

5.2 Workflow Builder Reference Guide

Bright Pattern Documentation

Generated: 10/16/2021 11:20 am

Content is available under license unless otherwise noted.

Table of Contents

Table of Contents	2
Purpose	8
Audience	8
Workflow Builder Overview	8
Graphical User Interface	9
Workflow Blocks	9
Conditional Exits	9
Settings	10
Workflow Entries Screen Properties	10
Name	10
Service	10
Triggers	10
Interaction ends with disposition	10
Non-final case disposition	11
Average sentiment	11
Assign Case to Agent	11
Settings	11
Title Text	11
Agent	12
Agent ID	12
Bright Pattern Search Object	12
How to Use This Block	12
Conditional Exits	12
Failed	13
No Data	13
Settings	13
Title Text	13
Object type	13
Search	13
Case	13
Company	13
Contact	14
Activity history	14
Return fields	15
Recordset name	15
Comment	15
Settings	15
Title text	15
Comment	15
DB Execute	16
Conditional Exits	16
No Data	16
Failed	16
Settings	16
Title text	17
DB Connection	17
Name	17
JDBC driver and connection string	17
Database username and password	17
SQL Statement	17
Recordset name	17
EMail	18
Conditional Exits	18
Settings	19
Title text	19
From / Display name	19
From / Address	19
To / Address(es)	19
Message / Subject	19
Message / Format	19
Message / Body	19
Exception Handler	20
Conditional Exits	20

Settings	21
Title text	21
Exit	22
Fetch URL	22
Conditional Exits	22
Failed	22
No Data	23
Settings	23
Title text	23
Request type	23
URL to fetch	23
Extra headers	23
URL parameters	23
Content Type	24
Body	24
Username	24
Password	24
Initial path in the result JSON	24
Scenario variable prefix for JSON data	24
Use GetNext block to loop through data	24
Response Data Handling	25
HTTP Response Codes	26
HTTP Redirect Response Handling	26
Get Next Record	27
Conditional Exits	27
Settings	27
Title text	27
Direction	27
Recordset name	27
Get User Configuration	28
Settings	28
Title text	28
Find user by	28
Value	28
User properties to return	28
Goto	29
If	30
Branches and Conditions	30
Block of Conditions	31
Typical Uses	31
Caller's Number	31
Current Time	32
Workflow Variable (String)	32
Internal Message	33
Conditional Exits	33
Settings	34
Title text	34
Username	34
Message	34
Log	34
Settings	34
Title text	34
Text message	35
Log Level	35
RightNow Create Object	35
Conditional Exits	35
Settings	35
Title text	35
Object type	36
Variable name of object ID	36
Set fields	36
Raw JSON	36
RightNow Search	36

Conditional Exits	37
Failed	37
No data	37
Settings	37
Title text	37
ROQL Query	37
Recordset name	37
RightNow Select Account	38
Settings	38
Title text	38
Account	38
RightNow Update Object	39
Conditional Exits	39
Failed	39
No data	39
Settings	39
Title text	39
Object Type	39
Object Identifier	39
Set fields	39
Raw JSON	40
Salesforce.com Delete	40
Conditional Exits	40
Failed	40
No Data	40
Settings	41
Title text	41
Object type name	41
Object ID	41
Salesforce.com Insert	41
Conditional Exits	42
Settings	42
Title text	42
Object type name	42
Variable name of object ID	42
Object fields	42
Salesforce.com Search	43
Conditional Exits	43
Failed	43
No Data	43
Settings	43
Title text	43
Name the retrieved recordset	43
Query language	43
Statement	44
Salesforce.com Select Account	44
Settings	45
Title text	45
Account	45
Salesforce.com Update	45
Conditional Exits	45
Failed	45
No Data	45
Settings	46
Title text	46
Object type name	46
Object ID	46
Fields to update	46
Send Message+	46
Conditional Exits	47
Settings	47
Message	47
Send from this number	47
Send to this number	47

ServiceNow Create Object	47
Conditional Exits	48
Settings	48
Title text	48
Object type	48
Variable name of object ID	48
Set fields	48
Raw JSON	48
ServiceNow Search	49
Conditional Exits	49
Failed	49
No data	49
Settings	50
Title text	50
Object type	50
Query	50
Recordset name	50
ServiceNow Select Account	51
Settings	51
Title text	51
Account	51
ServiceNow Update Object	51
Conditional Exits	51
Failed	52
No data	52
Settings	52
Title text	52
Object Type	52
Object Identifier	52
Set fields	52
Raw JSON	52
Set Variable	53
Settings	53
Variable name	53
Value	53
Start Another Workflow	54
Transfer Case to Service	54
Settings	55
Title Text	55
Service	55
Wait	55
Settings	55
Wait+	56
Settings	56
Maximum wait duration	56
Service	56
Stop waiting on	56
Zapier Invoke Zap	56
Conditional Exits	57
Settings	57
Title Text	57
Response in JSON	57
Zapier Select Account	57
Settings	58
Title Text	58
Account	58
Zendesk Create Object	58
Conditional Exits	58
Settings	58
Title text	58
Object type	59
Variable name of object ID	59

Set fields	59
Raw JSON	59
Zendesk Search	59
Conditional Exits	60
Failed	60
No Data	60
Settings	60
Title text	60
Recordset name	60
Max results	60
Search string	60
Zendesk Select Account	61
Settings	61
Title text	61
Account	61
Zendesk Update Object	62
Conditional Exits	62
Failed	62
No Data	62
Settings	62
Title text	62
Object type	62
Object ID	62
Set fields	63
Raw JSON	63
Workflow Variables	63
Common Variables	63
\$(user.id)	63
\$(user.loginId)	64
\$(user.team)	64
\$(user.firstName)	64
\$(user.lastName)	64
\$(item.caseId)	64
\$(item.caseNumber)	64
\$(item.contactId)	64
\$(item.firstName)	64
\$(item.lastName)	64
\$(global_interaction_id)	64
\$(disposition)	65
Variables for Voice	65
\$(LanguageAsked)	65
\$(NPS_raw)	65
\$(contact_satisfaction)	65
\$(destination)	65
\$(first_call)	65
\$(screenpopData)	65
\$(item.ANI)	65
\$(item.DNIS)	65
\$(item.cnam)	66
\$(item.customerPhone)	66
\$(item.from)	66
\$(item.interactionId)	66
\$(item.media)	66
\$(outbound_data)	66
String Expressions	66
Integer Expressions	67
Floating Point Expressions	67
Built-In Functions	67
Descriptions	68
formatdatetime(int unixtimestamp, string format)	68
formatduration(duration_in_seconds)	68
hmac(hash_function, key, message)	68
length(string)	68
now(string timezone)	68
parsedatetime(string datetime, string format)	68

replace(string, search_pattern, replace_pattern, flags)	68
round(floating_number, precision)	69
stripnondigits(string)	69
titlecase(string)	69
tostring(integer)	69
urlencode(string)	69
Pattern Types	69

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.

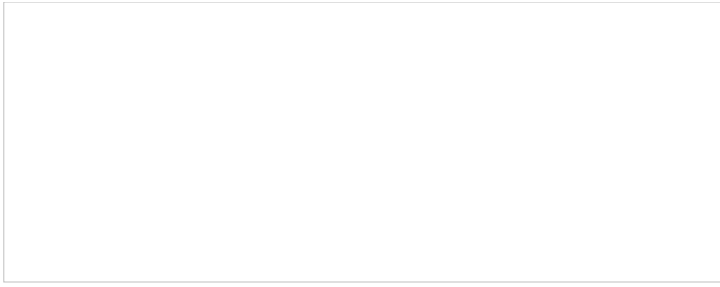
Trigger types include the following:

- Interaction ends with disposition
- Non-final case disposition
- Average sentiment

Interaction ends with disposition

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

- Problem solved
- Answer provided
- Spam
- Need more information
- Needs approval
- Approved
- Handled by other action

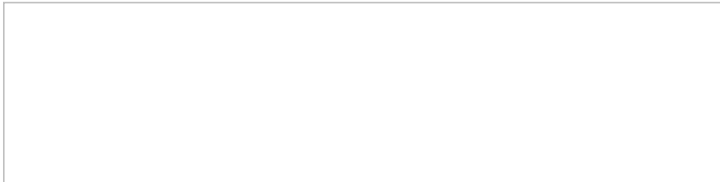


Interaction Ends with Disposition trigger

Non-final case disposition

If this trigger is selected, you must select a specific non-final case disposition from the drop-down list:

- Need more information
- Needs approval
- Approved



Non-Final Case Disposition trigger

Average sentiment

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.*

Assign Case to Agent

This block assigns the case to a selected agent.

[Workflow Builder Block](#)

Settings

Title Text

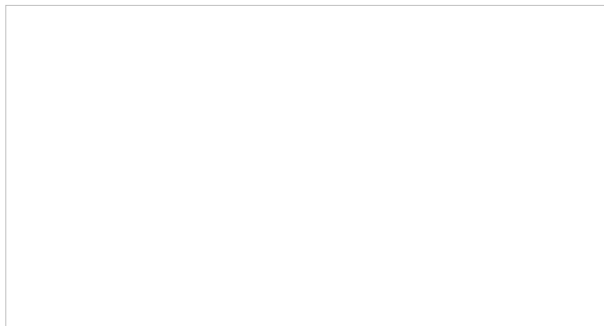
Title Text is the name of the instance of the block. Enter a name in the text field and the new name of the block will appear in the flowchart.

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

Bright Pattern Search Object

The *Bright Pattern Search Object* block finds an existing object in the database and retrieve activity data. Note this block will replace the [Retrieve Internal Record](#) block starting from version 5.2.3 of Bright Pattern Contact Center software.



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 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 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:

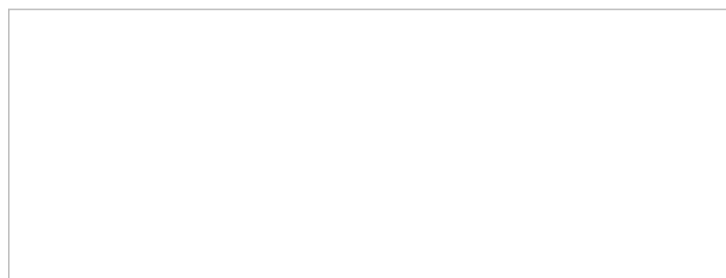
Search

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** 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.

A screenshot of a configuration window for 'Bright Pattern Search Object case settings'. The window is mostly empty, showing a large white rectangular area where settings would be configured. The title bar of the window is visible at the top.

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** 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.

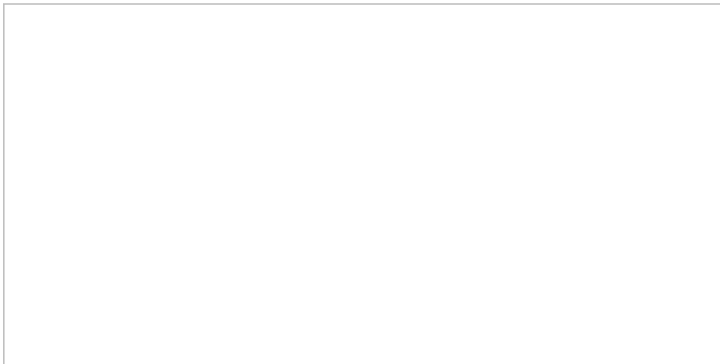


Bright Pattern Search Object company settings

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** 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.

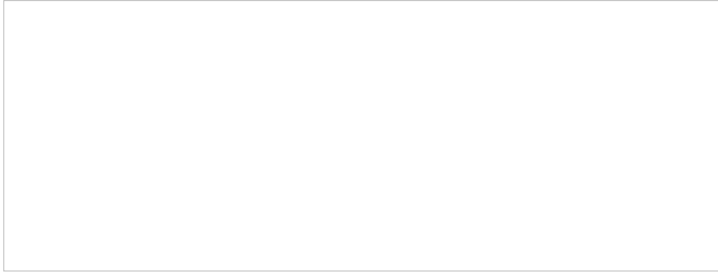


Bright Pattern Search Object contact settings

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** 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.

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](#)

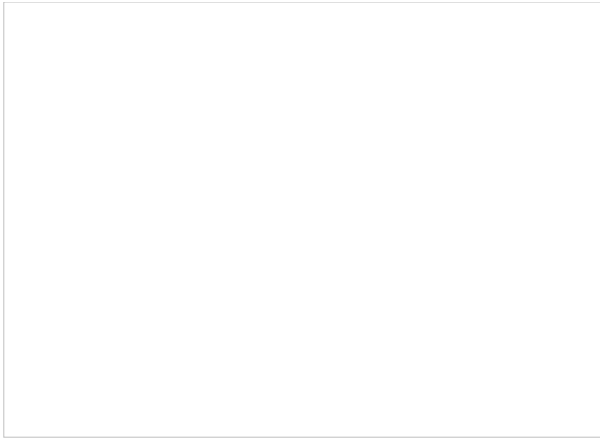
Settings

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

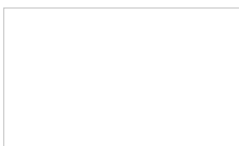
The DB Execute block may take one of the following conditional exits: Failed or No Data.

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

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*.

[EMail workflow block](#)

Conditional Exits

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).

[WB-EMail-CE-52.PNG](#)

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.

From / Display name

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

From / Address

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

To / Address(es)

This specifies 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.

Message / Subject

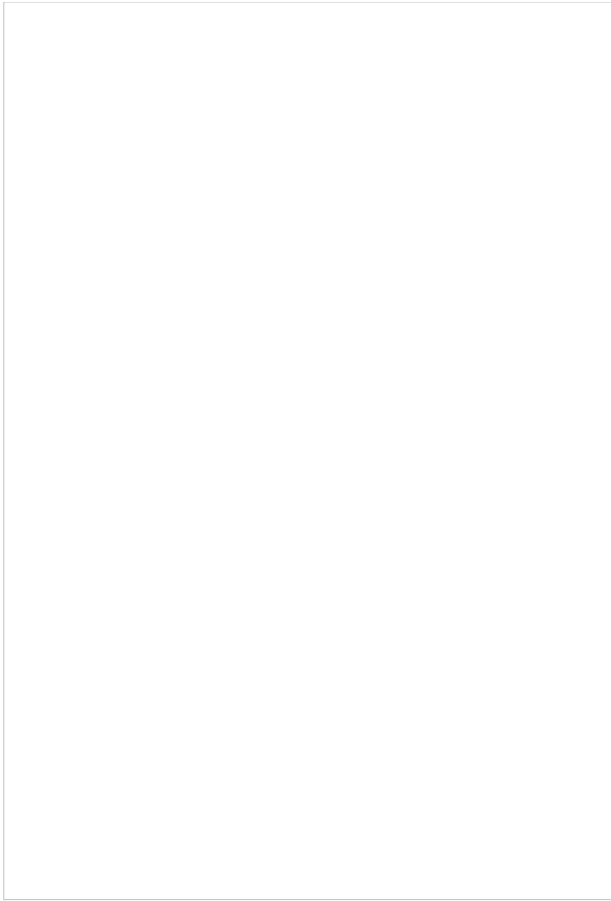
This is the subject line of the email. Can contain workflow variables in the $\$(varname)$ format.

Message / Format

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

Message / Body

This is 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.



EMail settings

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

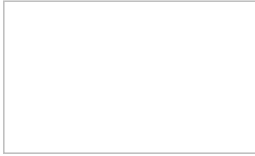
The Fetch URL block may take one of two conditional exits: *Failed* or *No Data*.

Failed

The Failed conditional exit is taken if an error occurred during the HTTP method execution. See [HTTP 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.

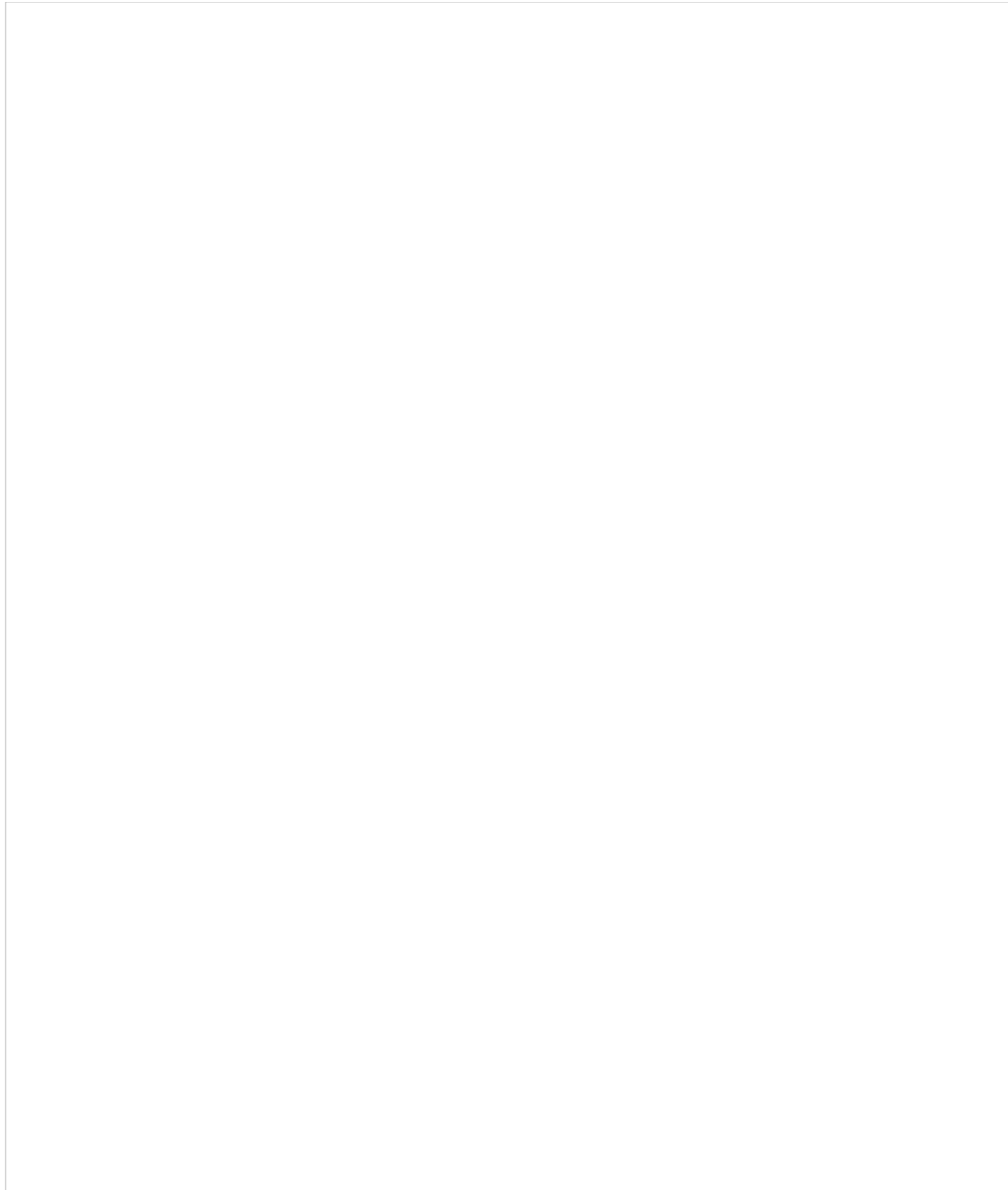
Scenario variable prefix for JSON data

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.

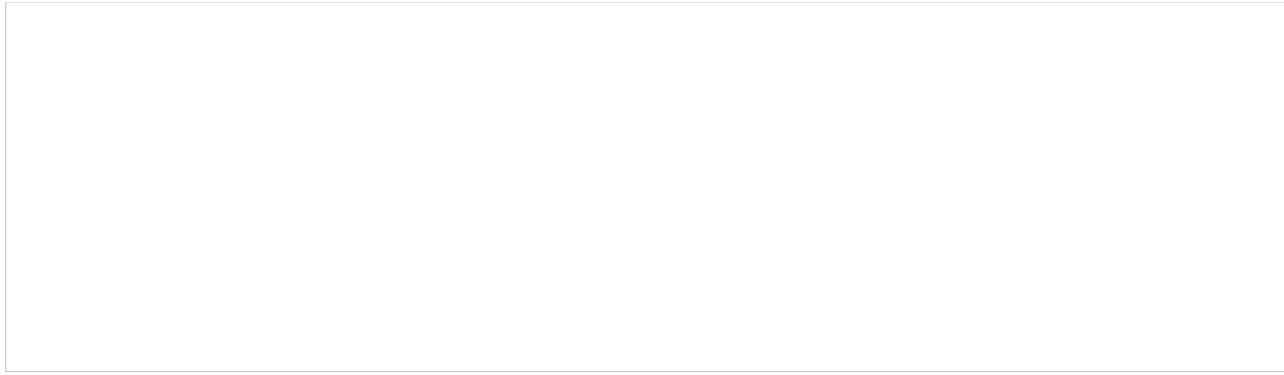
Direction

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.

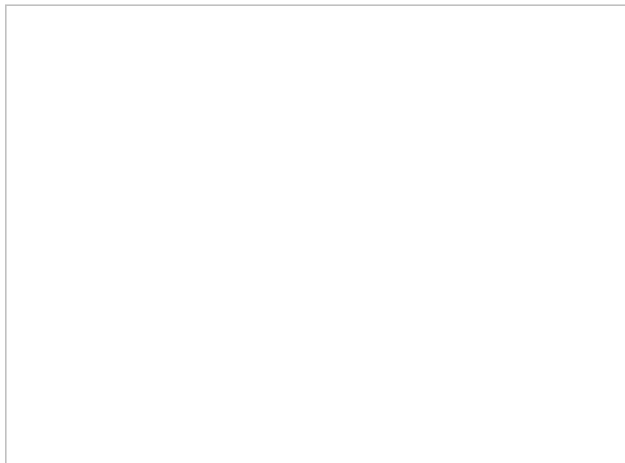
User properties to return

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

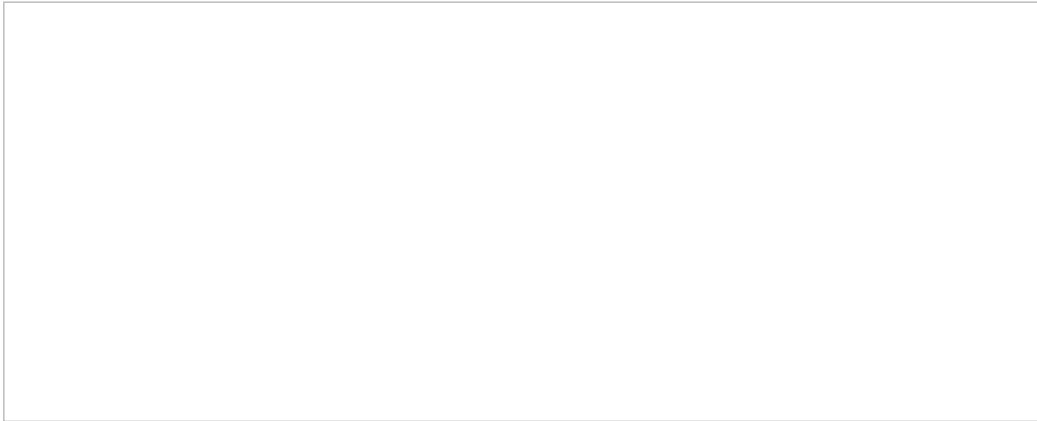
you define a destination for it.

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.



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.)

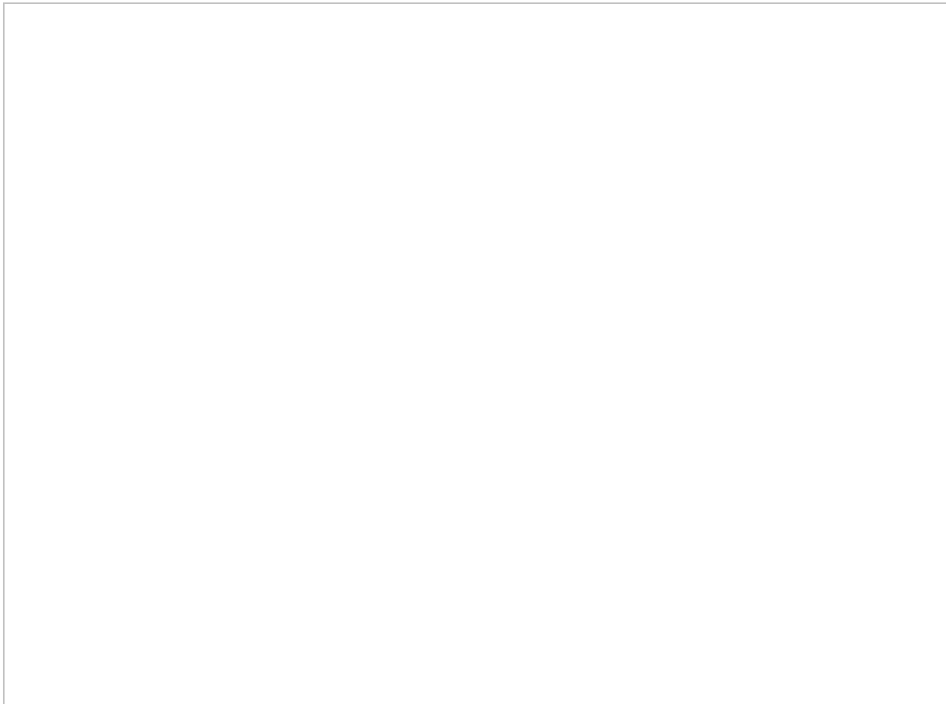
The branches are tried in the order in which they are defined in the block. If none of the branches leads to a positive verification, the block that directly follows the given If block in the flowchart is executed.

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."

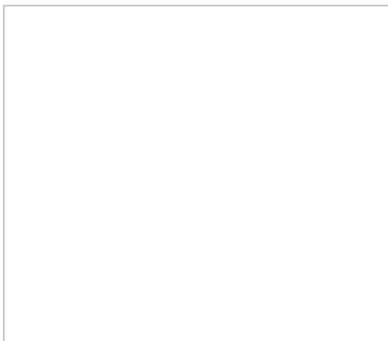
Workflow Variable (String)

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

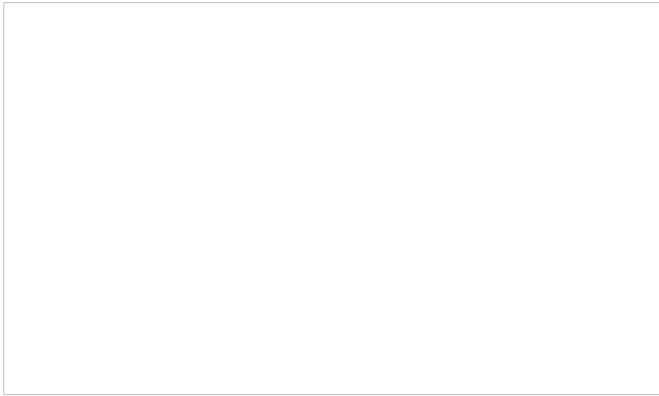
Title text is the name of the instance of the block. Enter a name in the text field and the new name of the block appears in the flowchart.

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.

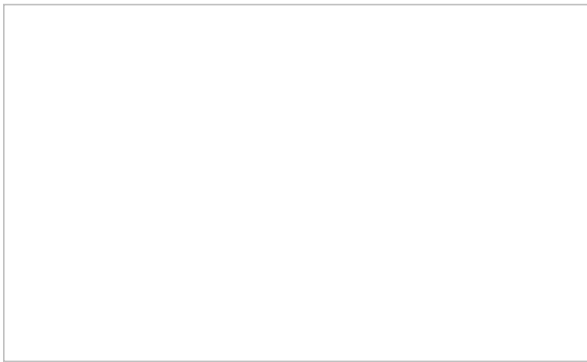
Text message

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

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.

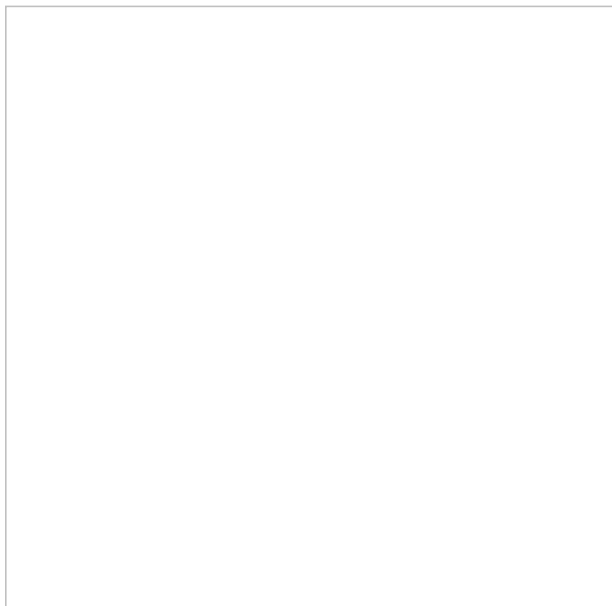
Set fields

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 `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 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.

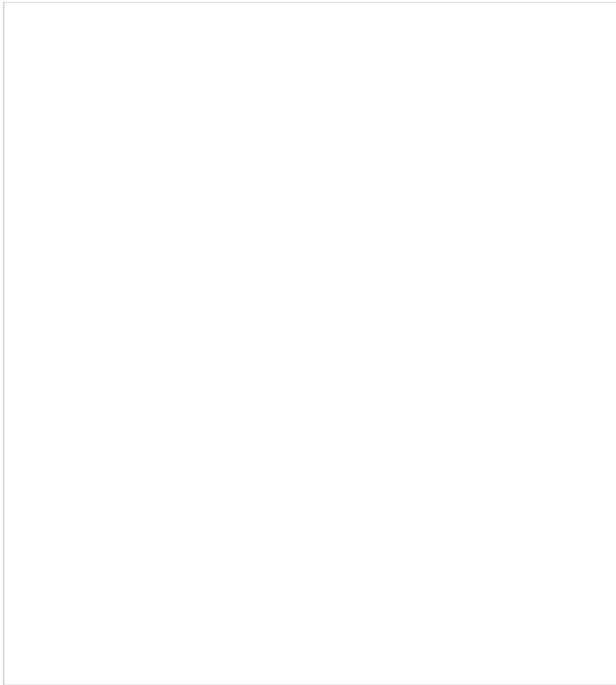
ROQL Query

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.

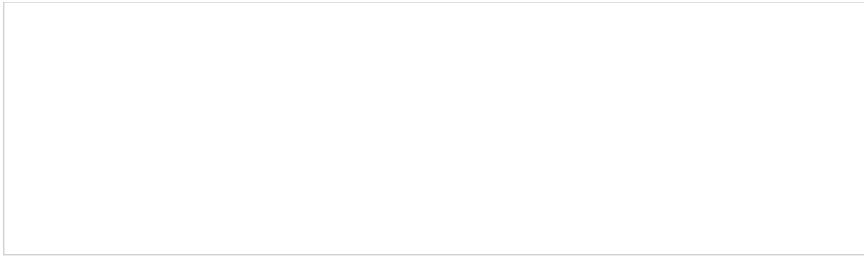
Settings

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.

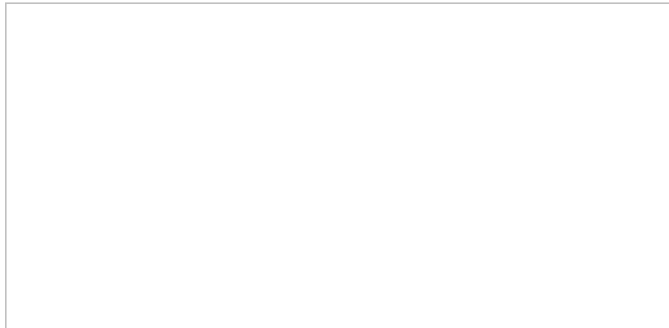
Object type name

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.

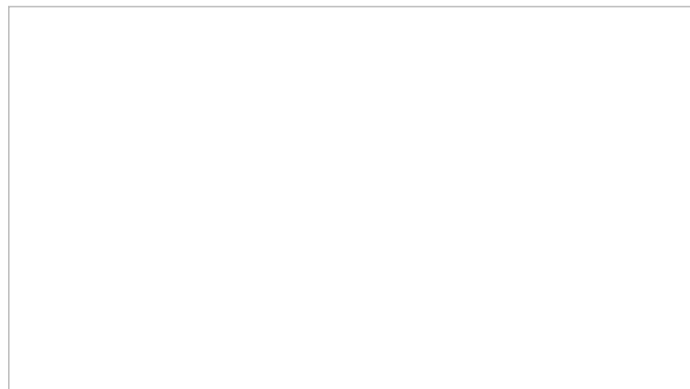
Variable name of object ID

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

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 $\langle recordset_name \rangle.\langle column_name \rangle$. For that statement, the results will be stored in variables *Recordset.id* and *Recordset.name*.

The number of returned records is stored in variable $\langle recordset_name \rangle._count_ (e.g., RS._count_)$. (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.

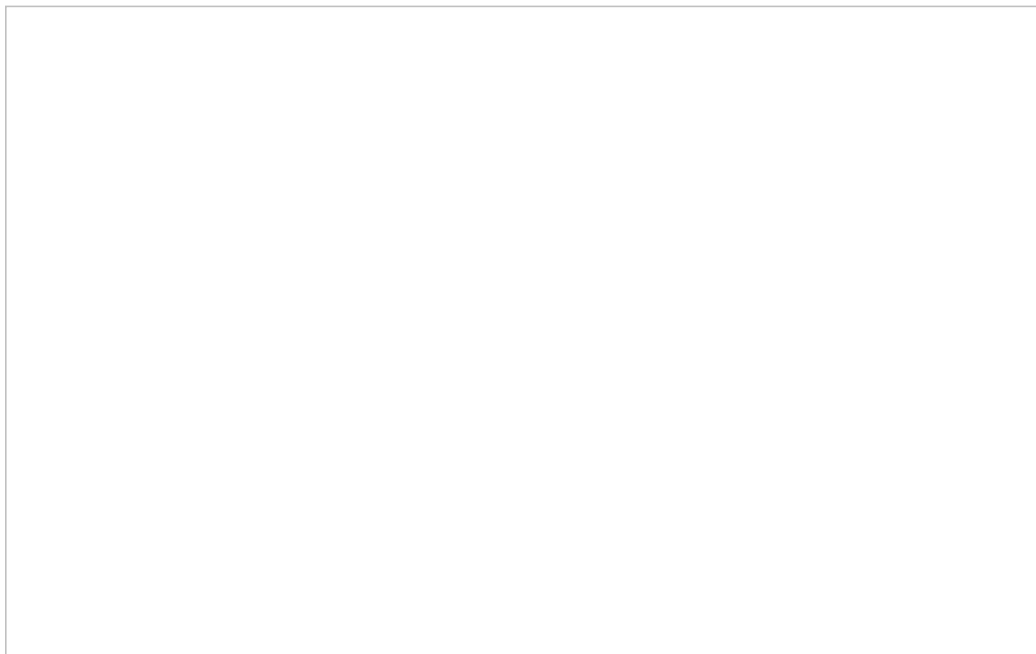
Query language

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)$, respectively. For troubleshooting purposes, use the [E-Mail](#) or [Internal Message](#) block to obtain the content of responses indicating a failed search attempt. For more information, see the description of the variable [\\$\(integrationResultBody\)](#).



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.



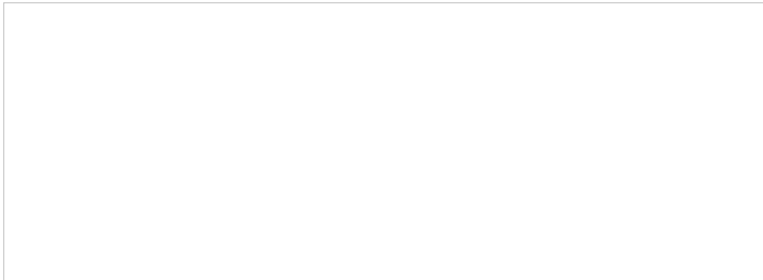
Settings

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.

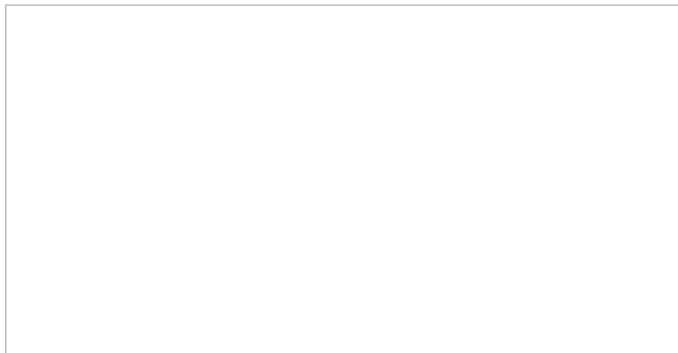
Object ID

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>](#)

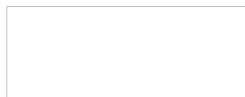
Send Message+

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

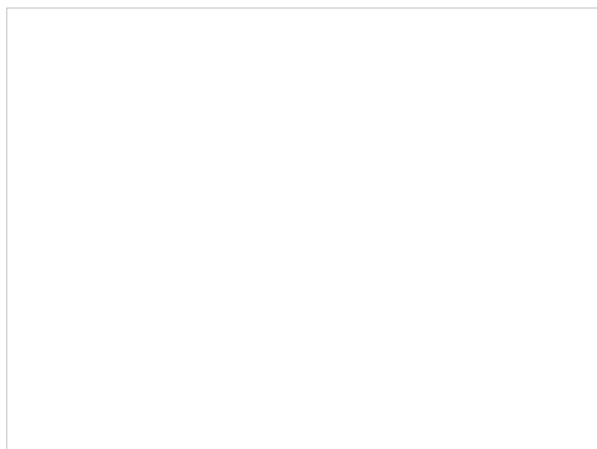
Message is the text of the message to be sent to the customer. Variables in the $\$(varname)$ format can be used in the message text.

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.

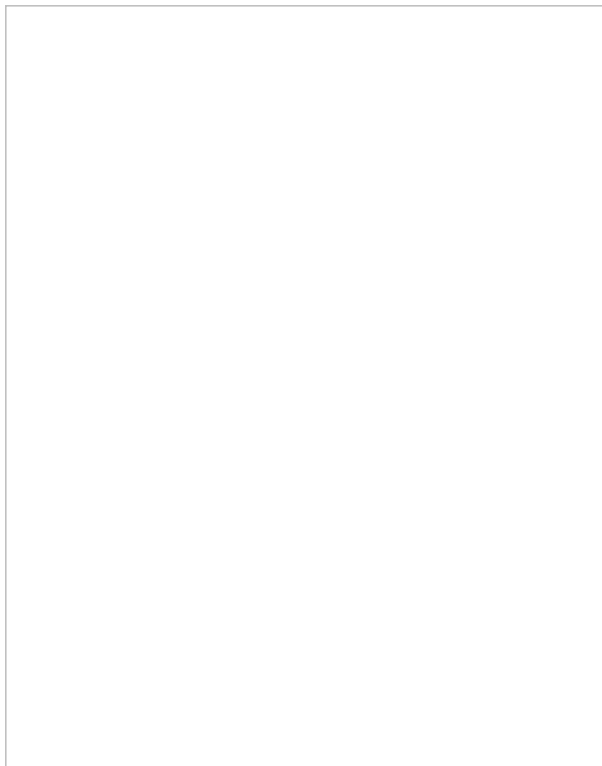
Query

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 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

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 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 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

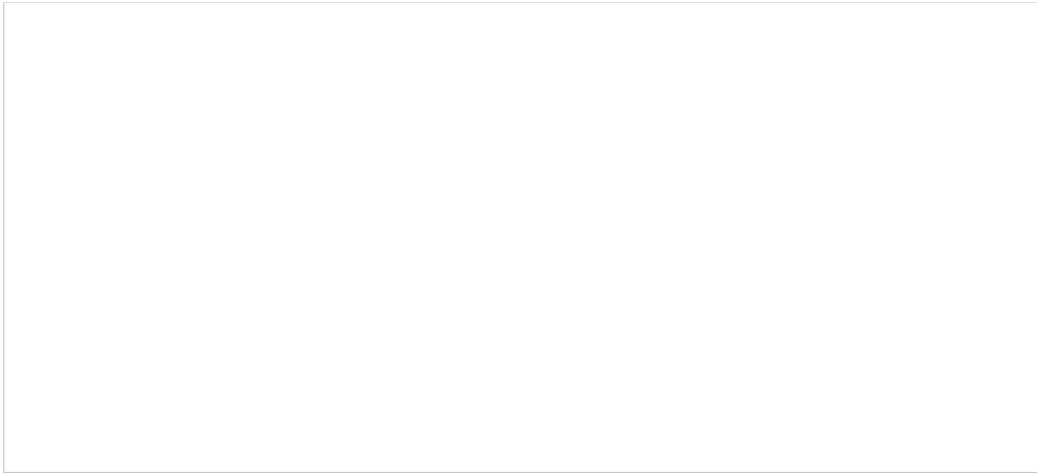
Variable name

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

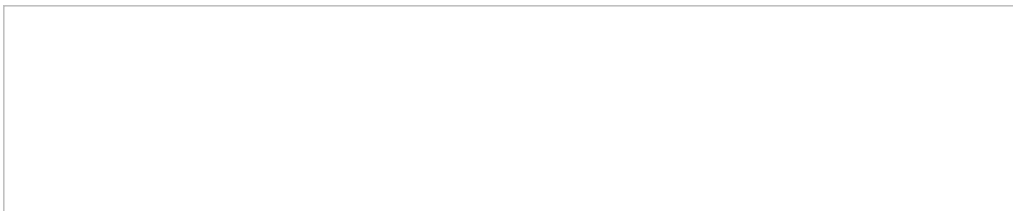
[<Previous](#) | [Next>](#)

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

The *Wait* workflow block will pause workflow execution for a specified number of seconds.

Settings

The *Wait* duration setting is where you specify the number of seconds 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

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

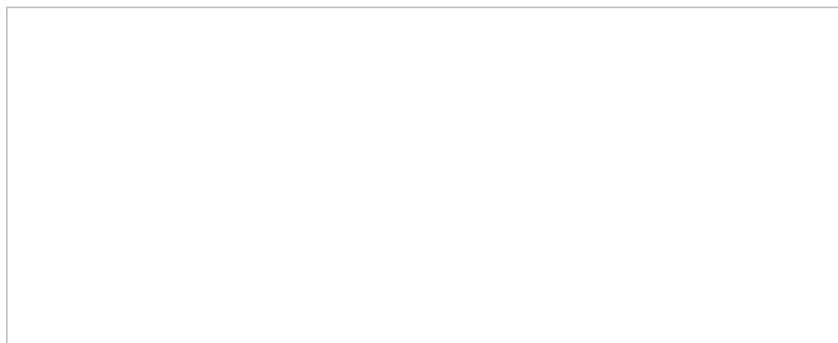
The *Maximum wait duration* setting is where you specify the number of seconds, minutes, hours, or days for which any workflow execution should be paused; the default time is 30 minutes. Decimal fractions are allowed. The maximum value is 15 days.

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 select the specific disposition; once these criteria are met, the wait will end.



Wait+ workflow settings

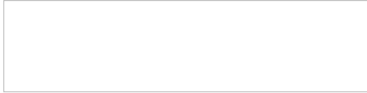
Zapier Invoke Zap

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.



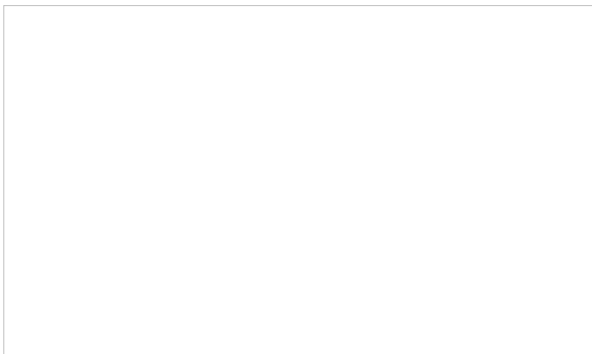
Settings

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 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.

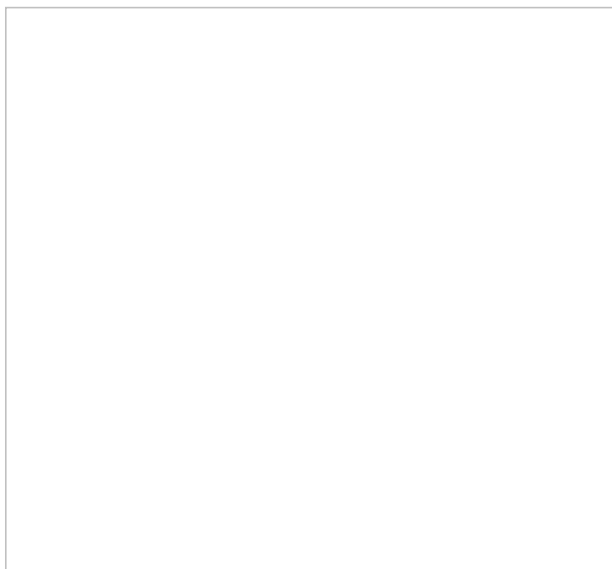
Set fields

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 [E-Mail](#) 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

[<Previous](#) | [Next>](#)

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 `Recordset.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

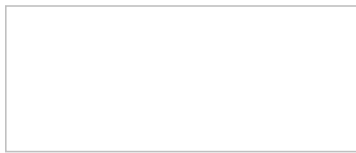
The Zendesk Search block may take one of the following conditional exits: *Failed* or *No Data*.

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.

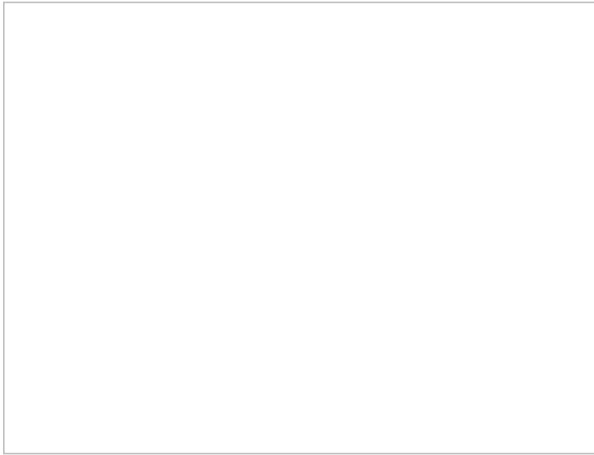
Max results

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 [E-Mail](#) 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.



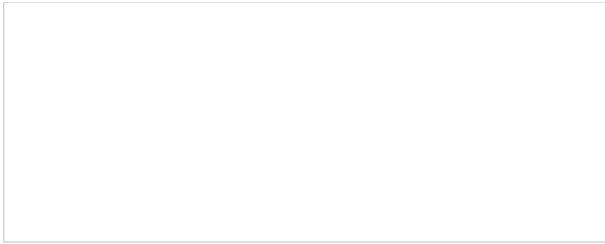
Settings

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.

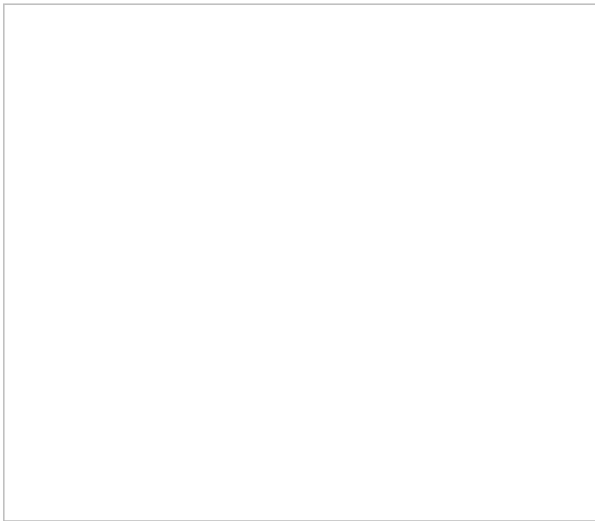
Set fields

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 [E-Mail](#) 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>](#)

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.id)`

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

Note: This variable is available in scenarios.

\$(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.team)

\$(user.team) is the name of the team to which the user is assigned.

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 customer's full name, if available.

\$(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.

[<Previous](#) | [Next>](#)

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., $=2 + \text{"string"}$ 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 Java [SimpleDateFormat](#), 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. \10 - \19 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](#)