route}</PositionInRoute>
     <TechnicianId>{user_login}</TechnicianId>
     <ResourceId>{resource_external_id}</ResourceId>
     <ResourceType>{resource_type}</ResourceType>
     <EventDateTime>{message_time_of_creation}</EventDateTime>
     </MessageItem>
```

`Complete Activity`**: different format**

```
    <values>
     <appt_number>{activity_number}</appt_number>
     <external_ID>{resource_external_id}</external_ID>
     <date>{route_date}</date>
     <site_id>{pr_SITEID}</site_id>
     <p>
     <l>astatus</l>
     <v>{activity_status}</v>
```

```
</p>
<p>
<l>TECH_DEPARTURE_TIMESTAMP</l>
<v>{pr_TECH_DEPARTURE_TIMESTAMP}</v>
</p>
<p>
<l>JOURNAL_ENTRY_DATE</l>
<v>{message_time_of_creation}</v>
</p>
<p>
<l>JOURNAL_ENTRY_TYPE</l>
<v>Work Log</v>
</p>
<p>
<l>JOURNAL_ENTRY_DESCRIPTION</l>
<v>{pr_JOURNAL_ENTRY_DESCRIPTION}</v>
</p>
<p>
<l>JOURNAL_NOTE_ENTRY</l>
<v>{pr_JOURNAL_NOTE_ENTRY}</v>
</p>
<p>
<l>INCIDENT_CASE_ID</l>
<v>{pr_INCIDENT_CASE_ID}</v>
</p>
<p>
<l>SITEID</l>
<v>{pr_SITEID}</v>
</p>
<p>
<l>INVOICE</l>
<v>{pr_INVOICE}</v>
</p>
<p>
<l>CABLE_FOOTAGE</l>
<v>{pr_CABLE_FOOTAGE}</v>
</p>
<p>
<l>DROP_DIRECT_BURY</l>
<v>{pr_DROP_DIRECT_BURY}</v>
</p>
<p>
<l>CLOSURE_CODE</l>
<v>{prnum_CLOSURE_CODE}{prnum_CLOSURE_CODE1}</v>
</p>
</values>
```

7 Blocking Conditions

About Blocking Conditions

You can define message blocking conditions that, when met, will block messages.

Any field used in the system can serve as a blocking condition. When blocked, a message will either be generated with its final status or not generated at all.

The condition value supports CSV format, such as `1, 2, 3, 4,`, `"1,1,1"`, `"2,s"`, and `"(""test"")"`. The following comparison functions are available: `IS NULL`, `IS NOT NULL`, `IN`, `NOT IN`, `CONTAINS`, `DOESN'T CONTAIN`, >, <, >=, <=.

In addition to the blocking conditions, the system allows you to block or shift messages that fall on a non-working day or a holiday. This functionality is configured in the **Settings** tab of the **Add notification scenario step** window.

Control the Message Flows for Specified Resources by Blocking Conditions

Here are some examples on how to control the message flow for a specific resource:

- Stop message delivery by specific resources

  If you need to block all notifications related to some resources (or bucket or any other resource tree item), you can use conditions based on a resource property to achieve it. For example, you can create a 'checkbox' property "Block notifications" and add it to the Resource Info screen. You must then configure blocking conditions for the message steps based on the value of this property ("block_notifications" is not null). After that all can set on or off the checkboxes for particular resources to control the delivery.
- Redirecting e-mail notifications for some resources only.

  If you need to change a normal flow for some selected resources then you may find it easy to achieve by using resource properties and message blocking conditions.

  For example, you need to create a flow where an email must be sent to resources, but for some reasons, it is also necessary to redirect these emails to another address for some resources. You may create an additional resource property (for example, 'alternative_email'), store the second address in this property for necessary resources, and leave it empty for others. You can specify the following message blocking conditions:
  - message step 'send_general_email': Add blocking condition 'alternative_email' is not null so that this message is not sent to normally used address if the alternative one was specified for this particular resource.
  - additional message step 'send_alternative_email' can be created with the blocking condition “alternative_email is null” so that the message is sent, only if the address was specified.

  Therefore, only one message is sent for either to the normally used address or to the alternative e-mail (if it is specified).
- Duplicating e-mail notifications to a dispatcher for some resources only

Use the resource property and message blocking conditions to duplicate e-mail notifications to a dispatcher for some resources only. The flow is similar to the redirect example (above) with one difference that you do not need to block the step with normal delivery but only have the blocking condition for alternative step.

- Stop delivery for testing environments

  A delivery channel (internal or external) can be made inactive when a client copies a production instance to create a test instance and would not like any message to be sent to a real customer or a production external system. Message from the channel that are inactive are not sent to real customers. Alternatively, you may configure the credentials of delivery channels to point to the test instances of external systems (available only for external delivery channels).

Blocking Conditions Reference

Use message blocking conditions to block messages from being sent then the blocking condition is met. Click the link to see the available blocking conditions for each of the following entities:

- *What are the blocking conditions available for activities?*
- *Inventory Blocking Conditions*
- *Resource Blocking Conditions*
- *Service Request Blocking Conditions*
- *Route Blocking Conditions*
- *Visit Blocking Conditions*
- *User Blocking Conditions*
- *Shift and Calendar Blocking Conditions*
- *What are the miscellaneous blocking conditions available for message scenarios?*

What are the blocking conditions available for activities?

You can use these blocking conditions with activities.

Activity Status Blocking Condition

Activity Status	
Condition name	`[appt_status]`
Condition type	Activity
Description	Checks activity status
Valid values/format	**`started`**, **`completed`**, **`suspended`**, etc.
Case sensitive?	No
Suggested functions	**`IN`**, **`NOT IN`**

Customer Number Blocking Condition

Account Number	
Condition name	`[customer_number]`
Condition type	Activity
Description	Checks the account number
Valid values/format	text

Activity Type Blocking Condition

Activity Type	
Condition name	`[appt_type]`
Condition type	Activity
Description	Checks the activity type
Valid values/format	`regular`, `reopened`, `prework`, `multiday_activity`, `multiday_activity_segment`
Case sensitive?	No
Suggested functions	`IN`, `NOT IN`

Activity Time of Assignment blocking condition

Activity Time of Assignment	
Condition name	`[atime_of_assignment]`
Condition type	Activity
Description	Checks the time the activity is assigned in the time zone of the assigned resource.
Valid values/format	`YYYY-MM-DD HH24:MI:SS`
Suggested functions	`IS NULL`, `IS NOT NULL`, `CONTAINS`, `DOESN'T CONTAIN`, <, >
Example 1	**Condition**: `'Time of assignment' CONTAINS '2012-05-21'` **Result**: Blocks all activities that were assigned during `'2012-05-21'`
Example 2	**Condition**: `'Time of assignment' <'2012-05-21'` **Result**: Blocks all activities that were assigned before `'2012-05-21'`

Activity Time of Booking Blocking Condition

Activity Time of Booking	
Condition name	`[atime_of_booking]`

Activity Time of Booking	
Condition type	Activity
Description	Checks the time the activity is booked in the time zone of the assigned resource.
Valid values/format	`YYYY-MM-DD`, `HH24:MI:SS`
Suggested functions	`IS NULL`, `IS NOT NULL`, `CONTAINS`, `DOESN'T CONTAIN`, <, >
Example 1	**Condition**: `'Time of booking' CONTAINS '2012-05-21'` **Result**: Blocks all activities that were booked during `'2012-05-21'`
Example 2	**Condition**: `'Time of booking' < '2012-05-21'` **Result**: Blocks all activities that were booked before `'2012-05-21'`

Activity Work Type Blocking Condition

Activity Type	
Condition name	`[aworktype]`
Condition type	Activity
Description	Checks the activity work type
Valid values/format	Activity type IDs
Notes	This condition can be used with both activity types and their groups.

Address Blocking Condition

Address	
Condition name	`[caddress]`
Condition type	Activity
Description	Checks the activity address
Valid values/format	text

Calendar Days from Activity Assignment blocking condition

Calendar Days from Activity Assignment	
Condition name	`[calendar_days_from_activity_assignment]`
Condition type	Activity
Description	Calculated as the number of calendar days from the activity assignment date to the current date of the resource.
Valid values/format	integer >= 0

Calendar Days from Activity Assignment	
Suggested functions	`IN`, `NOT IN`, <, >, <=, >=
Example 1	**Condition**: `'Calendar days from activity assignment' <= 1` **Result**: Blocks all activities that were assigned yesterday `(1)` or today `(0)`.
Notes	Both dates have the same time zone difference.

Calendar Days from Activity Booking blocking condition

Calendar Days from Activity Booking	
Condition name	`[calendar_days_from_activity_booking]`
Condition type	Activity
Description	Calculated as the number of calendar days from the activity booking date to the current date of the resource the activity is assigned to.
Valid values/format	integer >= 0
Suggested functions	`IN`, `NOT IN`, <, >, <=, >=
Example 1	**Condition**: `'Calendar days from activity booking' <= 1` **Result**: Blocks all activities that were booked or created yesterday `(1)` or today `(0)`.
Notes	Both dates have the same time zone difference.

Capacity Categories blocking condition

Capacity Categories	
Condition name	`[activity_capacity_categories]`
Condition type	Activity
Description	Selects activities that belong to the given capacity category.
Valid values/format	Capacity category IDs
Suggested functions	`IN` - the activity belongs to all provided categories. `NOT IN`- the activity belongs to none of the categories provided in the list.
Notes	The same rules apply as in Quota Management.

City blocking condition

City	
Condition name	`[ccity]`
Condition type	Activity
Description	Checks the city of the activity location.

City	
Valid values/format	text

Coordinate Status blocking condition

Coordinate Status	
Condition name	`[acoord_status]`
Condition type	Activity
Description	Checks whether or not the current activity coordinates were found
Valid values/format	`found`, `not_found`, `invalid`

Coordinate X blocking condition

Coordinate X	
Condition name	`[acoord_x]`
Condition type	Activity
Description	Checks the longitude of the activity location
Valid values/format	A longitude value, such as `37.40562` or `-79.42639`

Coordinate Y blocking condition

Coordinate Y	
Condition name	`[acoord_y]`
Condition type	Activity
Description	Checks the latitude of the activity location
Valid values/format	A latitude value, such as `22.831438`

Days from Activity Assignment blocking condition

Days from Activity Assignment	
Condition name	`[days_from_activity_assignment]`
Condition type	Activity
Description	The number of full days elapsed between the time the activity was assigned and the current time.
Valid values/format	integer >= 0

Days from Activity Assignment	
Suggested functions	`IN`, `NOT IN`, <, >, <=, >=
Example	**Condition**: `'Days from activity assignment' = 0` **Result**: Blocks all activities assigned within the last 24 hours. If the current time is '`2012-09-27 08:28:22`', the condition will return the activity that was assigned on '`2012-09-26 19:17:00`'.

Days from Activity Booking blocking condition

Days from Activity Booking	
Condition name	`[days_from_activity_booking]`
Condition type	Activity
Description	The number of full days elapsed between the time the activity was booked and the current time.
Valid values/format	integer >= 0
Suggested functions	`IN`, `NOT IN`, <, >, <=, >=
Example	**Condition**: `'Days from activity booking' = 0` **Result**: Blocks all activities booked within the last 24 hours. If the current time is '`2012-09-27 08:28:22`', the condition will return the activity that was booked on '`2012-09-26 19:17:00`'.

Days to Activity blocking condition

Days to Activity	
Condition name	`[days_to_appt]`
Condition type	Activity
Description	Returns the number of days from the message creation to the activity.
Valid values/format	integer >= 0
Suggested functions	`IN`, `NOT IN`

Activity Scheduled? blocking condition

Activity Scheduled?	
Condition name	`[is_activity_scheduled]`
Condition type	Activity
Description	Checks if the activity is scheduled
Valid values/format	`1` or `<empty>`
Suggested functions	`IS NULL`, `IS NOT NULL`

Activity Scheduled?	
Notes	This condition cannot be used in the `cancel activity` scenario. This constraint is related to the two-phase nature of the `Cancel non-scheduled activity` action.This action consists of two steps: 1. Making the activity `scheduled` (moving it to the current day). 2. Performing the`cancel` action. According to this scenario, the launch condition is invoked when the activity is already `scheduled`. Therefore, the condition is always `false`.

Message Language blocking condition

Message Language	
Condition name	`[clanguage]`
Condition type	Activity
Description	Checks the language of the message to be sent to the customer
Valid values/format	Language IDs
Suggested functions	`IS NULL`, `IS NOT NULL`
Notes	This condition is true for both segmentable activities and individual segments which can be distinguished by the value of the `Type` field. For segmentable activities the `Type` value is `multiday_activity`, while for segments it is `multiday_activity_segments`.

Customer Name blocking condition

Name	
Condition name	`[cname]`
Condition type	Activity
Description	Checks the customer name
Valid values/format	text

Not Ordered Activity blocking condition

Not Ordered Activity	
Condition name	`[all_day_flag]`
Condition type	Activity
Description	Checks if the activity is not ordered
Valid values/format	`1` or `<empty>`
Suggested functions	`IS NULL`, `IS NOT NULL`

Pending Activity Order blocking condition

Pending Activity Order	
Condition name	`[pending_activity_order]`
Condition type	Activity
Description	Checks the offset of a pending activity from the start of the route to which this activity belongs
Valid values/format	integer >= -1
Suggested functions	`IN`, `NOT IN`, <, >, <=, >=
Example	**Condition**: `'Pending activity order' = 1` **Result**: Blocks the first pending activity in the route.
Notes	The following logic applies:– "1..." pending ordered activity in the route (offset from the route start)– "1" pending ordered in the bucket– "1" pending ordered non-scheduled activity– "-1" pending not ordered activity– "0" started, completed, canceled, notdone, suspended activity

Phone blocking condition

Phone	
Condition name	`[phone]`
Condition type	Activity
Description	Checks the customer's phone number
Valid values/format	text
Notes	Takes into account whether the contact can be used with messages.

Email Address blocking condition

Email Address	
Condition name	`[email]`
Condition type	Activity
Description	Checks the customer's email address
Valid values/format	text
Notes	Takes into account whether the contact can be used with messages.

Cellular Phone blocking condition

Cellular Phone	
Condition name	`[cell]`

Cellular Phone	
Condition type	Activity
Description	Checks the customer's cell phone number
Valid values/format	text
Notes	Takes into account whether the contact can be used with messages.

Points blocking condition

Points	
Condition name	`[apoints]`
Condition type	Activity
Description	Checks the number of points assigned to the activity
Valid values/format	integer

Position in Route blocking condition

Position in Route	
Condition name	`[position_in_route]`
Condition type	Activity
Description	Checks the activity position in the route
Valid values/format	integer

Reminder blocking condition

Reminder	
Condition name	`[cmessagetime]`
Condition type	Activity
Description	Checks the number of minutes before the `Notification base` when reminder notification is to be generated
Valid values/format	integer

SLA Start blocking condition

SLA Start	
Condition name	`[sla_window_start]`

SLA Start	
Condition type	Activity
Description	Checks the start of the activity SLA window
Valid values/format	`YYYY-MM-DD`, `HH24:MI:SS`

SLA End blocking condition

SLA End	
Condition name	`[sla_window_end]`
Condition type	Activity
Description	Checks the end of the activity SLA window
Valid values/format	`YYYY-MM-DD`, `HH24:MI:SS`

Service Window End blocking condition

Service Window End	
Condition name	`[service_window_end]`
Condition type	Activity
Description	Checks the end of the activity service window
Valid values/format	`HH24:MI`

Service Window Start blocking condition

Service Window Start	
Condition name	`[service_window_start]`
Condition type	Activity
Description	Checks the start of the activity service window
Valid values/format	`HH24:MI`

Start blocking condition

Start	
Condition name	`[eta]`
Condition type	Activity

Start	
Description	Checks if the activity has an ETA
Valid values/format	`HH24:MI:SS`
Suggested functions	`IS NULL`, `IS NOT NULL`

State blocking condition

State	
Condition name	`[state]`
Condition type	Activity
Description	Checks the area name of the activity (state, county, land, etc.)
Valid values/format	text

Time Slot blocking condition

Time Slot	
Condition name	`[time_slot]`
Condition type	Activity
Description	Checks the label of the activity time slot

Time Zone blocking condition

Time Zone	
Condition name	`[c_zid]`
Condition type	Activity
Description	Checks the activity time zone
Valid values/format	Time zone IDs

Time Notified blocking condition

Time Notified	
Condition name	`[time_delivered]`
Condition type	Activity
Description	Checks the start time of arrival interval communicated to the customer

Time Notified	
Valid values/format	`HH24:MI`
Suggested functions	`IS NULL`, `IS NOT NULL`
Notes	This condition works both for a single activity and a visit. If the message is based on a visit, it refers to the visit's field instead of the activity's field.

Travel Area blocking condition

Travel Area	
Condition name	`[atravelarea]`
Condition type	Activity
Description	Checks the activity travel area
Valid values/format	Travel area IDs

Traveling Time blocking condition

Traveling Time	
Condition name	`[travel]`
Condition type	Activity
Description	Checks the travel time from the previous activity/location to the current activity
Valid values/format	integer

Work Order blocking condition

Work Order	
Condition name	`[appt_number]`
Condition type	Activity
Description	Checks the activity work skills
Valid values/format	Work skill IDs
Suggested functions	`IS NULL` - The activity doesn't require any work skills. `IS NOT NULL` - The activity requires at least one work skill. `IN` - The activity requires all the provided work skills. `NOT IN` - The activity requires none of the provided work skills.
Example 1	**Activity #1 work skills:** `Install IPTV (1)`, `English (10)` **Activity #2 work skills:** `Install IPTV (1), Spanish (11)` **Condition:** `Activity work skill IN (1)` **R**
Example 2	**Condition:** `Activity work skill IN (1, 10)` **Condition:** `Activity work skill NOT IN (1,10` **Result:** `<NONE>`

Work Order	
Notes	Groups of work skills are also supported. In this case a group is handled as any work skill it includes.

ZIP/Postal Code blocking condition

ZIP/Postal Code	
Condition name	`[czip]`
Condition type	Activity
Description	Checks ZIP (postal code)
Valid values/format	text

Time Delivered End blocking condition

Time Delivered End	
Condition name	`[ctime_delivered_end]`
Condition type	Activity
Description	Checks the end time of arrival interval communicated to the customer
Valid values/format	`YYYY-MM-DD`, `HH24:MI:SS`
Suggested functions	`IS NULL`, `IS NOT NULL`

Time Delivered Start blocking condition

Time Delivered Start	
Condition name	`[ctime_delivered_start]`
Condition type	Activity
Description	Checks the start time of arrival interval communicated to the customer
Valid values/format	`YYYY-MM-DD`, `HH24:MI:SS`
Suggested functions	`IS NULL`, `IS NOT NULL`

Delivery Window End blocking condition

Delivery Window End	
Condition name	`[delivery_window_end]`
Condition type	Activity

Delivery Window End	
Description	Checks the delivery window end
Valid values/format	`HH24:MI:SS`

Delivery Window Start blocking condition

Delivery Window Start	
Condition name	`[delivery_window_start]`
Condition type	Activity
Description	Checks the delivery window start
Valid values/format	`HH24:MI:SS`

Inventory Blocking Conditions

These are blocking conditions used with inventory.

Changed Inventory blocking condition

Fields	Description
Condition name	`[inv_change_invid]`
Condition type	Inventory
Description	Checks ID of the changed inventory
Valid values/format	IDs of the changed inventory

Inventory Pool blocking condition

Fields	Description
Condition name	`[invpool]`
Condition type	Inventory
Description	Checks the inventory pool
Valid values/format	`customer`, `install`, `deinstall`, `provider`

Inventory Type blocking condition

Fields	Description
Condition name	`[invtype]`
Condition type	Activity
Description	Checks the inventory type
Valid values/format	Inventory type IDs

Serial Number blocking condition

Fields	Description
Condition name	`[invsn]`
Condition type	Activity
Description	Checks the inventory serial number
Valid values/format	text

Resource Blocking Conditions

These are blocking conditions used with resources.

Date Format blocking condition

Fields	Description
Condition name	`[pdate_fid]`
Condition type	Resource
Description	Checks the date format of the resource
Valid values/format	Date/time format IDs

Email Address blocking condition

Fields	Description
Condition name	`[provider_email]`
Condition type	Resource
Description	Checks the resource's email address
Valid values/format	text

External ID blocking condition

Fields	Description
Condition name	`[external_id]`
Condition type	Resource
Description	Checks the resource's external ID
Valid values/format	text

Message Language blocking condition

Fields	Description
Condition name	`[planguage]`
Condition type	Resource
Description	Checks the language set for the resource
Valid values/format	Language IDs

Name blocking condition

Fields	Description
Condition name	`[pname]`
Condition type	Resource
Description	Checks the resource's name
Valid values/format	text

Phone blocking condition

Fields	Description
Condition name	`[provider_phone]`
Condition type	Resource
Description	Checks the resource's phone number
Valid values/format	text

Resource Type blocking condition

Fields	Description
Condition name	`[provider_type]`
Condition type	Resource
Description	Checks the resource's type
Valid values/format	Resource type IDs

Routing Profile blocking condition

Fields	Description
Condition name	`[p_rprid]`
Condition type	Resource
Description	Checks the ID of the routing profile assigned to the resource
Valid values/format	Routing profile IDs

Status blocking condition

Fields	Description
Condition name	`[pactive]`
Condition type	Resource
Description	Checks to see if the resource is active
Valid values/format	`1`, `0`
Suggested functions	`IS NULL`, `IS NOT NULL`

Time Format blocking condition

Fields	Description
Condition name	`[ptime_fid]`
Condition type	Resource
Description	Checks the resource's time format
Valid values/format	Date/time format IDs

Time Zone blocking condition

Fields	Description
Condition name	`[time_zone]`
Condition type	Resource
Description	Checks the time zone assigned to the resource
Valid values/format	Time zone IDs

Service Request Blocking Conditions

These are blocking conditions used with service requests.

Request Type blocking condition

Fields	Description
Condition name	`[service_request_type]`
Condition type	Service request
Description	Checks the service request type
Valid values/format	Service request type IDs

Route Blocking Conditions

These are blocking conditions used with routes.

Activated blocking condition

Fields	Description
Condition name	`[activated]`
Condition type	Activity
Description	Checks if the route is activated.
Valid values/format	`YYYY-MM-DD`, `HH24:MI:SS`
Suggested functions	`IS NULL`, `IS NOT NULL`

Deactivated blocking condition

Fields	Description
Condition name	`[deactivated]`
Condition type	Activity
Description	Checks if the route is deactivated.
Valid values/format	`YYYY-MM-DD`, `HH24:MI:SS`
Suggested functions	`IS NULL`, `IS NOT NULL`

Reactivated blocking condition

Fields	Description
Condition name	`[reactivated]`
Condition type	Activity
Description	Checks if the route is reactivated.
Valid values/format	`YYYY-MM-DD`, `HH24:MI:SS`
Suggested functions	`IS NULL`, `IS NOT NULL`

Visit Blocking Conditions

These are blocking conditions used with visits.

Based on Visit blocking condition

Fields	Description
Condition name	`[is_visit]`
Condition type	Visit
Description	Determines if the message is related to a visit.
Valid values/format	`1`, `<empty>`
Suggested functions	`IS NULL`, `IS NOT NULL`

Visit Status blocking condition

Fields	Description
Condition name	`[visit_status]`

Fields	Description
Condition type	Visit
Description	Checks the current visit status. It is mainly intended for the PAS scenario.
Valid values/format	`pending`, `started`, `canceled`, `notdone`, `completed`
Suggested functions	`IS NULL`, `IS NOT NULL`, `IN`, `NOT IN`

User Blocking Conditions

These are blocking conditions used with users.

User Type blocking condition

Fields	Description
Condition name	`[user_type_label]`
Condition type	User
Description	Checks whether the user which created the message belongs to the specified user type.
Valid values/format	User type labels

Shift and Calendar Blocking Conditions

These are blocking conditions used with shifts and calendars.

The calendar is retrieved for the date of the corresponding route (if applicable), or the service request.

Calendar Record Type blocking condition

Fields	Description
Condition name	`[calendar_record_type]`
Condition type	Shift and Calendar
Description	Checks the type of the `regular` calendar record.
Valid values/format	`shift`, `extra_shift`, `working`, `extra_working`, `non-working`
Suggested functions	`IN`, `NOT IN`

Non-Working Reason blocking condition

Fields	Description
Condition name	`[calendar_non_working_reason]`
Condition type	Shift and Calendar
Description	Checks the `regular` non-working reason
Valid values/format	Non-working reason IDs

On-Call Calendar Record Type blocking condition

Fields	Description
Condition name	`[calendar_oncall_record_type]`
Condition type	Shift and Calendar
Description	Checks the type of the `on-call` calendar record.
Valid values/format	`shift`, `extra_shift`, `working`, `extra_working`, `non-worki`
Suggested functions	`IN`, `NOT IN`

On-Call Non-Working Reason blocking condition

Fields	Description
Condition name	`[calendar_oncall_non_working_reason]`
Condition type	Shift and Calendar
Description	Checks the `on-call` non-working reason
Valid values/format	Non-working reason IDs

On-Call Shift blocking condition

Fields	Description
Condition name	`[calendar_oncall_shift]`
Condition type	Shift and Calendar
Description	Checks the `on-call` shift
Valid values/format	Shift IDs

On-Call Work Schedule blocking condition

Fields	Description
Condition name	`[calendar_oncall_work_schedule]`
Condition type	Shift and Calendar
Description	Checks the `on-call` work schedule
Valid values/format	Work schedule IDs

Shift blocking condition

Fields	Description
Condition name	`[calendar_shift]`
Condition type	Shift and Calendar
Description	Checks the `regular` shift
Valid values/format	Shift IDs

Work Schedule blocking condition

Fields	Description
Condition name	`[calendar_work_schedule]`
Condition type	Shift and Calendar
Description	Checks the `regular` work schedule
Valid values/format	Work schedule IDs

Other Blocking Conditions

There are some blocking conditions that aren't related to a specific area. For example, day of the week, day changed, and property blocking conditions. Messages aren't sent, or are blocked, when these conditions are true.

Application blocking condition

Fields	Description
Condition name	[application]
Condition type	Other
Description	Checks the application ID which generated the message.

Fields	Description
Valid values/format	Value of the application ID set on the Configuration, Applications screen.
Case sensitive?	Yes
Notes	The value in the Application field is empty in the following cases: • If the message is generated by a user action in the Core app or Legacy app. • If the message is generated by the Oracle Field Service application. • If the message is generated by a REST API call that uses the OAuth2 assertion grant with a user identity in the assertion for authentication. The value in the Application field is present in the following cases: • SOAP API call • REST API call authenticated through HTTP basic authentication • REST API call authenticated through OAuth2 with client_credentials grant or with the assertion grant without the user identity in the assertion.

Day of Week blocking condition

Fields	Description
Condition name	`[day_of_week]`
Condition type	Other
Description	Checks the day of the message
Valid values/format	**`sun`**, **`mon`**, **`tue`**, **`wed`**, **`thu`**, **`fri`**, **`sat`**
Case sensitive?	No
Suggested functions	**`IN`**, **`NOT IN`**

Interface blocking condition

Fields	Description
Condition name	`[interface]`
Condition type	Other
Description	Checks the interface where the message was created.
Valid values/format	**`web`**, **`soap`**, **`wap`**, **`xhtml`**, **`file_upload`**
Case sensitive?	No
Notes	The interface will be empty if the message is initiated by the server. • web: Corresponds to operations performed from Legacy Manage. • soap: Corresponds to both SOAP (except Inbound API) and REST APIs.

Fields	Description
	• wap: Corresponds to operations performed from Core Application and Android and iOS applications. • xhtml: not in use. • file_upload: Corresponds to Inbound API. • mobile: not in use.

Day Changed? blocking condition

Fields	Description
Condition name	**`[is_day_changed]`**
Condition type	Other
Description	Checks whether the day has changed.
Valid values/format	**`1`**, **`<empty>`**
Suggested functions	**`IS NULL`**, **`IS NOT NULL`**
Notes	This function is mainly intended to be used in the **`move`** scenario. It can't be used to detect the **`move/reschedule`** actions for the **`Add`** launch condition messages.

Resource Changed? blocking condition

Fields	Description
Condition name	**`[is_provider_changed]`**
Condition type	Other
Description	Checks whether the resource has changed.
Valid values/format	**`1`**, **`<empty>`**
Suggested functions	**`IS NULL`**, **`IS NOT NULL`**
Notes	This function is mainly intended to be used in the **`move`** scenario. It can't be used to detect the **`move/reschedule`** actions for the **`Add`** launch condition messages.

Message Day blocking condition

Fields	Description
Condition name	**`[message_day]`**
Condition type	Other
Description	Checks if the day of the message is a holiday
Valid values/format	**`holiday`**, **`regular`**

Fields	Description
Case sensitive?	No
Suggested functions	IN, NOT IN

Previous Message Data blocking condition

Fields	Description
Condition name	[prev_data]
Condition type	Other
Description	Checks the data of the previous message
Valid values/format	Message data
Notes	Can be used to implement the following condition: all failed except failed/INVALID_PHONE_NUMBER

Previous Message Description blocking condition

Fields	Description
Condition name	[prev_desc]
Condition type	Other
Description	Checks the description of the previous message
Valid values/format	Message descriptions
Notes	Can be used to implement the following condition: all failed except failed/INVALID_PHONE_NUMBER

Previous Message Status blocking condition

Fields	Description
Condition name	[prev_status]
Condition type	Other
Description	Checks the status of the previous message
Valid values/format	new, sending, failed, sent, delivered, falsemethod, obsolete
Notes	Can be used to implement the following condition: all failed except failed/INVALID_PHONE_NUMBER

Property blocking condition

Fields	Description
Condition type	Other
Description	Checks the property value for an entity related to the message. Supports resource, activity, inventory, and support request properties.
Valid values/format	Format according to a property type.
Notes	Use **`index`** values for enumeration properties, rather than their translation.

8 Other Reference

Configure mTLS Connection

You can configure the application to send the messages using mTLS connection. With mutual TLS authentication (mTLS), not only does the service side prove its identity by exposing a certificate, but also the clients prove their identity to the servers by exposing a client-side certificate.

In this reference the term "client" refers to Applications Interface, which performs an API call to a "server" (external system), which receives the message and provides the result back in the response. To use mTLS connection, perform these steps and settings:

- Generate root certificate (root ca) and private key for the certificate.
- Generate client private key and client certificate signed with root ca.
- Generate server private key and client certificate signed with root ca.

To set up a channel through Oracle Field Service, follow these steps.

1. Click Configuration, Message scenarios, Delivery Channels to open the Delivery Channels screen. Select or create the necessary delivery channel.

   For more information see *How do I add a delivery channel?*.
2. Configure the connection point with the host and port of the server.
3. Set "TLS 1.2" value in the Connection menu.
4. For the x509 Trust File set content of the root ca file.
5. Set content of client certificate for Client Certificate.
6. Set content of client private key for Client Private Key

   > **Note:** root ca, client certificate and client key files must be in pem format.

7. Check if your server uses mTLS connection:
   a. To check that server supports mtls, run the following command (in linux terminal) against the server:

      ```
      openssl s_client -connect
      SERVER_HOST:PORT -key
      /path_to_client_key_dir/client.key.pem -cert
      /path_to_client_cert_dir/client.cert.pem -CAfile
      /path_to_rootca_cert_dir/cacert.pem -state
      ```

      Open ssl will print information about mtls connection establishing, the output should not have any error messages.
   b. To check that generated certificates are ok, run openssl server:

      ```
      openssl s_server -accept PORT -CAfile
      /path_to_rootca_dir/cacert.pem -cert
      /path_to_server_cert_dir/server.cert.pem -key
      /path_to_server_key_dir/server.key.pem -state
      ```

   c. Make request from openssl client and then check logs.

Notification Methods

These are the supported notification methods and their functional features.

Recipient Method

Supported Features	Description
Email	Yes
External launch condition	No. Always `Customer`.
External System	Yes
Set Property	Yes

Send At Method

Supported Features	Description
Email	Yes
External launch condition	No. Always `time of event`.
External System	Yes
Set Property	Yes

Within Method

Supported Features	Description
Email	Yes
External launch condition	Yes
External System	Yes
Set Property	Yes

Block messages for specific days Method

Supported Features	Description
Email	Yes
External launch condition	Yes
External System	Yes

Supported Features	Description
Set Property	Yes

Block messages for Holidays Method

Supported Features	Description
Email	Yes
External launch condition	Yes
External System	Yes
Set Property	Yes

Shift Blocked Messages Method

Supported Features	Description
Email	Yes
External launch condition	Yes
External System	Yes
Set Property	Yes

Number of Attempts on Failed Status Method

Supported Features	Description
Email	Yes
External launch condition	No
External System	Yes
Set Property	No

Number of Attempts on Sent Status Method

Supported Features	Description
Email	No
External launch condition	No
External System	Yes
Set Property	No

Customer Notification Time Method

Supported Features	Description
Email	Yes
External launch condition	Yes
External System	Yes
Set Property	No

Email Address Source/Phone Number Source Method

Supported Features	Description
Email	Yes
External launch condition	No
External System	No
Set Property	No

Reply Address Method

Supported Features	Description
Email	Yes
External launch condition	No
External System	No
Set Property	No

Survey-PAS Survey Support Method

Supported Features	Description
Email	No
External launch condition	No
External System	Yes
Set Property	No

Default Encoding Method

Supported Features	Description
Email	-
External launch condition	XML
External System	XML
Set Property	-

Time zone of date/time values in subject/body of messages Method

Supported Features	Description
Email	based on recipient
External launch condition	based on recipient
External System	based on recipient
Set Property	based on recipient

Message Removal Cases

There are several actions in the application that might remove the existing messages, based on certain conditions. The messages are removed only if they haven't been sent.

Activity Start action

Event/Action	Description
Removed Messages	`Reminder`, `Change`, `Day before`
Status	`obsolete`
Description	`ACTIVITY_WAS_STARTED`

Activity Cancelation action

Event/Action	Description
Removed Messages	All
Status	`obsolete`
Description	`ACTIVITY_WAS_CANCELED`

Activity Notdone action

Event/Action	Description
Removed Messages	All
Status	`obsolete`
Description	`ACTIVITY_WAS_NOT_DONE`

Deletion of a Pending Activity action

Event/Action	Description
Removed Messages	All
Status	`obsolete`
Description	`ACTIVITY_WAS_DELETED`

Activity Suspend action

Event/Action	Description
Removed Messages	`Started`
Status	`obsolete`
Description	`ACTIVITY_WAS_SUSPENDED`

Activity Reschedule action

Event/Action	Description
Removed Messages	All except `SLA Warning`
Status	`obsolete`
Description	`ACTIVITY_WAS_RESCHEDULED`

Activity Move action

Event/Action	Description
Removed Messages	`Reminder`, `Change`, `Not started`, `Service window warning`, `Call ahead`, `Add`
Status	`obsolete`
Description	`ACTIVITY_WAS_MOVED`

Convert an activity to not ordered

Event/Action	Description
Removed Messages	**Reminder**, **Change**
Status	**obsolete**
Description	**ACTIVITY_IS_NOT_ORDERED**

Reminder message creation action

Event/Action	Description
Removed Messages	**Customer** messages: Recipient = **Customer**
Status	**obsolete**
Description	**NEW_CUSTOMER_MESSAGE_WAS_CREATED**
Notes	The **Reminder** launch condition is not invoked if the existing customer messages cannot be dropped using the **drop_message** call (if required).

Change message creation action

Event/Action	Description
Removed Messages	**Customer** messages: Recipient = **Customer**
Status	**obsolete**
Description	**NEW_CUSTOMER_MESSAGE_WAS_CREATED**
Notes	The **Change** launch condition is not invoked if an incomplete **Reminder** exists, or if the existing customer messages cannot be dropped using the **drop_message** call (if required).

Cancel visit action

Event/Action	Description
Removed Messages	All (visit related)
Status	**obsolete**
Description	**VISIT_WAS_CANCELED**

Delete visit action

Event/Action	Description
Removed Messages	All (visit related)

Event/Action	Description
Status	`obsolete`
Description	`VISIT_WAS_DELETED`

Start visit action

Event/Action	Description
Removed Messages	`Visit reminder`, `Visit change #`, `Visit day before`
Status	`obsolete`
Description	`VISIT_WAS_STARTED`

Applying new visit formulas action

Event/Action	Description
Removed Messages	All (visit related)
Status	`obsolete`
Description	`VISIT_WAS_RECALCULATED`
Notes	The messages are only removed if the visit is removed as the result of applying formula changes.

Block/Shift messages action

Event/Action	Description
Removed Messages	N/A
Status	`false`
Method Description	`NONWORKING_DAY`
Notes	This removal is performed if message sending of is not allowed for a non-working day (or a holiday) and such message cannot be shifted to an appropriate working day.

Revision History

This document will continue to evolve as existing sections change and new information is added.

Date	What's Changed	Notes
February 2022	Minor updates	