5.3 Workflow Builder Reference Guide

Bright Pattern Documentation

Generated: 6/16/2021 2:08 pm

Content is available under license unless otherwise noted.

Table of Contents

Purpose

The Bright Pattern Contact Center Workflow Builder Reference Guide describes the building blocks of the Bright Pattern workflow language and how those blocks are managed in the Workflow Builder application. The function of each workflow block, its parameters, and usage are explained.

For information about workflow management in the context of contact center configuration, such as the association of workflows with interaction access points, refer to the Bright Pattern Contact Center Administrator Guide.

Audience

The Bright Pattern *Workflow Builder Reference Guide* is intended for professionals responsible for the design, development, and testing of interaction processing logic in your contact center.

Participants are expected to be familiar with general principles of computer programming and to have a solid understanding of contact center operations and resources that are involved in such operations, including agents and teams, services and skills, schedules, and access points.

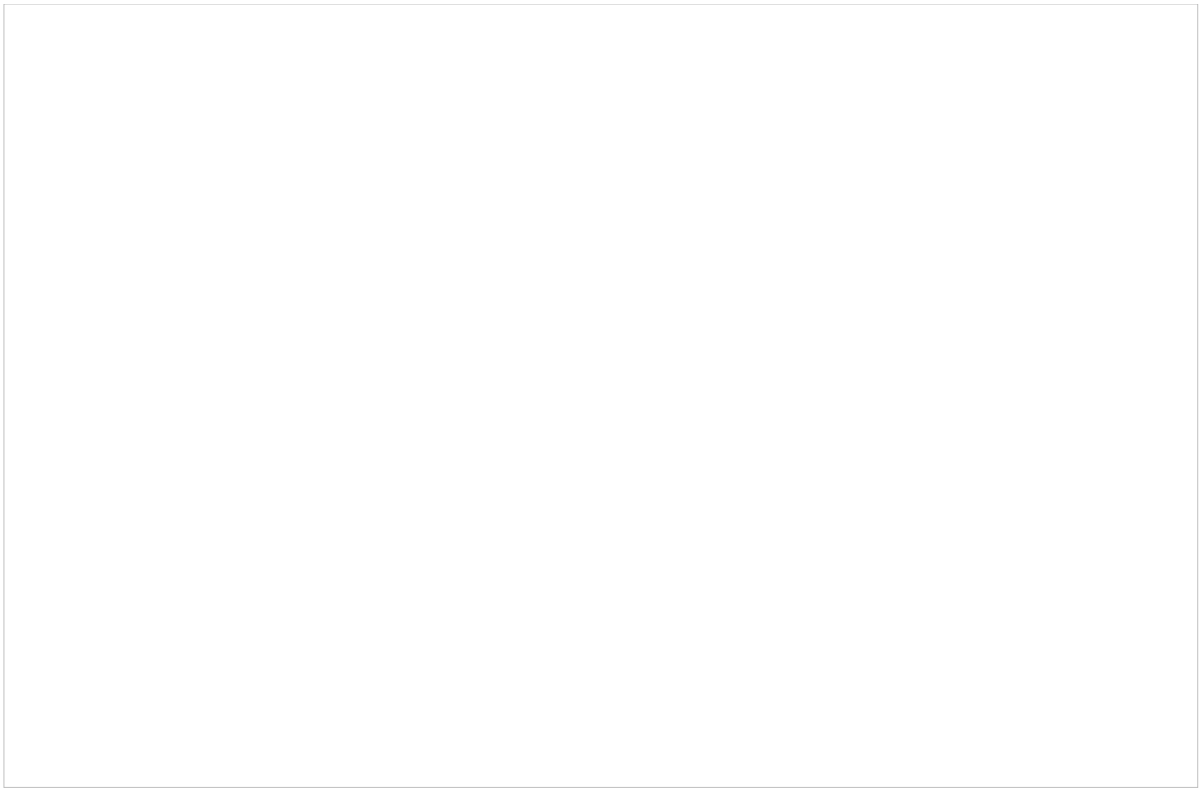
Workflow Builder Overview

Workflows are a series of events configured to launch automatically after an interaction has been assigned a disposition. For example, setting a particular disposition for an interaction could trigger the following series of events to happen:

- Send a survey to the customer
- Wait a specified amount of time (e.g., 30 minutes, 1 day, 1 week, etc.)
- Send a follow-up email to the customer.

Workflows are designed and edited in the Workflow Builder application. This application is launched from the Contact Center Administrator application when you either add a new workflow or select an existing one for editing. Workflows handle any necessary follow-ups, depending on the interaction's disposition type. Such follow-up actions include setting context variables, sending an email, sending a text message, scheduling an event, clearing a scheduled event, and making an external request (or internal API call). Workflows reduce the workload of agents while following up with customers and gathering data in a consistent way.

While similar to the Scenario Builder application, the Workflow Builder application only dictates the follow-up actions that happen after an interaction has ended.



Workflow Builder

Graphical User Interface

Workflow Builder incorporates a graphical user interface (GUI) with which you can visually connect a sequence of functional blocks, thus building your workflow. These blocks are known as *workflow blocks*. Workflows are created using a flowchart format that represents the sequence of interaction processing steps in the workflow. Different workflow blocks perform different functions, such as sending internal messages, waiting a period of time to take an action, or looking for available agents.

To add a block to the workflow, select it from the list on the left and drag it to the desired location within the flowchart. To remove a block from a workflow, select the block and drag it back to the list of blocks on the left.

Workflow Blocks

Each block has its own configuration attributes, which appear in the edit pane on the right when the block is added to the flowchart or selected within the flowchart. The attributes specify the function represented by the block. The scenario blocks described in this guide may have configuration attributes related to conditional exits, prompts, and/or settings.

Conditional Exits

The workflow typically processes blocks sequentially; however, some blocks have multiple paths that the scenario can take after processing the block. These paths are called conditional exits. Conditional exits enable you to determine how the voice scenario responds to certain conditions that may occur during the processing of an interaction, such as an agent not responding to a call. Each conditional exit appears in the flowchart as green text beneath the block to which it applies. A conditional exit may contain a flow of blocks to handle specific situations.

Settings

Settings, also known as configuration attributes, for workflow blocks appear in the edit pane on the right when a block is added to the flowchart or selected within the flowchart. These settings specify the function represented by the block. The subsequent sections of this guide describe specific workflow blocks, their attributes, and usage. The blocks are listed in alphabetical order.

Workflow Entries Screen Properties

Workflows are added to services as workflow entries. To add a workflow, navigate to *Configuration* > *Workflows* and select the **+** button. A new workflow will open in the Workflow Builder application, where you customize the workflow scenario by dragging and dropping blocks onto your scenario.

Name

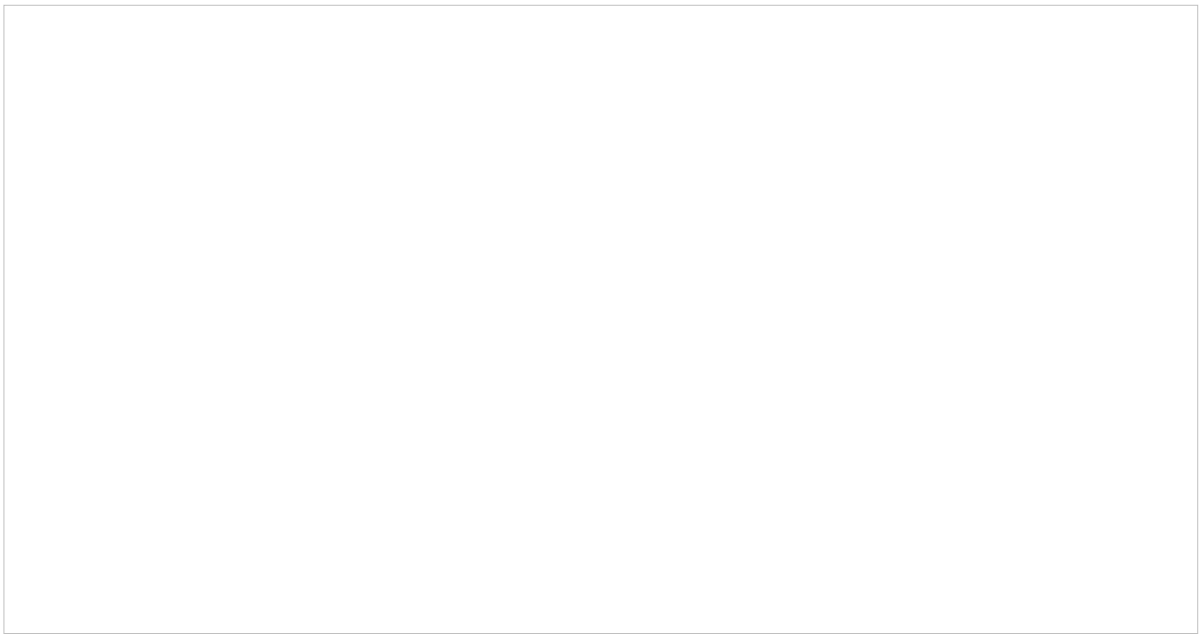
Every workflow entry needs a *name*. This field is mandatory.

Service

Service is the name of the service with which the workflow entry should be associated.

Triggers

Triggers are the events that cause a configured workflow to be launched. To add triggers, click **add**, select the available triggers and their properties, and select the green check mark to apply your changes.



Workflow triggers

Agent completes interaction with disposition

This trigger causes the workflow to start if the interaction ends at the specified disposition set only by the agent.

If this trigger is selected, you must select a specific disposition from the drop-down list, which can include the following:

- Any disposition
- Product sold

- Offer rejected
- Bad record
- Add to DNC
- Record matches DNC
- Number matches DNC
- Wrong person
- Wrong number
- Dropped while talking
- Busy
- Fast busy
- No answer
- Fax or modem
- Announcement
- Network announcement
- Answering machine
- Silence
- Invalid number
- Other connection issue
- Abandoned
- Congestion
- No agent
- No disposition
- Skipped
- System failure
- Agent failure
- Phone network unavailable
- Finalized by dial rule
- No numbers left to call
- Disposition not found
- Record expired
- Call failed
- Erase content

Interaction ends with average sentiment

This trigger causes the workflow to start if the average sentiment matches the specified statement value.

If this trigger is selected, you must show the percentage of the average sentiment. This is done by selecting< or > and moving the percentage slider to the complete the following type of example statement: *A workflow is triggered if the average sentiment is greater than 0.51.*

Interaction ends with last disposition

This trigger causes the workflow to start if the interaction ends with the specified disposition. Note that this trigger differs from "Agent completes interaction with disposition" in that the disposition for this trigger can be set by either the agent or the scenario (i.e., some interactions can take place without the participation of an agent).

If this trigger is selected, you must select a specific disposition from the drop-down list, which can include the following:

- Any disposition
- Product sold
- Offer rejected
- Bad record
- Add to DNC
- Record matches DNC
- Number matches DNC
- Wrong person

- Wrong number
- Dropped while talking
- Busy
- Fast busy
- No answer
- Fax or modem
- Announcement
- Network announcement
- Answering machine
- Silence
- Invalid number
- Other connection issue
- Abandoned
- Congestion
- No agent
- No disposition
- Skipped
- System failure
- Agent failure
- Phone network unavailable
- Finalized by dial rule
- No numbers left to call
- Disposition not found
- Record expired
- Call failed
- Erase content

How to Solicit Post-Transactional Surveys via Email

Survey forms are designed to be emailed your customers after interactions are completed. Email interactions are not dictated by scenario, so while it is not possible to utilize the interactive survey features associated with chat or voice scenarios, survey forms provide you an opportunity to request satisfaction input.

Survey forms are created and edited in the Survey Form Editor application and can be distributed by post-transaction workflows. Additionally, a single survey form may be configured in any number of required languages. Note that survey forms are associated with the quality management features of Bright Pattern Contact Center software and are different from the forms configured in the Form Builder application.

Note: Although this article outlines how to solicit surveys based on email interactions, the process may be applied for voice or chat interactions, too.

Prerequisites

This example requires you to have a configured email service with dispositions, an email scenario entry, as well as a working SMTP configuration.

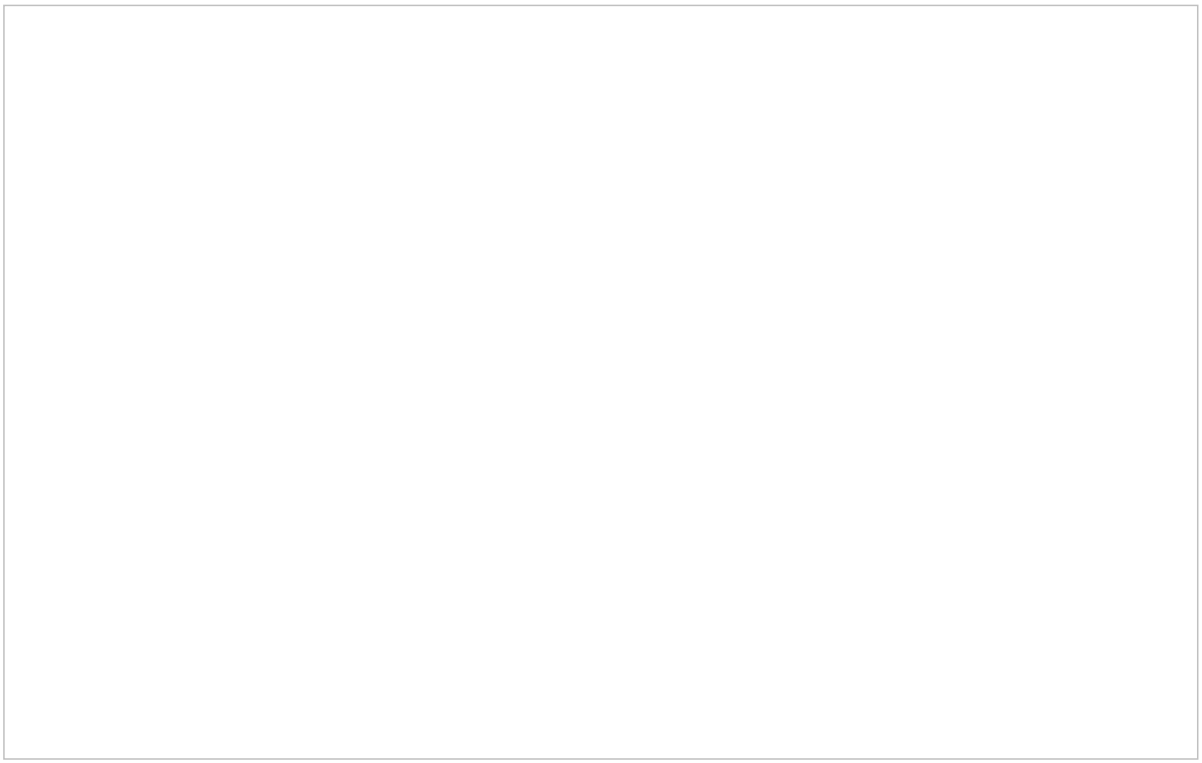
Additionally, if you wish to map custom fields to questions on your form, you will need to configure them first in the Contact Center Administrator application, section Custom Survey Fields.

Procedure

The following sequence of actions illustrates the steps necessary to take in the various Bright Pattern Contact Center applications in order to send post-transactional surveys via email.

Step 1: Define a Form in Contact Center Administrator

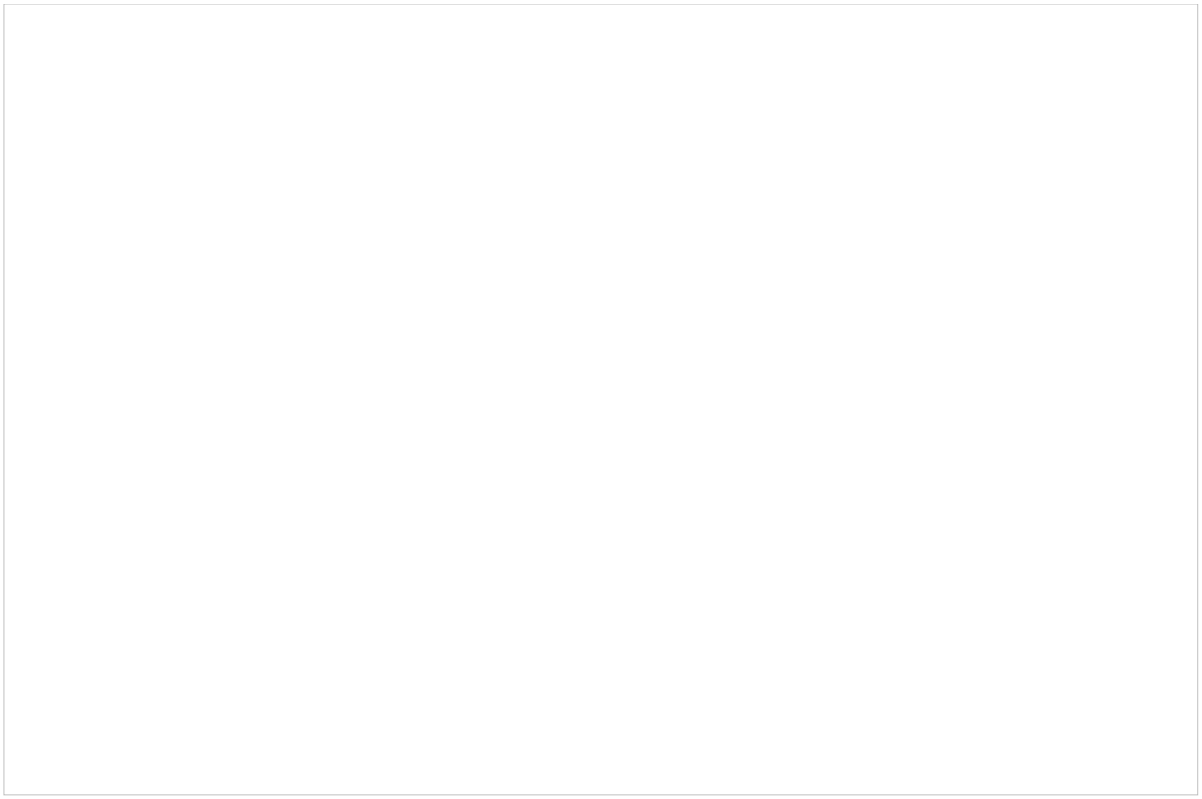
In the Contact Center Administrator application, section Quality Management > Survey Forms, click the **Add new form** button; this will launch the Survey Form Editor application.



Quality Management > Survey Forms

Step 2: Create Your Form in Survey Form Editor

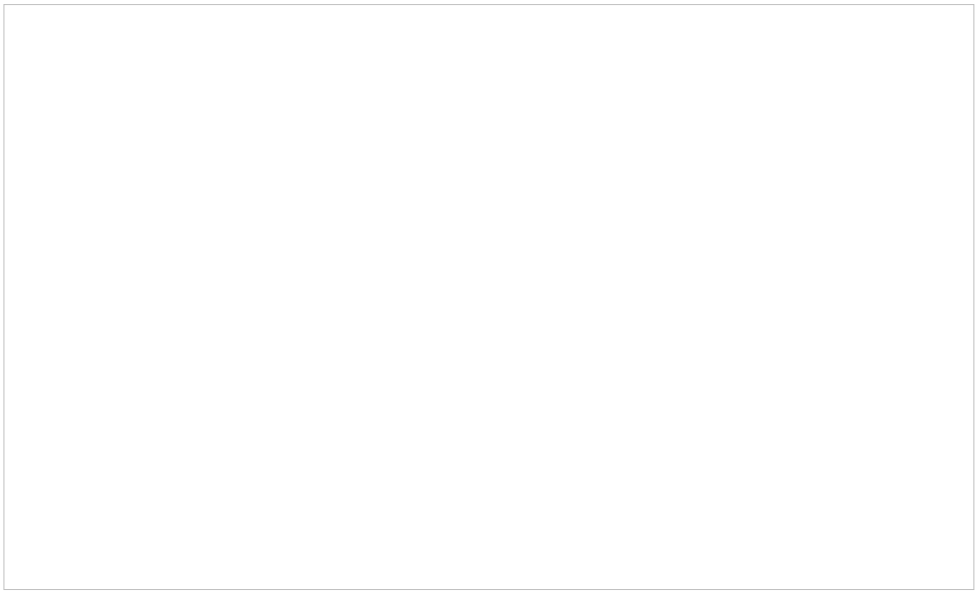
In the Survey Form Editor application, you will configure how your form looks, what questions it will have, and what languages it will be available in; this is accomplished by using the form controls and settings. Note that certain questions can have reporting metrics or fields mapped to them; these values will then appear on reports.



How a completed form looks in the Survey Form Editor

Step 2a: Understanding Basic Form Controls

To define a new question for your form, select the corresponding control from the control palette, drag it to the desired location on the canvas, and specify its settings in the properties editor.



Check Box Elements properties editor

Notice these form controls and elements:

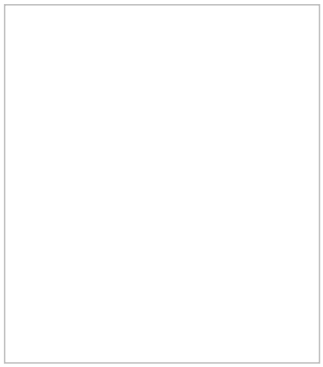
- **edit** lets you add or edit a question's settings in the properties editor.
- **delete** lets you remove a question from the form.
- **copy** lets you create a copy of a question; the copied question will appear below the original.
- **Save** saves the form.
- **Text** is the field where you add the text of your option/answer.
- **Value** is the field where you may enter a number or unique value that will be associated with this option; values can be passed to reports.
- **N/A** checkbox, when selected, prevents any value being associated with the option.
- **Field** drop-down menu allows you to select reporting metrics/fields to map to a given question. The following field options are available:
  - CSAT
  - FCR
  - NPS
  - Any custom fields created in the Contact Center Administrator application, section Custom Survey Fields

When you are finished configuring your form, click **Save**. If you are saving a form for the first time, you will be prompted to name your form in a pop-out window.

Step 2b: (Optional) Configure Your Form in Different Languages

When creating your form, the default language is the language of your contact center. To create a new version of your current form in a different language, do the following:

1. After having saved your form in the original language, select a new language from the drop-down language selector.

   Language selector

2. Begin changing the language of the current questions to the new language; also, you may add new questions.
3. Click the **Save** icon; your form will be configured in two languages.
4. Repeat this process for as many languages as you require.

Step 2c: (Optional) Customize the Look of Your Form with Custom CSS

Note that it is possible to edit the look of your form by using custom CSS. To launch the CSS editor, click the **Edit form properties** icon. To preview your form, select the **Preview** icon.

Custom CSS editor

Step 3: Configure a Workflow

When your survey form is complete, you will need a way to distribute it via email that happens post-interaction; this can be accomplished with a workflow.

Configuring a workflow to send a survey form requires an EMail block. It is recommended that you add a Wait block as well (see Wait Block Settings, below).

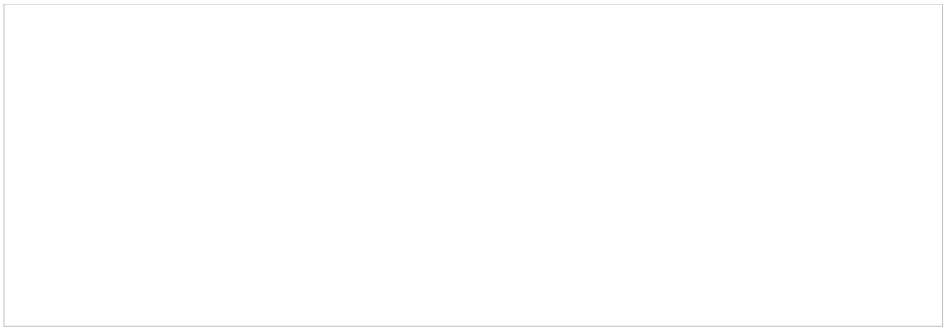
EMail Block Settings

When configuring the settings for this block note the following:

- In order for the survey to be sent to the correct customer's email address, enter the $(item.from) variable in the To: Address(es) field; this variable pulls in the email address of the customer whose email has been assigned the correct disposition.
- In the **Template language** field, you should enter the ISO language code associated with the selected language (e.g., "en" for English, "ja" for Japanese, etc.).
- When composing the message body, click the **Survey link** button, and select the appropriate form from the drop-down menu; selecting the form inserts the survey URL into the body of your message.

Wait Block Settings

Note that for surveys sent via workflows, the survey arrives faster than email. When soliciting post-transactional surveys via email, we recommend adding a Wait block to the workflow and specifying a Wait duration of 24 hours, as shown. Doing so will delay the workflow for 24 hours before sending the survey, ensuring that the survey is available for the customer to read when they answer the email.



Wait block configuration

Step 3b: (Optional) Send a survey form in a different language

If you configured your survey form in a different language (from step 2b of this procedure), follow these steps to configure an EMail block in that specific language and insert the survey form link associated with it. In this example, we will be sending a survey in Japanese.

1. Create a Japanese version of your survey form (from Step 2b above).

2. Configure a workflow with an EMail block (from Step 3 above).

3. In the EMail block settings, make sure that the *Template language* field contains the correct language value (e.g., "ja"), or that it refers to a variable containing the correct language.

4. Create a Japanese version of the email body and insert survey link (from Step 3 above). The link will look like this:

   https://<tenant>.brightpattern.com/clientweb/survey/$(item.globalInteractionId)/$(item.interactionId)/<survey_name>?lang=ja&tenantId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

5. Save the workflow with a new name (e.g., "CupIQ Survey - Japanese") so you can differentiate this survey from others in different languages.

Step 4: Add a Workflow Entry to your Workflow

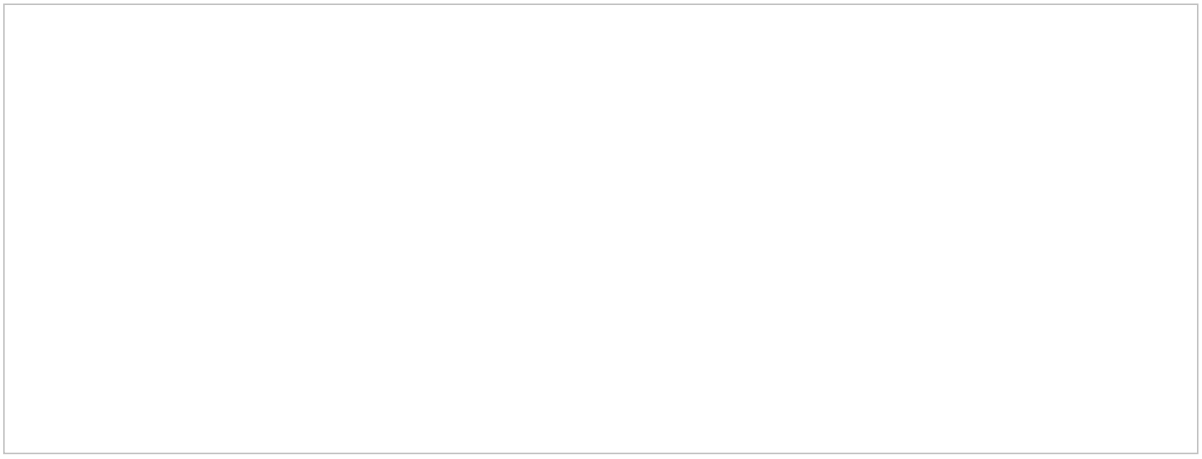
When you are finished configuring your workflow, you will need to create a workflow entry; this associates the workflow with your configured service. To send the email post-transaction, do the following:

- Select the service from the drop-down menu
- Configure what triggers (i.e., dispositions) will launch the workflow.

A workflow entry

Troubleshooting

If your workflow is giving you trouble, you can configure the EMail block'sMail not sent conditional exit with the Internal Message block; this sends a message to a specific Agent Desktop user if the email was not sent.



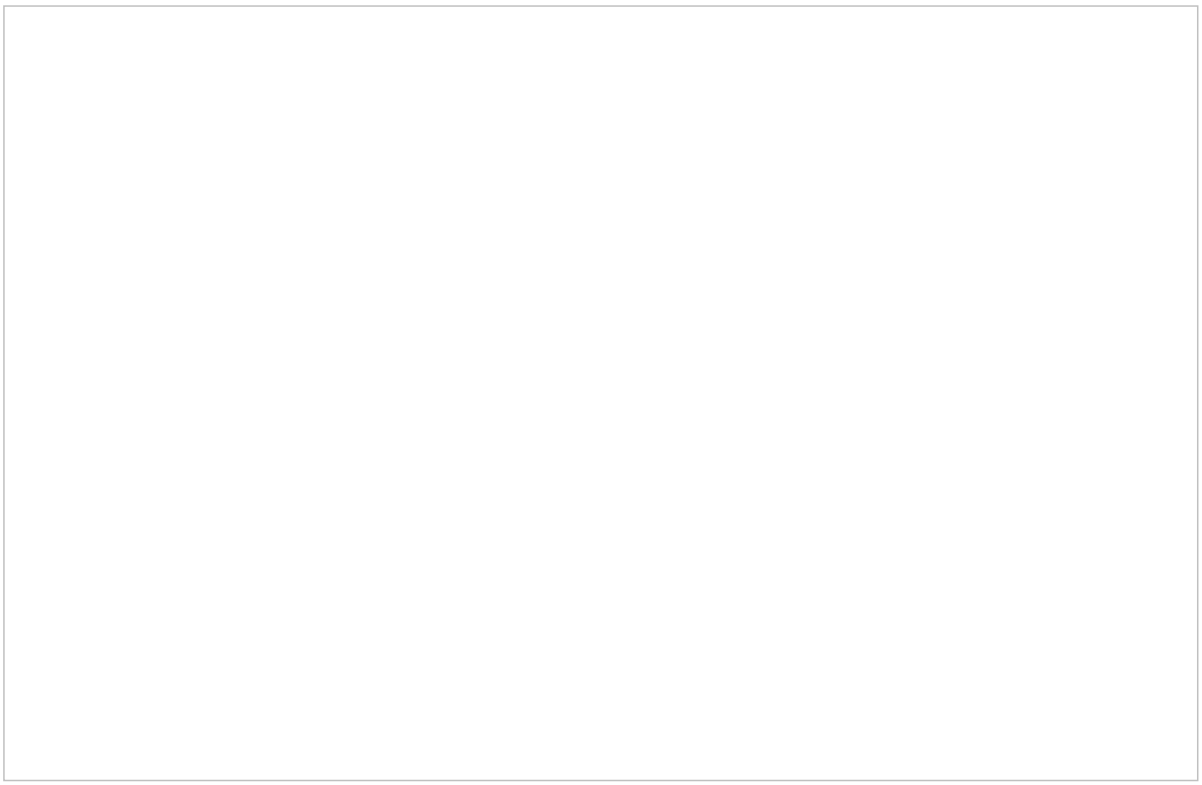
Use the Internal Message block to see if your email was not sent

How to Create a New Zendesk Ticket on Behalf of Another

Zendesk User in Workflows

You can use the Zendesk API Request workflow block to make API requests on behalf of any end user, and because you are an authenticated admin, the end user's email does not need to be verified. If a record is created, updated, or deleted, the change is associated with the end user, not you.

This article will show you how to use the Zendesk API Request workflow block to create a new Zendesk ticket on someone else's behalf.



Example workflow in the Workflow Builder application

Prerequisites

- Have administrator privileges for Bright Pattern Contact Center.
- Use Bright Pattern Contact Center version 5.5.x or later.
- Have a Zendesk integration account configured for your contact center.
- In the integration account properties, set OAuth access token as the authorization type.
- Have administrator access to the Zendesk user interface and API.

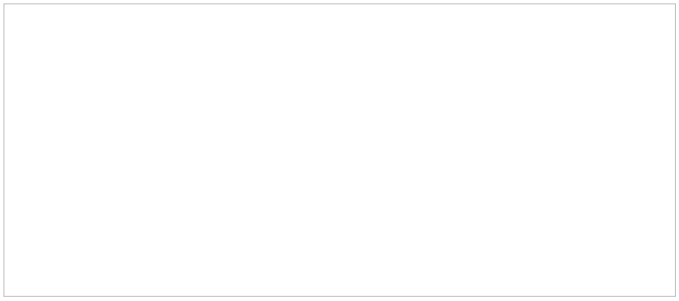
Procedure

Step 1: Create a workflow

In the Contact Center Administrator application, section *Workflows*, create a new workflow that includes the following workflow blocks: Zendesk Select Account and Zendesk API Request.

Step 2: Specify a Zendesk integration account

In the Zendesk Select Account block properties, select a Zendesk integration account that is configured for your contact center.



Zendesk Select Account scenario block properties

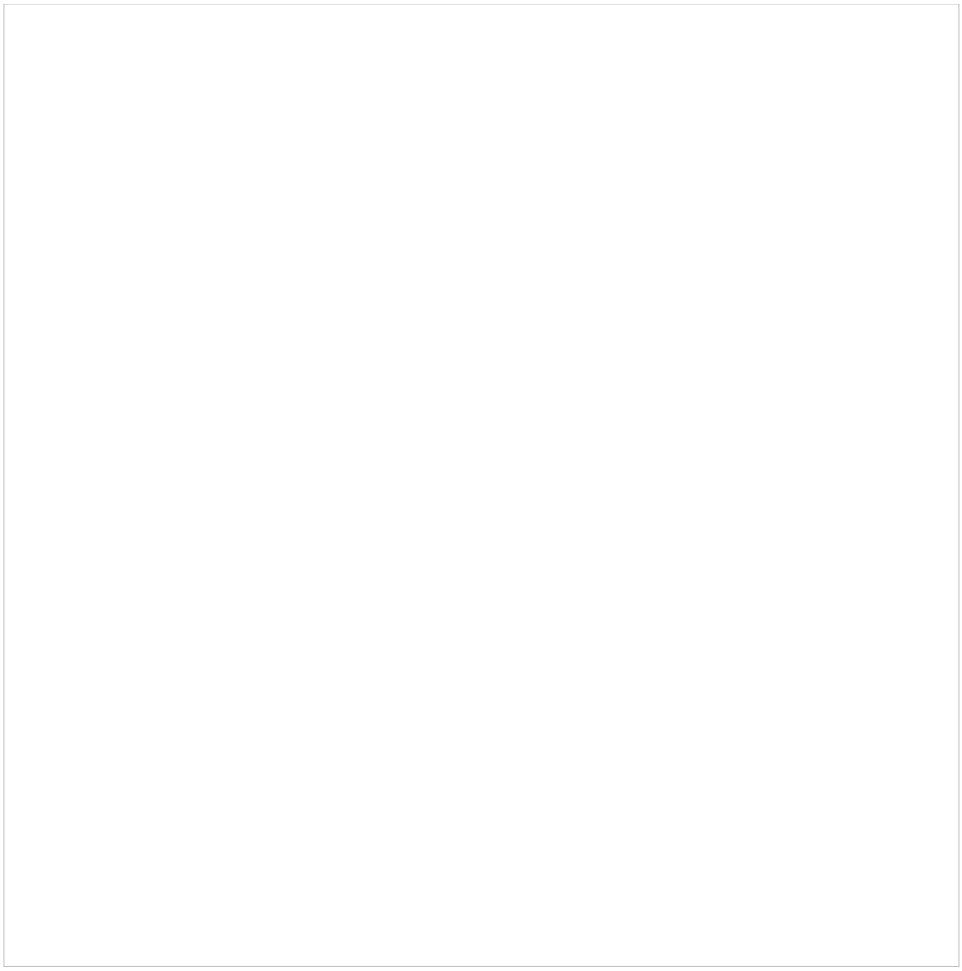
Step 3: Configure Zendesk API Request properties

In the Zendesk API Request scenario block properties, set the following:

- **Title text** - Type a name for this block.
- **Request type' - *Set** POST*.
- **URL** - Set **tickets.json**.
- **Extra headers** - Click **add** to add the an extra header with the following name (key) and value:
- - **Name** - Set **X-On-Behalf-Of**.
- - **Value** - Set the email address of the user who you are making the request for in the Zendesk system (e.g., "first.last@email.com"); note that the value must be a real user in the Zendesk system.
- **Raw JSON** - Paste your ticket object properties as raw JSON (see *Sample Raw JSON* below):
- **Initial path in the result JSON** - Set **subject**.
- **Scenario variable prefix for JSON data** - Set **zendesk.subject**.
- **Use GetNext block to loop through data** - Leave unchecked.

Sample Raw JSON:

```
{
"ticket": {
"subject": "Behalf Of Header",
"comment": { "body": "first.last@email.com" },
"priority": "urgent"
}
}
```



Zendesk Request API workflow block properties with completed fields

Step 4: Save the workflow

When you are done setting up the workflow, click**Save** to save your changes.

Step 5: Test the workflow

Test the workflow by doing the following steps:

1. In Zendesk, log in to the integrated Agent Desktop widget as an agent who is assigned to a team that can handle service calls, and make yourself Ready.

2. From a separate phone, call your contact center's access number. The call should be routed to you and at the same time, the Zendesk system should then contain a new ticket with the correct information. The ticket requester should be the user that was set in the extra header of the Zendesk Request API block.

3. Look up the ticket in Zendesk and check that the requester is the correct user.

Assign Case to Agent

This block assigns the case to a selected agent.

Workflow Builder Block

Settings

Title Text

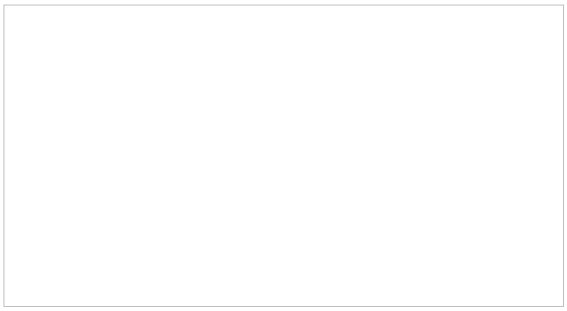
Title Text is the name of the instance of the block. Enter a name in the text field and the new name of the block will appear in the flowchart.

Agent

The *Agent*field where you may select from a drop-down menu the name of the agent you would like to assign cases to.

Agent ID

Agent ID is where you may enter the ID of the agent you would like to assign cases to. You may also enter variables in this field (i.e., $agentID). Note the Agent ID field overrides the Agent field.



Assign Case to Agent settings

AWS Lambda

The AWS Lambda block allows you to invoke functions (i.e., programs) you have created in your Amazon AWS Lambda account. Note that you must have an Amazon AWS Lambda integration account configured in order to invoke functions.

AWS Lambda block

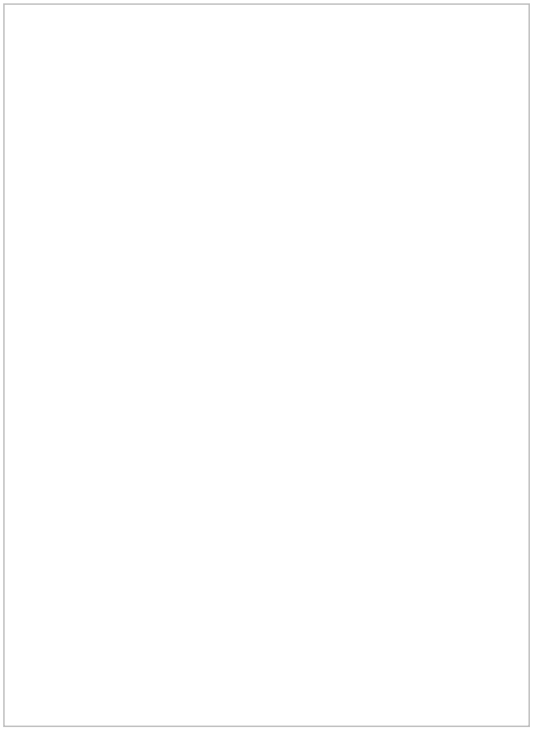
Conditional Exits

Failed

This exit is taken if the function invocation failed due to invalid parameters, timeout, or network connectivity to Amazon AWS Lambda.

For troubleshooting purposes, use the EMail or Internal Message block to obtain content of responses indicating a failed attempt to create an object. For more information, see the description of the variable $(integrationResultBody).

Settings



AWS Lambda block settings

Title text

The name of the instance of the block (any name).

Integration account

The name of the Amazon AWS Lambda integration account that will be used for access to your Lambda functions.

Function name

The name of the specific Amazon AWS Lambda function that will be invoked in this workflow.

Invocation type

RequestResponse

Requests a response from the invoked function.

Event

Does not wait for a response from the invoked function.

DryRun

Acts as a diagnostics tool, allowing you to test everything except the actual function code.

Parameters

The parameters to invoke from the function. You may choose *Field*, which allows you to enter in the specific function *names* and *values*, or *JSON*, which allows you to enter the parameters in JSON format. Note that the Field option will be converted to JSON.

Response variable name

The variable name that contains a parsed JSON object from the invoked function (i.e., it does not include the payload).

Response payload name

The variable name that contains the payload (i.e., the output) from the invoked function; the payload is limited to 50 KB. Note that the name will always be a string.

Treat payload as

Allows you to choose the way the payload is received; you may choose either *JSON* or *String*. Note that the default option is JSON.

Bright Pattern Create Object

The *Bright Pattern Create Object* block inserts a new object into the internal contact database. The allowed object types are Case, Contact, Company, and Activity History.

Conditional Exit

The Bright Pattern Create Object block may take the Failed conditional exit.

Failed

The *Failed* conditional exit is executed if the creation operation failed.

Settings

Title Text

Title Text is the name of the instance of the block. Enter a name in the text field and the new name of the block will appear in the flowchart.

Object Type

The *Object Type* drop-down menu allows you to choose the type of object you want to create in the internal database. You may choose **Case**, **Contact**, **Company**, or **Activity history**. Depending on what object type is selected, the settings change as follows:

Case Settings

Case title

This is the text you would like associated with the case title.

Category name

The Category names that have been configured in section Case & Contact Management > Case Categories will appear here in a drop-down menu.

Priority

This field allows you to enter a text or variable item associated with the case's priority.

Reporter contact ID

This field allows you to enter a text or variable item associated with the the reporter contact ID (i.e., customer).

Custom Case Fields

Custom Case Fields that have been configured in Case & Contact Management > Custom Fields > Case will appear here; you may enter text items in these fields.

Return variables

The *Return variables* settings allow entry of text or variable items in the form of *Generate case numbers* or *Generated case IDs*.

Generated case number

This is the variable item associated with the case number.

Generated case ID

This is the variable item associated with the case ID.

Contact Settings

First name

This is the text associated with the first name of the contact.

Last name

This is the text associated with the last name of the contact.

Title

This is the text associated with the title of the contact.

Position

This is the text associated with the position of the contact.

Summary

This is the text associated with the summary of the contact.

Segment

This is the text associated with the segment of the contact.

Date of birth

This is the text associated with the contact's date of birth.

Company ID

This is the ID of the company record.

Emails

When you click the **add** option, you may select the type of email from the drop-down menu (**Primary**, **Business**, or **Private**), then enter the email address. It is possible to add more than one type of email address; the order the emails are listed in is insignificant. Note: Gaps are not allowed (i.e. you must create the email with the index [0] if you want to populate the email with the index [1]).

Phones

When you click the **add** option, you may select the type of phone from the drop-down menu (**Business**, **Home**, **Mobile**, or **Fax**), then enter the phone number. It is possible to add more than one type of phone number; the order the numbers are listed in is insignificant. Note: Gaps are not allowed (i.e. you must create phone numbers with the index [0] if you want to populate the email with the index [1]).

Custom Contact Fields

Custom Contact Fields that have been configured in Case & Contact Management > Custom Fields > Contact will appear here; you may enter text items in these fields.

Return variables

The Return variables settings allow entry of text or variable items in the form of Generate record.

Generated record

This is the variable item associated with the contact record number.

Company Settings

Company name

This is the text associated with the company name.

Web URL

This is the text associated with the web URL of the company.

Revenue

This is the text associated with the revenue of the company.

Employees

This is the data associated with the employees of the company; the allowed range is from 0 to 2147483647.

Custom Company Fields

Custom Company Fields that have been configured in Case & Contact Management > Custom Fields > Company will appear here; you may enter text items in these fields.

Return variables

The Return variables settings allow entry of text or variable items in the form of Generate record ID.

Generated record ID

This is the variable item associated with the contact record ID.

Activity History Settings

Notes

This is the text associated with activity history notes; notes are written on behalf of a party. The notes should support HTML links, bullets, and bold, italic, underline, and strikethrough text.

Disposition

This is the text associated with activity history dispositions; dispositions are written on behalf of a party.

Custom Activity History Fields

Custom Activity History Fields that have been configured in Case & Contact Management > Custom Fields > Activity History will appear here; you may enter text items in these fields.

Return variables

The Return variables settings allow entry of text or variable items in the form of Generate record ID.

Generated record ID

This is the variable item associated with the activity history record ID.

Bright Pattern Delete Object

The *Bright Pattern Delete Object* deletes an object from the internal database. The object types allowed for deletion are Case, Contact, and Company. Note the following about the block:

- The deletion of cases deletes all activity history related to the case.
- The deletion of contacts deletes activity history associated with this contact that is not linked to any cases.
- The deletion of contacts erases reference to the contact from cases and activity history items linked to the cases but keeps the cases.
- The deletion of contacts erases references to the contact from the calendar but does not remove calendar entries.
- The deletion of contact erases references to contact from emails in queue (queue_items.reporter_id), but keeps emails in queue.
- The deletion of companies erases reference to it from the contacts but does not delete any contacts.

Conditional Exits

Failed

The *Failed* conditional exit is executed if the deletion operation failed, either through timeout or failed network connectivity to an external CRM (e.g. Salesforce).

No Data

The *No Data* conditional exit is executed if no data matching the specified deletion criteria is found.

Settings

Title text

Title Text is the name of the instance of the block. Enter a name in the text field and the new name of the block will appear in the flowchart.

Object type

The *Object type* drop-down menu allows you to choose the type of object you want to delete in the internal database. You may choose **Case**, **Contact**, or **Company**.

Object ID

This is the database record ID.

Bright Pattern Search Object

The *Bright Pattern Search Object* block finds an existing object in the database and retrieves activity data.

Note: For a list of standard fields returned by this block, see section Standard Fields for CRM Objects

How to Use This Block

This block may be used to retrieve activity data. To do this, create a workflow using this block with an Activity history object, *Global Interaction ID* criterion. If the workflow is configured to start immediately after an interaction, make sure to add a reasonable delay (e.g., the Wait block) as it takes a few seconds to write activity history. Next, use the Get Next Record block to browse through recordset returned by this block.

After configuring the workflow, configure an activity form with custom activity history fields. When you run an interaction using this activity form and trigger the workflow, make sure the agent creates a case and adds a few notes during the interaction.

Conditional Exits

Failed

The *Failed* conditional exit is executed if the search operation failed, due to invalid parameters, timeout, or network connectivity to external CRM (e.g. Salesforce).

No Data

The *No data* conditional exit is executed if no data matching the specified search criteria is found.

Settings

Title Text

Title Text is the name of the instance of the block. Enter a name in the text field and the new name of the block will appear in the flowchart.

Object type

The *Object type* drop-down menu allows you to choose the type of object you want to search for in the internal database. You may choose **Case**, **Contact**, **Company**, or **Activity History**. Depending on what object type is selected, the settings change as follows:

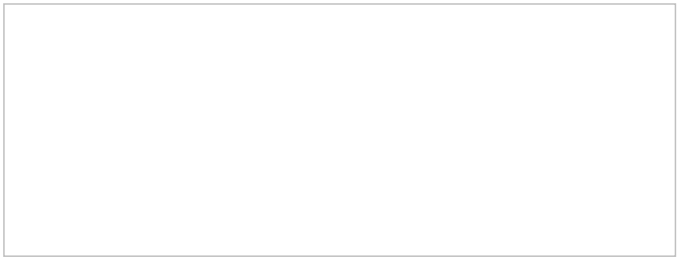
Search

Search settings will vary depending on what object type you are searching for. The options are listed as follows:

Case

To search by case, select either **ID**, **Number**, or a **custom case field** from the first drop-down menu. Note: Custom case fields will not appear in the *Search* menu unless the checkbox **Searchable in scenarios and workflows** has been selected. For more information, see Custom Fields.

For any of these options, you may then select **Equal**, **Empty**, or **Not empty** from the second drop-down menu. If you select **Equal**, a field will appear and you will have the option to enter a text or variable item.

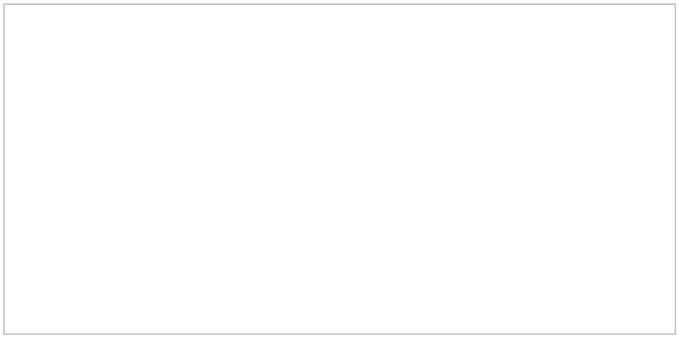


Bright Pattern Search Object case settings

Company

To search by company, select **ID**, **Name**, **URL**, a **custom company field** from the first drop-down menu. Note: Custom company fields will not appear in the *Search* menu unless the checkbox **Searchable in scenarios and workflows** has been selected. For more information, see Custom Fields.

For all these options, you may then select **Equal**, **Empty**, or **Not empty** from the second drop-down menu. If you select **Equal**, a field will appear and you will have the option to enter a text or variable item.

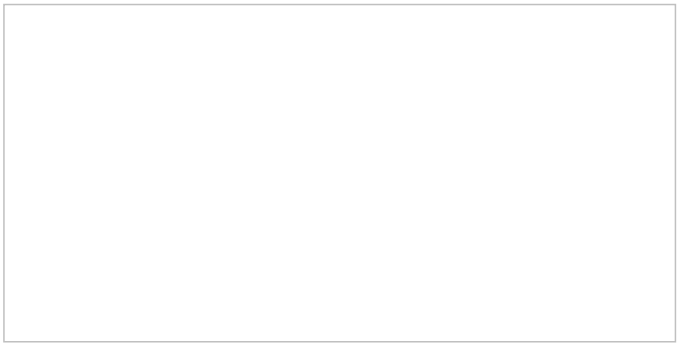


Bright Pattern Search Object company settings

Contact

To search by contact, select **Email address**, **External ID**, **ID**, **Messenger ID**, **Phone**, or a **custom contact field** from the first drop-down menu. Note: Custom contact fields will not appear in the *Search* menu unless the checkbox **Searchable in scenarios and workflows** has been selected. For more information, see Custom Fields.

For all these options, you may then select **Equal**, **Empty**, or **Not empty** from the second drop-down menu. If you select **Equal**, a field will appear and you will have the option to enter a text or variable item.

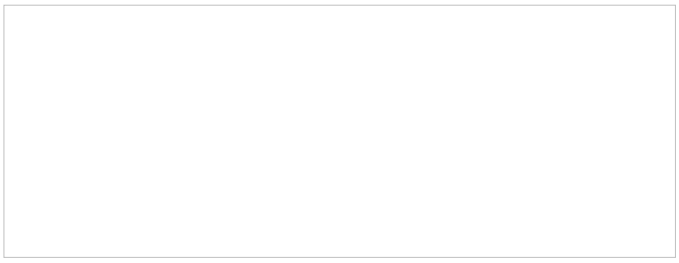


Bright Pattern Search Object contact settings

Activity history

To search by activity history, select **Case ID**, **Contact ID**, **Global Interaction ID**, **ID**, or a **custom activity history field** from the first drop-down menu. Note: Custom activity history fields will not appear in the *Search* menu unless the checkbox **Searchable in scenarios and workflows** has been selected. For more information, see Custom Fields.

For all these options, you may then select **Equal**, **Empty**, or **Not empty** from the second drop-down menu. If you select **Equal**, a field will appear and you will have the option to enter a text or variable item.



Bright Pattern Search Object activity history settings

Return fields

Select this option to add return field to your search. When you click the**add** option, you may enter a text or variable item in the field. Note you may add multiple return fields.

Recordset name

This is the text associated with the name of the recordset.

Bright Pattern Update Object

The *Bright Pattern Update Object* block updates an existing object in the database. The allowed object types are Case, Contact, and Company.

Settings

Title text

The name of the instance of the block.

Object type

The *Object type* drop-down menu allows you to choose the type of object you want to update in the internal database. You may choose **Case**, **Contact**, or **Company**. Depending on what object type is selected, the settings change as follows:

Case Settings

Record Id

The database record case ID.

Case title

The text associated with the case title.

Category name

The category names that have been configured in the Contact Center Administrator application, section *Case & Contact Management* > *Case Categories* > *Add new*, will appear here in the drop-down menu. These category names are the same set as in the Bright Pattern Create Object block.

Priority

Updates an existing case object with the priority level set in this field. This setting is optional.

Pending reason

Updates an existing case object with the pending reason set in this field. This setting is optional.

New status

Updates an existing case object with the status set in this field. This setting is optional.

Reporter contact ID

Updates an existing case object with the reporter contact ID set in this field. This setting is optional.

Custom Case Fields

Custom Case Fields that have been configured in Case & Contact Management > Custom Fields > Case will appear here; you may enter text items in these fields.

Contact Settings

Record ID

The database record contact ID.

First name

The text associated with the first name of the contact.

Last name

The text associated with the last name of the contact.

Title

The text associated with the title of the contact.

Position

The text associated with the position of the contact.

Summary

The text associated with the summary of the contact.

Segment

The text associated with the segment of the contact.

Date of Birth

The text associated with the age and date of birth of the contact.

Company ID

The ID of the company record.

Emails

When you click the **add** option, you may select the type of email from the drop-down menu (**Primary**, **Business**, or **Private**), then enter the email address. It is possible to add more than one type of email address; the order the emails are listed in is insignificant. Note: Gaps are not allowed (i.e. you must create the email with the index [0] if you want to populate the email with the index [1]).

Phones

When you click the **add** option, you may select the type of phone from the drop-down menu (**Business**, **Home**, **Mobile**, or **Fax**), then enter the phone number. It is possible to add more than one type of phone number; the order the numbers are listed in is insignificant. Note: Gaps are not allowed (i.e. you must create phone numbers with the index [0] if you want to populate the email with the index [1]).

Note the following for existing phone numbers:

- Existing phones in the contact are searched by phone number.
- If an existing phone is found, its type is updated.
- If an existing phone is not found, it is added to the list of phones in the contact list.
- There is no way to delete a phone number (or change a phone number) via the *Bright Pattern Update Object* block and must be done via Agent Desktop; deleting phone numbers is deemed too complex for this iteration of scenario block design.

Custom Contact Fields

Custom Contact Fields that have been configured in Case & Contact Management > Custom Fields > Contact will appear here; you may enter text items in these fields.

Company Settings

Record ID

The database record company ID.

Company name

The text associated with the company name.

Web URL

The text associated with the web URL of the company.

Revenue

The text associated with the revenue of the company.

Employees

The text associated with the number of employees in the company. The allowed range is from 0 to 2,147,483,647.

Custom Company Fields

Custom Company Fields that have been configured in Case & Contact Management > Custom Fields > Company will appear here; you may enter text items in these fields.

Conditional Exits

Failed

The *Failed* conditional exit is executed if the update operation failed, due to invalid parameters, timeout, or network connectivity to external CRM (e.g., Salesforce).

No Data

The *No Data* conditional exit is executed if no data matching the specified update criteria is found.

Comment

The *Comment* workflow block allows you to enter internal comments related to this workflow. Comments have no effect on run-time operations.

Workflow Comment block

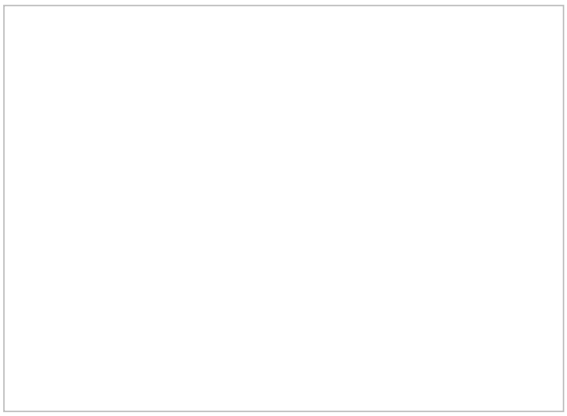
Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Comment

Comment is the field where you enter free-form text comments.



Comment block settings

DB Execute

The *DB Execute* workflow block provides a way for a workflow to execute SQL statements on a specified database. This block can be used to share data between workflows.

DB Execute workflow block

Conditional Exits

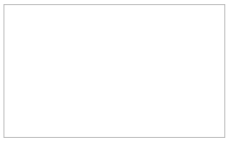
The DB Execute block may take one of the following conditional exits: Failed or No Data.

No Data

The SELECT statement successfully executed but did not return any records.

Failed

An error occurred during SQL statement execution. No error details are provided.



DB Execute Block conditional exits

Settings

Title text

Title Text is the name of the instance of the block. Enter a name in the text field and the new name of the block will appear in the flowchart.

DB Connection

DB connection represents the desired database connection. See the other settings for details.

Name

This is the name of the database connection. This name is shown in the DB Execute block connection selection menu. If no options are shown, click **Manage DB connections** to add new DB connections.

JDBC driver and connection string

Specify the JDBC driver and connection string that will be used to access this database. Note that templates are provided for some widely used DBMS systems.

Database username and password

Specify database access credentials.

The database connection selector allows the following operations:

- Add a new connection (**Add New** button).
- Edit and save the selected connection (**Save** button).
- Delete the selected connection (**Delete** button).
- Select the connection to be used in the DB Execute block (**Select** button).
- Clear the connection selected for the DB Execute block (**Select None** button).
- Close the window without changing the DB Execute block connection (**Close** button).

SQL Statement

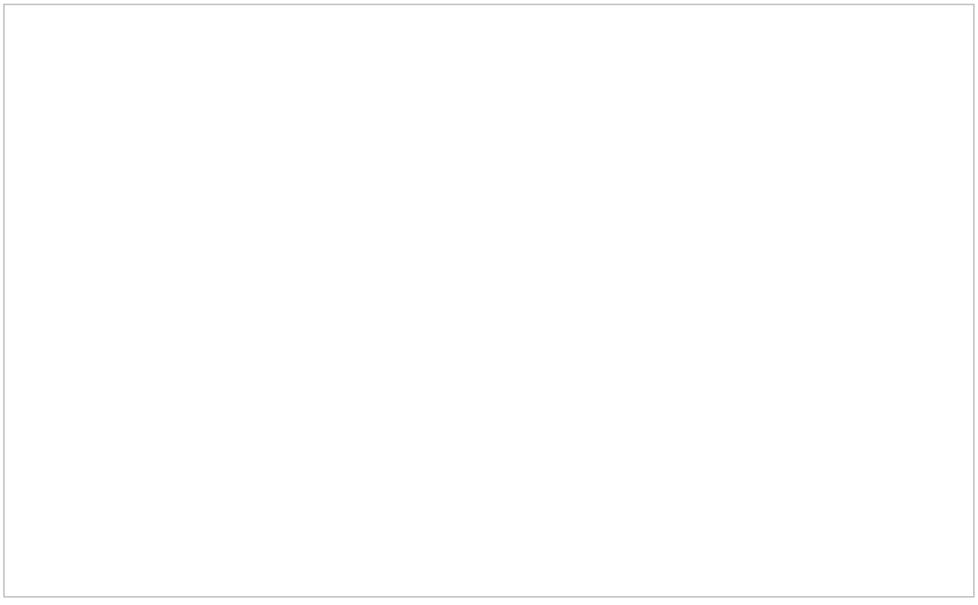
This is the SQL statement to be executed. SQL statements may use workflow variables. For example:*SELECT id, name FROM customers WHERE phone='$(item.from)'*

Recordset name

For SELECT statements, the name of the retrieved recordset should be specified. This allows workflows to have more than one recordset and to choose the recordset to iterate on.

- The columns of the first retrieved record (if any) are stored in the workflow variables*<recordset_name>.<column_name> (e.g., RS.id)*.
- The number of returned records is stored in variable*<recordset_name>.__count__ (e.g., RS.__count__)*. Note the double underscores in front and after count; they are used to reduce the chance of confusing the name of this variable with a column name in a recordset.
- To iterate through the recordset, use the Get Next Record block.
- The number of records in the retrieved recordset is limited to 25.

The database connection is selected from a list of connections. Click the**Manage DB connections** button. The pop-up window will display all database connections defined in this workflow. For each connection, the following data should be defined:



Scenario Builder DB Execute scenario block settings

EMail

EMail workflow block

The *Email* workflow block sends an email, with an attachment if so configured.

For SMTP server configuration, see section Email Settings of the *Contact Center Administrator Guide*.

Settings

EMail settings

Title text

The name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

From / Display name

The display name of the email sender. A workflow variable can be specified as the value using the *$(varname)* format.

From / Address

The email address of the email sender. A workflow variable can be specified as the value using the *$(varname)* format.

To / Address(es)

The email addresses of the intended recipients. If sending to multiple email addresses, separate email addresses using a comma or a semicolon. Workflow variables can be specified as values using the *$(varname)* format.

Template language

The ISO language code associated with the selected language (e.g., "en" for English, "ja" for Japanese, etc.).

Template

The language template to be used for the email (e.g., "English", "Japanese", etc.). The email's text will be displayed in the selected language.

Message / Subject

The subject line of the email. Can contain workflow variables in the *$(varname)* format.

Message / Format

The format for email: HTML (default) or .TXT (plain text).

Message / Body

The text to be sent as a message body. It can contain workflow variables in the *$(varname)* format. Bright Pattern does not impose any limits on the size of the email.

Insert $()

This button allows you to insert a variable in the *$(varname)* format into the body of the email. Choose from the following:

- $(user.firstName)
- $(user.lastName)
- $(user.email)
- $(from.name)
- $(from.emailAddress)
- $(case.number)
- $(app.emailAddress)

Survey link

This button allows you to select a configured survey form and insert a link to it into the body of the email. Note that if your contact center has no existing survey forms, there will be no survey link to select.

To learn how to create a survey, see How to Solicit Post-Transactional Surveys via Email.

Remove <Language> Template

Clicking this button deletes any text present in the message body.

Test <Language> Template

Clicking this button sends a test email, with the specified message, to a specified email recipient.

Conditional Exits

WB-EMail-CE-52.PNG

The EMail block may take the *Mail not sent* conditional exit, in which the message cannot be sent (if, for example, the SMTP server is down).

Exception Handler

The *Exception Handler* workflow block provides an alternative branch the workflow can execute if an exception, a block error, or a disconnection occurs. This allows the workflow to continue executing instead of terminating as it normally would under such circumstances without the Exception Handler block. Use this block in any part of a workflow in which you expect exceptions, block errors, or caller disconnects in order to ensure continued processing.

Conditional Exits

The Exception Handler block may take one of the following conditional exits: Try or Catch.

Try In the Try conditional exit, enter the sequence of blocks that you predict might generate an exception, block error, or disconnect.

Catch In the Catch conditional exit, enter the sequence of blocks that you want the workflow to execute if an exception, block error, or disconnect occurs during the Try conditional exit.

Workflow Builder Exception

Handler Try/Catch conditional exits

The Exception Handler block initially executes the Try conditional exit.

- If an exception, block error, or caller disconnect occurs while executing any block in the Try conditional exit, the workflow executes the Catch conditional exit. After executing the Catch conditional exit, the workflow executes the next block in the flowchart.
- If no exception, block error, or disconnect occurs, the workflow does not execute the Catch conditional exit. Instead, the workflow processes the next block in the flowchart.

After execution, the Exception Handler block stores one of the following values in the Exception interaction property:

- **Disconnect** - The caller disconnected.
- **Error** - A block error or exception occurred.
- **No** - No exception occurred (normal processing).

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Exception Handler settings

Exit

The *Exit* workflow block disconnects the currently active interaction and exits the workflow.

Exit workflow block

The following image shows the Exit block at the end of a workflow.

Exit block

Fetch URL

The *Fetch URL* workflow block fetches web content, including JSON-formatted data, from a specified URL, using a specified method and parses it into workflow variables. Both HTTP and HTTPS protocols are supported, as well as basic authentication.

Fetch URL workflow block

Conditional Exits

The Fetch URL block may take one of two conditional exits: *Failed* or *No Data*.

Failed

The Failed conditional exit is taken if an error occurred during the HTTP method execution. SeeHTTP Response Codes below for details.

No Data

The No Data conditional exit is executed if no data is returned in the body of the HTTP response.

Fetch URL

conditional exits

Settings

Title text

Title text is the name of the instance of the block, displayed in the flowchart.

Request type

Request type is the type of method to be used to fetch web content from the specified URL. Request types are defined in the pull-down menu.

Select from the following request types, or write in another method manually:

- GET (default)
- POST
- PUT
- PATCH
- DELETE

URL to fetch

URL to fetch is the HTTP/HTTPS URL of the web content to get. Query string parameters are specified separately.

Extra headers

Extra headers are the HTTP headers to add to the request (e.g. for authentication purposes). Functions can be used by inserting them as a value. Click **add** to define a header, type in the name, and type in the value.

For example, a payment gateway may have a RESTful interface that requires authentication via "Authorization" header and SHA-256 hash of time, username, and password. To enable authentication, you would provide a request header with the *name* "Authorization" and the value *"accessid==hmac("SHA-256", key, message)"*. Functions are described in the *Workflow Builder Reference Guide*, section Built-in Functions.

URL parameters

These are the URL parameters to be URL encoded and appended into URL. Workflow variables can be used by inserting them as *$(varname)*. Click **add** to define URL parameters, type in the name, and type in the value.

Content Type

Content type is the type of data to be submitted in the request body. The setting is displayed only if request type POST is selected.

Select *application/json* for a JSON data structure, or select *application/x-www-form-urlencoded* for a URI-encoded data in the body of the message.

Body

Body is the data to be submitted in the request body. The setting is displayed only if request type POST is selected.

Body is the data to be submitted. The format of the data must correspond to the *Content Type* above. Workflow variable substitutions are allowed.

Username

Username is the request authentication username. Variable substitutions are allowed.

Password

Password is the request authentication password. Variable substitutions are allowed.

Initial path in the result JSON

If the response body contains JSON, this setting can be used to save into workflow variables a specific part of the data. Example: *myobject.node.list[4]*. The default is "none"; the path starts from the root of the returned JSON.

Scenario variable prefix for JSON data

This string will be used as the name of the variable to receive parsed JSON data. Note that if the initial path above points to an array, depending on the value of the *GetNext* option below, this variable would either contain the array or its first (and subsequent) elements.

Use GetNext block to loop through data

This box is selected if the JSON response data (at the initial path) is an array. The workflow variable will be set to the first element of the array and *GetNext* block could be used to iterate over the array elements, setting workflow variable to the next element.

Note: If you enable this option, the behavior of the Fetch URL block will be the same as it was for Bright Pattern Contact Center version 3.13 and earlier.

Fetch URL settings

Response Data Handling

In all cases, the response data is limited to 50 KB.

If the response data is not JSON-encoded, it could be accessed as string via the*$(integrationResultBody)* workflow variable described below.

If the response data is JSON, the following will happen:

1. It will be parsed.
2. If the "GetNext" option is enabled and the item the initial path is pointing to is a regular, non-associative array:
    1. The workflow variable will be set to the first item of the array.
    2. The GetNext block could be used to initialize the variable to the next and subsequent items.
3. Otherwise, the workflow variable will be set to the item to which the initial path is pointing.

Possible syntax for the initial path setting and the access to the data saved in the workflow variable is as follows:

- *item.subitem*
- *item[index]*
- *item[attr1=value].attr2*

- combinations (e.g., *item.subitem.array[index].value.array[attr=x].value*)

HTTP Response Codes

The status code and the body of the received HTTP response will be stored in local variables*$(integrationResultCode)* and *$(integrationResultBody)*, respectively. For troubleshooting purposes, you may use the EMail or Internal Message block to obtain content of responses indicating a failed attempt. For more information, see the description of variable *$(integrationResultBody)*.

For backward compatibility reasons, the code and the body of the received HTTP response are also stored in local variables *$(fetchURLResultCode)* and *$(fetchURLResultBody)*.

The possible values of the *$(fetchURLResultCode)* variable are as follows:

- **0**: 200 OK response (i.e., a successful HTTP request response)
- **-1**: 200 OK response, but unable to parse the body into recordset
- **-2**: 200 OK response, but body length exceeds 50 KB
- **-3**: Unable to connect to HTTP server or other connection errors
- **-4:** Incorrect JSON syntax in the request body when using PUT or POST requests
- **Other**: Actual non-200 response code

HTTP Redirect Response Handling

The Fetch URL block handles 3xx Hypertext Transfer Protocol (HTTP) response codes in the following way.

When the following codes are received, the block retries the request to the provided redirect URL using the*GET* request method:

- 301 Moved Permanently
- 302 Found
- 303 See Other (since HTTP/1.1)

When the following codes are received, the block retries the request to the provided redirect URL using the originally specified request method:

- 307 Temporary Redirect (since HTTP/1.1)
- 308 Permanent Redirect (RFC 7538)

When the following status codes are received, the conditional exit Failed will be selected:

- 300 Multiple Choices
- 304 Not Modified (RFC 7232)
- 305 Use Proxy (since HTTP/1.1)
- 306 Switch Proxy

Get Next Record

The *Get Next Record* workflow block provides a way for a workflow to retrieve the next or previous record from a recordset created by a previously executed blocks DB Execute, Fetch URL, RightNow Search, Salesforce.com Search, and Zendesk Search.

Conditional Exits

The Get Next Record block may take the *No more items* conditional exit if no more items can be retrieved from the specified recordset.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

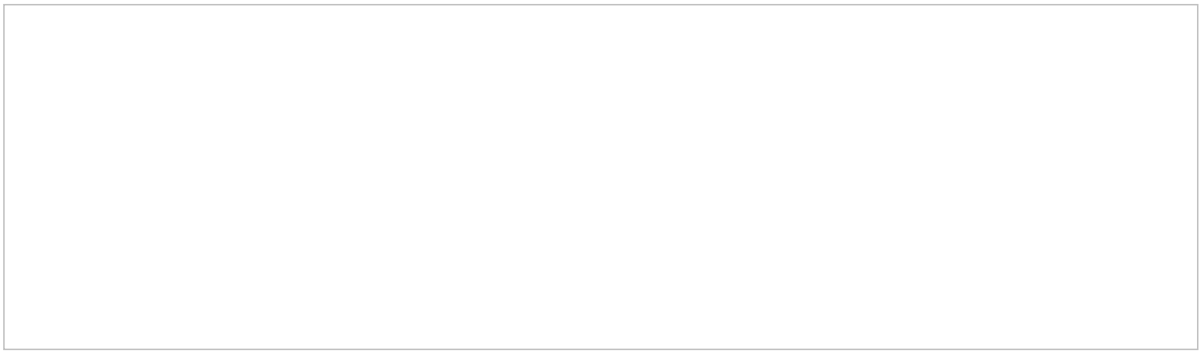
Direction

Choose whether the next or the previous record should be retrieved.

Recordset name

Recordset name is the recordset from which the record should be retrieved. The name is selected from the list of available recordsets, which is populated from all DB Execute, Fetch URL, RightNow Search, Salesforce.com Search, and Zendesk Search blocks of the workflow.

The columns of the retrieved records (if any) are stored in the workflow variables *RS.[name]* (e.g., *RS.id*).



Get Next Record block settings

Get User Configuration

The *Get User Configuration* workflow block obtains one or more of the properties of a user, such as username, phone number, email address, or unique identifier, and saves them in workflow variables for future use.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Find user by

Find user by specifies the type of the property that will be used to find the user. You can find a user by using the following properties:

- **Login ID** – The username
- **User ID** – The global unique identifier of this user in system configuration; global identifiers can be used, for example, to associate contacts/cases/interactions in third-party systems with agents who handled them
- **Email address** – The user's email address
- **Phone (hard or soft)** – The user's assigned phone extension or default hardphone number

Value

Value specifies the value of the property that will be used to find the user.

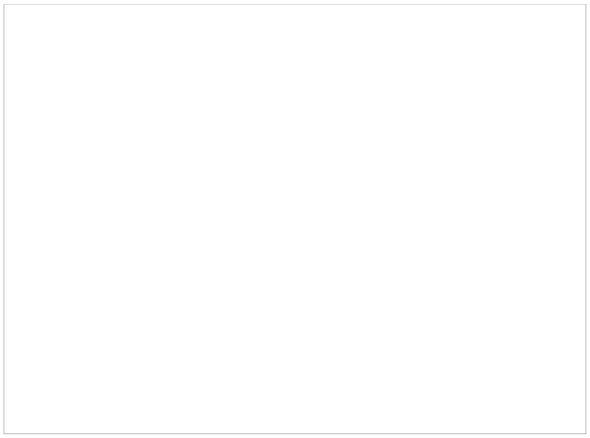
User properties to return

Click **add** to select the type of property that you want to use in the workflow and the name of the variable where its value will be stored. The following property descriptions refer to their names as they appear in the Contact Center Administrator application.

- **First Name** – The user's first name
- **Last Name** – The user's last name
- **Login ID** – The user's username
- **Email address** – The user's email address
- **Softphone number** – The user's phone extension
- **Hardphone number** – The user's default hardphone number
- **Team ID** – The global unique identifier of the user's team in system configuration
- **Team name** – The name of the user's team
- **Voice Mail Greeting URL** – The link to user's currently selected voicemail greeting
- **Skill** – The name of user's skill from the specified group; this parameter can be used for group-based routing
- **User ID** – The global unique identifier of this user in system configuration; user ID can be used, for example, to associate contacts/cases/interactions in third-party systems with agents who handled them in order to facilitate personal routing.

Workflow variables can be specified as values using the *$(varname)* format.

For more information, see *Contact Center Administrator Guide* sections Users, Teams, and Skill Levels.



Get User Configuration settings

Goto

The *Goto* workflow block redirects the processing flow of the workflow to a specified destination in the flowchart.

Goto workflow block

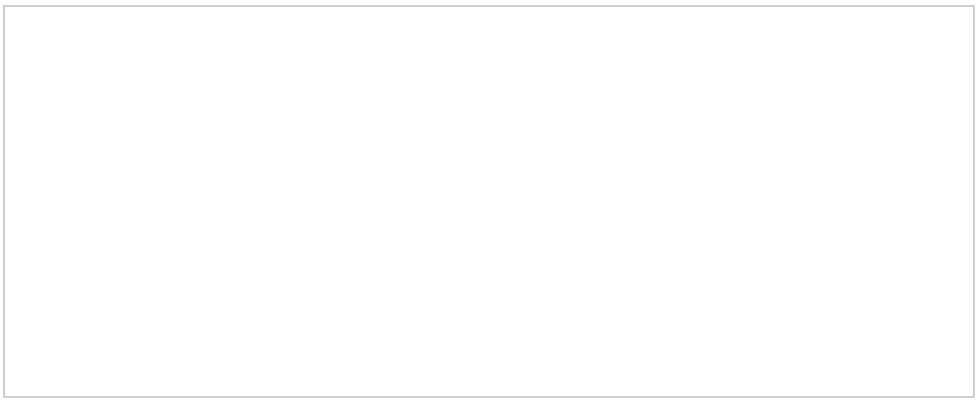
To redirect the flow using this block, follow these steps:

1. Add a Goto block to the desired location in the flowchart. The Workflow Builder will mark the block in red until you define a destination for it.
2. Select the Goto block in the flowchart. The Edit Pane will display a copy of the flowchart.
3. In the Edit Pane, click the desired Goto destination (i.e., the building block to which you want to redirect the flow using this Goto block).

The flowchart displays the new name of the Goto block, which indicates the location in the flowchart to which the block redirects the processing flow. The format of the name is Goto *"[destination block title]"*.

The Workflow Builder will highlight the Goto block in red if you remove its destination block during editing.

In the following example, a Goto block is used to redirect the workflow back to the beginning.



Goto workflow block settings

If

The *If* workflow block allows branching of a workflow based on verification of some specified conditions. Multiple conditional exits (branches) can be configured in the same block.

If workflow block

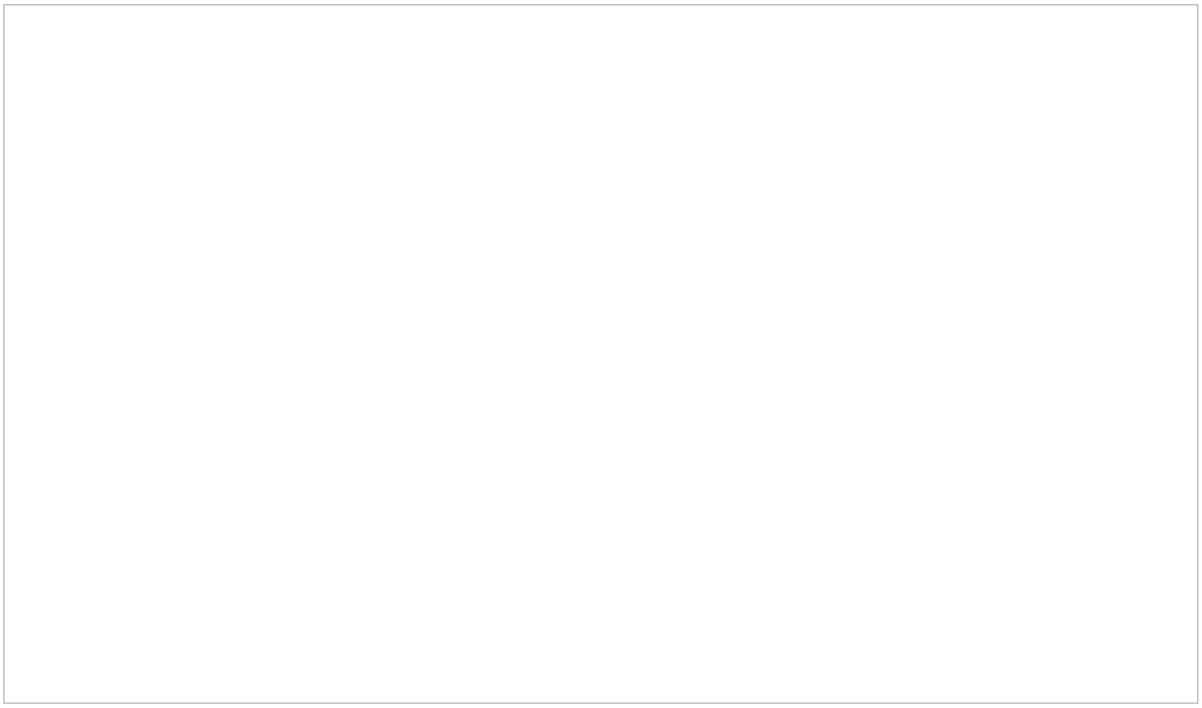
Branches and Conditions

A branch can include one or more logical expressions (conditions), where each condition verifies one of the following:

- Dialed number
- Caller's number
- Current time
- Day
- Current date
- Workflow variable (number)
- Workflow variable (string)

Use the **Add branch** button to add a branch corresponding to the desired conditional exit. Provide a label that will identify the corresponding conditional exit in the flowchart.

Click the **add condition** link to define a logical expression for verification of one of the above parameters.



If workflow block settings

Block of Conditions

A block of conditions in a branch can be joined by either the AND (default) or OR operator.

- AND is used if all specified conditions in a branch must be met in order for the workflow to take the given branch exit.
- OR is used when it is sufficient for one of the specified conditions to be met in order for the workflow to take the given branch exit.

If necessary, add more branches as described. (Note that there is a limit of 20 branches per If block.)

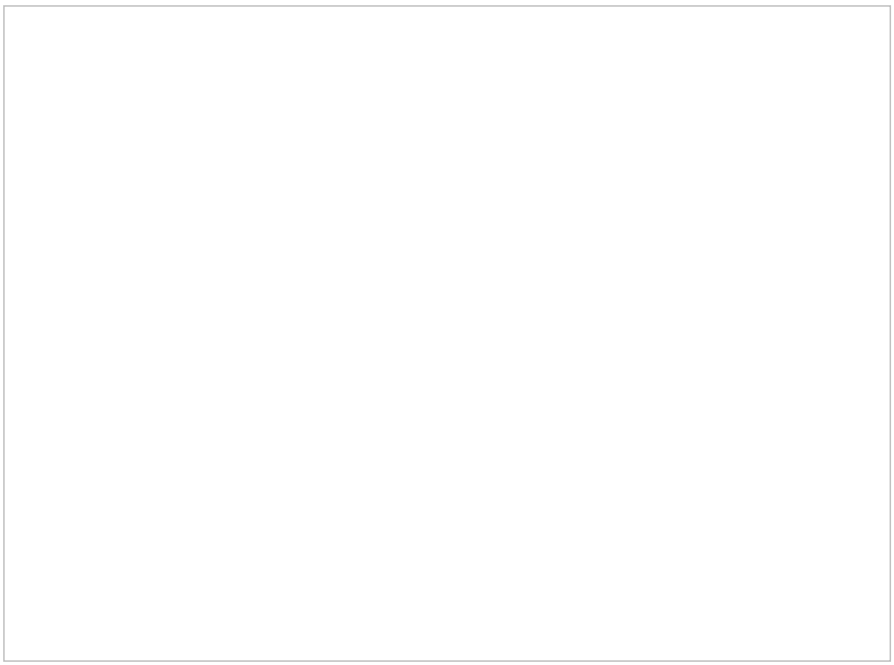
The branches are tried in the order in which they are defined in the block. If none of the branches leads to a positive verification, the block that directly follows the given If block in the flowchart is executed.

Typical Uses

The following are examples of some typical uses of the If block.

Caller's Number

The *Caller's number* condition may be used to filter incoming calls in a number of ways. When configuring this branch, the caller's number either *is* or *is not* equal to the following values: *In range*, *equal*, *starts with*, *contains*, *ends with*, <, and >. Once configured, you may add additional workflow blocks to a given branch, such as Assign Case to Agent.



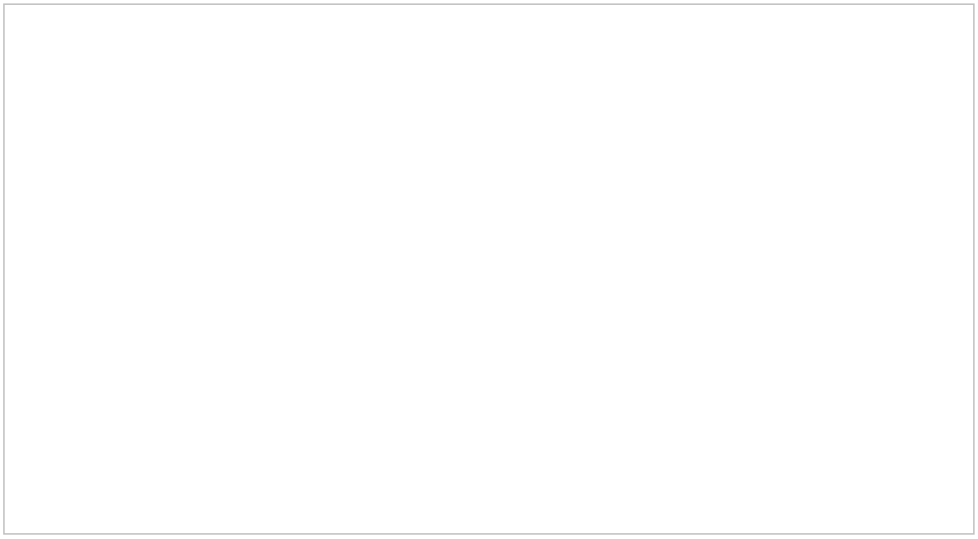
The Caller's number condition may be used to filter incoming calls by area code

Current Time

The *Current Time* condition can be used along with an Assign Case to Agent block to set queue limits based on the time of day. For example, the Current Time condition may be set in the If block to find an agent and set queue limits: "If 10-5 (where 10-5 refers to 10:00 am to 5:00 pm), find an agent with queue limit of 20" and/or "If 5-7 (where 5-7 refers to 5:00 pm to 7:00 pm), find an agent with queue limit of 10."

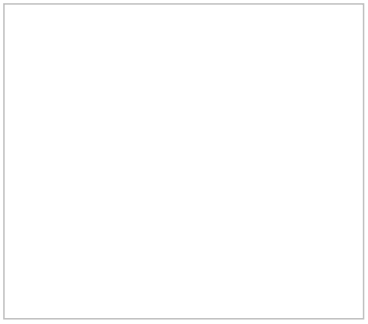
Workflow Variable (String)

Using the *Workflow variable (string)* and $(disposition) variable, you can send different emails to customers based on what disposition their interaction received (i.e., send one follow-up email if the disposition is "Product sold" and a different follow-up email if the disposition is "Service provided").



The workflow variable (string) condition configured with the $(disposition) variable

To do this, add a branch, then add a condition with a workflow variable (string), the $(disposition) variable, and the name of the disposition (e.g., "Need more information"). For every branch configured with a different disposition, you will add and configure an EMail block.



Different dispositions can receive different emails

Internal Message

The *Internal Message* workflow block is used to send an internal chat message to a specified user.

Conditional Exits

The Internal Message block may take the *Failed* conditional exit if the attempt to send the message has failed.

Settings

Title text

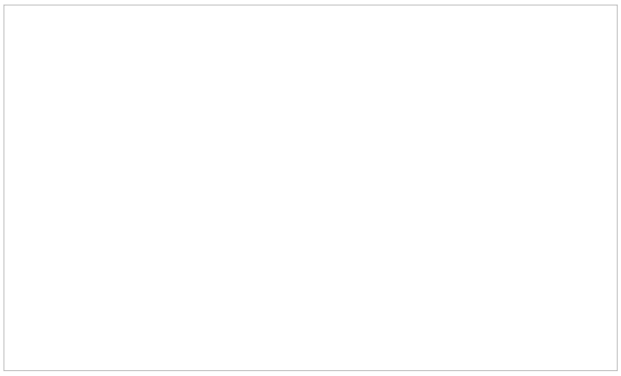
Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Username

This is the username of the message recipient.

Message

The message is the text of the message to be sent. Variables in the *$(varname)* format can be used in the message text.



Internal Message settings

Log

The *Log* workflow block adds a message to the Workflow Engine log file. This block is intended for debugging and testing purposes only, and it may be removed in production versions of workflows.

Log workflow block

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

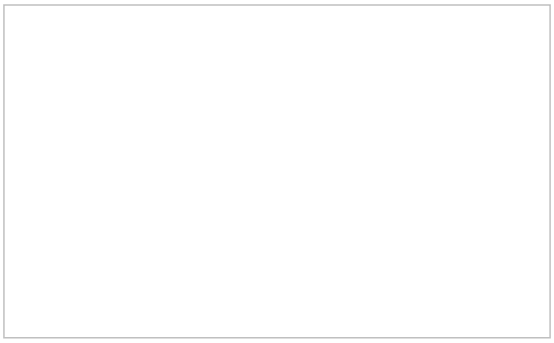
Text message

Text message is the free-form content of the message. You can use workflow variables in the text (e.g.,*This is a test log message for call from $(item.from) to $(item.to)*).

Log Level

Log Level specifies the Workflow Engine log level where this message will appear. You may choose from the following options:

- DEBUG (default)
- INFO
- WARNING
- ERROR
- PANIC



Log workflow settings

Microsoft Dynamics Create Object

The Microsoft Dynamics Create Object workflow block creates a specified object in the Dynamics 365 database. You may find it useful to pair this block with the Microsoft Dynamics Search Object workflow block; if you search the Dynamics 365 CRM database for an object and do not find it, you can opt to create that object.

Properties

Microsoft Dynamics Create Object properties

Title text

The name of the instance of the block (any name).

Object type

The type of object you want to create in the Dynamics 365 database. You may enter a custom object type or select from the drop-down menu one of the following types:

- Account
- Case
- Contact
- Contract
- Invoice
- Lead
- Opportunity
- Order
- Quote

Variable name for object ID

The name of the variable that will be used as the identifier for the Dynamics 365 object to be created. The variable name of the object ID will be set only if the block succeeds.

Set fields

This setting is reserved.

Raw JSON

Where object properties are specified in JSON format.

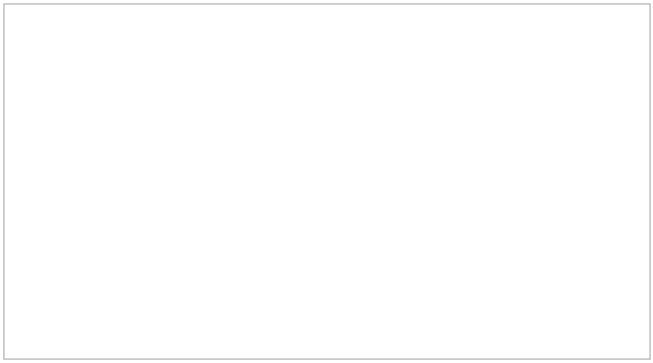
Conditional Exits

If the create operation has failed, the block will take the *Failed* exit.

Microsoft Dynamics Delete Object

The Microsoft Dynamics Delete Object workflow block is used for deleting a specified object in the Dynamics 365 database. You may find it useful to pair this block with the Microsoft Dynamics Search Object workflow block; you can search the Dynamics 365 CRM database for an object and then use the Delete Object block to remove it.

Properties



Microsoft Dynamics Delete Object properties

Title text

The name of the instance of the block (any name).

Object type

The type of object you want to delete in the Dynamics 365 database. You may enter a custom object type or select from the drop-down menu one of the following types:

- Account
- Case
- Contact
- Contract
- Invoice
- Lead
- Opportunity
- Order

- Quote

Object ID

The object ID is the name of the variable that will be used to identify the Dynamics 365 object to be deleted. The variable name of the object ID will be set only if the block succeeds.

Conditional Exits

If the create operation has failed, the block will take the*Failed* or *No Data* exit.

Failed

This exit is taken if the search operation failed due to invalid parameters, timeout, or network connectivity to the Dynamics 365 CRM.

No Data

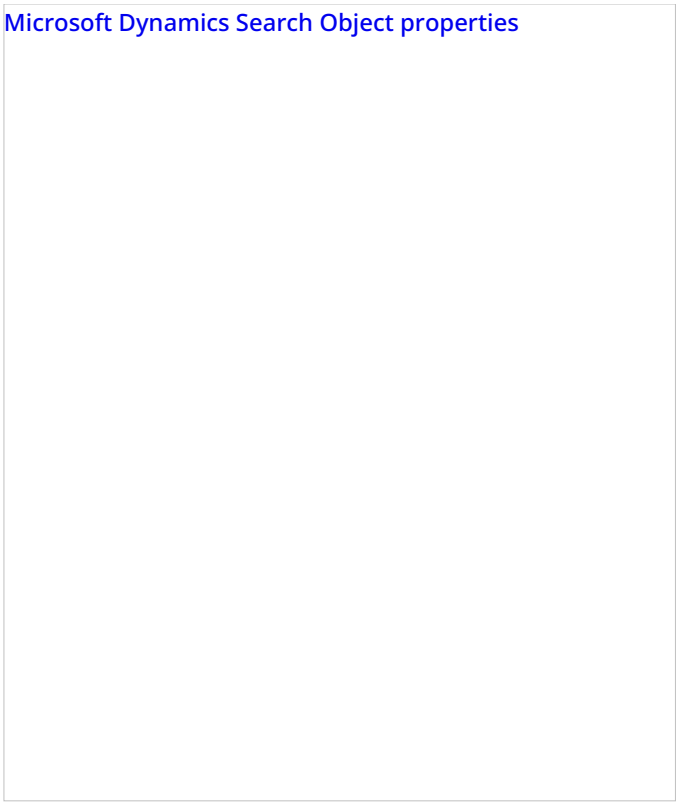
This exit is taken if no data matching the specified search criteria is found.

Microsoft Dynamics Search Object

The Microsoft Dynamics Search Object block finds an existing object in the Dynamics 365 database. When using this block, we assume that we have some information about a contact, such as case state, account number, or name, and we use that information for data lookup. You can use this block to search for a variety of data object types.

Properties

Microsoft Dynamics Search Object properties



Title Text

The name of the instance of the block (any name).

Object type

The type of object you want to search for in the Dynamics 365 database. Values may be manually entered in this field or selected from the drop-down menu. Note that the value in the *Object type* field is the Dynamics **Entity Set Name**. For more information, see Query Data using the Web API.

Select from one of the following types:

- Account
- Case
- Contact
- Contract
- Invoice
- Lead
- Opportunity
- Order
- Quote

Query

This field allows free-form entry of of text in Dynamics 365 search URL syntax.

Recordset name

A recordset is a set of records (presented either as a table or a query from a table) in JSON format, which holds the results of the search that this block is doing.

The recordset name is the name of the recordset in JSON format. When entering the recordset name (e.g., "MScase"), you are declaring your recordset in a field name, and the value entered in this field can then be used in a number of subsequent blocks to dictate further actions (e.g., passing the value to a Microsoft Dynamics Create Object block if information about a contact is not found).

Conditional Exits

Failed

This exit is taken if the search operation failed due to invalid parameters, timeout, or network connectivity to the Dynamics 365 CRM.

No Data

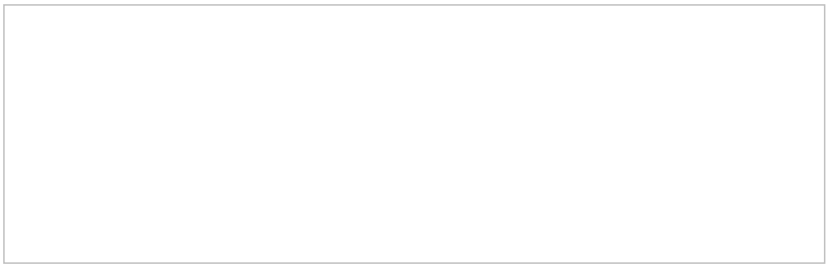
This exit is taken if no data matching the specified search criteria is found.

Microsoft Dynamics Select Account

Your contact center configuration may contain multiple Microsoft Dynamics 365 integration accounts for access to different Dynamics 365 apps. Use the Microsoft Dynamics Select Account block to specify the integration account that will be used by subsequent Microsoft Dynamics blocks in the given workflow.

If this block is not used, all Microsoft Dynamics blocks in the given workflow will use access data from the integration account marked as *Default account*. For more information, see the *Contact Center Administrator Guide*, section Microsoft Dynamics 365 Integration.

Properties



Microsoft Dynamics Select Account workflow block properties

Title text

The name of the instance of the block.

Enter a name in the text field and click the**Update** button at the bottom of the Edit pane. The new name of the block appears in the flowchart.

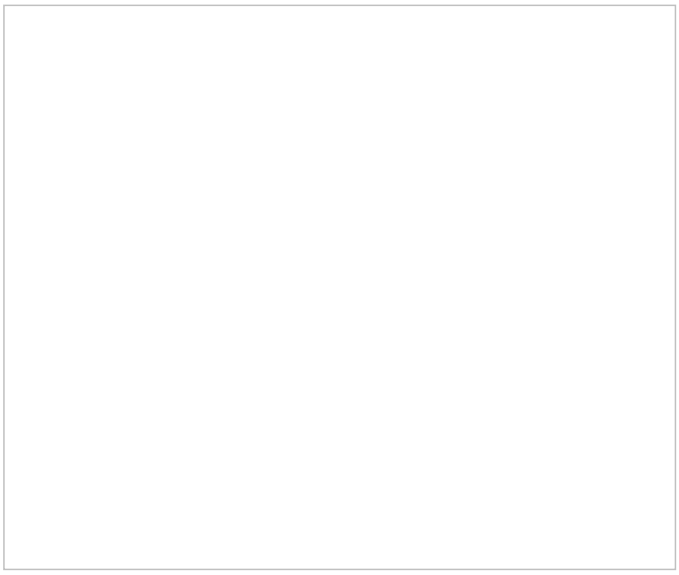
Account

The name of the Microsoft Dynamics Select Account integration account that will be used for access to Dynamics 365 data by subsequent Microsoft Dynamics blocks in the given workflow.

Microsoft Dynamics Update Object

The Microsoft Dynamics Update Object workflow block is used for making changes to a specified object in the Dynamics 365 database. You may find it useful to pair this block with the Microsoft Dynamics Search Object block; if you search the Dynamics 365 CRM database for an object and want to edit it, you can opt to update that object.

Properties



Microsoft Dynamics Update Object properties

Title text

The name of the instance of the block (any name).

Object type

The type of object you want to create in the Dynamics 365 database. You may enter a custom object type or select from the drop-down menu one of the following types:

- Account
- Case
- Contact
- Contract
- Invoice
- Lead
- Opportunity

- Order
- Quote

Object identifier

The object ID is the name of the variable that will be used to identify the Dynamics 365 object to be updated. The variable name of the object ID will be set only if the block succeeds.

Set fields

This setting is reserved.

Raw JSON

Where object properties are specified in JSON format.

Conditional Exits

If the create operation has failed, the block will take the*Failed* or *No Data* exit.

Failed

This exit is taken if the search operation failed due to invalid parameters, timeout, or network connectivity to the Dynamics 365 CRM.

No Data

This exit is taken if no data matching the specified search criteria is found.

RightNow Create Object

The *RightNow Create Object* workflow block creates a specified object in the RightNow database.

Conditional Exits

The RightNow Create Object block may take the *Failed* conditional exit if the create operation has failed.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Object type

Object type is the type of RightNow object to be created. You can either select one of the standard objects from the drop-down menu, or you can enter the name of the desired custom object type.

Variable name of object ID

This is the name of the variable that will be used as the identifier for the RightNow object to be created. The variable name of object ID will be set only if the block succeeds.

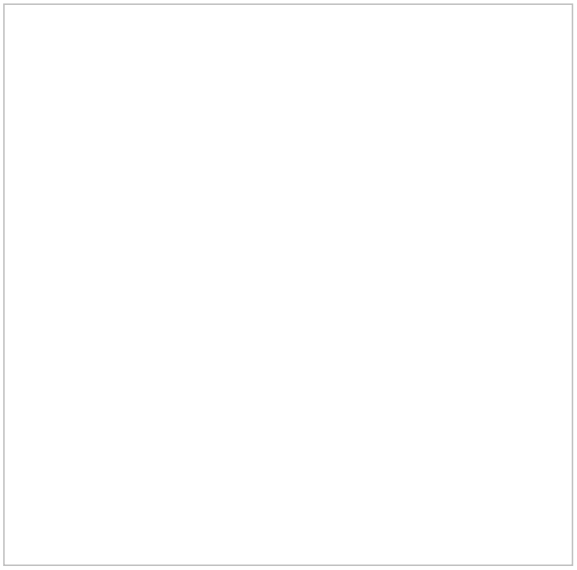
Set fields

This setting is reserved.

Raw JSON

The *Raw JSON* parameter is where object properties are specified in JSON format.

The code and the body of the received HTTP response will be stored in local variables*$(integrationResultCode)* and *$(integrationResultBody)*, respectively. For troubleshooting purposes, use the EMail or Internal Message block to obtain content of responses indicating a failed attempt to create an object. For more information, see the description of variable $(integrationResultBody).



RightNow Create Object workflow settings

RightNow Search

The *RightNow Search* workflow block executes the specified RightNow record selection statement written in the RightNow Object Query Language (ROQL).

The columns of the first record of retrieved recordset are stored in variables *<recordset_name>.<column_name>.* For that statement, the results will be stored in variables *Recorset.id* and *Recordset.name*.

The number of returned records is stored in variable*<recordset_name>.__count__* (e.g., *RS.__count__*.). Note the double underscores in front and after count; they are used to reduce the chance of confusing the name of this variable with a column name in a recordset.

To iterate through the recordset, use the Get Next Record block.

Conditional Exits

The RightNow Search block may take one of the following conditional exits: *Failed* or *No Data*.

Failed

The Failed conditional exit is executed if the search operation failed.

No data

The No data conditional exit is executed if no data matching the specified search criteria is found.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

ROQL Query

ROQL Query is the record selection statement in the RightNow Object Query Language. It may contain application variables specified as *$(varname)*.

Recordset name

Recordset name is the name of the recordset that will be retrieved via this search operation.

The code and the body of the received HTTP response is stored in local variables*$(integrationResultCode)* and *$(integrationResultBody)* respectively. For troubleshooting purposes, use the EMail or Internal Message block to obtain the content of responses indicating a failed search attempt. For more information, see the description of the variable $(integrationResultBody).

RightNow Search workflow settings

RightNow Select Account

Your contact center configuration may contain multiple RightNow integration accounts for access to different RightNow systems. Use the *RightNow Select Account* block to specify the integration account that will be used by subsequent RightNow blocks in the given workflow. If this block is not used, all RightNow blocks in the given workflow will use access data from the integration account marked as the Default account.

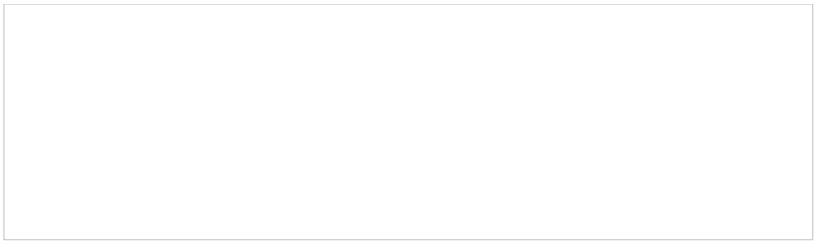
Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Account

Account is the name of the RightNow integration account that will be used for access to RightNow data by subsequent RightNow blocks in the given workflow.



RightNow Select Account workflow settings

RightNow Update Object

The *RightNow Update Object* workflow block updates the properties of the specified RightNow object.

Conditional Exits

The RightNow Update Object block may take one of the following conditional exits: *Failed* or *No data*.

Failed

The Failed conditional exit is executed if the update operation failed.

No data

The No data conditional exit is executed in the specified object is not found.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Object Type

Object type is the type of RightNow object to be created. You can either select one of the standard objects from the drop-down menu or enter the name of the desired custom object type.

Object Identifier

Object Identifier is the identifier of the object to be updated.

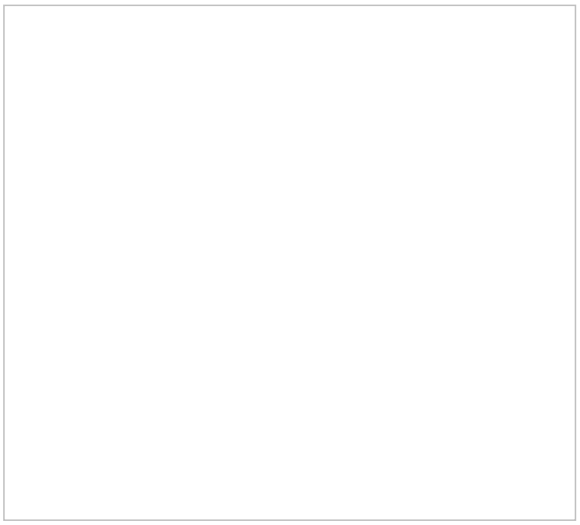
Set fields

This setting is reserved.

Raw JSON

Raw JSON is where the object properties to be updated are specified in JSON format.

The code and the body of the received HTTP response is stored in local variables*$(integrationResultCode)* and *$(integrationResultBody)* respectively. For troubleshooting purposes, use the EMail or Internal Message block to obtain the content of responses indicating a failed attempt to update an object. For more information, see the description of the variable $(integrationResultBody).



RightNow Update Object workflow settings

<Previous | Next>

Salesforce.com Delete

The *Salesforce.com Delete* workflow block is the part of Salesforce.com Integration with Bright Pattern Contact Center. This block deletes the specified Salesforce.com (SFDC) object from the SFDC database.

Conditional Exits

The Salesforce.com Delete block may take one of the following conditional exits:*Failed* or *No Data*.

Failed

The Failed conditional exit is executed if the delete operation failed.

No Data

The No Data conditional exit is executed in the specified object is not found.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

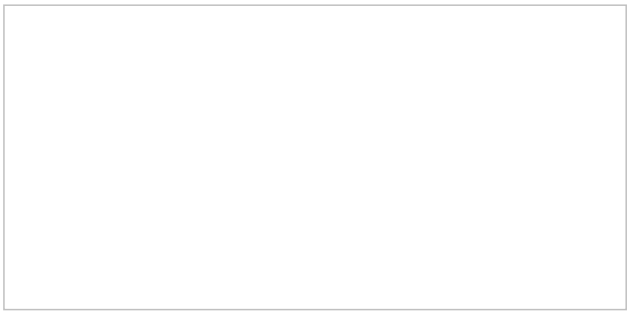
Object type name

Object type name is the type of the SFDC object to be deleted as defined in the SFDC application. It may be specified as an application variable in the *$(varname)* format.

Object ID

Object ID is the identifier of the SFDC object to be deleted as defined in the SFDC application. The Object ID may be specified as an application variable in the *$(varname)* format.

The code and the body of the received HTTP response is stored in local variables*$(integrationResultCode)* and *$(integrationResultBody)*, respectively. For troubleshooting purposes, use the EMail or Internal Message block to obtain the content of responses indicating a failed attempt to delete an object. For more information, see the description of the variable $(integrationResultBody).



Salesforce.com Delete workflow settings

<Previous | Next>

Salesforce.com Insert

The *Salesforce.com Insert* workflow block is the part of Salesforce.com Integration with Bright Pattern Contact Center

This block creates the specified Salesforce.com (SFDC) object in the SFDC database.

Conditional Exits

The Salesforce.com Insert block may take the *Failed* conditional exit if the insert operation fails.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Object type name

This is the type of the SFDC object to be created as defined in the SFDC system. It may be specified as an application variable in the *$(varname)* format.

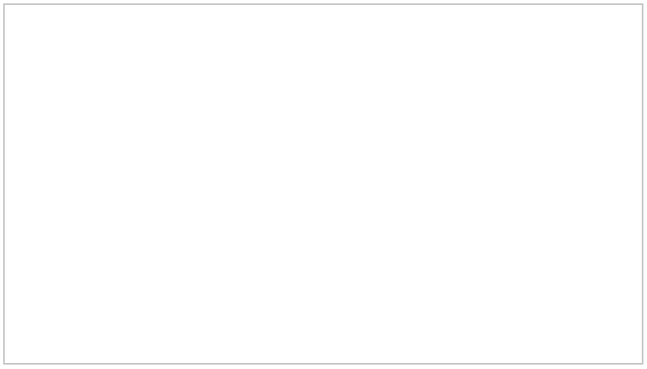
Variable name of object ID

This is the name of the variable that will be used as the identifier for the SFDC object to be created. It will be set only if the block succeeds.

Object fields

Object fields are object properties. Click **add** and specify the property *Name* as defined in the SFDC system. Then specify the desired *Value*. Repeat for the remaining object properties. Field values may be specified as application variables in the *$(varname)* format.

The code and the body of the received HTTP response will be stored in local variables*$(integrationResultCode)* and *$(integrationResultBody)*, respectively. For troubleshooting purposes, use the EMail or Internal Message block to obtain the content of responses indicating a failed attempt to create an object. For more information, see the description of the variable $(integrationResultBody).



Salesforce.com Insert workflow settings

<Previous | Next>

Salesforce.com Search

The *Salesforce.com Search* workflow block is the part of Salesforce.com Integration with Bright Pattern Contact Center.

This block executes the specified Salesforce.com (SFDC) record selection statement written in either the Salesforce Object Query Language (SOQL) or Salesforce Object Search Language (SOSL). Workflow variables can be used in this statement in the *$(varname)* from. Consider this example of an SOQL statement that uses the *$(ANI)* variable: *SELECT id, name FROM Accounts WHERE phone = '$(ANI)'.*

The columns of the first record of retrieved recordset are stored in variables*<recordset_name>.<column_name>*. For that statement, the results will be stored in variables *Recorset.id* and *Recordset.name*.

The number of returned records is stored in variable*<recordset_name>.__count__ (e.g., RS.__count__)*. (Note the double underscores in front and after count; they are used to reduce the chance of confusing the name of this variable with a column name in a recordset.)

To iterate through the recordset, use Get Next Record block.

Conditional Exits

The Salesforce.com Search block may take one of the following conditional exits:*Failed* or *No Data*.

Failed

The Failed conditional exit is executed if the search operation failed.

No Data

The No Data conditional exit is executed if no data matching the specified search criteria is found.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Name the retrieved recordset

This is the name of the recordset that will be retrieved via this search operation. Name of the retrieved recordset is a mandatory field.

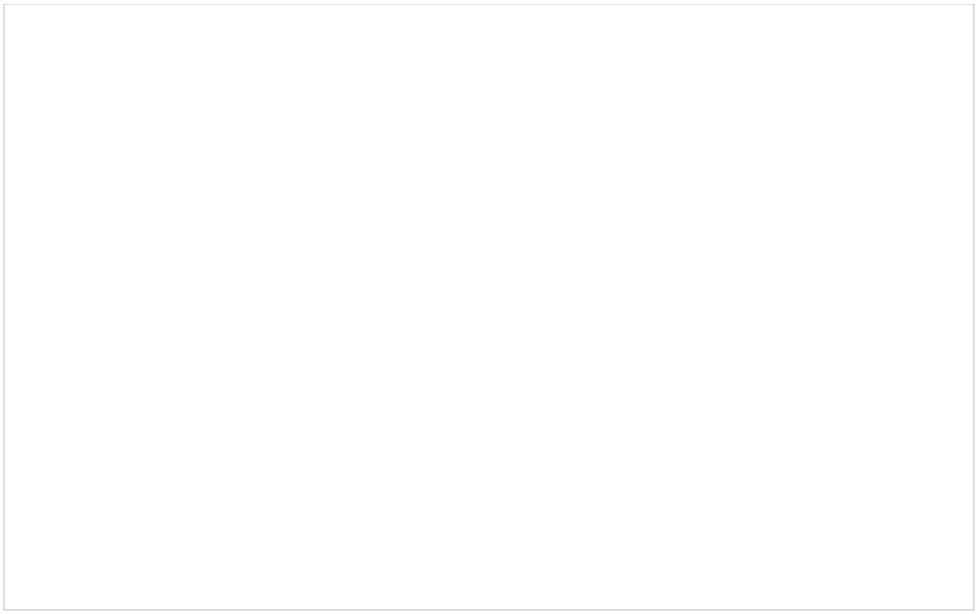
Query language

Query language is the language used for the data query. Possible language options include the Salesforce Object Query Language (SOQL) and Salesforce Object Search Language (SOSL). Unlike SOQL, which can only query one object at a time, a single SOSL query can be used to search multiple objects, which can both simplify and improve the efficiency of searches, especially in large SFDC environments. For more information, see Salesforce docs.

Statement

Statement refers to the record selection statement in the selected Query language. It may contain application variables in the *$(varname)* format.

The code and the body of the received HTTP response is stored in local variables*$(integrationResultCode)* and *$(integrationResultBody)*, respectively. For troubleshooting purposes, use the EMail or Internal Message block to obtain the content of responses indicating a failed search attempt. For more information, see the description of the variable $(integrationResultBody).



Salesforce.com Search workflow settings

<Previous | Next>

Salesforce.com Select Account

Your contact center configuration may contain multiple Salesforce integration accounts for access to different Salesforce systems. Use the *Salesforce.com Select Account* block to specify the integration account that will be used by subsequent Salesforce blocks in the given workflow. If this block is not used, all Salesforce blocks in the given workflow will use access data from the integration account marked as the Default account.

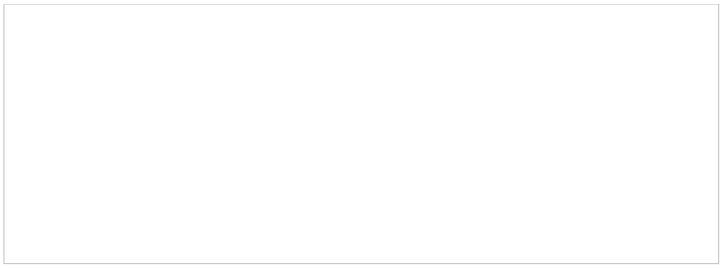
Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Account

Account is the name of the Salesforce integration account that will be used for access to Salesforce data by subsequent Salesforce blocks in the given workflow.



Salesforce.com Select Account workflow settings

<Previous | Next>

Salesforce.com Update

The *Salesforce.com Update* workflow block is the part of Salesforce.com Integration with Bright Pattern Contact Center.

This block updates properties of the specified Salesforce.com (SFDC) object.

Conditional Exits

The Salesforce.com Update block may take one of the following conditional exits:*Failed* or *No Data*.

Failed

The Failed conditional exit is executed if the update operation failed.

No Data

The No Data conditional exit is executed if the specified object is not found.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Object type name

Object type name is the type of Salesforce.com (SFDC) object to be created as defined in the SFDC system. It may be specified as an application variable in the *$(varname)* format.

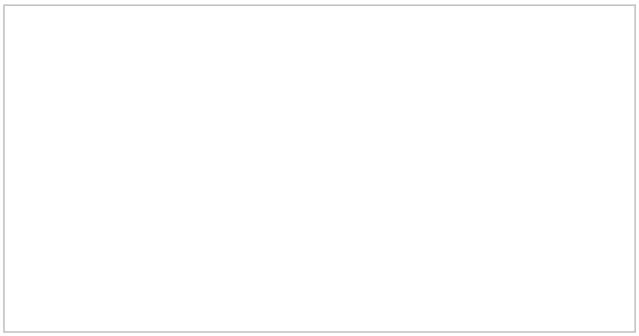
Object ID

Object ID is the identifier of the SFDC object to be updated. It may be specified as an application variable in the*$(varname)* format.

Fields to update

Fields to update are the object properties to be updated. Click **add** and specify the property *Name* as defined in the SFDC system, then, specify the desired new *Value*. If necessary, repeat for the other object properties to be updated. Field values may be specified as application variables in the *$(varname)* format.

The code and the body of the received HTTP response is stored in local variables*$(integrationResultCode)* and *$(integrationResultBody)*, respectively. For troubleshooting purposes, use the EMail or Internal Message block to obtain the content of responses indicating a failed attempt to update an object. For more information, see the description of the variable $(integrationResultBody).



Salesforce.com Update workflow settings

<Previous | Next>

Send Message+

The *Send Message+* workflow block is used to send a text message to a chat customer or to a mobile customer via SMS.

Conditional Exits

The Send Message block may take the *Send Error* conditional exit if the attempt to send a text message returned an error. This conditional exit may be used for SMS messages only.

Settings

Message

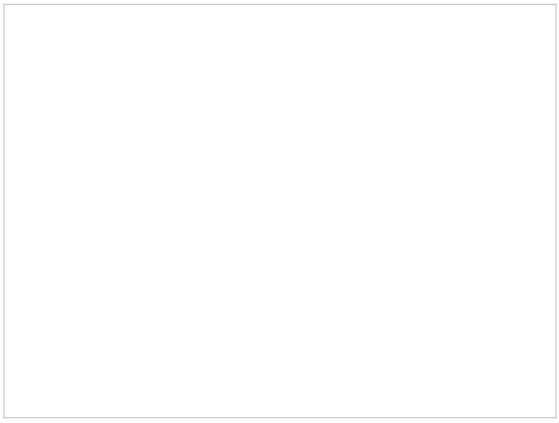
Message is the text of the message to be sent to the customer. Variables in the*$(varname)* format can be used in the message text.

Send from this number

If sending an SMS message, specify the number from which the text message will be sent. Variables in the *$(varname)* format can be used.

Send to this number

If sending an SMS message, specify the number to which the text message will be sent. Variables in the *$(varname)* format can be used. The default variable is *$(item.from)*.



Send Message+ workflow settings

ServiceNow Create Object

The *ServiceNow Create Object* workflow block creates a specified object in the ServiceNow database.

Conditional Exits

The *ServiceNow Create Object* block may take the *Failed* conditional exit if the create operation has failed.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Object type

Object type is the type of ServiceNow object to be created. You can either select one of the standard objects from the drop-down menu (e.g., “problem” or “incident”), or you can enter the name of the desired custom object type.

Variable name of object ID

This is the name of the variable that will be used as the identifier for the ServiceNow object to be created. The variable name of object ID will be set only if the block succeeds.

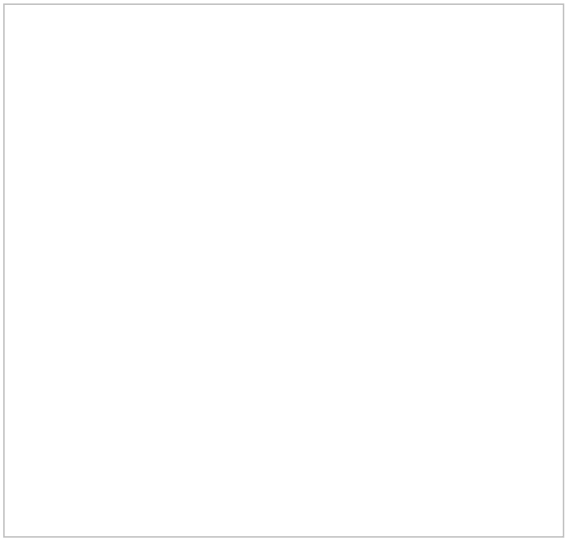
Set fields

This setting is reserved.

Raw JSON

Clicking *Raw JSON* enables object properties to be specified in JSON format.

The code and the body of the received HTTP response will be stored in local variables*$(integrationResultCode)* and *$(integrationResultBody)*, respectively. For troubleshooting purposes, use the EMail or Internal Message block to obtain the content of responses indicating a failed attempt to create an object. For more information, see the description of variable $(integrationResultBody).



RightNow Create Object workflow settings

<Previous | Next>

ServiceNow Search

The *ServiceNow Search* workflow block is used to obtain ServiceNow data. The block executes the specified ServiceNow record selection statement written in the ServiceNow Encoded Query String.

The columns of the first record of the retrieved recordset are stored in variables *<recordset_name>.<column_name>*. For that statement, the results will be stored in variables *Recordset.id* and *Recordset.name*.

The number of returned records is stored in variable *<recordset_name>.__count__* (e.g., *RS.__count__*.). Note the double underscores in front and after count; they are used to reduce the chance of confusing the name of this variable with a column name in a recordset.

To iterate through the recordset, use the Get Next Record block.

Conditional Exits

The ServiceNow Search block may take one of the following conditional exits: *Failed* or *No Data*.

Failed

The Failed conditional exit is executed if the search operation failed.

No data

The *No data* conditional exit is executed if no data matching the specified search criteria is found.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Object type

Object type is the type of ServiceNow object to be created. You can either select one of the standard objects from the drop-down menu (e.g., “problem” or “incident”), or you can enter the name of the desired custom object type.

Query

Query is the record selection statement. It may contain application variables specified as *$(varname)*.

Recordset name

Recordset name is the name of the recordset that will be retrieved via this search operation. The recordset name is the same as the value that you entered for the *Variable* name of object ID in the ServiceNow Create Object block.

The code and the body of the received HTTP response is stored in local variables *$(integrationResultCode)* and *$(integrationResultBody)* respectively. For troubleshooting purposes, use the EMail or Internal Message block to obtain the content of responses indicating a failed search attempt. For more information, see the description of the variable $(integrationResultBody).

ServiceNow Search workflow settings

<Previous | Next>

ServiceNow Select Account

Your contact center configuration may contain multiple ServiceNow integration accounts for access to different ServiceNow systems. Use the *ServiceNow Select Account* block to specify the integration account that will be used by subsequent ServiceNow blocks in the given workflow. If this block is not used, all ServiceNow blocks in the given workflow will use access data from the integration account marked as the *Default* account.

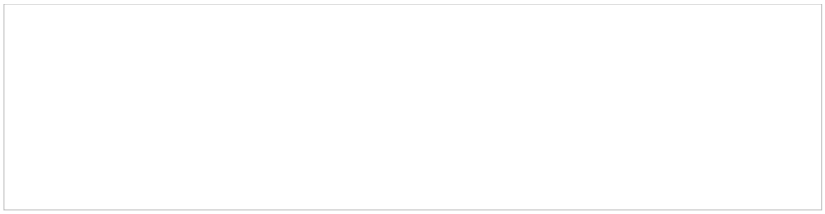
Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Account

Account is the name of the ServiceNow integration account that will be used for access to ServiceNow data by subsequent ServiceNow blocks in the given workflow.



ServiceNow Select Account workflow settings

<Previous | Next>

ServiceNow Update Object

The *ServiceNow Update Object* workflow block updates the properties of the specified ServiceNow object.

Conditional Exits

The ServiceNow Update Object block may take one of the following conditional exits: *Failed* or *No data*.

Failed

The Failed conditional exit is executed if the update operation failed.

No data

The No data conditional exit is executed in the specified object is not found.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Object Type

Object type is the type of ServiceNow object to be created. You can either select one of the standard objects from the drop-down menu (e.g., “problem” or “incident”), or you can enter the name of the desired custom object type.

Object Identifier

Object Identifier is the identifier of the object to be updated.

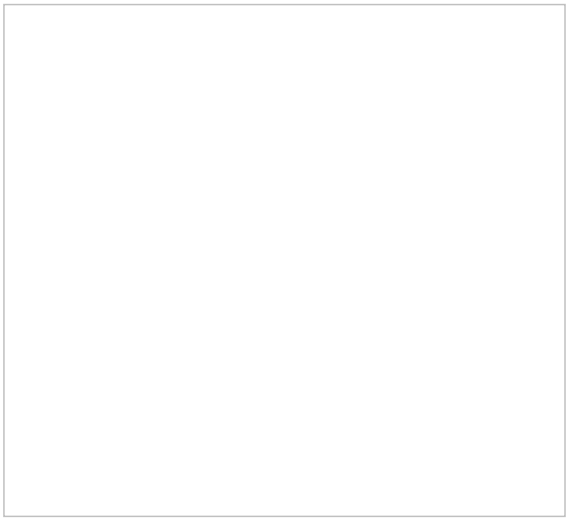
Set fields

This setting is reserved.

Raw JSON

Raw JSON enables object properties to be specified in JSON format.

The code and the body of the received HTTP response is stored in local variables*$(integrationResultCode)* and $(integrationResultBody) respectively. For troubleshooting purposes, use the EMail or Internal Message block to obtain the content of responses indicating a failed attempt to update an object. For more information, see the description of the variable $(integrationResultBody).



ServiceNow Update Object workflow settings

<Previous | Next>

Set Variable

The *Set Variable* block sets a value for a workflow variable.

Set Variable workflow block

Settings

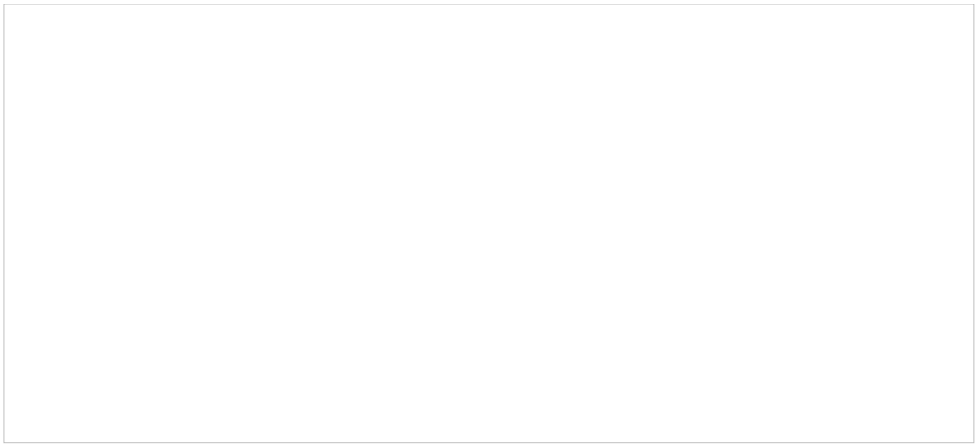
Variable name

This is the name of the variable. The Variable name can be set to be anything you like.

Value

Value is the desired variable value. Variables in the *$(varname)* format can be used as values. Values can be specified as either expressions or literal strings. Literal strings are passed exactly as entered.

Expressions must begin with assignment sign = as the first character. For example, 2+2 will produce 2+2, whereas =2+2 will produce 4. The expression result produces one of the following data types: strings, integers, and floating point numbers.



Set Variable workflow settings

<Previous | Next>

Start Another Workflow

The *Start Another Workflow* block starts the specified workflow from within the given (parent) workflow. Variables defined in the parent workflow carry over to the sub workflow. After the sub workflow finishes executing, control returns to the parent workflow (except when terminated by error or disconnect). The parent workflow will resume by executing the next block in the flowchart.

To configure the block, select the desired sub workflow in the drop-down list. The label of the block in the flowchart will display Start Another Workflow "[workflow name]". In the workflow shown, you can see that the Start Another Workflow block is used if the call is disconnected.

Start Another Workflow block settings

<Previous | Next>

Transfer Case to Service

The *Transfer Case to Service* workflow block transfers the active case to a different selected service.

Settings

Title Text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Service

Service allows you to select from a drop-down menu the new service you want the case transferred to.

Transfer Case to Service workflow settings

<Previous | Next>

Wait

Wait workflow block

The *Wait* workflow block will pause workflow execution for a specified amount of time (e.g., XX seconds, minutes, hours, days).

Settings

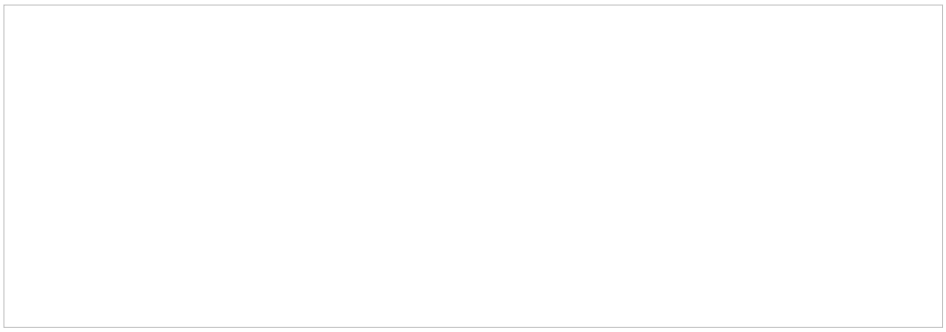
The Wait duration setting is where you specify the number of seconds, minutes, hours, or days for which any workflow execution should be paused; the default time is 10 seconds. Decimal fractions are allowed. The maximum value is 15 days.

Wait workflow settings

Example Usage

The Wait block can be used along with an EMail block in a workflow for soliciting post-transactional surveys via email.

Note that for surveys sent via workflows, the survey arrives faster than email. When soliciting post-transactional surveys via email, we recommend specifying a Wait duration of 24 hours, as shown. Doing so will delay the workflow for 24 hours before sending the survey, ensuring that the survey is available for the customer to read when they answer the email.



Wait block configuration

Wait+

The *Wait+* workflow block will pause workflow execution for a specified amount of time. Additionally, you may also select the service for which the wait applies and when to cancel the wait (based on dispositions).

Wait+ workflow block

Settings

Maximum wait duration

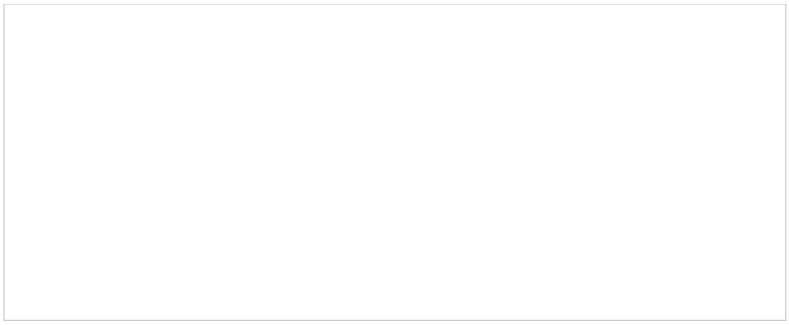
The *Maximum wait duration* setting is where you specify the number of seconds, minutes, hours, or days for which any workflow execution should be paused; the default time is 30 minutes. Decimal fractions are allowed. The maximum value is 15 days.

Service

Service allows you to select from a drop-down menu the service you would like the wait to apply to.

Stop waiting on

Stop waiting on allows you to configure a point to cancel the wait based on the service reaching a specific disposition before the maximum wait duration is reached. To configure this, first, select either *Non-final case disposition* or *Final case disposition*, then the select the specific disposition; once these criteria are met, the wait will end.



Wait+ workflow settings

<Previous | Next>

Zapier Invoke Zap

The *Zapier Invoke Zap* workflow block sets a trigger in Zapier that invokes a Zap workflow.

Conditional Exits

The Zapier Invoke Zap block will take the *Failed* conditional exit if the invoke operation has failed.

Settings

Title Text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Response in JSON

Response in JSON returns a value as a JSON string.

Zapier Invoke Zap workflow settings

<Previous | Next>

Zapier Select Account

Your contact center configuration may contain multiple Zapier integration accounts for access to different Zapier systems. Use the *Zapier Select Account* block to specify the integration account that will be used by subsequent Zapier blocks in the given workflow. If this block is not used, all Zapier blocks in the given workflow will use access data from the integration account marked as the *Default* account.

Settings

Title Text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Account

Account is the name of the Zapier integration account that will be used for access to Zapier data by subsequent Zapier blocks in the given workflow.

Zapier Select Account workflow settings

<Previous | Next>

Zendesk API Request

The Zendesk API Request workflow block allows you to make a free-form Zendesk API request to Zendesk using JSON, making it possible to take a variety of actions with your integrated account. The block uses your preconfigured Zendesk integration account for authorization and so forth. Note that the HTTP method is configurable and the returned results are processed the same way as they are with the Fetch URL workflow block.

Conditional Exits

Failed

The *Failed* conditional exit is taken if an error occurred during the API method execution.

No Data

The *No Data* conditional exit is executed if no data is returned in the body of the HTTP response.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and click the **Save** button at the bottom of the Edit pane. The new name of the block appears in the flowchart.

Request type

Request type is the type of method to be used to retrieve content from the specified Zendesk URL. Request types are defined in the drop-down menu.

Select from the following request types, or write in another method manually:

- GET (default)
- POST
- PUT
- PATCH
- DELETE

URL

URL is the URL of the Zendesk API you are requesting. Query string parameters are specified separately. Note that this overrides the **URL setting** in your Zendesk integration account.

Set fields

This setting is reserved.

Raw JSON

Raw JSON is where object properties are specified in JSON format.

Note: The code and the body of the received HTTP response will be stored in local variables *$(integrationResultCode)* and *$(integrationResultBody)*, respectively. For troubleshooting purposes, use the EMail or Internal Message block to obtain the content of responses indicating a failed attempt to create an object. For more information, see the description of the variable *$(integrationResultBody)*.

Initial path in the result JSON

If the response body contains JSON, this setting can be used to save into workflow variables a specific part of the data. Example: *myobject.node.list[4]*. The default is "none"; the path starts from the root of the returned JSON.

Scenario variable prefix for JSON data

This string will be used as the name of the variable to receive parsed JSON data. Note that if the initial path above points to an array, depending on the value of the following *GetNext* option, this variable would either contain the array or its first (and subsequent) elements.

Use GetNext block to loop through data

This box is selected if the JSON response data (at the initial path) is an array. The workflow variable will be set to the first element of the array and GetNext block could be used to iterate over the array elements, setting workflow variable to the next element.

Zendesk API Request workflow block settings

Zendesk Create Object

The *Zendesk Create Object* workflow block creates a specified object in the Zendesk database. Objects that can be created are tickets and users. For more information, see the following articles on creating tickets and creating users.

Conditional Exits

The Zendesk Create Object block will take the *Failed* conditional exit if the create operation has failed.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Object type

Object type is the type of the Zendesk object to be created. An object can be either a ticket or a user.

Variable name of object ID

This is the name of the variable that will be used as an identifier for the Zendesk object to be created. The variable name of object ID will be set only if the Zendesk Create Object block succeeds.

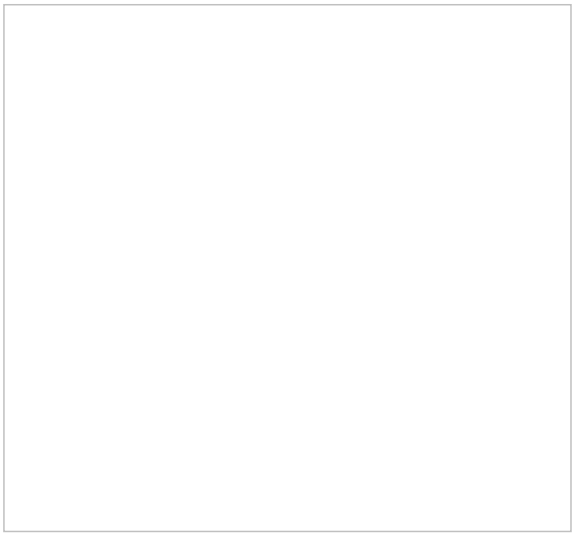
Set fields

This setting is reserved.

Raw JSON

Raw JSON is where object properties are specified in JSON format.

The code and the body of the received HTTP response will be stored in local variables*$(integrationResultCode)* and *$(integrationResultBody)*, respectively. For troubleshooting purposes, use the EMail or Internal Message block to obtain the content of responses indicating a failed attempt to create an object. For more information, see the description of the variable $(integrationResultBody).



Zendesk Create Object workflow settings

Zendesk Search

The *Zendesk Search* workflow block executes the specified Zendesk record selection statement. (For more information, refer to Zendesk's Search API and Zendesk Support search reference articles.)

The columns of the first record of the retrieved recordset are stored in the variables <*recordset_name*>.<*column_name*>. For that statement, the results will be stored in the variables *Recorset.id* and *Recordset.name*.

The number of returned records is stored in the variable<*recordset_name*>.__*count*__ (e.g., *RS.__count__*). (Note the double underscores in front and after count; they are used to reduce the chance of confusing the name of this variable with a column name in a recordset.)

To iterate through the recordset, use the Get Next Record block.

Conditional Exits

The Zendesk Search block may take one of the following conditional exits:*Failed* or *No Data*.

Failed

The Zendesk Search block will execute the Failed conditional exit if the search operation failed.

No Data

The No Data conditional exit is executed if no data matching the specified search criteria is found.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Recordset name

Recordset name is the name of the recordset that will be retrieved via this search operation.

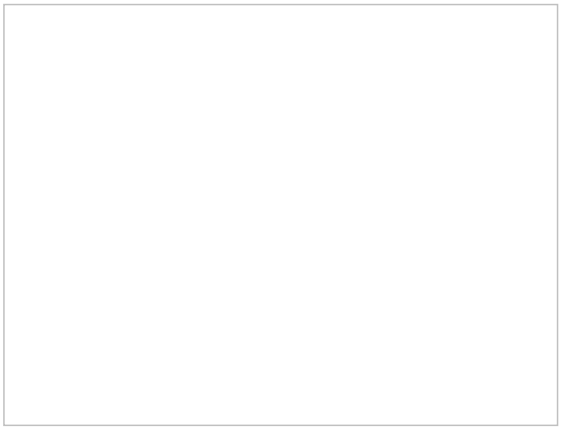
Max results

Max results is the maximum number of records in the recordset.

Search string

Search string is the record selection statement, and it may contain application variables specified in the*$(varname)* format.

The code and the body of the received HTTP response is stored in local variables*$(integrationResultCode)* and *$(integrationResultBody)*, respectively. For troubleshooting purposes, use the EMail or Internal Message block to obtain the content of responses indicating a failed search attempt. For more information, see the description of the variable $(integrationResultBody).



Zendesk Search workflow settings

<Previous | Next>

Zendesk Select Account

Your contact center configuration may contain multiple Zendesk integration accounts for access to different Zendesk systems. Use the Zendesk Select Account workflow block to specify the integration account that will be used by subsequent Zendesk blocks in the given workflow. If this block is not used, all Zendesk blocks in the given workflow will use access data from the integration account marked as the *Default* account.

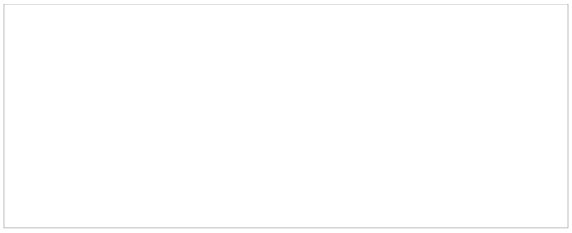
Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Account

Account is the name of the Zendesk integration account through which Zendesk data will be accessed by subsequent Zendesk blocks in the given workflow.



Zendesk Select Account workflow settings

<Previous | Next>

Zendesk Update Object

The *Zendesk Update Object* workflow block updates the properties of the specified Zendesk object. Objects that can be updated are tickets and users. (For more information, refer to Zendesk's articles on updating tickets and updating users.)

Conditional Exits

The Zendesk Update Object block may take one of the following conditional exits:*Failed* or *No Data*.

Failed

The Failed conditional exit is executed if the update operation failed.

No Data

The No Data conditional exit is executed if the specified object is not found.

Settings

Title text

Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

Object type

Object type is the type of Zendesk object to be created. The object can be either a ticket or a user.

Object ID

Object ID is the identifier of the object to be updated.

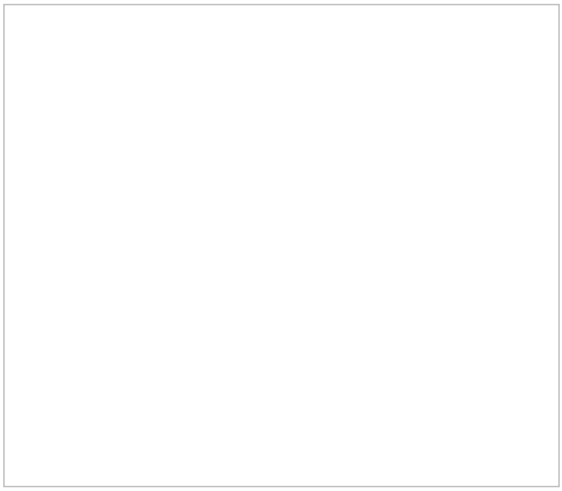
Set fields

This setting is reserved.

Raw JSON

Raw JSON is where the object properties to be updated are specified in JSON format.

The code and the body of the received HTTP response is stored in local variables*$(integrationResultCode)* and *$(integrationResultBody)*, respectively. For troubleshooting purposes, use the EMail or Internal Message block to obtain the content of the responses indicating a failed attempt to update an object. For more information, refer to the description of the variable $(integrationResultBody).



Zendesk Update Object workflow settings

<Previous | Next>

Standard Fields for CRM Objects

The following is a list of standard fields from CRM objects that are returned in theBright Pattern Search Object block.

Note: Custom fields are stored with the**custom**_ prefix.

Activity History

_id

Identifier of the given activity history object

Example: _id: ObjectId("58c37dc25be74141236fc250")

account_id

For email activities, returns the internal identifier of the email scenario entry

Example: account_id: "B5A69B62-9E37-48A0-8864-5E71E19148EE"

assigned_by_first_name

Reserved

assigned_by_last_name

Reserved

assigned_by_user_id

Reserved

assigned_from_first_name

Reserved

assigned_from_last_name

Reserved

assigned_from_user_id

Reserved

case_ids

Returns identifiers of cases that the given activity relates to; note the following:

- For chat and voice interactions, it may take multiple interactions to resolve a case
- For email interactions and notes, a single case is allowed

Example: case_ids: [ObjectId("58b744a27477eb6d1076e645")]

created_time

For chat and voice interactions, returns the time at which the interaction started (i.e., when it was first detected in the system); for email interactions and notes, returns the time at which this record was created (e.g., inbound email was received, outbound email was sent)

Example: created_time: ISODate("2016-10-14T01:39:38.326Z")

direction

For interaction-handling activities, provides a media-specific direction; it displays one of the following: INBOUND, OUTBOUND, REPLY, FORWARD, AUTO_ACK

Note the following:

- For chat interactions, it displays either INBOUND or OUTBOUND
- For email interactions, it may display INBOUND, OUTBOUND, REPLY, FORWARD, or AUTO_ACK
- For voice interactions, it displays either INBOUND or OUTBOUND

Example: direction: "OUTBOUND"

email_id

For email activities, returns the ID of the current email interaction

Example: email_id: ObjectId("579a651f7477eb3c65b652d8")

event

Reserved

flagged

Indicates whether an activity history record is flagged in the activity history of a case

Example: flagged: false

global_interaction_id

For interaction-handling activities, returns the global interaction identifier (GIID)

Example: global_interaction_id: "E9FF0804-A1E9-47CF-8D2F-60E684B21657"

has_voice_recording

If voice recordings are enabled, this indicates whether the interaction has a voice recording.

Example: has_voice_recording: false

media_type

Returns a value that indicates the type of interaction, which include the following: EMAIL, CHAT, VOICE, CASE, or NOTES

Example: media_type: "EMAIL"

original_email

For email activities, returns the ID of the original email interaction in the case

Example: original_email_id: ObjectId("5790099c7477eb673f327780")

parties

This returns an array of values associated with the parties involved in the activity; the party type will be one of the following: CONTACT, SCENARIO, USER, or UNIDENTIFIED. Note that USER applies to agents and supervisors.

Example:

```
parties: [
 {
   party_type: 'USER',
   user_id: "93B68CAB-9271-4B0A-AFB6-B9238CD36875",
   first_name: "Edna",
   last_name: "Partee",
   notes: "",
   disposition: "Product sold",
   party_role: 'CALLEE'
   party_id: "58c1e0297477eb3f8d6552ec"
  start_time: ISODate("2017-03-14T19:15:06.784Z"),
   duration: 338,
   custom_address_verified: true,
   custom_name_verified: true
   service_id: "72452D33-A7B4-4A1A-914A-AFA4076A76C3",
 }

 {
   party_type: 'CONTACT',
   contact_id: ObjectId("552856a7e4b0acb4156ddd11"),
   first_name: "John",
   last_name: "Doe",
   email: "john.doe@example.com"
   party_id: "58c1e7137477eb3f8d655322"
   start_time: ISODate("2017-03-14T19:15:44.784Z"),
   duration: 300
}

 {
   party_type: 'UNIDENTIFIED',
   first_name: 'Wireless',
   last_name:'Caller',
   phone: '14155551212'
```

```
  }

  {
    party_type: 'SCENARIO',
    custom_address_verified: true,
    custom_name_verified: true
  }
]
```

pinned

Indicates whether an activity history record is pinned in the activity history of a case

Example: pinned: false

services

Provides a list of services that were involved in an activity; note that this is separate from the*parties* property because services may exist on abandoned/self-service attempts where there were no agent parties.

Example:

```
services: [
 {
  service_id: "72452D33-A7B4-4A1A-914A-AFA4076A76C3",
  service_name: "Maintenance Renewal",
 }
]
```

subject

Returns the subject of the conversation

Example: subject: "TEST SUBJECT"

tenant_id

Returns the ID of the contact center

Example: tenant_id: "EB95E6C6-A7A9-4581-83E7-7336FB8FC377"

thread_id

Returns the email thread ID, if configured

Example: thread_id: "JX54YYN91FZH"

transferred_from_first_name

Reserved

transferred_from_last_name

Reserved

transferred_from_user_id

Reserved

Case

case_number

The case number as defined in the Agent Desktop application

Example: case_number: "2451"

case_status

Returns one of the following case states: New, Open, Pending, Resolved, or Closed

Example: case_status: "Pending"

case_title

Case title, for emails, is copied from the original email subject line; agents can edit it when editing cases

Example: case_title: "Mid-april Bulk email #1060 on Mon Oct 19 15:49:28 PDT 2015"

category_id

Returns the ID of the case category

Example: category_id: "560AC234-D124-458F-8DFB-C11ADF48F1D2"

category_name

Returns the configured case category

Example: category_name: "Maintenance Renewal"

cc

Provides a list of on-copy contacts

Example:

```
cc: [
 {
  contact_id: "5800375a7477eb4f25c630a7",
  first_name: "Jane",
  last_name: "Doe"
 }
]
```

created_time

When the case was first created; time is in GMT

Example: created_time: ISODate("2015-10-19T22:50:22.565Z")

customer_update_time

For email, this states when the last customer email was received. For voice and chat interactions, it coincides with the *modified_time* property.

Example: customer_update_time: ISODate("2016-11-03T23:49:41.454Z")

is_flagged

Indicates if a case is flagged; it is unrelated to the interaction flag

Example: is_flagged: true

is_pinned

Indicates if the case is pinned; it is unrelated to email pins

Example: is_pinned: true

modified_time

This provides the time when the last update on the case was made or a new activity was posted on the case. Typically, this reflects the end of the interaction (i.e., when an update is done); time is in GMT.

Example: modified_time: ISODate("2016-11-03T23:49:41.454Z")

open_time

Provides the time when the case was opened by the agent (i.e., when the its state is changed from theNew state to the Open state)

Example: open_time: ISODate("2015-10-19T22:51:22.565Z")

pending_reason

Returns the contact center's configured Pending reasons; these are only present if a case is in the Pending state.

Example: pending_reason: "Waiting for info from customer"

pending_time

The time the case state was last set to Pending. The *pending_time* property is only present in cases in the Pending, Resolved, and Closed states. For multiple transitions to the Pending state, it will provide the time the last transition was used.

Example: pending_time: ISODate("2015-10-19T22:53:22.565Z")

priority

Reserved

reporter_first_name

The reporter's (i.e., the customer/ person who contacted your call center) first name as it was at the moment it was collected

Example: reporter_first_name: "John"

reporter_id

The ID of the case reporter (i.e., the customer/ person who contacted your call center); it is taken from the contact on the original interaction from the case

Example: reporter_id: ObjectId("5581ef0de4b02187dd0a555f")

reporter_last_name

The reporter's (i.e., the customer/ person who contacted your call center) last name as it was at the moment it was collected

Example: reporter_last_name: "Doe"

resolved_time

The time the case state was set to Resolved; it is used for automatic transition to the Closed state. The *resolved_time* property is only present in cases in the Resolved and Closed states. For multiple transitions to the Resolved state, it will provide the time the last transition was used.

Example: resolved_time: ISODate("2016-11-03T23:49:41.454Z")

response_sla_start_time

Reserved

response_sla_target_time

Reserved

response_sla_time

Reserved

sentiment

Provides the sentiment from the last customer interaction in numerical form; the score indicates how negative or positive the interaction was, based on sentiment analysis

Example: sentiment: 0.97

tenant_id

Returns the ID of the contact center

Example: tenant_id: "EB95E6C6-A7A9-4581-83E7-7336FB8FC377"

users

This provides a list of agents who have participated in the handling of this case.

Example:

```
users: [
  user_id: "93B68CAB-9271-4B0A-AFB6-B9238CD36875"
]
```

Company

company_name

The name of the company as defined in Agent Desktop

Example: company_name: "Warehousing Inc. 4"

created_time

When this company object was first created; time is in GMT

Example: created_time: ISODate("2015-03-24T23:57:13.440Z")

employees

The number of employees the company has as defined in Agent Desktop

Example: employees: "123"

modified_time

This provides the time when the last update for this company object was made; time is in GMT.

Example: modified_time: ISODate("2016-10-06T21:22:29.850Z")

revenue

The revenue of the company as defined in Agent Desktop

Example: revenue: "100 million"

tenant_id

Returns the ID of the contact center

Example: tenant_id: "EB95E6C6-A7A9-4581-83E7-7336FB8FC377"

web_url

The company's web URL as defined in Agent Desktop

Example: web_url: "www.example.com"

Contact

addresses

The contact's addresses as defined in Agent Desktop; possible values are PRIMARY, BILLING, SHIPPING, and OTHER

Example:

```
addresses: [
 {
  type: "PRIMARY",
  postcode: "90670",
  state: "CA",
  city: "Santa Fe Springs",
  address_line1: "1111 Bayhill Dr.",
  address_line2: "Suite 275",
  country: "USA",
  id: "579936537477eb39496a4bea"
 }
]
```

bpo_client_id

Reserved

company_id

Returns the ID of the company that this contact is associated with

Example: company_id: ObjectId("5511f9d9e4b0033ff9b8bc99")

created_time

When the contact was first created; time is in GMT

Example: created_time: ISODate("2015-04-10T23:03:03.322Z")

dob

The contact's date of birth (DOB) as defined in Agent Desktop

Example: dob: ISODate("2001-08-14T00:00:00.000Z")

emails

The contact's email addresses as defined in Agent Desktop; possible values are PRIMARY, BUSINESS, and PRIVATE

Example:

```
emails: [
 {
  type: "PRIMARY",
  email_address: "something@brightpattern.com",
  id: "579936537477eb39496a4be9"
 }
]
```

external_ids

Reserved

first_name

The contact's first name as defined in Agent Desktop

Example: first_name: "Jeanne"

last_name

The contact's last name as defined in Agent Desktop

Example: last_name: "Wengler"

messengers

Reserved

modified_time

This provides the time when the last update on the contact was made; time is in GMT.

Example: modified_time: ISODate("2016-09-24T00:11:48.205Z")

phone

The contact's phone numbers as defined in Agent Desktop; possible values are BUSINESS, HOME, MOBILE, and FAX

Example:

```
phones: [
 {
  type: "MOBILE",
  phone: "16505551212",
  id: "579fff287477eb45790af319"
 },
 {
  type: "HOME",
  phone: "14155551212",
  id: "57e2d6437477eb660c92899d"
 },
]
```

picture

Contact's photo

position

The contact's position as defined in Agent Desktop

Example: position: "Sr. Support Engineer"

segment

The contact's segment as defined in Agent Desktop

Example: segment: "Gold"

social_links

Reserved

summary

The summary of the contact as defined in Agent Desktop

Example: summary: "Jeanne works in the main office M - Th; off-site office F"

tenant_id

Returns the ID of the contact center

Example: tenant_id: "EB95E6C6-A7A9-4581-83E7-7336FB8FC377"

title

The contact's title as defined in Agent Desktop

Example: title: "Miss"

Workflow Variables

This section describes the variables that are used in Bright Pattern Contact Center workflows and scenarios.

Variables are accessed using the common *$(varname)* format. They can be used in integer and string expressions.

Note: If you would like to pass information from scenarios to workflows, note that not all scenario variables pass information to workflows; however, local variables are always passed to workflows. *Local variables* do not contain a prefix (e.g., *item.*, *user.*, etc.) and their values are available to the scenario where they are defined and to the sub-scenarios that are started from that parent scenario using the Start Another Scenario block. For any variables that are not passed to workflows, you may use a Set Variable block to rename the required scenario variable into a local variable (e.g., set "varDNIS" to *$(item.DNIS)*).

Common Variables

$(user.loginId)

$(user.loginId) specifies the agent's login ID if the workflow is started due to agent action.

Note: This variable is available in scenarios.

$(user.firstName)

$(user.firstName) specifies the agent's first name.

Note: This variable is available in scenarios.

$(user.lastName)

$(user.lastName) specifies the agent's last name.

Note: This variable is available in scenarios.

$(item.caseId)

$(item.caseId) specifies the case ID of the interaction, if available.

$(item.caseNumber)

$(item.caseNumber) specifies the case number of the interaction, if available.

$(item.contactId)

$(item.contactId) specifies the contact ID of the customer, if available.

Note: This variable is available in scenarios.

$(item.firstName)

$(item.firstName) specifies the first name of the customer, if available.

Note: This variable is available in scenarios.

$(item.lastName)

$(item.lastName) specifies the last name of the customer, if available.

Note: This variable is available in scenarios.

$(global_interaction_id)

$(global_interaction_id) specifies the Global interaction identifier. This variable is also known as *$(item.globalInteractionId)*.

Note: This variable is available in scenarios.

$(disposition)

$(disposition) specifies the disposition that was used in an interaction.

Note: This variable is available in scenarios.

Variables for Voice

$(LanguageAsked)

$(LanguageAsked) specifies whether the customer asked for a specific language (e.g., "Yes").

$(NPS_raw)

$(NPS_raw) specifies the Net Promoter Score (NPS) value (e.g., "11").

$(contact_satisfaction)

$(contact_satisfaction) specifies the customer's contact satisfaction rating (e.g., "1").

$(destination)

$(destination) specifies the destination for the Connect Call block (e.g., "2042").

Note: This variable is available in scenarios.

$(first_call)

$(first_call) specifies whether this voice interaction is the first placed call (e.g., "1").

$(screenpopData)

$(screenpopData) specifies the list of the screen pop data received or set by interactive voice response (IVR). An actual list of available screen pop data elements depends on the particular IVR and integration.

Note: This variable is available in scenarios.

$(item.ANI)

$(item.ANI) specifies the Automatic Number Identification (ANI), a telephone company service providing a calling party with a number of the calling party. "ANI" is often used instead of “calling party number.”

Note: This variable is available in scenarios.

$(item.DNIS)

$(item.DNIS) specifies the Dialed Number Identification Service (DNIS), a telephone company service providing the called party with a number that was dialed by the calling party. "DNIS" is often used as a shorthand for "called party number."

Note: This variable is available in scenarios.

$(item.cnam)

$(item.cnam) is the caller ID display name (if provided by the SIP network).

$(item.customerPhone)

$(item.customerPhone) specifies the customer phone number. This variable enables passing customer phone information from scenarios of primary inbound and outbound customer calls to scenarios of the associated consultations and blind transfers.

- In scenarios launched for new incoming calls, its value matches the value of the *$(item.from)* variable.
- In scenarios launched for new outgoing calls, its value matches the value of the *$(item.to)* variable.
- In scenarios launched for consult calls and blind transfers, the value of the *$(item.customerPhone)* is inherited from the scenarios of the associated primary calls.

Note: This variable is available in scenarios.

$(item.from)

$(item.from) specifies the origination address (i.e., phone number or chat user display name). This variable is also known as *ANI*.

$(item.interactionId)

$(item.interactionId) specifies the interaction identifier.

Note: This variable is available in scenarios.

$(item.media)

$(item.media) specifies the media type (e.g., "voice").

Note: This variable is available in scenarios.

$(outbound_data)

$(outbound_data) contains data from calling lists and is available if the workflow is triggered from outbound campaign.

$(item.transcript.JSON)

$(item.transcript.JSON) specifies the full JSON transcript of the chat session or voice call.

Note: This variable can pass information from scenarios.

$(item.transcript.HTML)

$(item.transcript.HTML) specifies the HTML formatted transcript of the chat session or voice call.

Note: This variable can pass information from scenarios.

$(item.transcript.text)

$(item.transcript.text) specifies the text transcript of the chat session or voice call.

Note: This variable can pass information from scenarios.

String Expressions

In Bright Pattern Contact Center, you may work with data as variables stored as values, which may be specified as strings or expressions. A string is a sequence of characters that is generally understood as a data type; often a string is implemented as an array of bytes (i.e., words) that store a sequence of elements. An expression will produce a data type such as string.

What follows is a list of tips regarding string expression structure.

- Strings are enclosed in double quotes (e.g. *"sample string"*).
- Backslash can be used to embed a double quote phrase within a string (e.g.,*"sample string \"embedded quote\" sample string"*).
- Backslash can also be used to insert literal new-line and carriage-return symbols using the*\n* and *\r* notation (e.g., *"sample string\n with new line in it"*).
- Strings can be concatenated; that is, strings can be linked together as in a chain (e.g., *"string 1"*+" "+*"string2"* produces *"string1 string2"*).
- Strings themselves or string expressions cannot span multiple lines (i.e., while embedded, *\n* is OK, but the actual new line is not).

<Previous | Next>

Integer Expressions

Some helpful information on using integer expressions in the Workflow Builder application is given as follows.

- Numbers can have the unary minus operator (e.g., *-2*).
- The four arithmetic operations and parentheses, including nested ones, are supported (e.g., *(2+3)*((7-1)/2+1)*).
- Division by zero produces errors in the log; the operation result is undefined.
- Strings cannot be mixed in one expression with numbers (e.g., *=2 + "string"* is invalid).

<Previous | Next>

Floating Point Expressions

Some helpful information on using floating point expressions in the Workflow Builder application is given as follows.

- Either the point or exponent can be used (e.g., *2.00* or *2E00* or *20E-1* or *.2E1*).
- The four arithmetic operations and parentheses, including nested ones, are supported.
- A mix of floats and integers produces a floating expression.
- Division by zero produces errors in the log; the operation result is undefined.

<Previous | Next>

Built-In Functions

A number of built-in functions may be used in the Workflow Builder application.

To invoke a function from a text field, prefix it with an equal sign. Example: *=now("UTC")*

Descriptions

The functions are described as follows.

formatdatetime(int unixtimestamp, string format)

This function formats the time as specified in the format argument. This format is the same as in JavaSimpleDateFormat, as implemented by the International Components for Unicode (ICU)library.

Example: *"yyyy-MM-dd'T'HH:mm'Z'"* yields *2012-07-20T20:45:44.0973928Z*

formatduration(duration_in_seconds)

This function converts duration in specified in seconds into *MM:SS* or *HHH:MM:SS* formats. It produces formatted string as output (e.g., *formatduration(121)* will return *"02:01"*).

hmac(hash_function, key, message)

This function is used to create an authentication hash (HMAC) for MD5, SHA-1, and SHA-256, where*Hash_function = (“MD5” | “SHA-1” | “SHA-256”)*. Returned value is a string with base64-encoded hash.

length(string)

This function returns the number of characters in a string.

now(string timezone)

This function returns the current time in the specified time zone in Unix format (number of seconds elapsed since 1/1/1970, 00:00:00 UTC). Time zone is optional; if not specified, the current time will be returned in the tenant’s default time zone.

parsedatetime(string datetime, string format)

This function returns the specified date and time in Unix format (number of seconds elapsed since 1/1/1970, 00:00:00 UTC). The date and time input is expected in the ICU’s Java SimpleDateFormat.

replace(string, search_pattern, replace_pattern, flags)

- This function performs search and replace in the input string. It returns the string with replacements performed.
- Parameters are as follows:
  - **string** is the input string to be searched.
  - **search_pattern** is the regular expression pattern to be matched in the input string. The list of supported patterns can be found in the table below. Note the extra \ in escapes. This is necessary in order to allow literal insertions of " and new line symbols.
  - **replace_pattern** is the text to insert instead of matched text according to search pattern. *\\0 - \\9* are allowed in replacements.
  - **flags** denote one or more of the following additional conditions:
    - **i** – ignore case in the search
    - **g** – replace all matches because otherwise only the first match will be replaced (e.g., *replace("abcdefg","c","z","ig")* produces *"abzdefg"*).

Example that takes first name from fullname variable and could be used in value section of Set Variable block:
=replace("$(fullname)","(.*)\s+(.*)","\1","i")

round(floating_number, precision)

This function rounds the number to the <*precision*> number of digits after the point. The result is still a floating point number.

stripnondigits(string)

This function removes non-digit characters from string, leaving only digits from 0 to 9, * and # symbols (e.g., *stripnondigits("123abc456")* will return *"123456"*).

titlecase(string)

This function converts string to title case (i.e., each word is capitalized).

tostring(integer)

This function converts an integer to a string (e.g., *tostring(-2+1)* should return *"-1"* as a string).

urlencode(string)

URL encodes a string, replacing special characters using the *%dd* notation. This is a conservative implementation that replaces all characters that are not explicitly in allowed characters.

Pattern Types

Pattern types are described as follows.

Pattern	Description
^	Match beginning of a buffer
$	Match end of a buffer
()	Grouping and substring capturing
[...]	Match any character from the set
[^...]	Match any character but the ones from the set
\\s	Match whitespace
\\S	Match non-whitespace
\\d	Match decimal digit
\\D	Match anything but decimal digit
\\r	Match carriage return
\\n	Match new line
+	Match one or more times (greedy)
+?	Match one or more times (non-greedy)
*****	Match zero or more times (greedy)
***?**	Match zero or more times (non-greedy)
?	Match zero or once
\\meta	Match one of the meta characters: ^$().[*+?\

<Previous