

Message Flow Designer Guide

Yeastar P-Series Cloud Edition

Version: 1.0

Date: 2026-09-21



Contents

Message Flow Designer Overview.....	1
Message Flow Setup.....	5
Set up a Message Flow from Template.....	5
Set up a Message Flow from Scratch.....	11
Set up a Message Flow from Existing Flow.....	17
Message Flow Management.....	24
Export Message Flows.....	24
Enable or Disable a Message Flow.....	25
Components Basics.....	26
Message Components.....	26
Send Message.....	26
Send WhatsApp Template Message.....	31
Wait for Reply.....	39
Menu.....	44
Process Components.....	54
Business Hours.....	54
Wait.....	63
Condition.....	67
Loop.....	72
Developer Components.....	81
Internal Data Ops Component.....	81
Email Sender.....	102
Database Access.....	108
HTTP Request.....	115
End Components.....	122
Transfer.....	122
Close.....	128
Expression Basics.....	131
Variables and Functions in Yeastar Expression.....	131

Message Flow Designer Overview

Yeastar Message Flow Designer enables you to easily build intelligent and customizable message routes to automatically perform time-condition routing, auto-replies, database lookups, and more.

Requirements

Item	Requirement
Firmware	84.24.0.122 or later
Plan	Ultimate Plan

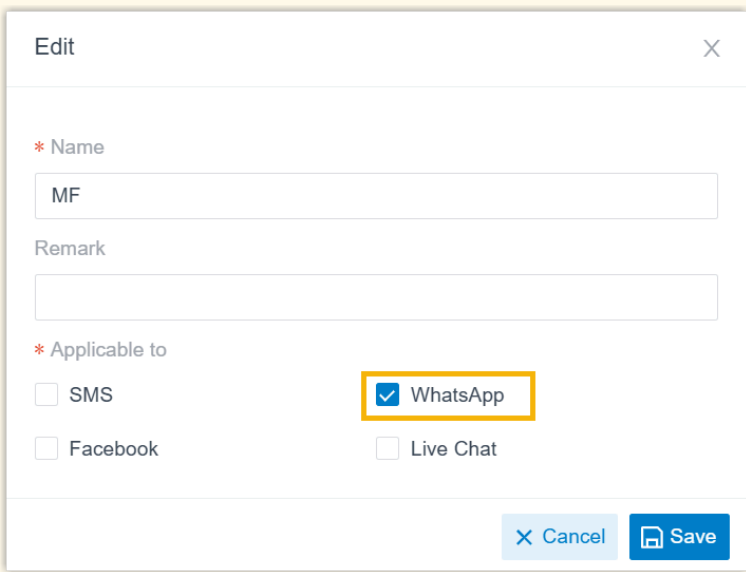
Limitations

- **Maximum number of message flows:** 80
- **Maximum number of components and branches per message flow:** 300
- **Maximum number of steps per message flow:** 50

Message flow creation methods

Creation Method	Description
Create Message Flow by Template	Yeastar P-Series Cloud Edition provides built-in templates that cover common use cases. You can use a ready-made template to build a custom message flow. For more information, see Set up a Message Flow from Template .
Create Message Flow from Scratch	Build a message flow from the ground up, with full control over each component to meet your business needs. For more information, see Set up a Message Flow from Scratch .
Create Message Flow from Existing Flow	Build a message flow by starting from an existing one - either by cloning a message flow or importing a JSON file, and making modifications to fit your needs. For more information, see Set up a Message Flow from Existing Flow .

Supported components

Component	Description
Session Start	Session Start component is the entry point of a message flow. This component is uneditable.
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message.
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates.  Important: This component is available only when the message flow is only applicable to WhatsApp channel type.  For more information, see Send WhatsApp Template Message.
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.

Component	Description
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services. For more information, see HTTP Request.
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

Message flow logs

All message sessions handled by the flow are recorded in [External Chat Logs](#) (Path: **Report and Recordings > External Chat Logs > External Chat Logs**).

Message Flow Setup

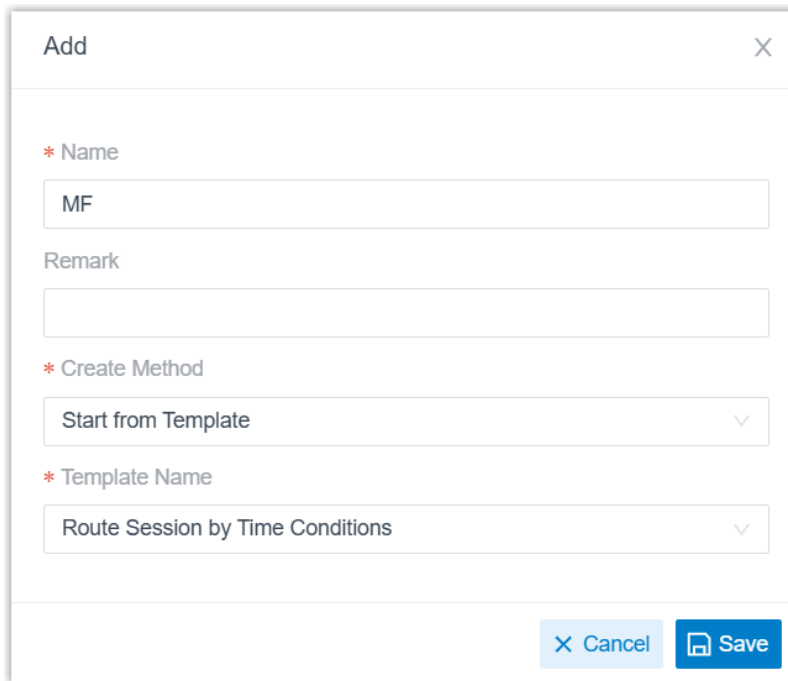
Set up a Message Flow from Template

Yeastar P-Series Cloud Edition provides built-in templates that cover common use cases. You can use a ready-made template to build a custom message flow.

Introduction

Yeastar P-Series Cloud Edition provides the following three templates.

Template	Description	Screenshot
Route Session by Time Template	A message flow allows you to transfer inbound messages to different destinations based on the business hours configured for default time zone.	
Route Session by Conversation Content	A message flow allows you to transfer inbound messages to different destinations based on the conversation content.	



The screenshot shows a modal dialog box titled "Add" with a close button (X) in the top right corner. The dialog contains four labeled input fields, each preceded by a red asterisk (*):

- Name:** A text input field containing the value "MF".
- Remark:** An empty text input field.
- Create Method:** A dropdown menu with "Start from Template" selected and a downward arrow.
- Template Name:** A dropdown menu with "Route Session by Time Conditions" selected and a downward arrow.

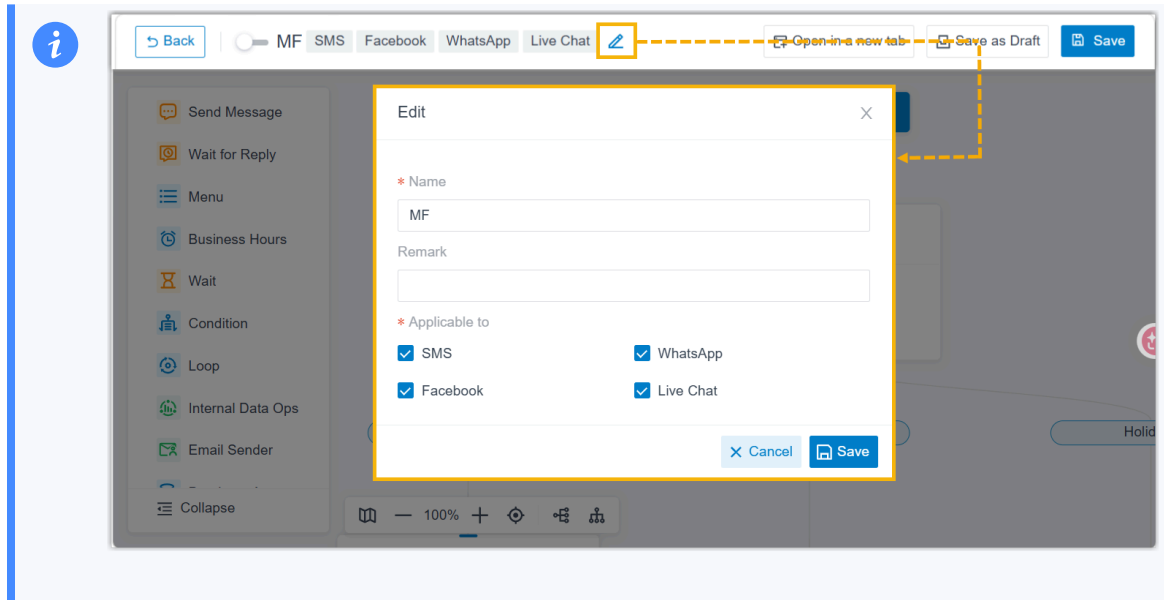
At the bottom right of the dialog, there are two buttons: a light blue "Cancel" button with an 'X' icon and a dark blue "Save" button with a floppy disk icon.

- **Name:** Enter a name to help you identify the message flow.
- **Remark:** Optional. Add a brief description.
- **Create Method:** Select **Start from Template**.
- **Template Name:** Select a template.

The message flow is created, and the webpage is redirected to the configuration page.

**Tip:**

If needed, you can click  at the top of the configuration page to modify the flow basic information, including name, remark, and the applicable channel types.

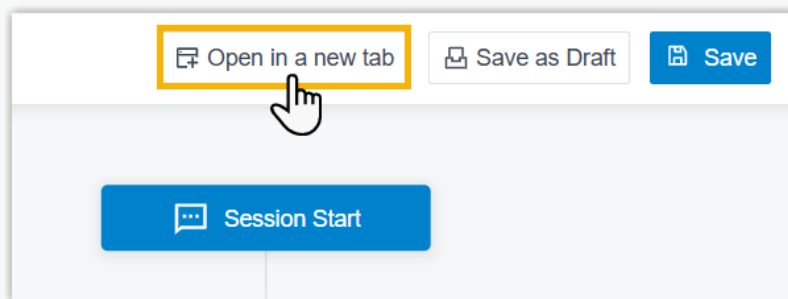


Step 2. Edit components



Tip:

If you need to configure PBX features while editing components, click **Open in a new tab** at the top-right corner to continue editing in a separate tab.



1. Click component to edit its settings.



Note:

To add more components, click **+** on the desired branch, then select a component.

Refer to the table below for introduction and instruction on each component.

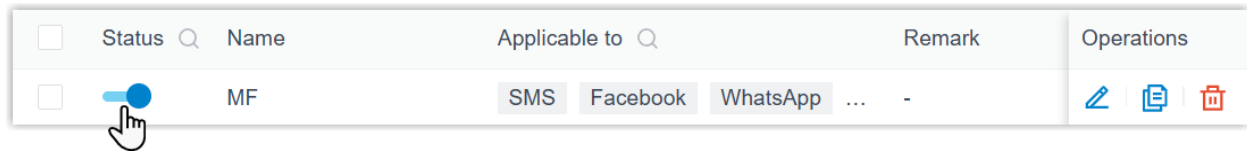
Component	Description
Session Start	Session Start component is the entry point of a message flow. This component is uneditable.
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message .
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates. ! Important: This component is available only when the message flow is only applicable to WhatsApp channel type. Edit × * Name MF Remark * Applicable to ☐ SMS ☒ WhatsApp ☐ Facebook ☐ Live Chat × Cancel Save
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply .
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu .
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day.

Component	Description
	For more information, see Business Hours .
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services. For more information, see HTTP Request.
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

2. At the top-right corner, click **Save**.

Step 3. Enable the message flow

On the message flow list, turn on the switch of the desired message flow and click **OK** in the pop-up window.



What to do next

Go to **Messaging > Message Channel**, and set the message flow as the destination for message channels matching the flow's applicable channel type.

After configuration, inbound messages received via the channels will enter the message flow directly, and the system will automatically perform the routing strategy in the flow.

Set up a Message Flow from Scratch

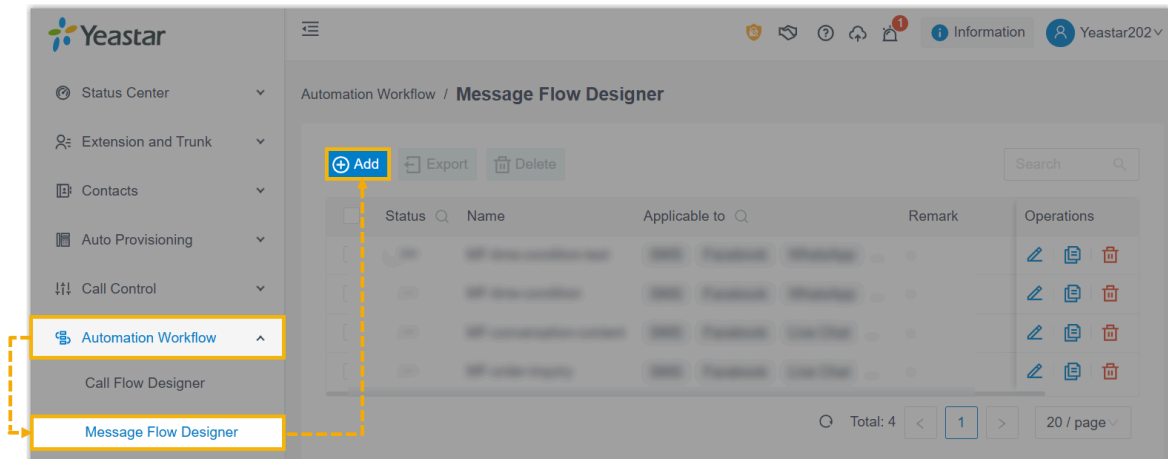
You can build a message flow from the ground up, with full control over each component to meet your business needs.

Procedure

- [Step 1. Create a message flow from scratch](#)
- [Step 2. Add and connect components](#)
- [Step 3. Enable the message flow](#)

Step 1. Create a message flow from scratch

1. Log in to PBX web portal, go to **Automation Workflow > Message Flow Designer**, then click **Add**.



2. In the pop-up window, complete the following settings, then click **Save**.

The 'Add' pop-up window contains the following fields and options:

- Name:** A text input field containing 'MF'.
- Remark:** A text input field.
- Create Method:** A dropdown menu set to 'Start from Scratch'.
- Applicable to:** A group of checkboxes:
 - ☒ SMS
 - ☐ WhatsApp
 - ☐ Facebook
 - ☐ Live Chat

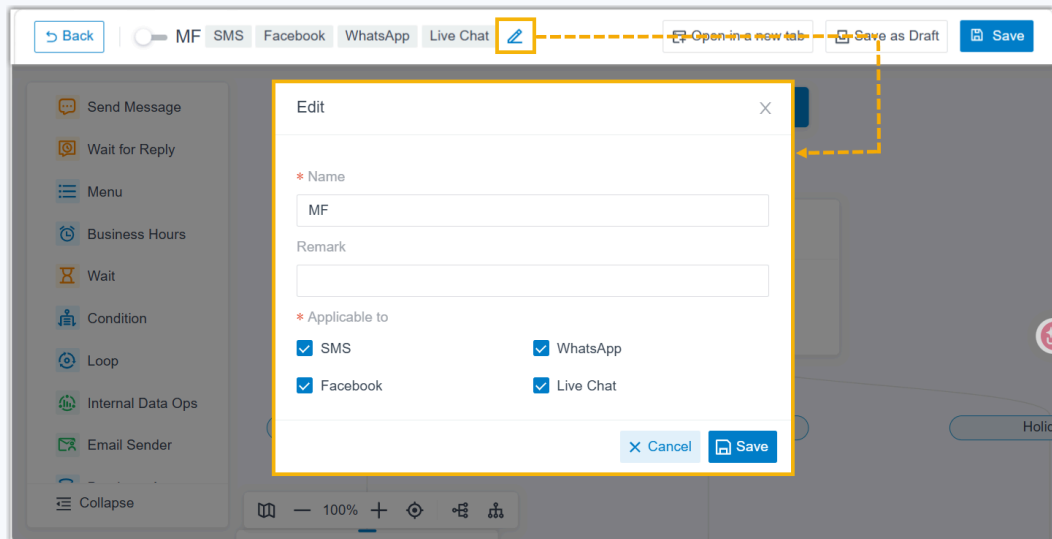
At the bottom right, there are 'Cancel' and 'Save' buttons.

- **Name:** Enter a name to help you identify the message flow.
- **Remark:** Optional. Add a brief description.
- **Create Method:** Select **Start from Scratch**.
- **Applicable to:** Select the channel type to which the message flow is applicable.

The message flow is created, and the webpage is redirected to the configuration page.

**Tip:**

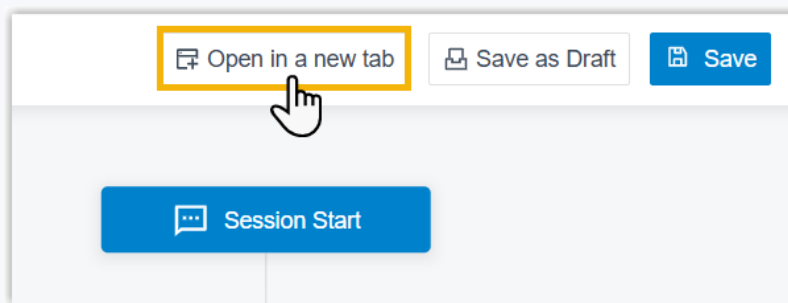
If needed, you can click  at the top of the configuration page to modify the flow basic information, including name, remark, and the applicable channel types.



Step 2. Add and connect components

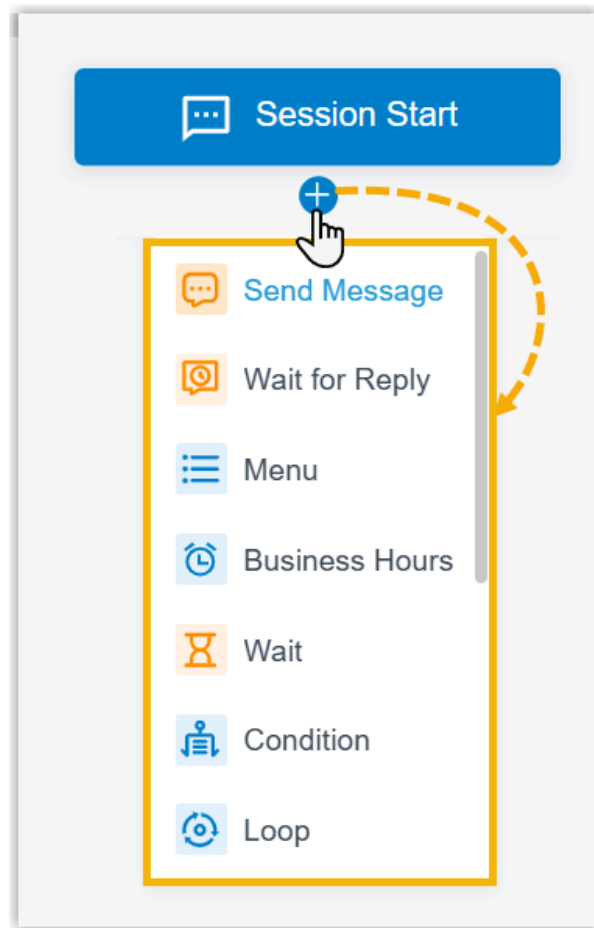
**Tip:**

If you need to configure PBX features while editing components, click **Open in a new tab** at the top-right corner to continue editing in a separate tab.



1. Add components using one of the following methods:

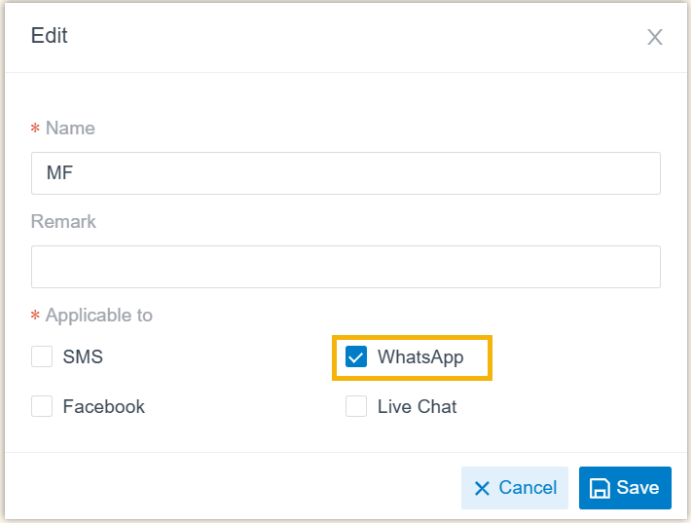
- Click  on an component, then select a desired component.



- From the left component toolbox, click or drag a component to the workspace, then click  and drag to connect components.
2. Click component to edit its settings.

Refer to the table below for introduction and instruction on each component.

Component	Description
Session Start	Session Start component is the entry point of a message flow. This component is uneditable.
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message .
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates.
	 Important:

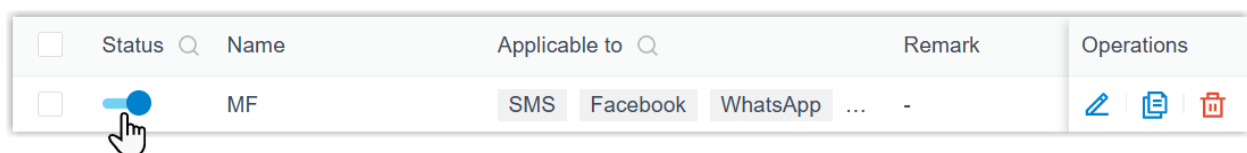
Component	Description
	 This component is available only when the message flow is only applicable to WhatsApp channel type.  For more information, see Send WhatsApp Template Message.
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.

Component	Description
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services. For more information, see HTTP Request.
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

3. At the top-right corner, click **Save**.

Step 3. Enable the message flow

On the message flow list, turn on the switch of the desired message flow and click **OK** in the pop-up window.



What to do next

Go to **Messaging > Message Channel**, and set the message flow as the destination for message channels matching the flow's applicable channel type.

After configuration, inbound messages received via the channels will enter the message flow directly, and the system will automatically perform the routing strategy in the flow.

Set up a Message Flow from Existing Flow

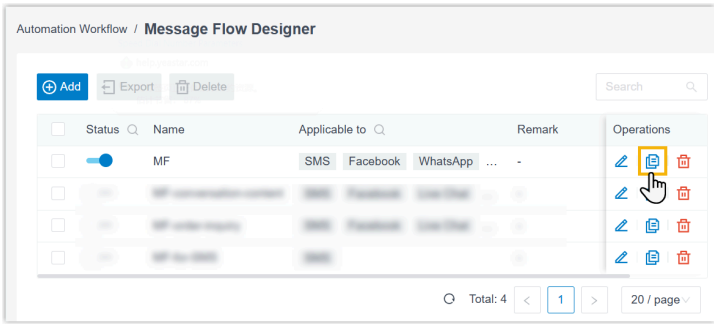
You can build a message flow by starting from an existing one - either by cloning a message flow or importing a JSON file, and making modifications as needed.

Procedure

- [Step 1. Create a message flow from an existing flow](#)
- [Step 2. Edit components](#)
- [Step 3. Enable the message flow](#)

Step 1. Create a message flow from an existing flow

1. Log in to PBX web portal, go to **Automation Workflow > Message Flow Designer**.
2. Create a message flow using one of the following methods.

Method	Instruction
Clone an existing message flow	a. On the right of the desired message flow, click .  b. In the pop-up window, complete the following settings for the new message flow as needed, then click Save.

Method	Instruction
	Add * Name MF(Copy) Remark * Create Method Copy * Applicable to ☒ SMS ☒ WhatsApp ☒ Facebook ☒ Live Chat Cancel Save • Name: Edit the flow name. • Remark: Edit the brief description. • Applicable to: Select the channel type to which the message flow is applicable.
Import an existing message flow (JSON file)	Note: Prepare a JSON file that does not exceed 50 MB. For quick reuse, you can export message flows from another P-Series PBX System. - At the top of the list, click Add. - In the pop-up window, complete the following settings, then click Save.

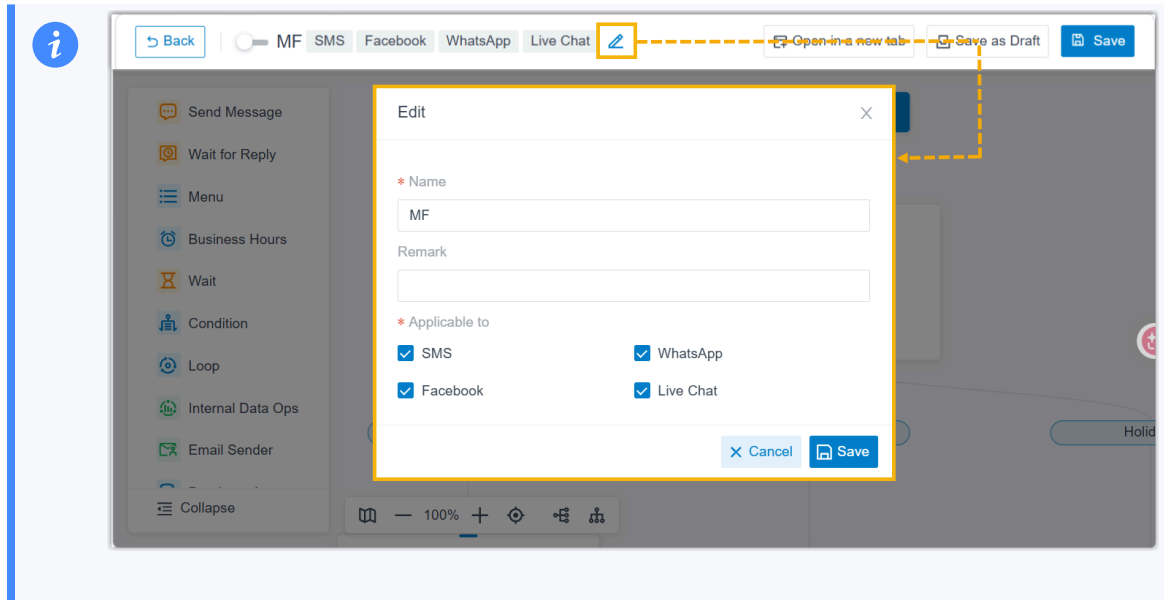
Method	Instruction
	Add * Name MF Remark * Create Method Import from File * Upload File -export-2026080617470 Upload Cancel Save

The message flow is created, and the webpage is redirected to the configuration page.



Tip:

If needed, you can click  at the top of the configuration page to modify the flow basic information, including name, remark, and the applicable channel types.

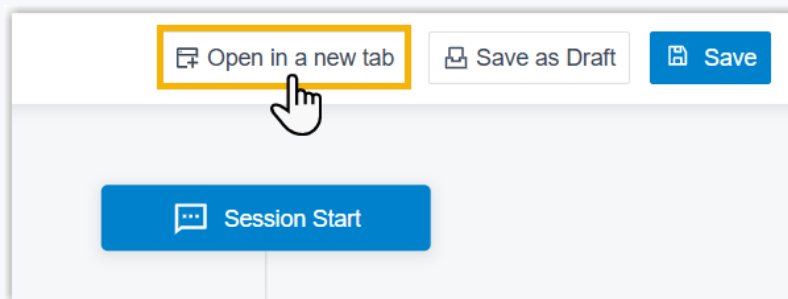


Step 2. Edit components



Tip:

If you need to configure PBX features while editing components, click **Open in a new tab** at the top-right corner to continue editing in a separate tab.



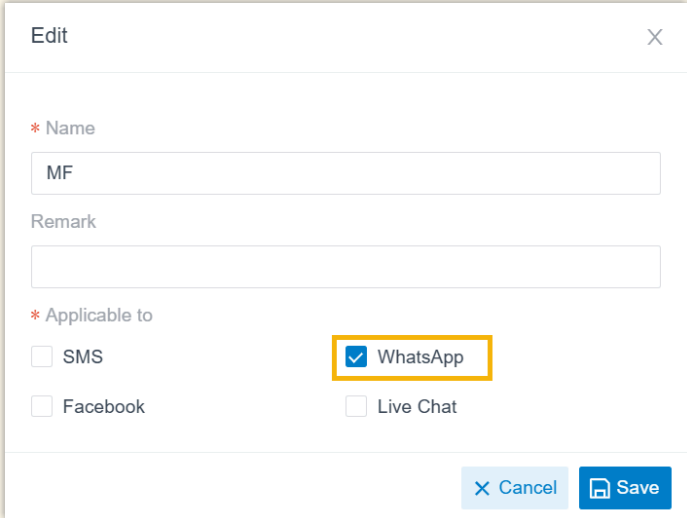
1. Click component to edit its settings.



Note:

To add more components, click **+** on the desired branch, then select a component.

Refer to the table below for introduction and instruction on each component.

Component	Description
Session Start	Session Start component is the entry point of a message flow. This component is uneditable.
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message.
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates.  Important: This component is available only when the message flow is only applicable to WhatsApp channel type.  For more information, see Send WhatsApp Template Message.
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day.

Component	Description
	For more information, see Business Hours .
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services. For more information, see HTTP Request.
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

2. At the top-right corner, click **Save**.

Step 3. Enable the message flow

On the message flow list, turn on the switch of the desired message flow and click **OK** in the pop-up window.

☐	Status	Name	Applicable to	Remark	Operations
☐		MF	SMS Facebook WhatsApp ...	-	

What to do next

Go to **Messaging > Message Channel**, and set the message flow as the destination for message channels matching the flow's applicable channel type.

After configuration, inbound messages received via the channels will enter the message flow directly, and the system will automatically perform the routing strategy in the flow.

Message Flow Management

Export Message Flows

Message flows configured on Yeastar P-Series Cloud Edition can be exported and imported to another P-Series PBX System for quick reuse. This topic describes how to export message flows.

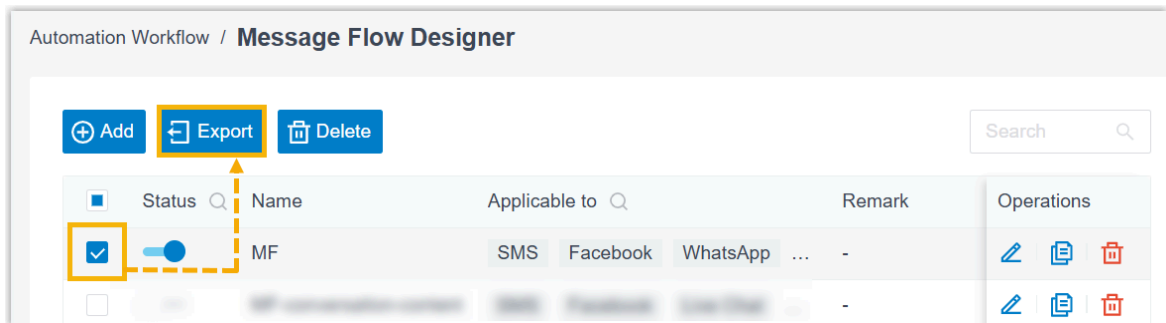


Important:

After exporting message flows, do NOT modify the exported file. Otherwise, the message flow may NOT work when imported into another P-Series PBX system.

Procedure

1. Log in to PBX web portal, go to **Automation Workflow > Message Flow Designer**.
2. Select the checkbox(es) of the message flow(s) that you want to export, then click **Export**.



Result

- If you export a single message flow, it is saved as a `.json` file.
- If you export multiple message flows, they are saved as a `.zip` file. You can unzip the file to access individual `.json` files.



Tip:

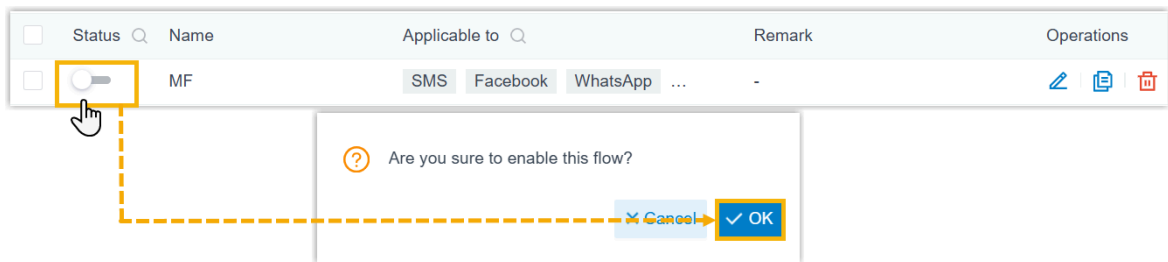
To import the `.json` file to another P-Series PBX System, see [Set up a Message Flow from a JSON File](#).

Enable or Disable a Message Flow

This topic describes how to enable or disable a message flow.

Enable a message flow

1. Log in to PBX web portal, go to **Automation Workflow > Message Flow Designer**.
2. Turn on the switch of the desired message flow, and click **OK** in the pop-up window.



The message flow is enabled, you can set it as the destination for applicable channels to route inbound messages received via the channels to this flow.

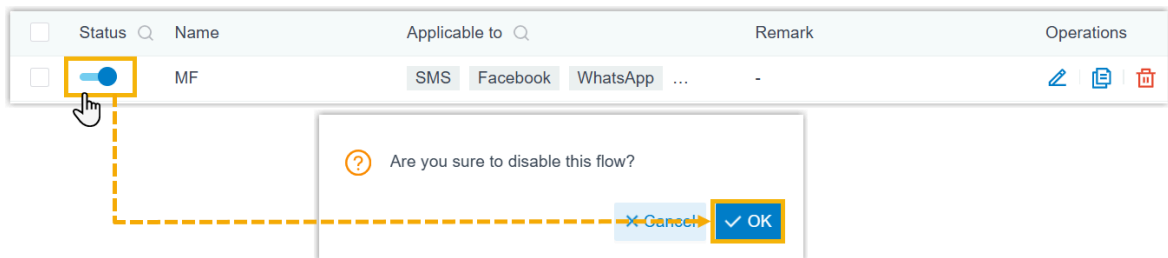
Disable a message flow



Note:

If a message flow has been set as the destination for channels, deleting the flow will render these channels' destination configurations invalid. You must reconfigure new destinations for the affected channels.

1. Log in to PBX web portal, go to **Automation Workflow > Message Flow Designer**.
2. Turn off the switch of the desired message flow, and click **OK** in the pop-up window.



Components Basics

Message Components

Send Message

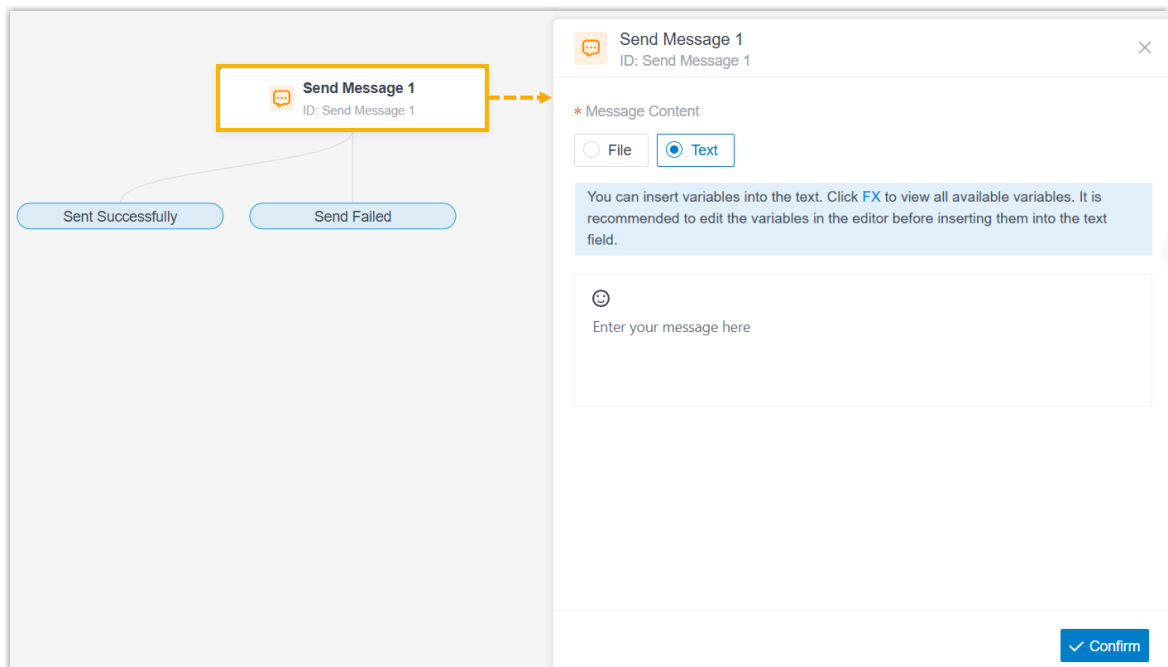
This topic provides an overview of the Send Message component, and describes its configuration as well as supported connections.

Component introduction

Send Message component enables the system to automatically send messages to the customer once the message session is routed to the component.

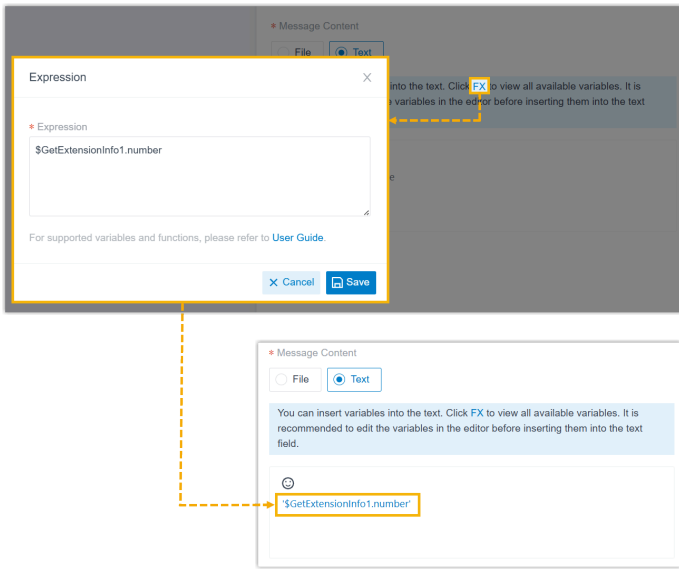
Component configuration

1. After adding **Send Message** component to a message flow, click the component to proceed with the configuration.



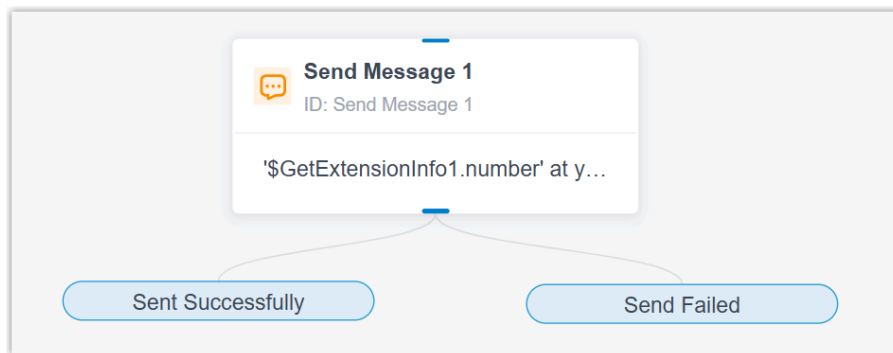
2. In the pop-up right panel, complete the following settings.
 - a. Fill in the message content to be sent to the customer using one of the following methods.

Method	Instruction
File upload	Click File and upload a file that includes the content to be sent.  Note: The file must not exceed 100 MB. 
Text input	Click Text and fill in the content to be sent in the text field. • To insert fixed text, enter plain text and emojis directly. • To insert expression, click FX, enter an expression and click Save.  Note: An expression may include variables, constants, and functions. You can enter FX to view and select system-supported functions, and enter \$ to view and select variables already stored in the current message flow. For more information about functions and variables, see Variables and Functions in Yeastar Expression.

Method	Instruction
	 The expression added in the editor is automatically filled in the text field.

b. Click **Confirm**.

The configured content appears on the component.



3. **Optional:** To route the message session or trigger other actions when the component's branch conditions are met, hover over the desired branch and click  to add the next component.

For more information about components that support to be connected, see [Component connections](#).

Component connections

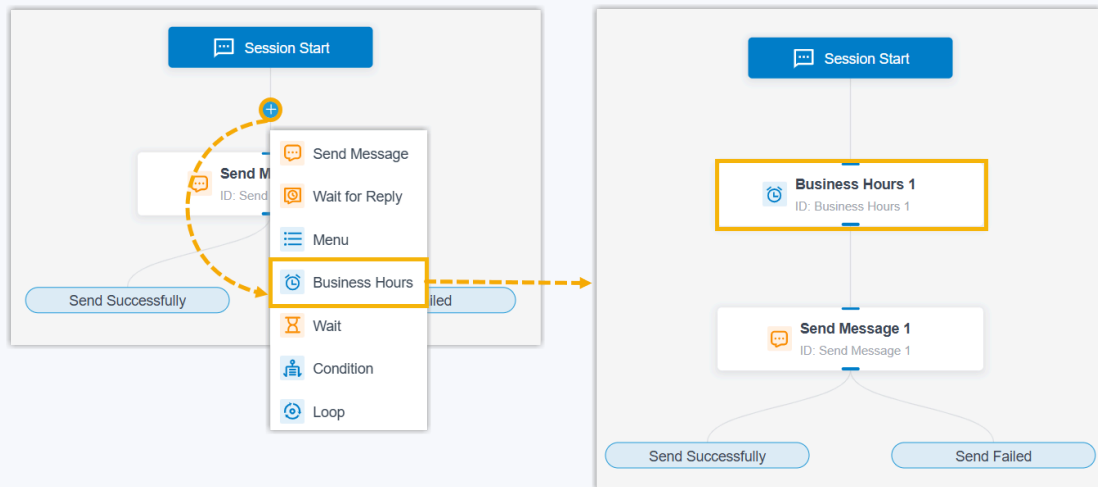
Send Message component comes with two built-in branches **Send Successfully** and **Send Failed**. Each branch can be connected on **one** component, which can be any of the components listed below.



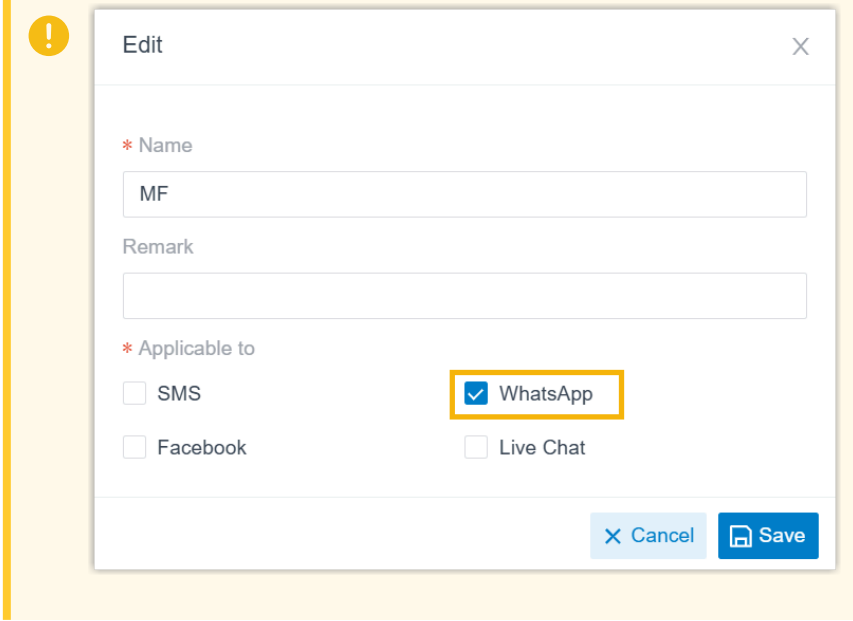
Note:

If a component is already connected to the branch, and you need to insert a new component between them, hover over the line, click **+** and select the target component. The selected component is inserted between them with connections auto-established on both ends, without the need to delete the connection, add the component and reconnect manually.

End components **Transfer** and **Close** are unavailable for this insertion method.



Component	Description
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message.
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates.  Important: This component is available only when the message flow is only applicable to WhatsApp channel type.

Component	Description
	 For more information, see Send WhatsApp Template Message.
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.

Component	Description
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services. For more information, see HTTP Request.
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

Send WhatsApp Template Message

This topic provides an overview of the Send WhatsApp Template Message component, and describes its configuration as well as supported connections.

Component introduction

Send WhatsApp Template Message component enables the system to automatically send template messages to the customer once the message session is routed to the component.



Important:



This component is available only when the message flow is only applicable to **WhatsApp** channel type.

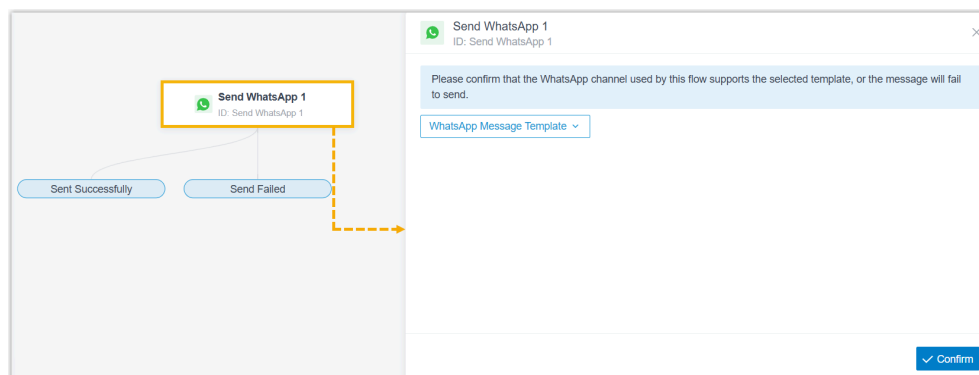
Component configuration

Prerequisites

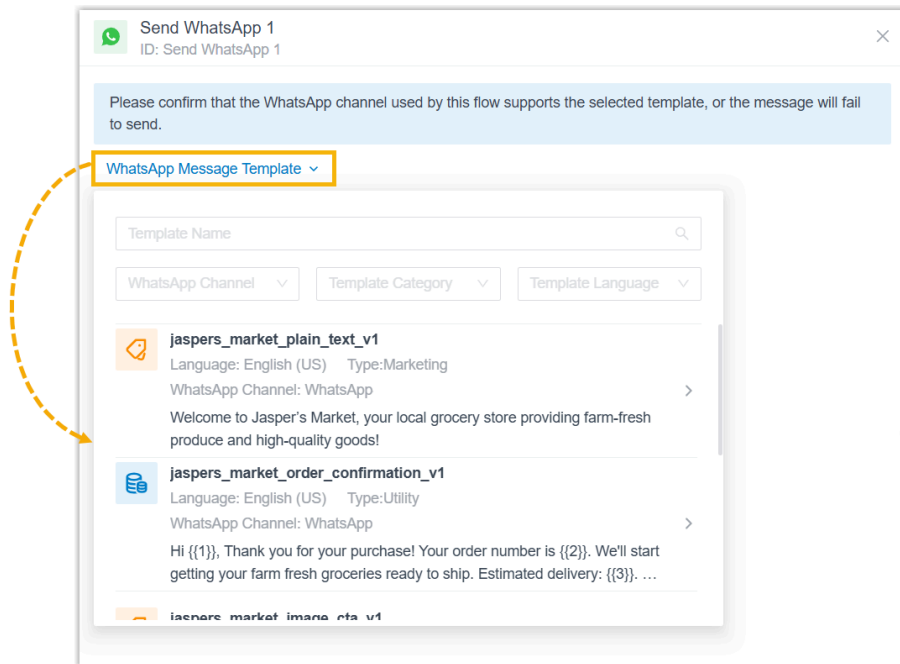
You have [set up at least one WhatsApp messaging channel](#) and [synchronized the desired templates](#) to the PBX.

Procedure

1. 将**Send WhatsApp Template Message** component to a message flow, click the component to proceed with the configuration.

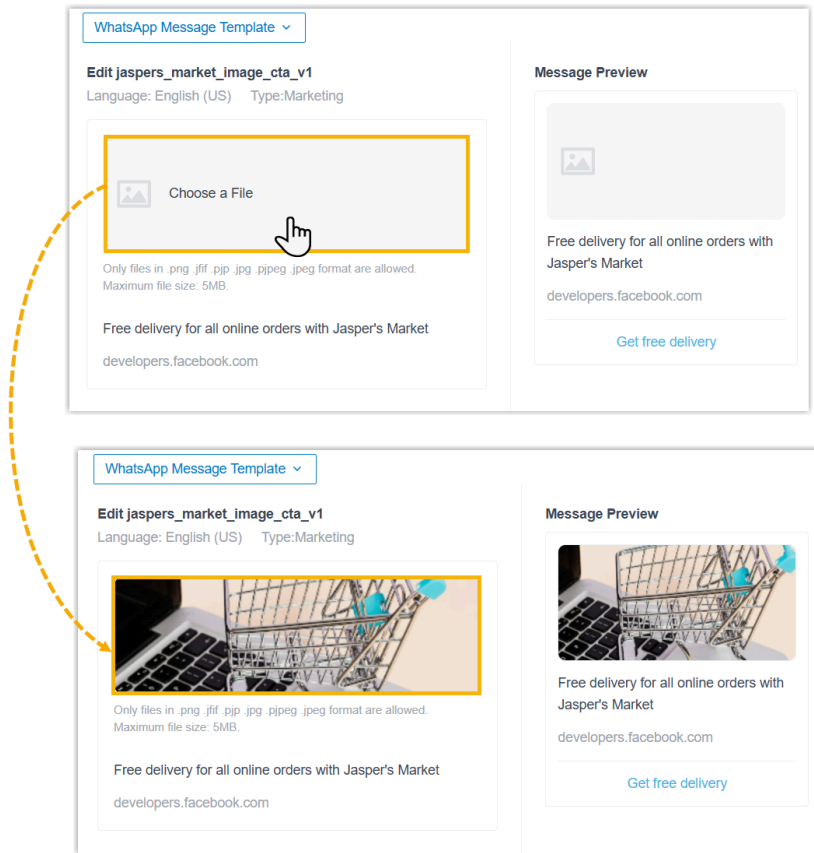


2. In the pop-up right panel, complete the following settings.
 - a. Click **WhatsApp Message Template**, filter and select the desired template from the list.

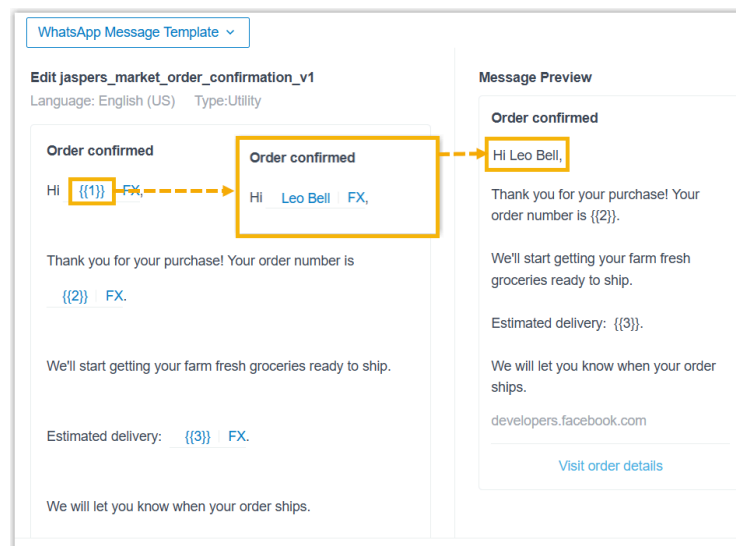


The configuration and preview for the message in select template are displayed on the webpage.

- b. If the template includes media or variable placeholders, upload the required media files and fill in values for all variables.
 - Upload media files.



- Fill in values for all variables.
 - To specify a constant as a variable value, directly click **{{variable}}** and enter plain text.



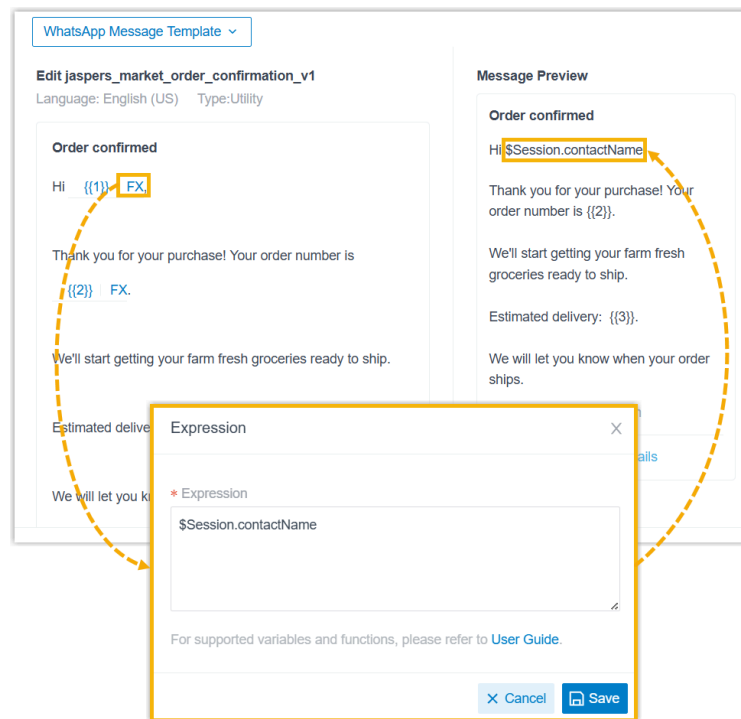
- To specify an expression as a variable value, click **FX** beside the `{{variable}}`, enter an expression and click **Save**.



Note:

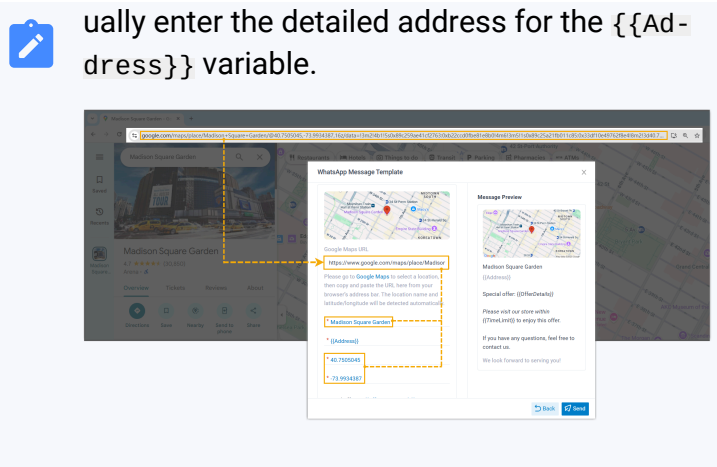
An expression may include variables, constants, and functions. You can enter **FX** to view and select system-supported functions, and enter **\$** to view and select variables already stored in the current message flow.

For more information about functions and variables, see [Variables and Functions in Yeastar Expression](#).



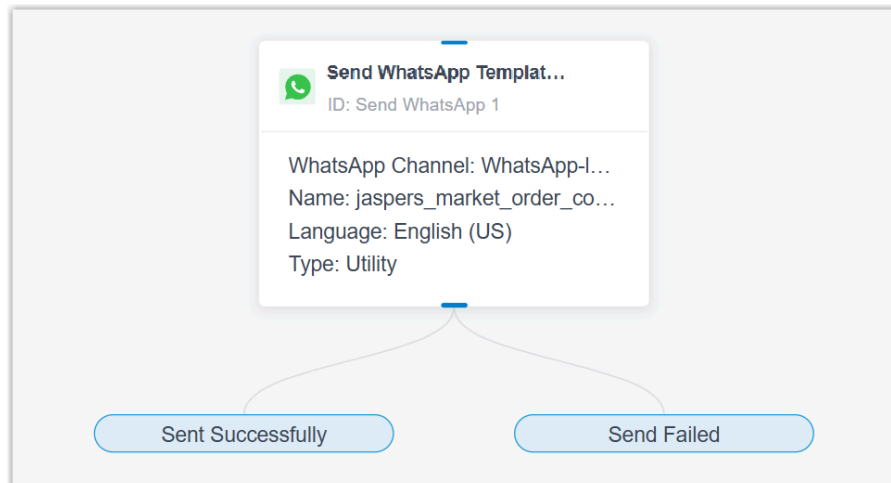
Note:

For variables about location information, you can simply paste the [Google Maps](#) URL, the system will auto-fill the name and coordinates based on the URL. However, you need to man-



c. Click **Confirm**.

The configured template appears on the component.



3. **Optional:** To route the message session or trigger other actions when the component's branch conditions are met, hover over the desired branch and click  to add the next component.

For more information about components that support to be connected, see [Component connections](#).

Component connections

Send WhatsApp Template Message component comes with two built-in branches **Send Successfully** and **Send Failed**. Each branch can be connected on **one** component, which can be any of the components listed below.

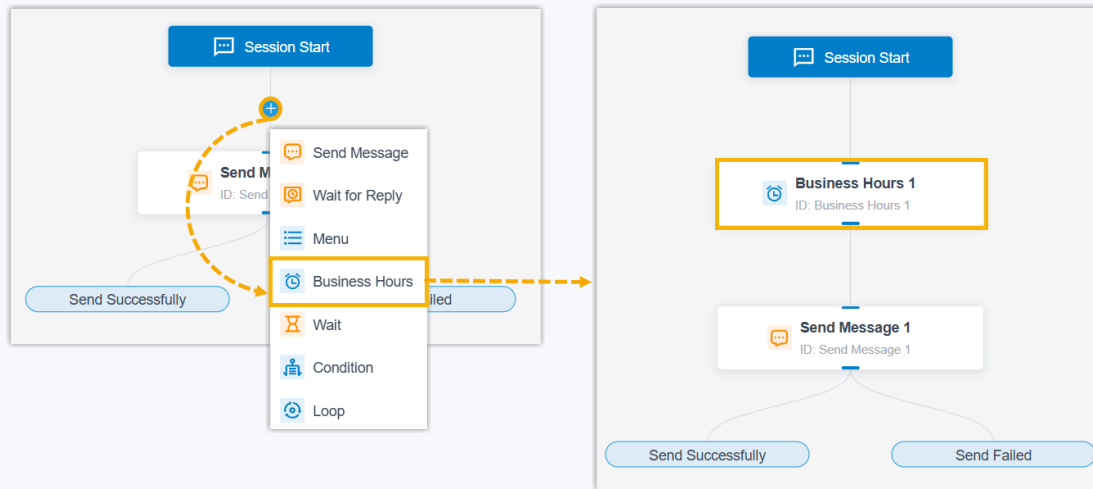


Note:

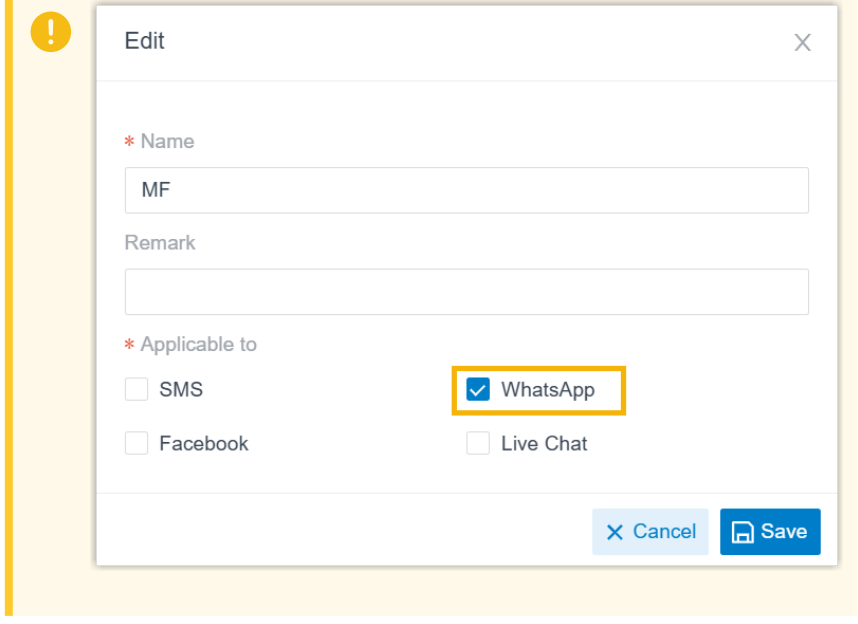


If a component is already connected to the branch, and you need to insert a new component between them, hover over the line, click **+** and select the target component. The selected component is inserted between them with connections auto-established on both ends, without the need to delete the connection, add the component and reconnect manually.

End components **Transfer** and **Close** are unavailable for this insertion method.



Component	Description
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message.
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates. Important: This component is available only when the message flow is only applicable to WhatsApp channel type.

Component	Description
	 For more information, see Send WhatsApp Template Message.
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.

Component	Description
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services. For more information, see HTTP Request.
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

Wait for Reply

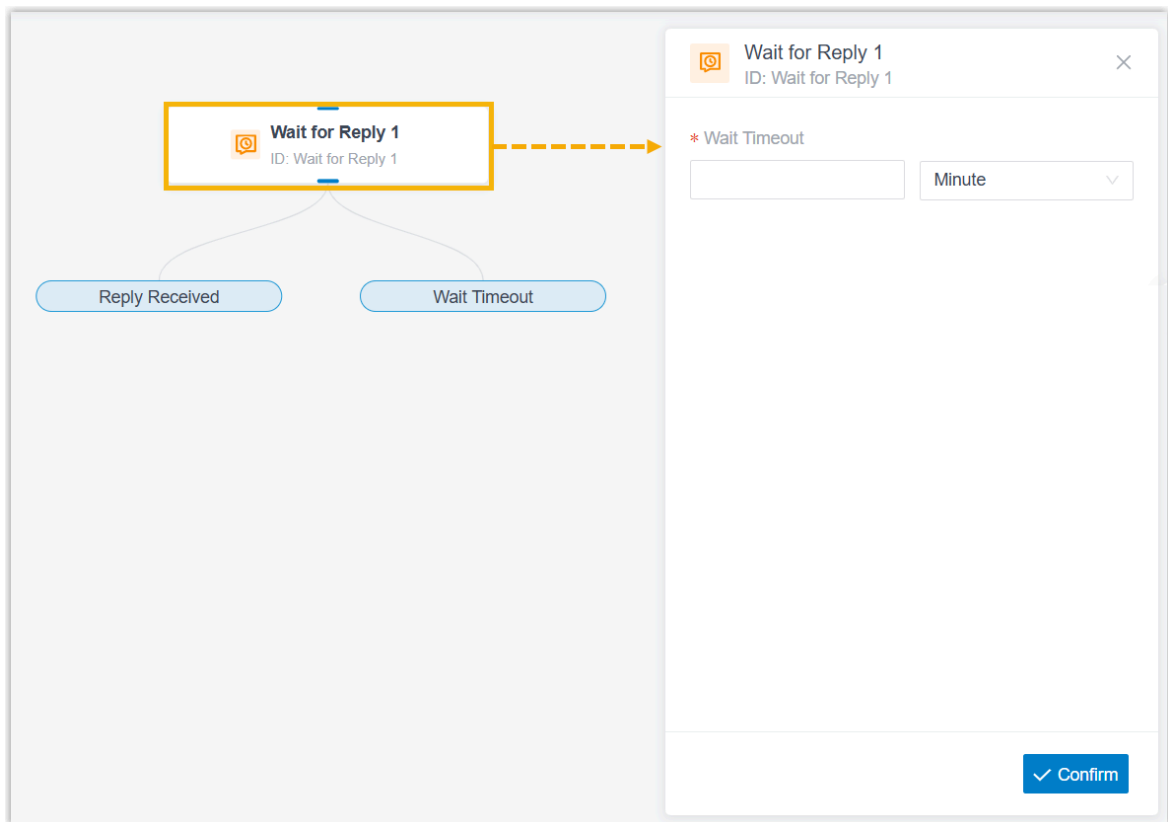
This topic provides an overview of the Wait for Reply component, and describes its configuration, variables, as well as supported connections.

Component introduction

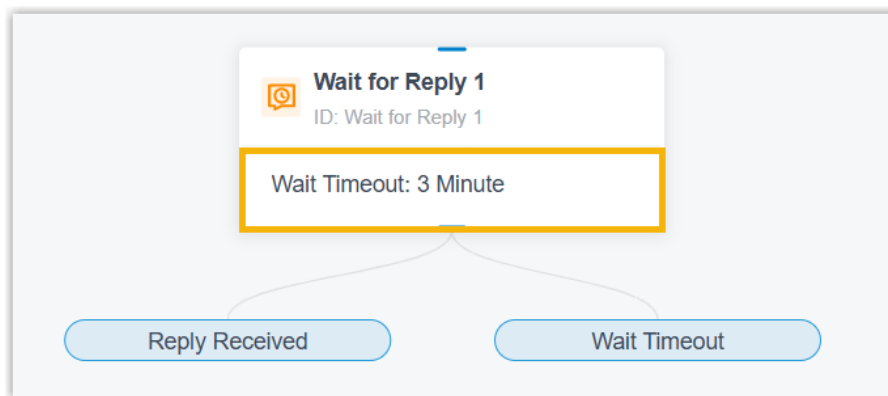
Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. The reply message is stored in variables, which can be used as input for subsequent components or for condition evaluation.

Component configuration

1. After adding **Wait for Reply** component to a message flow, click the component to proceed with the configuration.



2. In the pop-up right panel, configure the wait timeout, and then click **Confirm**.
The configured timeout appears on the component.



3. **Optional:** To route the message session or trigger other actions when the component's branch conditions are met, hover over the desired branch and click  to add the next component.

For more information about components that support to be connected, see [Component connections](#).

Component variables

When a **Wait for Reply** component is added to a message flow, the system stores message content that the customer reply in variables. These variables can be referenced in expression-supported components (such as **Condition**) to retrieve the input value and flexibly route the session.



Note:

Since **Wait for Reply** component can be added multiple times in a message flow, an index is appended to the unique ID of each component (e.g. Wait for Reply 1, Wait for Reply 2) based on the order in which the components are added. To retrieve data from the right component, the component variable must contain the corresponding index.

Variable	Type	Description	Example Value
<code>\$WaitForReply{index}.receivedMessage</code>	String	The message content that the customer replies.	"Transfer to a live agent."

Component connections

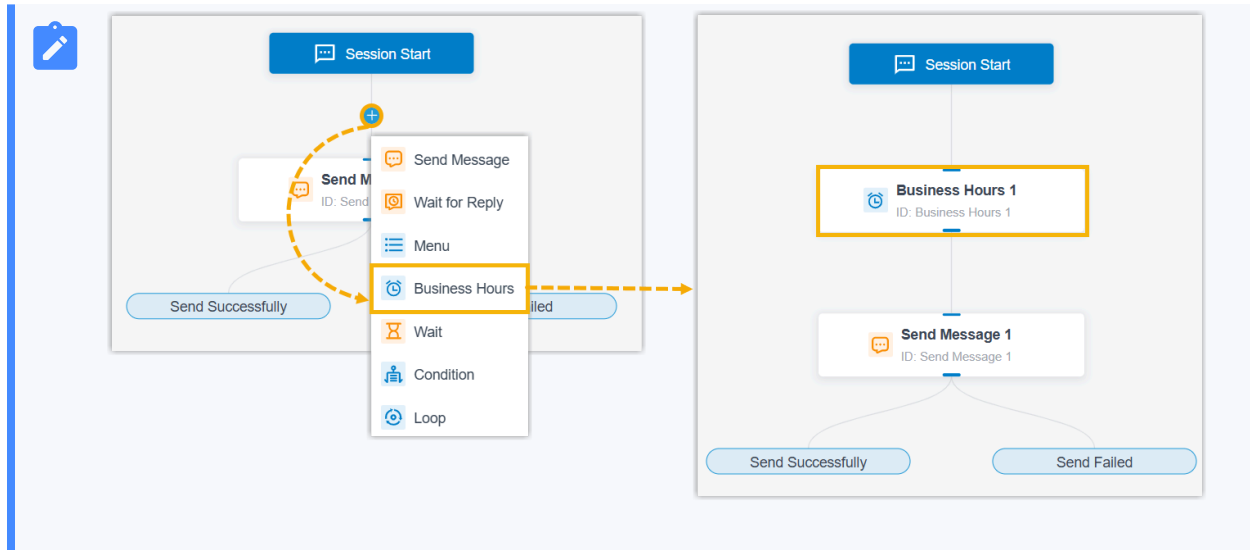
Wait for Reply component comes with two built-in branches **Reply Received** and **Wait Timeout**. Each branch can be connected on **one** component, which can be any of the components listed below.



Note:

If a component is already connected to the branch, and you need to insert a new component between them, hover over the line, click  and select the target component. The selected component is inserted between them with connections auto-established on both ends, without the need to delete the connection, add the component and reconnect manually.

End components **Transfer** and **Close** are unavailable for this insertion method.



Component	Description
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message.
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates. Important: This component is available only when the message flow is only applicable to WhatsApp channel type. Edit * Name MF Remark * Applicable to ☐ SMS ☒ WhatsApp ☐ Facebook ☐ Live Chat Cancel Save

Component	Description
	For more information, see Send WhatsApp Template Message .
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services.

Component	Description
	For more information, see HTTP Request .
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

Menu

This topic provides an overview of the Menu component, and describes its configuration, variables, as well as supported connections.

Component introduction

Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. The reply message is stored in variables, which can be used as input for subsequent components or for condition evaluation.

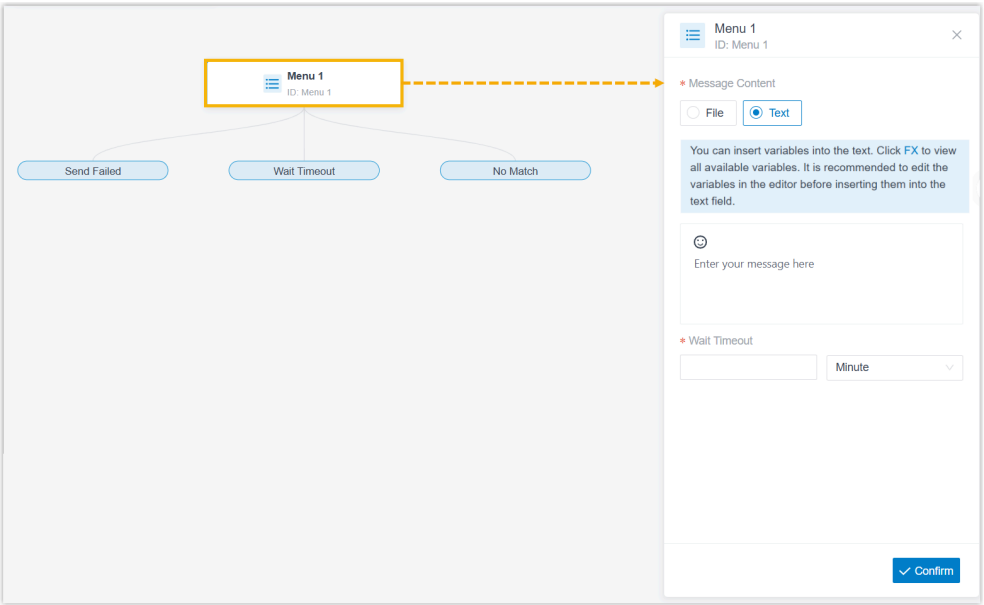
Component configuration

After adding **Menu** component to a message flow, you can configure reply timeout, retry strategies for no match or no input, and keys for menu options.

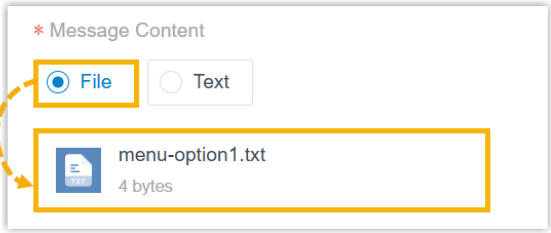
- [Configure menu content and reply timeout](#)
- [Set timeout retry strategy](#)
- [Set retry strategy for no match](#)
- [Set keys for menu options](#)
- [\(Optional:\) Add next components on branches](#)

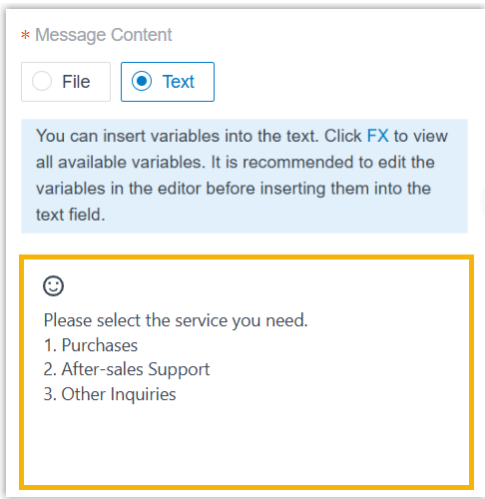
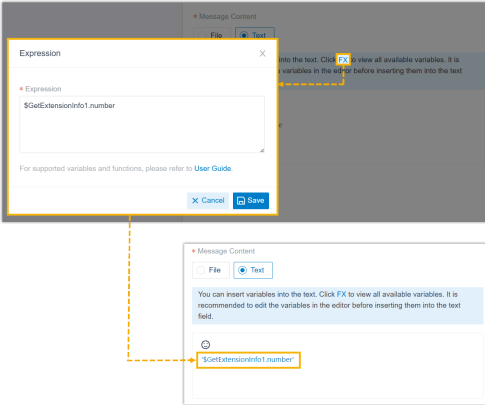
Configure menu content and reply timeout

1. Click **Menu** component to proceed with the configuration.



2. In the pop-up right panel, complete the following settings.
- a. Configure menu content to be send to the customer using one of the following methods.

Method	Instruction
File upload	Click File and upload a file that includes the content to be sent.  Note: The file must not exceed 100MB. 
Text input	Click Text and fill in the content to be sent in the text field. • To insert fixed text, enter plain text and emojis directly.

Method	Instruction
	 * Message Content ☐ File ☒ Text You can insert variables into the text. Click FX to view all available variables. It is recommended to edit the variables in the editor before inserting them into the text field. Please select the service you need. 1. Purchases 2. After-sales Support 3. Other Inquiries
	To insert expression, click FX, enter an expression and click Save.
	 Note: An expression may include variables, constants, and functions. You can enter FX to view and select system-supported functions, and enter \$ to view and select variables already stored in the current message flow. For more information about functions and variables, see Variables and Functions in Yeastar Expression.
	 Expression * Expression \$GetExtensionInfo1.number For supported variables and functions, please refer to User Guide. Cancel Save * Message Content ☐ File ☒ Text You can insert variables into the text. Click FX to view all available variables. It is recommended to edit the variables in the editor before inserting them into the text field. ☐ \$GetExtensionInfo1.number

Method	Instruction
	The expression added in the editor is automatically filled in the text filed.

b. Set wait timeout.

* Wait Timeout

5

Minute

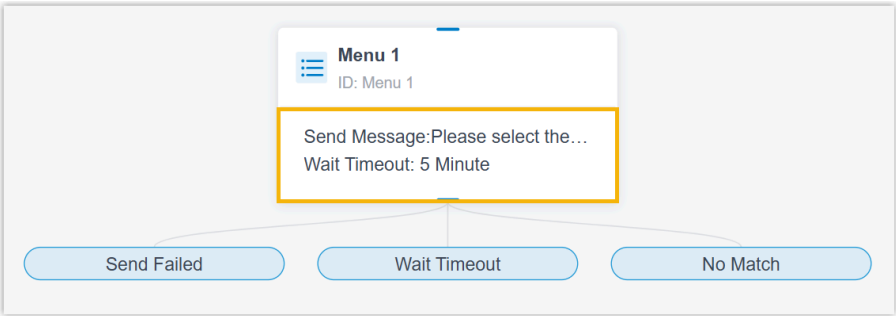
Day

Hour

Minute

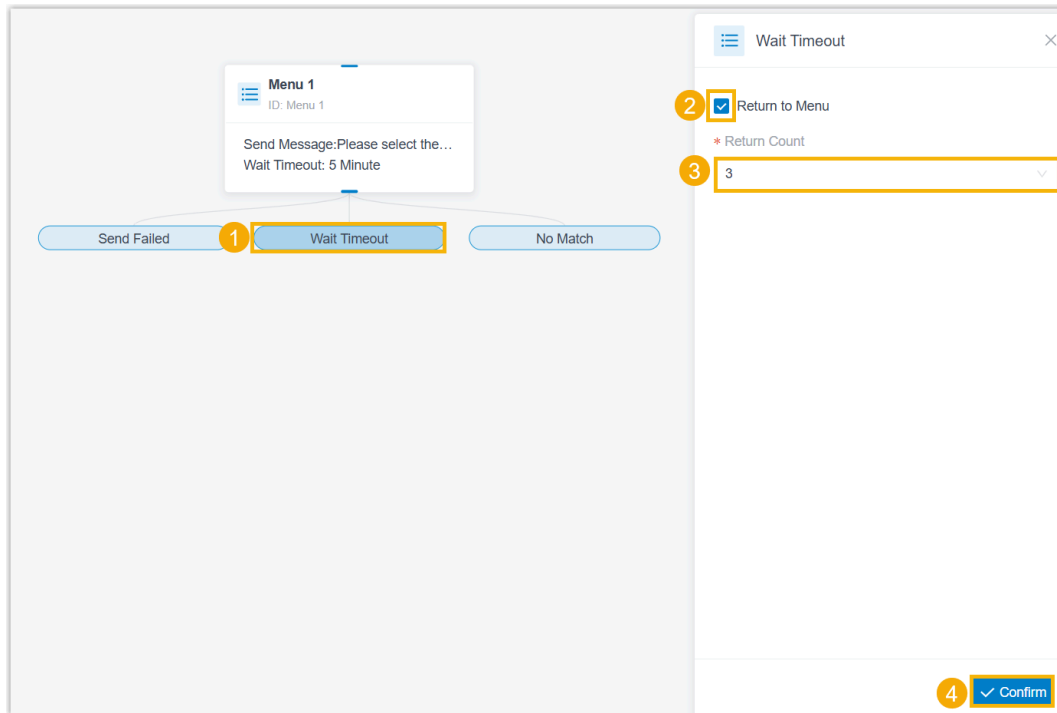
c. Click **Confirm**.

The configured content and timeout appears on the component.



Set timeout retry strategy

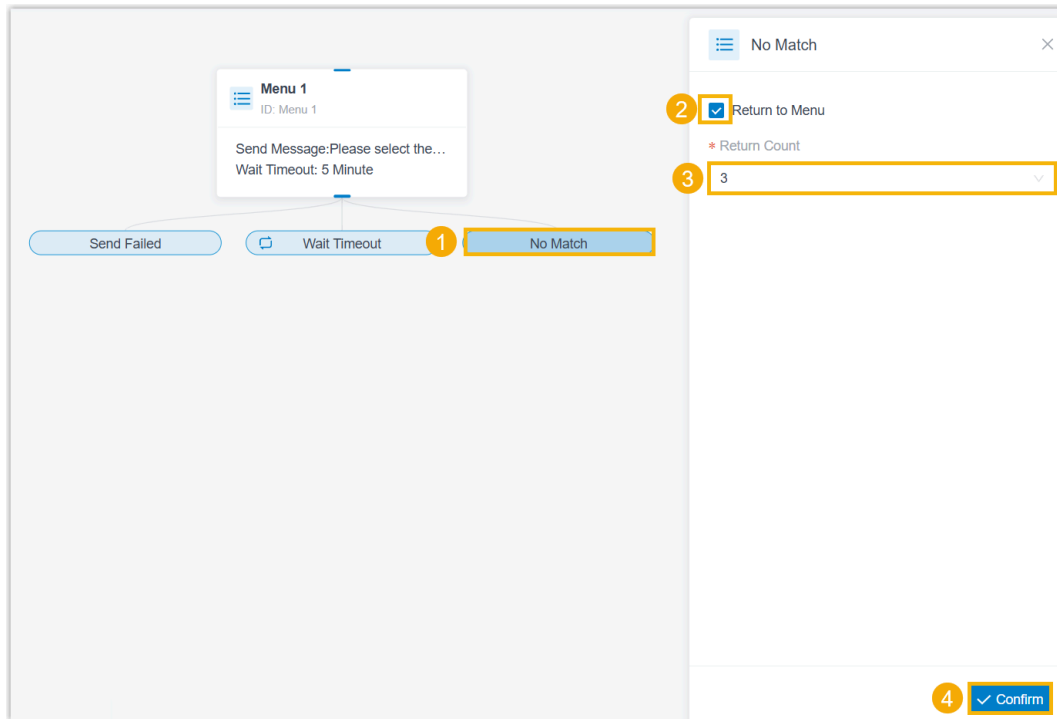
To allow the customer to return to the component for a retry upon timeout, complete the following settings.



1. Click **Wait Timeout** branch.
2. Select the checkbox of **Return to Menu**.
3. In the **Return Count** drop-down list, select how many times the customer can return to the component upon timeout.
4. At the bottom-right corner, click **Confirm**.

Set retry strategy for no match

To allow the customer to return to the component for a retry when no match is found, complete the following settings.



1. Click **No Match** branch.
2. Select the checkbox of **Return to Menu**.
3. In the **Return Count** drop-down list, select how many times the customer can return to the component upon timeout.
4. At the bottom-right corner, click **Confirm**.

Set keys for menu options

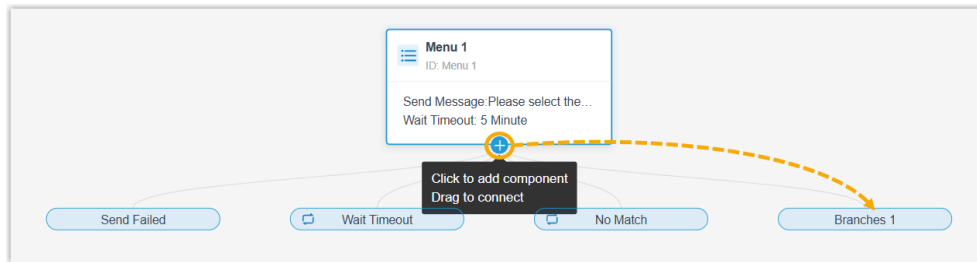
At least one key is required for **Menu** component.

1. Click  on the component to add a branch for a key.

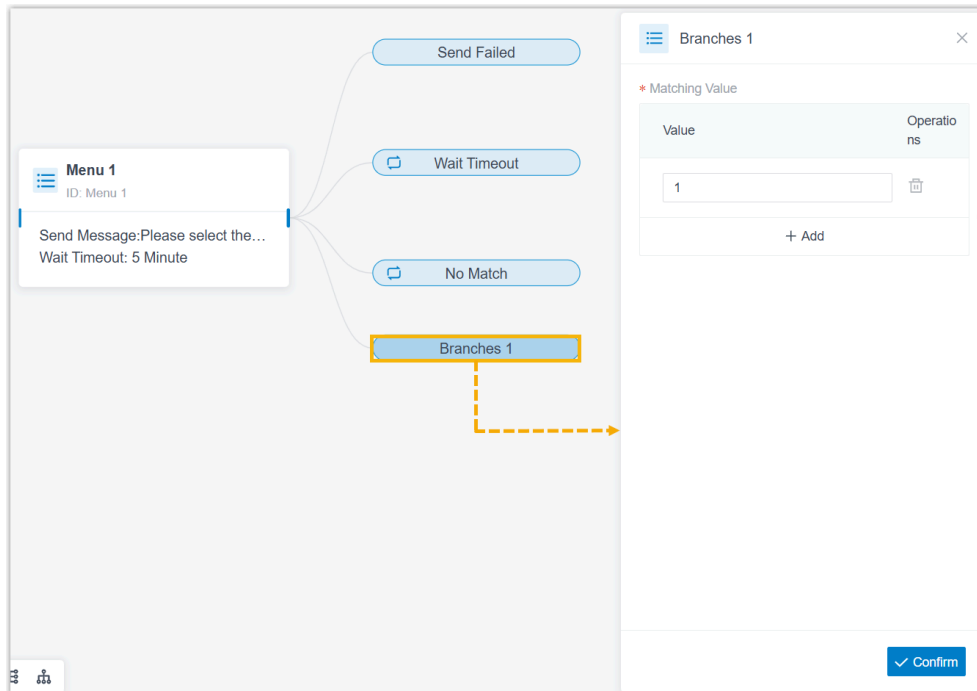


Note:

You can configure up to **100 branches**, each mapped to a different key.



2. Click the added branch to proceed with the configuration.



3. In the pop-up right panel, enter a value in the **Value** column and click **Confirm**.



Note:

To allow multiple values to enter this branch and trigger its subsequent strategy, click **Add** and enter the other desired values. Each branch supports up to 20 different values.

(Optional:) Add next components on branches

To route the message session or trigger other actions when the branch conditions are met, hover over the desired branch and click  to add the next component.

For more information about components that support to be connected, see [Component connections](#).

Component variables

When a **Menu** component is added to a message flow, the system stores message content that the customer reply in variables. These variables can be referenced in expression-supported components (such as **Condition**) to retrieve the input value and flexibly route the session.



Note:

Since **Menu** component can be added multiple times in a message flow, an index is appended to the unique ID of each component (e.g. Menu 1, Menu 2) based on the order in which the components are added. To retrieve data from the right component, the component variable must contain the corresponding index.

Variable	Type	Description	Example Value
\$Menu{ <i>index</i> }.receivedMessage	String	The message content that the customer sends in response to the menu.	"1"

Component connections

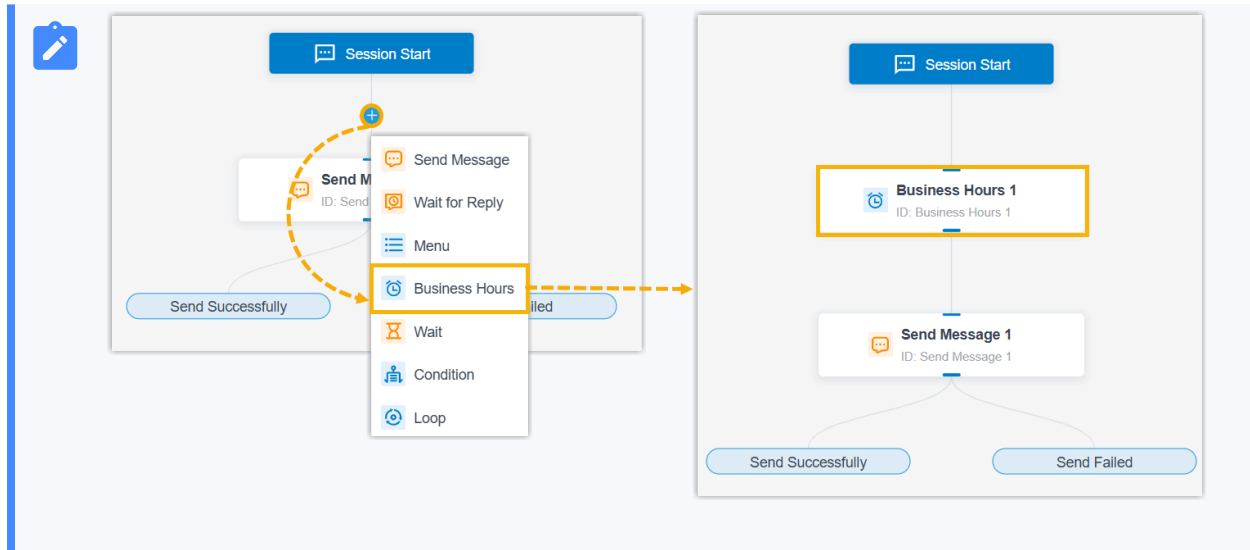
Menu component comes with three built-in branches **Send Failed**, **Wait Timeout** and **No Match**, and supports up to 100 additional key branches. Each branch can be connected to **one** component, which can be any of the components listed below.



Note:

If a component is already connected to the branch, and you need to insert a new component between them, hover over the line, click  and select the target component. The selected component is inserted between them with connections auto-established on both ends, without the need to delete the connection, add the component and reconnect manually.

End components **Transfer** and **Close** are unavailable for this insertion method.



Component	Description
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message.
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates. Important: This component is available only when the message flow is only applicable to WhatsApp channel type. Edit * Name MF Remark * Applicable to ☐ SMS ☒ WhatsApp ☐ Facebook ☐ Live Chat Cancel Save

Component	Description
	For more information, see Send WhatsApp Template Message .
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services.

Component	Description
	For more information, see HTTP Request .
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

Process Components

Business Hours

This topic provides an overview of the Business Hours component, and describes its configuration as well as supported connections.

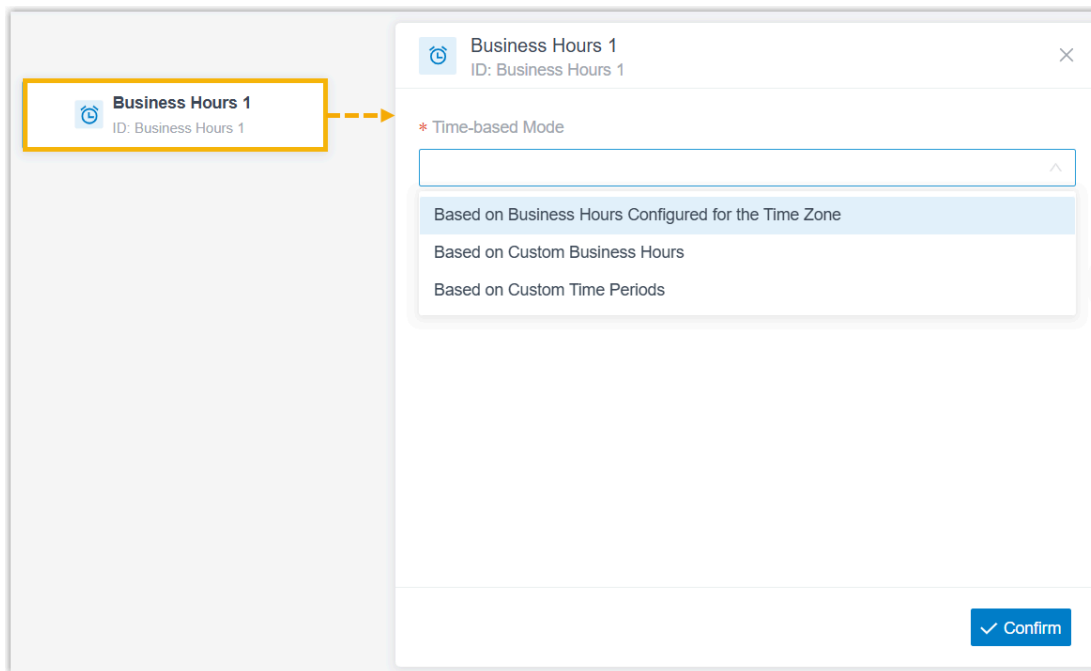
Component introduction

Business Hours component allows you to route the message session to different destinations based on the time of day. A variety of time-based modes listed in the table below are available to flexibly define office hours and implement time-based message routing.

Type	Description
Business Hours Configured for Time Zone	Use pre-configured business hours for a specific time zone.  Note: This requires you to configure business hours in advance. To configure this, go to Call Control > Business Hours and Holidays.
Custom Business Hours	Create custom business hours based on a specific time zone to specify uniform or varied schedule through the week, or full time (24x7) schedule.
Custom Time Periods	Create custom time periods based on a specific time zone to accommodate granular scheduling needs.

Component configuration

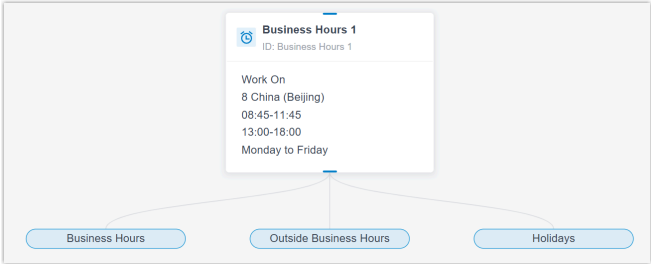
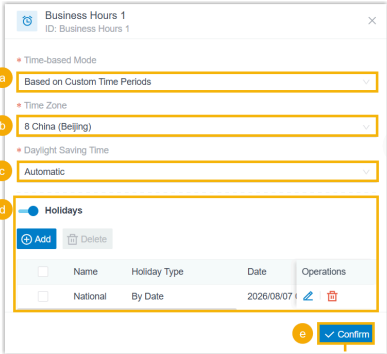
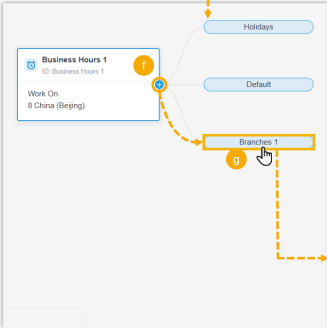
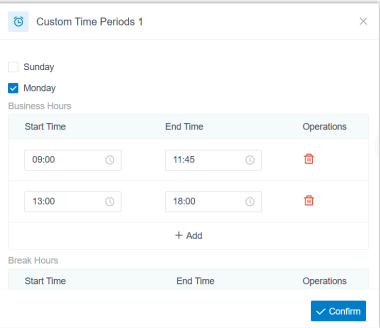
1. 将 **Business Hours** component to a message flow, click **Business Hours** component to proceed with the configuration.

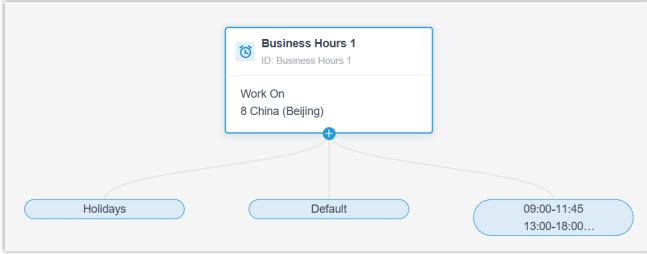


2. In the pop-up wight panel, select a time-based mode according to your business schedule, then configure its related settings.

Scenario	Instruction
	 Business Hours 1 ID: Business Hours 1 Work On Default Time Zone [8 China (Beijing)] Business Hours Outside Business Hours Holidays

Scenario	Instruction																							
Use Custom Business Hours	a. In the Time-based Mode drop-down list, select Based on Custom Business Hours. b. In the Time Zone drop-down list, select a time zone. c. In the Daylight Saving Time drop-down list, select an option to enable or disable the feature. d. In the Business Hours section, select Same Business Hours (Monday to Friday). Configure the business hours table as follows: <table border="1"> <thead> <tr> <th>Start Time</th> <th>End Time</th> <th>Operations</th> </tr> </thead> <tbody> <tr> <td>08:45</td> <td>11:45</td> <td></td> </tr> <tr> <td>13:00</td> <td>18:00</td> <td></td> </tr> </tbody> </table> Below the business hours table, click + Add. Configure the break hours table as follows: <table border="1"> <thead> <tr> <th>Start Time</th> <th>End Time</th> <th>Operations</th> </tr> </thead> <tbody> <tr> <td>11:45</td> <td>13:00</td> <td></td> </tr> </tbody> </table> Below the break hours table, click + Add. e. In the Holidays section, click the Add button. Configure the holidays table as follows: <table border="1"> <thead> <tr> <th>Name</th> <th>Holiday Type</th> <th>Date</th> <th>Operations</th> </tr> </thead> <tbody> <tr> <td>National</td> <td>By Date</td> <td>2026/08/07</td> <td></td> </tr> </tbody> </table> f. Click the Confirm button.	Start Time	End Time	Operations	08:45	11:45		13:00	18:00		Start Time	End Time	Operations	11:45	13:00		Name	Holiday Type	Date	Operations	National	By Date	2026/08/07	
Start Time	End Time	Operations																						
08:45	11:45																							
13:00	18:00																							
Start Time	End Time	Operations																						
11:45	13:00																							
Name	Holiday Type	Date	Operations																					
National	By Date	2026/08/07																						

Scenario	Instruction
	d. Select a business hour type, then configure its related settings to customize business hours.  Note: Up to 20 time periods are supported for business hours and break hours respectively. e. Optional: Turn on the switch of Holidays, then add holidays.  Note: Up to 50 holidays are supported. f. At the bottom-right corner, click Confirm. The custom business hours appear on the component and multiple associated time branches are automatically created. 
Use Custom Time Periods	  

Scenario	Instruction
	a. In the Time-based Mode drop-down list, select Based on Custom Time Periods. b. In the Time Zone drop-down list, select a time zone. c. In the Daylight Saving Time drop-down list, select an option to enable or disable the feature. d. Optional: Turn on the switch of Holidays, then add holidays. Note: Up to 50 holidays are supported. e. At the bottom-right corner, click Confirm. f. Hover over component and click + to add and configure custom time periods. Note: Up to 50 custom time periods are supported. The time zone and multiple time branches will appear on the component.  Note: Default branch indicates the failover destination for the message session that don't match any configured time conditions.

3. **Optional:** To route the message session or trigger other actions when the component's branch conditions are met, hover over the desired branch and click **+** to add the next component.

For more information about components that support to be connected, see [Component connections](#).

Component connections

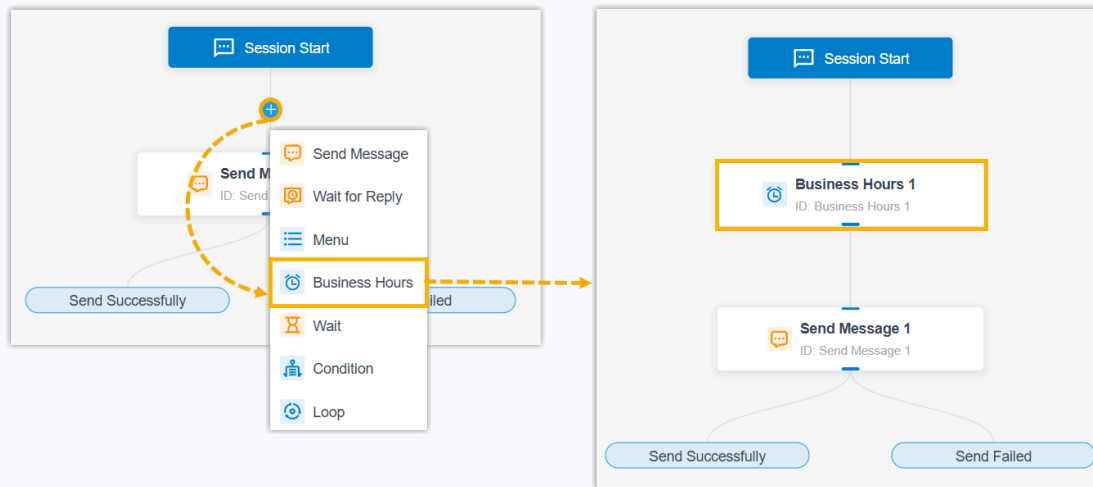
Business Hours component comes with built-in time branches, such as **Business Hours**, **Outside Business Hours**, and **Holidays**. Each branch can be connected to **one** component, which can be any of the components listed below.



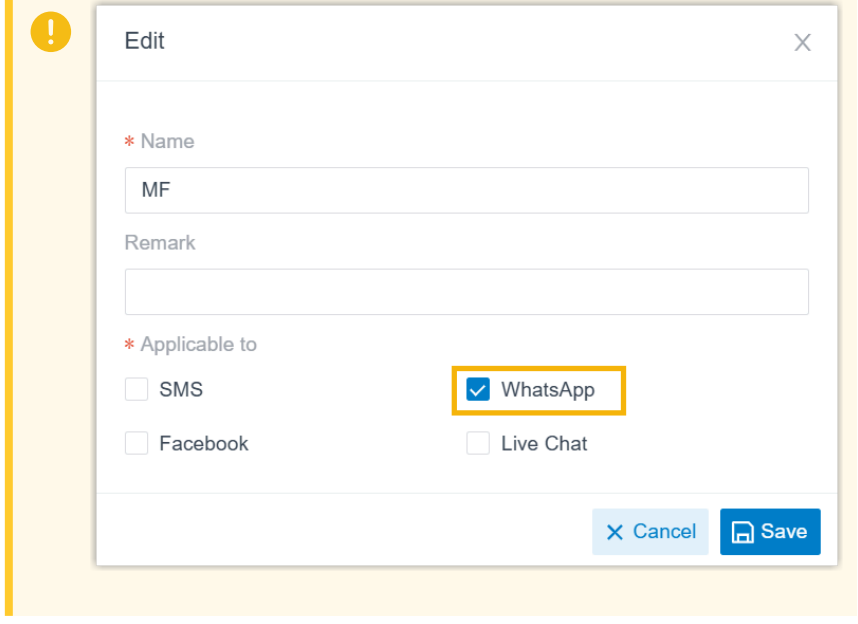
Note:

If a component is already connected to the branch, and you need to insert a new component between them, hover over the line, click **+** and select the target component. The selected component is inserted between them with connections auto-established on both ends, without the need to delete the connection, add the component and reconnect manually.

End components **Transfer** and **Close** are unavailable for this insertion method.



Component	Description
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message.
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates.  Important: This component is available only when the message flow is only applicable to WhatsApp channel type.

Component	Description
	 For more information, see Send WhatsApp Template Message.
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.

Component	Description
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services. For more information, see HTTP Request.
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

Wait

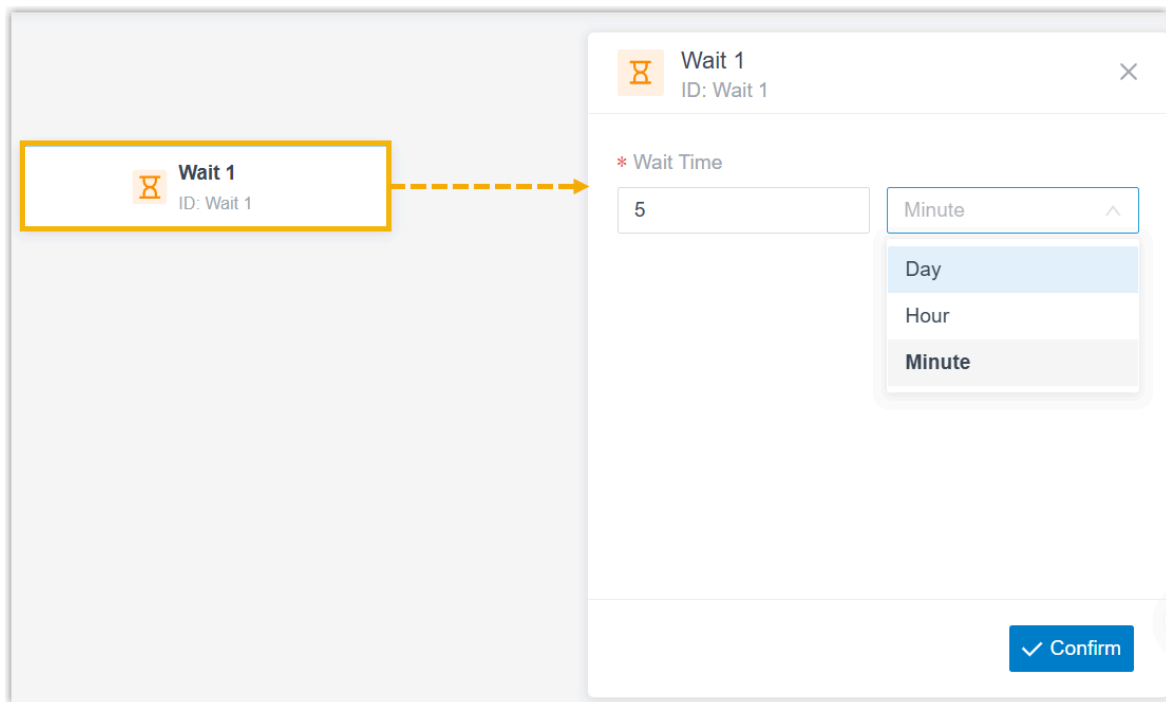
This topic provides an overview of the Wait component, and describes its configuration as well as supported connections.

Component introduction

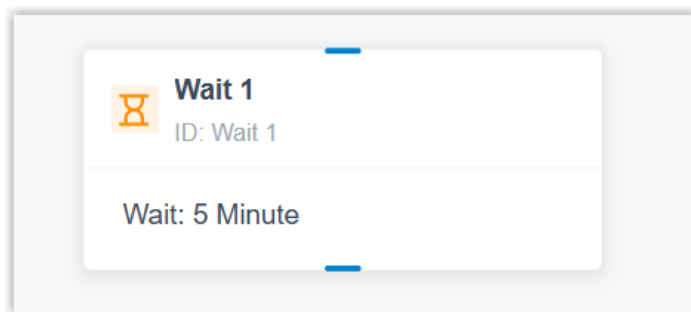
Wait component enables the system to automatically route the message session to the connected component or execute specified actions upon reply timeout.

Component configuration

1. After adding **Wait** component to a message flow, click the component and configure wait timeout.



The configured timeout appears on the component.



2. **Optional:** To route the message session or trigger other actions, hover over the component and click  to add the next component.

For more information about components that support to be connected, see [Component connections](#).

Component connections

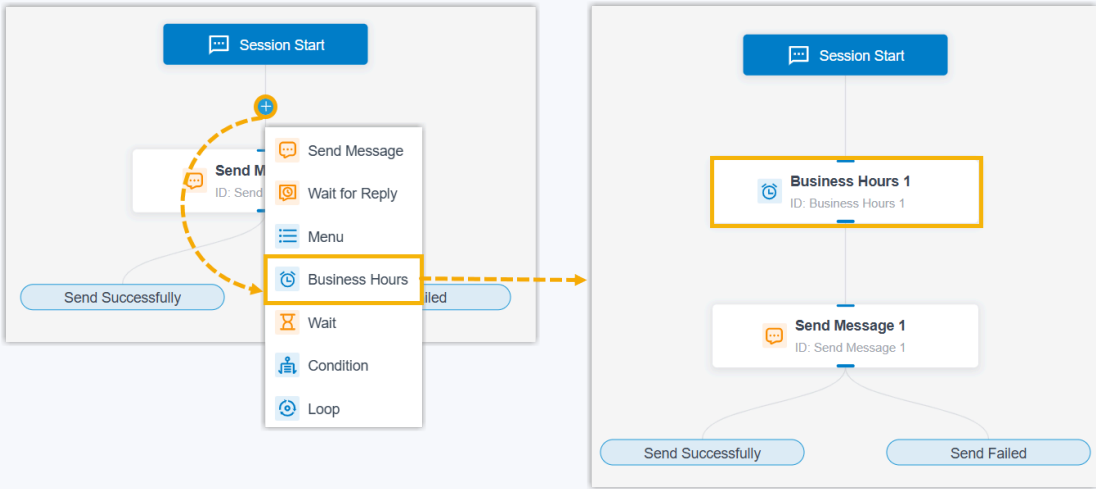
Wait component can be connected on **one** component, which can be any of the components listed below.



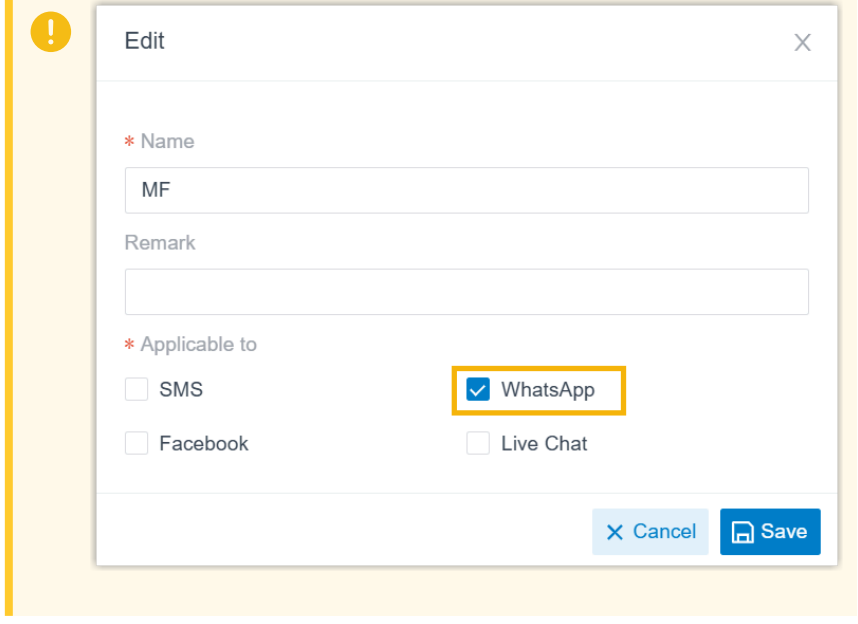
Note:

If a component is already connected to the component, and you need to insert a new one between them, hover over the line between them, click **+** and select the target component. The selected one is inserted between them with connections auto-established on both ends, without the need to delete the connection, add the component and reconnect manually.

End components **Transfer** and **Close** are unavailable for this insertion method.



Component	Description
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message .
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates.  Important: This component is available only when the message flow is only applicable to WhatsApp channel type.

Component	Description
	 For more information, see Send WhatsApp Template Message.
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.

Component	Description
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services. For more information, see HTTP Request.
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

Condition

This topic provides an overview of the Condition component, and describes its configuration as well as supported connections.

Component introduction

Condition component allows routing the message session based on logical expressions. You can use variables, constants, and functions in an expression to evaluate specific conditions. When a condition is met, the session will be routed to the connected component.

Component configuration

After adding **Condition** component to a message flow, you can configure branches with expressions to route calls based on specific conditions.

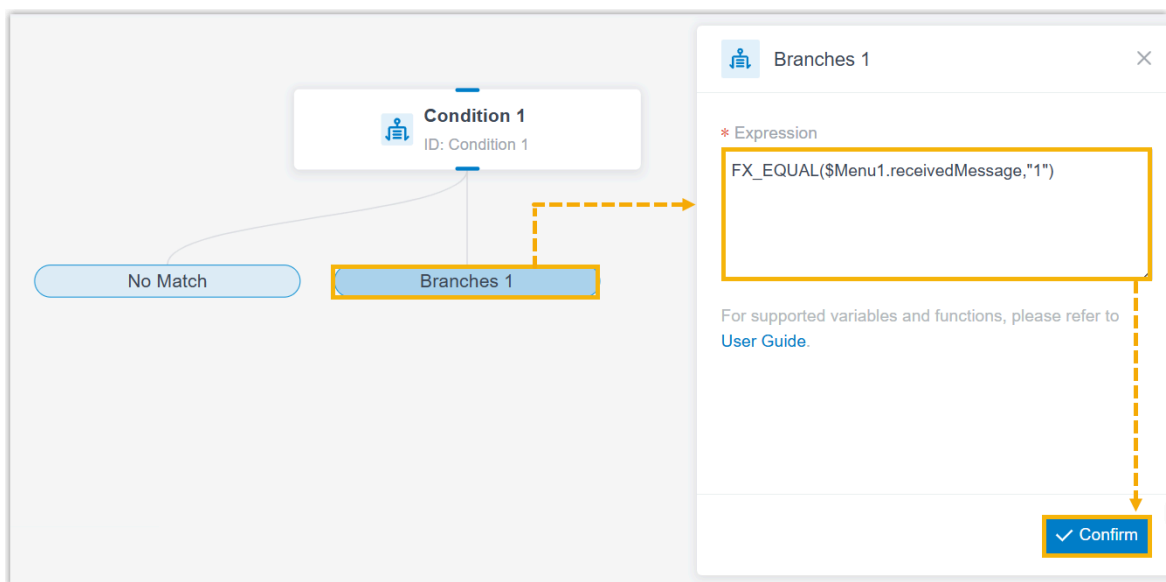
1. Click the branch to configure expression, then click **Confirm**.



Note:

- Each branch supports only one expression.
- An expression may include variables, constants, and functions. You can enter FX to view and select system-supported functions, and enter \$ to view and select variables already stored in the current message flow.

For more information about functions and variables, see [Variables and Functions in Yeastar Expression](#).



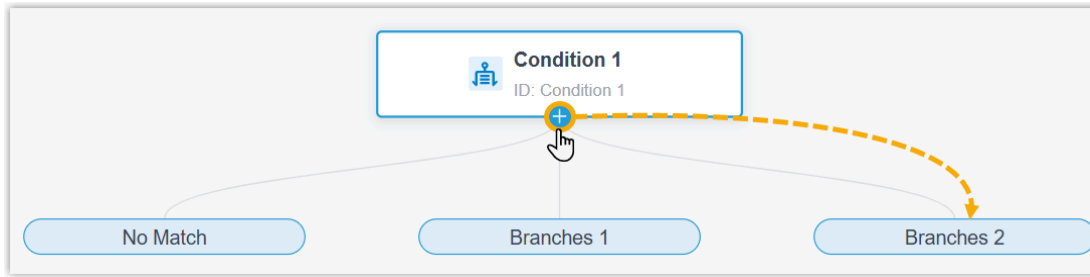
In this example, enter `FX_EQUAL($Menu1.receiveMessage, "1")` used to check whether the message content that the customer sends in response to the menu with ID **Menu 1** is 1. If they match, the session will be routed to the component connected to the branch.

2. **Optional:** To add more conditions, hover over the component and click  to add more, then repeat [Step 1](#).



Note:

You can add up to **9** additional branches.



3. **Optional:** To route the message session or trigger other actions when the component's branch conditions are met, hover over the desired branch and click  to add the next component.

For more information about components that support to be connected, see [Component connections](#).

Component connections

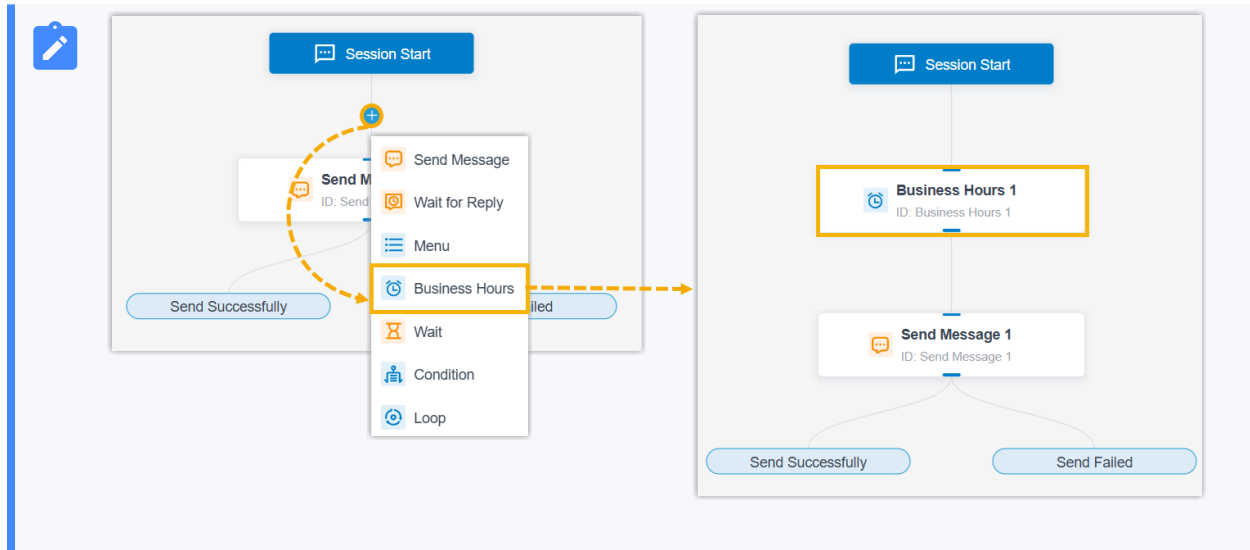
Condition component comes with two built-in branches **No Match** and **Branch1**, and supports up to 9 additional branches for specific conditions. Each branch can be connected on **one** component, which can be any of the components listed below.



Note:

If a component is already connected to the branch, and you need to insert a new component between them, hover over the line, click  and select the target component. The selected component is inserted between them with connections auto-established on both ends, without the need to delete the connection, add the component and reconnect manually.

End components **Transfer** and **Close** are unavailable for this insertion method.



Component	Description
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message.
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates. Important: This component is available only when the message flow is only applicable to WhatsApp channel type. Edit * Name MF Remark * Applicable to ☐ SMS ☒ WhatsApp ☐ Facebook ☐ Live Chat Cancel Save

Component	Description
	For more information, see Send WhatsApp Template Message .
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services.

Component	Description
	For more information, see HTTP Request .
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer .
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close .

Loop

This topic provides an overview of the Loop component, and describes its configuration as well as supported connections.

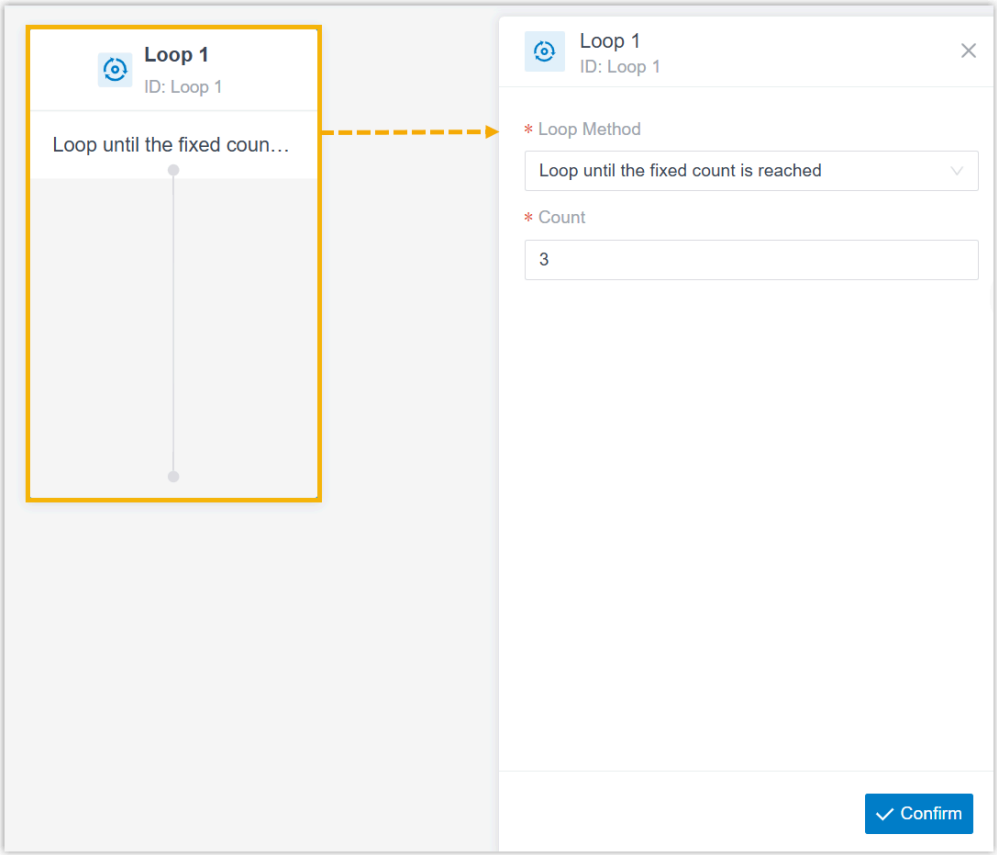
Component introduction

Loop component allows a group of components to be executed repeatedly based on one of the following loop strategies, helpful in repeating menu options or returning to the previous menu.

- Run a loop until a fixed count is reached.
- Run a loop while the configured condition is true.

Component configuration

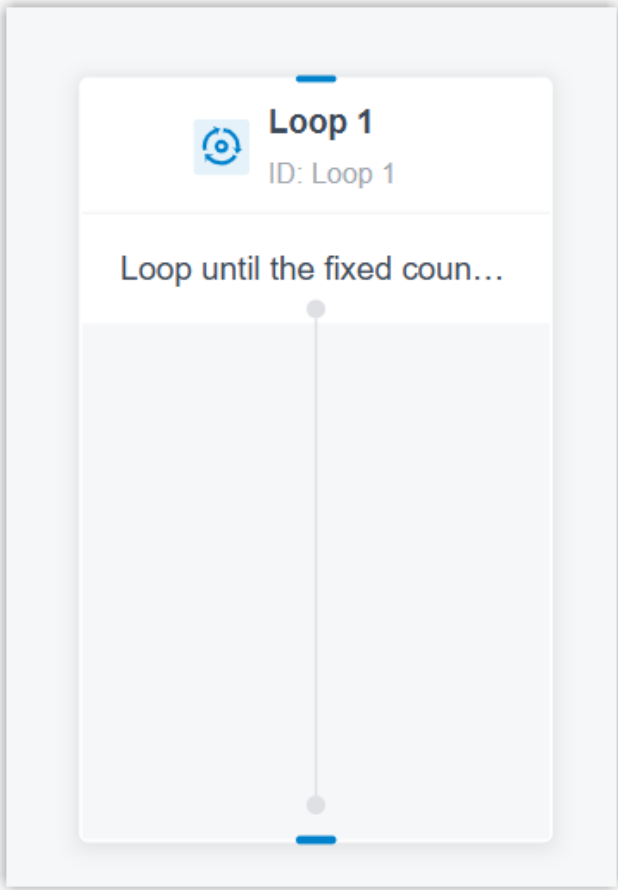
1. After adding **Loop** component to a message flow, click **Loop** component to proceed with the configuration.

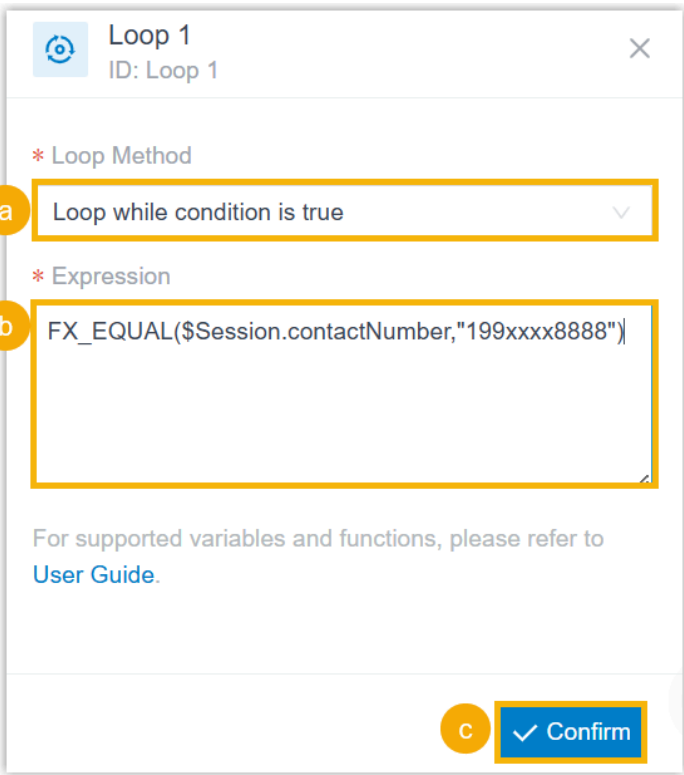


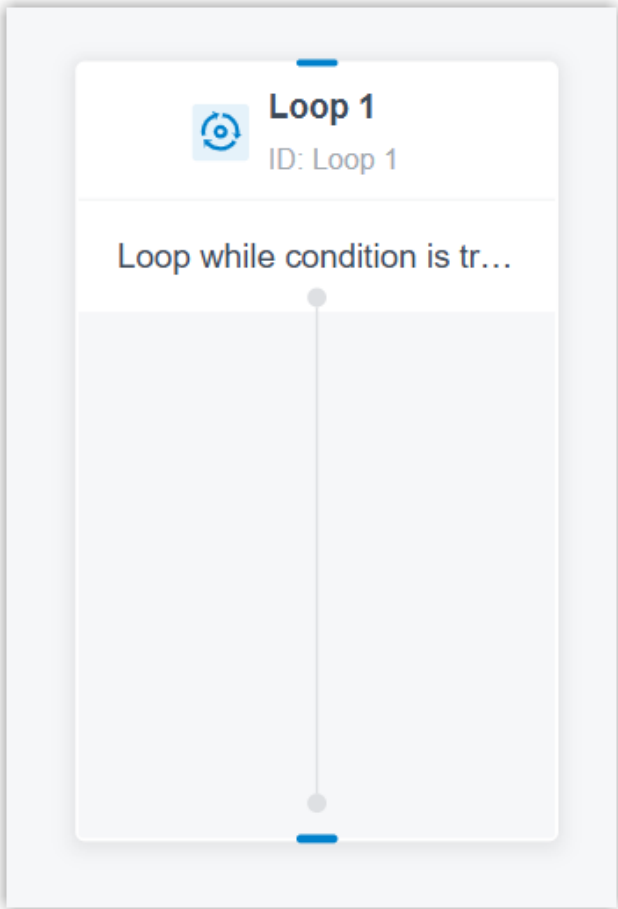
2. In the pop-up right panel, configure the loop strategy as needed.

Loop Method	Instruction
Loop until the fixed count is reached	To run the loop until a fixed count is reached, complete the following settings.

Loop Method	Instruction
	 Loop 1 ID: Loop 1 * Loop Method a Loop until the fixed count is reached * Count b 3 c ✓ Confirm a. In the Loop Method drop-down list, select Loop until the fixed count is reached. b. In the Count field, specify how many times the loop should repeat.  Note: You can enter a value between 1 and 99. c. At the bottom-right corner, click Confirm. The configured strategy appears on the component.

Loop Method	Instruction
	
Loop while condition is true	To run the loop while the condition is true, complete the following settings.

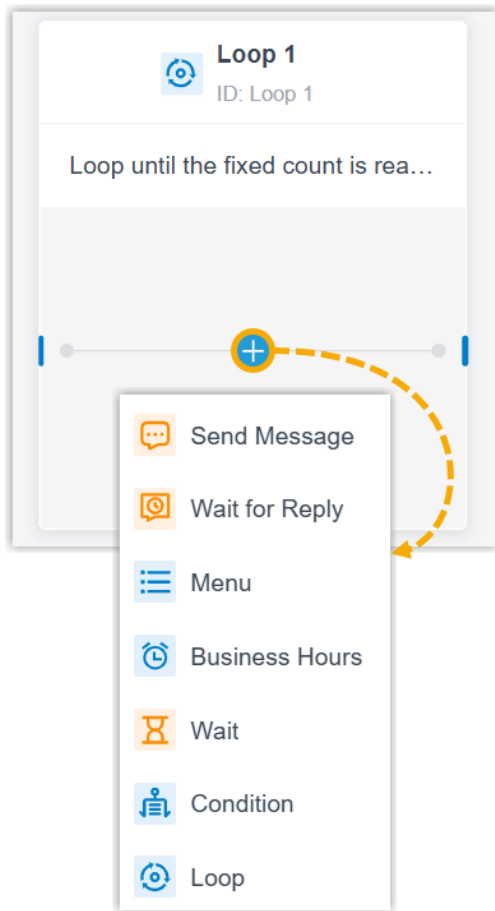
Loop Method	Instruction
	 Loop 1 ID: Loop 1 * Loop Method a Loop while condition is true * Expression b FX_EQUAL(\$Session.contactNumber,"199xxx8888") For supported variables and functions, please refer to User Guide. c Confirm a. In the Loop Method drop-down list, select Loop while condition is true. b. In the Expression field, enter an expression that returns a Boolean value. The loop runs when the result is true and is skipped when false. For example, enter FX_EQUAL(\$Session.contactNumber,"199xxx8888") as the loop condition, the session will enter the loop when the customer's number is 199xxx8888.  Note: An expression may include variables, constants, and functions. You can enter FX to view and select system-supported functions, and enter \$ to view and select variables already stored in the current message flow. For more information about functions and variables, see Variables and Functions in Yeastar Expression. c. At the bottom-right corner, click Confirm.

Loop Method	Instruction
	The configured strategy appears on the component. 

3. Click  inside the loop to add the components that you want to repeat.



Note:
If you add a nested **Loop** component, up to **3 levels of nesting** are supported.



4. **Optional:** To route the message session or trigger other actions when the loop ends, click  on the loop component border to add the next component.

For more information about components that support to be connected, see [Component connections](#).

Component connections

Loop component can be connected to **one** component, which will be executed after the loop completes. You can connect any of the components listed below.

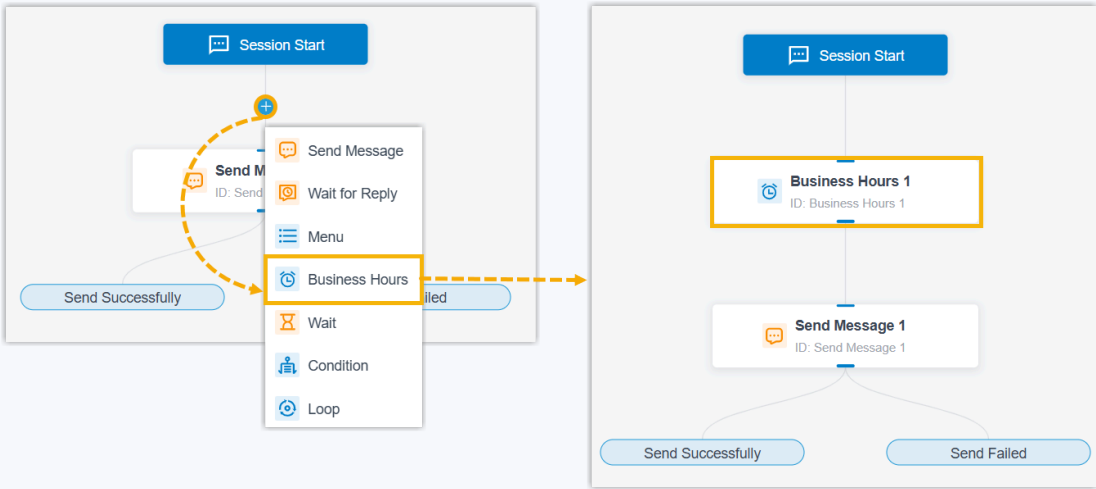


Note:

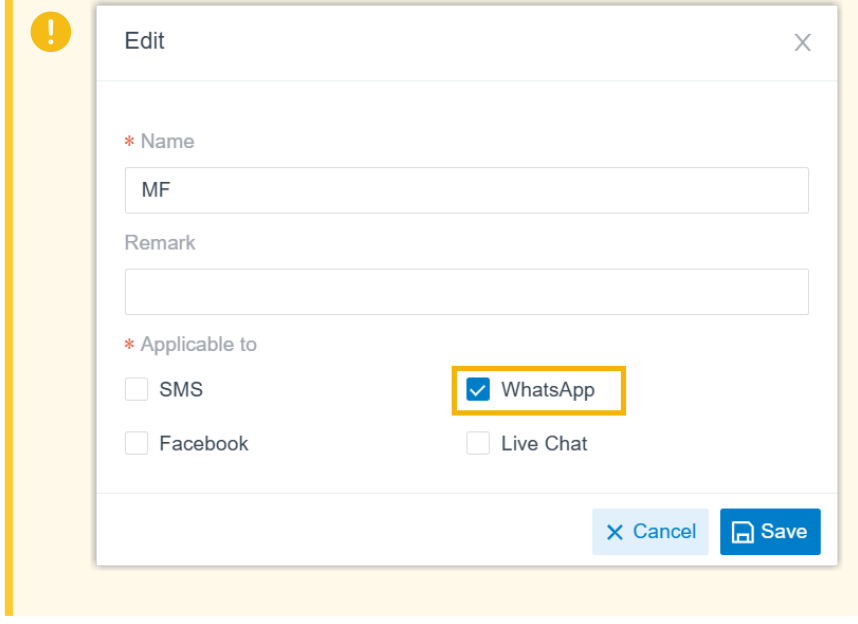
If a component is already connected to the component, and you need to insert a new one between them, hover over the line between them, click  and select the target component. The selected one is inserted between them with connections auto-established on both ends, without the need to delete the connection, add the component and reconnect manually.



End components **Transfer** and **Close** are unavailable for this insertion method.



Component	Description
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message.
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates. Important: This component is available only when the message flow is only applicable to WhatsApp channel type.

Component	Description
	 For more information, see Send WhatsApp Template Message.
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.

Component	Description
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services. For more information, see HTTP Request.
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

Developer Components

Internal Data Ops Component

Get Extension Presence Status

This topic provides an overview of the Get Extension Presence Status component, and describes its configuration, variables, as well as supported connections.

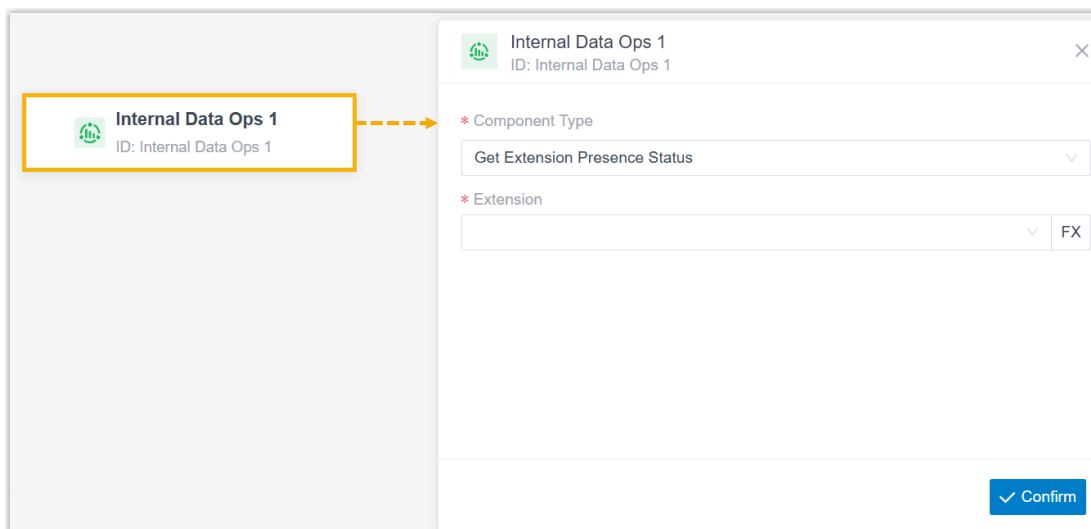
Component introduction

Get Extension Presence Status component allows you to query current presence status of a specific extension.

You can either select a specific extension directly, or configure an expression to dynamically specify the target extension. The query results are stored in variables, which can be used as input for subsequent components or for condition evaluation.

Component configuration

1. After adding **Internal Data Ops** component to a message flow, click **Internal Data Ops** component to proceed with the configuration.



2. In the pop-up right panel, complete the following settings.

Internal Data Ops 1
ID: Internal Data Ops 1

* Component Type

a Get Extension Presence Status

* Extension

b 1800001-1800001 FX

c Confirm

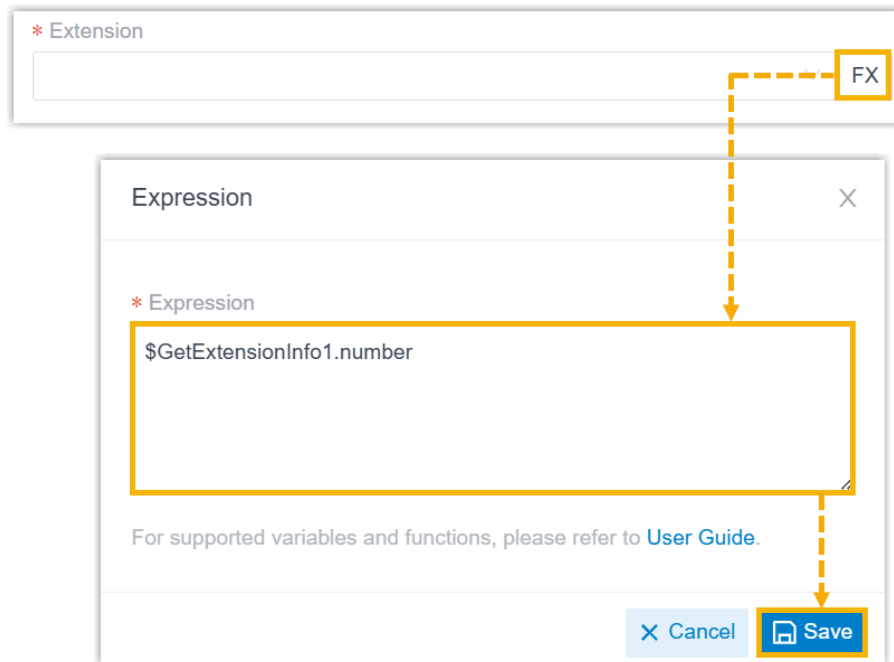
- a. In the **Component Type** drop-down list, select **Get Extension Presence Status**.
- b. In the **Extension** field, specify the target extension using one of the following methods.
 - Select an extension from the drop-down list.
 - Click **FX** beside the drop-down list, enter an expression and click **Save**.



Note:

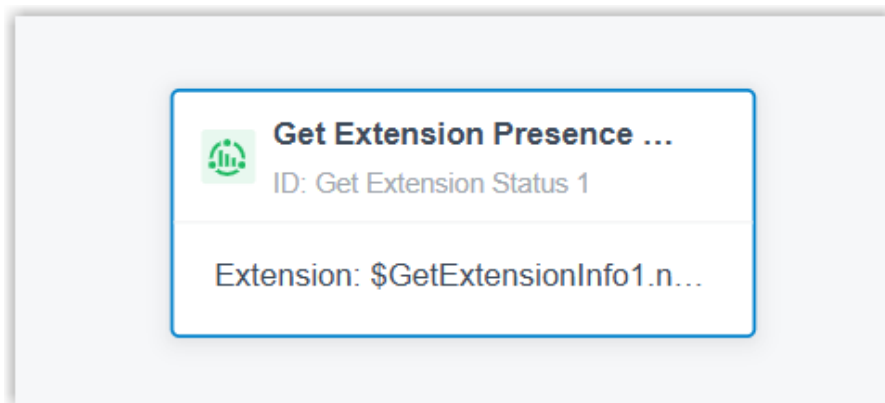
An expression may include variables, constants, and functions. You can enter **FX** to view and select system-supported functions, and enter **\$** to view and select variables already stored in the current message flow.

For more information about functions and variables, see [Variables and Functions in Yeastar Expression](#).



c. At the bottom-right corner, click **Confirm**.

The configured information appears on the component.



3. **Optional:** To route the message session or trigger other actions, hover over the component and click  to add the next component .

For more information about components that support to be connected, see [Component connections](#).

Component variables

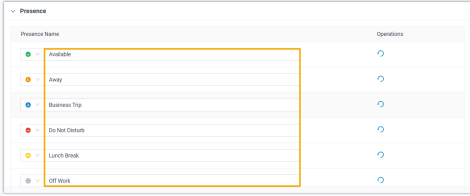
When a **Get Extension Presence Status** component is added to a message flow, the system stores the extension's statuses in variables. These variables can be referenced in ex-

pression-supported components (such as **Condition**) to retrieve the status values and respond accordingly based on the result.



Note:

Since **Get Extension Presence Status** component can be added multiple times in a message flow, an index is appended to the unique ID of each component (e.g. Get Extension Presence Status 1, Get Extension Presence Status 2) based on the order in which the components are added. To retrieve data from the right component, the component variable must contain the corresponding index.

Variable	Type	Description	Example Value
<code>\$GetExtensionStatus{ <i>index</i>}.currentProfileName</code>	String	Name of the extension's current presence status.  Note: When using functions to evaluate an extension's presence status, ensure that the status constant matches one of the values defined in PBX Settings > Preferences > Presence. 	"Available"
<code>\$GetExtensionStatus{ <i>index</i>}.currentProfile</code>	String	Name and additional information of the extension's current presence status.	"Available,work"
<code>\$GetExtensionStatus{ <i>index</i>}.isUnavailable</code>	Boolean	Whether the extension is unregistered.	False

Component connections

Get Extension Presence Status component can be connected to **one** component, which can be any of the components listed below.



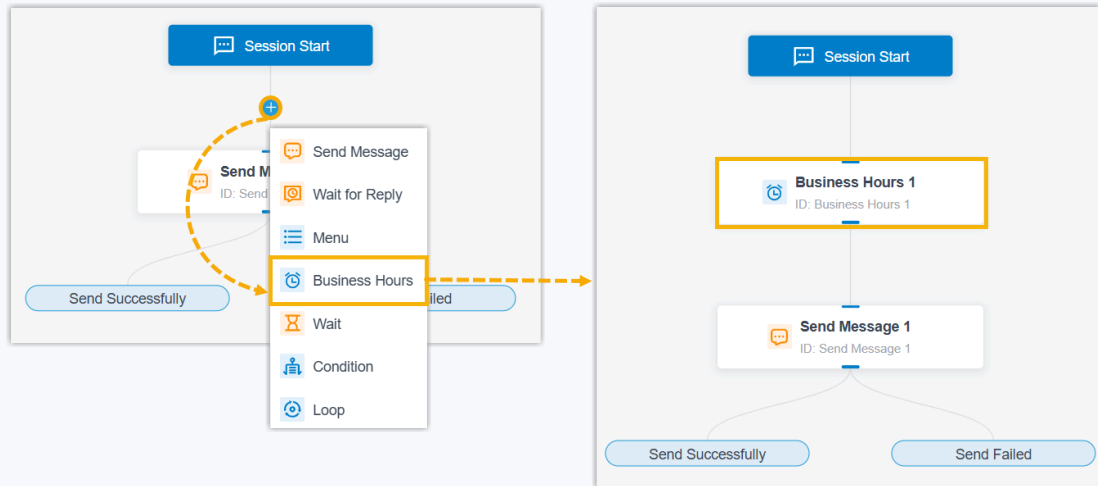
Note:

If a component is already connected to the component, and you need to insert a new one between them, hover over the line between them, click  and select the target component. The selected one is inserted between them with connections

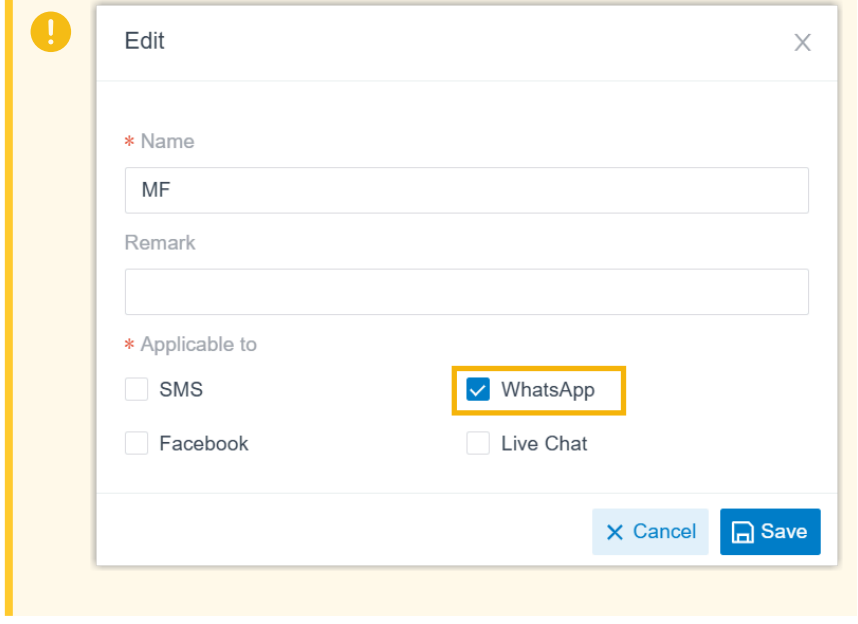


auto-established on both ends, without the need to delete the connection, add the component and reconnect manually.

End components **Transfer** and **Close** are unavailable for this insertion method.



Component	Description
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message.
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates.  Important: This component is available only when the message flow is only applicable to WhatsApp channel type.

Component	Description
	 For more information, see Send WhatsApp Template Message.
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.

Component	Description
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services. For more information, see HTTP Request.
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

Get Extension Info

This topic provides an overview of the **Get Extension Info** component, and describes its configuration, variables, as well as supported connections.

Component introduction

Get Extension Info component allows you to query extension information, including extension number, extension name, email address, and mobile number.

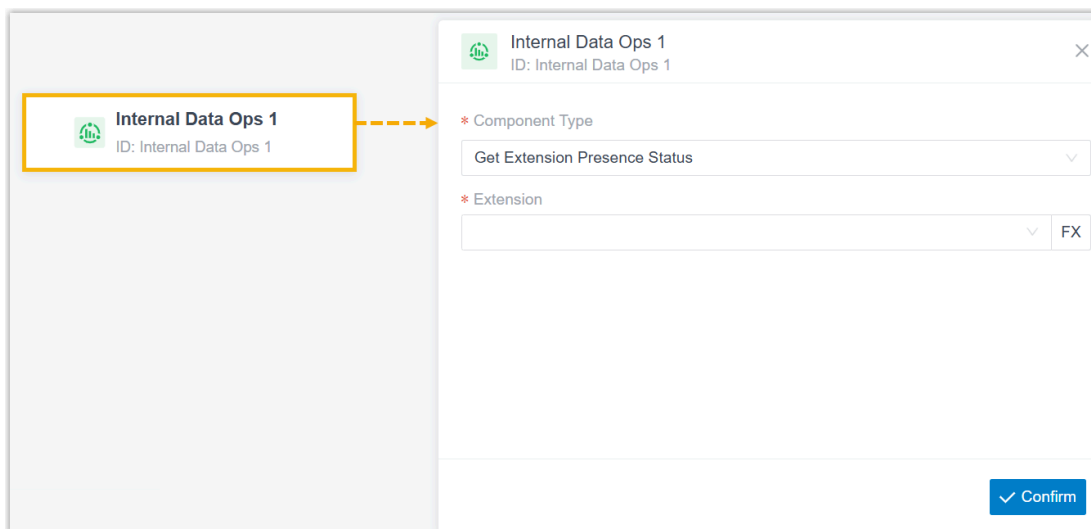
You can either specify the extension directly, or configure expressions to dynamically specify the target extension. The query results are stored in variables, which can be used as input for subsequent components or for condition evaluation.

**Note:**

If multiple matches are found, the system returns the information of the most recently created extension.

Component configuration

1. After adding **Internal Data Ops** component to a message flow, click **Internal Data Ops** component to proceed with the configuration.



2. In the pop-up right panel, complete the following settings.



Internal Data Ops 1
 ID: Internal Data Ops 1

* Component Type

a

Get Extension Info
 

b

Keyword

FX

Name

FX

Email

FX

Number

FX

Extension Group



Organization

c

 Confirm

- a. In the **Component Type** drop-down list, select **Get Extension Info**.
- b. Specify the target extension using the following filters.

Filter	Description
Keyword	Enter a keyword to perform a fuzzy search for extensions by first name, last name, email address, or extension number.
Name	Enter extension's first name or last name.
Email	Enter extension's email address.
Number	Enter extension number.
Extension Group	Select extension group(s) to search extensions in these groups.  Note: If no groups are configured, the system searches across all extension groups.
Organization	Select departments(s) to search extensions in these departments.

Filter	Description
	 Note: • This option is available only when Organization Management is enabled (Path: PBX Settings > Preferences > Organization Management). • If configured, the system searches extensions within the selected department(s) and their sub-departments. Otherwise, the system searches across all departments.


Note:

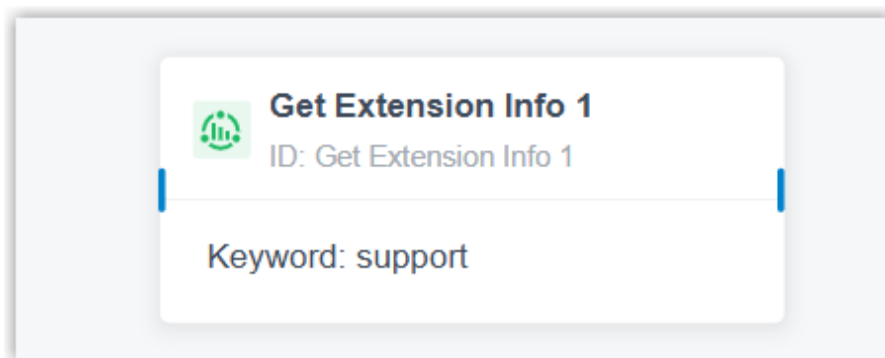
For **Keyword**, **Name**, **Email** and **Number** filters, you can click **FX** to configure expressions to dynamically filter the extensions.

An expression may include variables, constants, and functions. You can enter **FX** to view and select system-supported functions, and enter **\$** to view and select variables already stored in the current message flow.

For more information about functions and variables, see [Variables and Functions in Yeastar Expression](#).

c. At the bottom-right corner, click **Confirm**.

The configured information appears on the component.



3. **Optional:** To route the message session or trigger other actions, hover over the component and click  to add the next component.

For more information about components that support to be connected, see [Component connections](#).

Component variables

When a **Get Extension Info** component is added to a message flow, the system stores extension information in variables. These variables can be referenced in expression-supported components (such as **Condition**) to retrieve the extension information and respond accordingly based on the result.

**Note:**

Since **Get Extension Info** component can be added multiple times in a message flow, an index is appended to the unique ID of each component (e.g. Get Extension Info 1, Get Extension Info 2) based on the order in which the components are added. To retrieve data from the right component, the component variable must contain the corresponding index.

Variable	Type	Description	Example Value
<code>\$GetExtensionInfo{index}.number</code>	String	Extension number.	"1001"
<code>\$GetExtensionInfo{index}.name</code>	String	Extension name.	"Phillip Huff"
<code>\$GetExtensionInfo{index}.email</code>	String	Extension's email address.	"phillip@sample.com"
<code>\$GetExtensionInfo{index}.mobileNumber</code>	String	Extension's mobile number.	"15880123456"

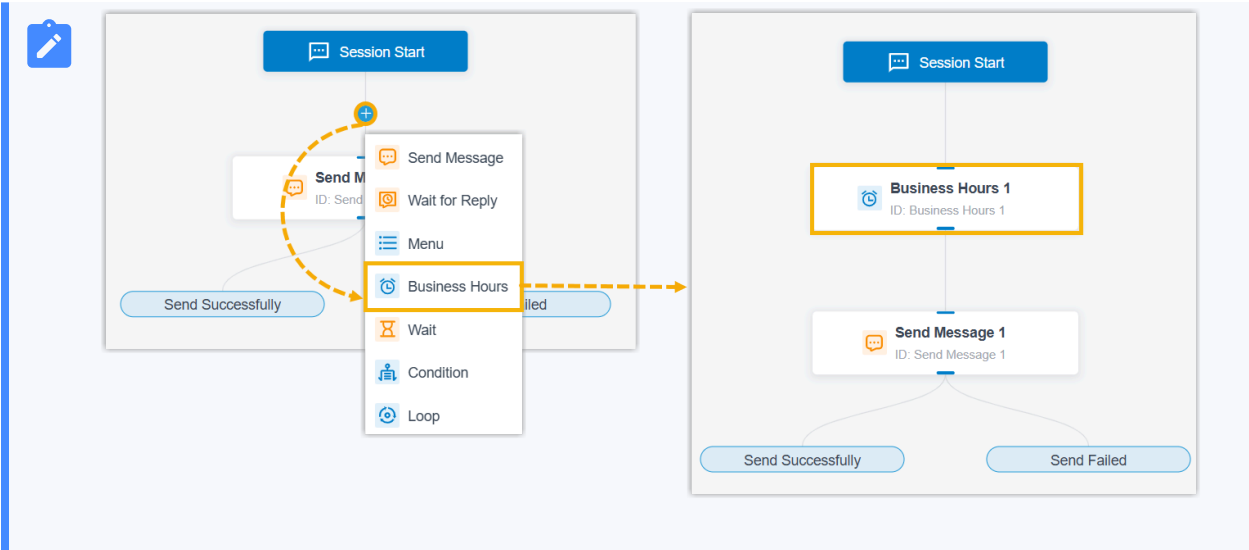
Component connections

Get Extension Info component can be connected to **one** component, which can be any of the components listed below.

**Note:**

If a component is already connected to the component, and you need to insert a new one between them, hover over the line between them, click  and select the target component. The selected one is inserted between them with connections auto-established on both ends, without the need to delete the connection, add the component and reconnect manually.

End components **Transfer** and **Close** are unavailable for this insertion method.



Component	Description
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message.
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates. Important: This component is available only when the message flow is only applicable to WhatsApp channel type. Edit * Name MF Remark * Applicable to ☐ SMS ☒ WhatsApp ☐ Facebook ☐ Live Chat Cancel Save

Component	Description
	For more information, see Send WhatsApp Template Message .
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services.

Component	Description
	For more information, see HTTP Request .
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

Get Company Contact Info

This topic provides an overview of the **Get Company Contact Info** component, and describes its configuration, variables, as well as supported connections.

Component introduction

Get Company Contact Info component allows you to query company contacts information, such as contact name, phone number, email address, etc.

You can either specify the company contact directly, or configure expressions to dynamically specify the target company contact. The query results are stored in variables, which can be used as input for subsequent components or for condition evaluation.

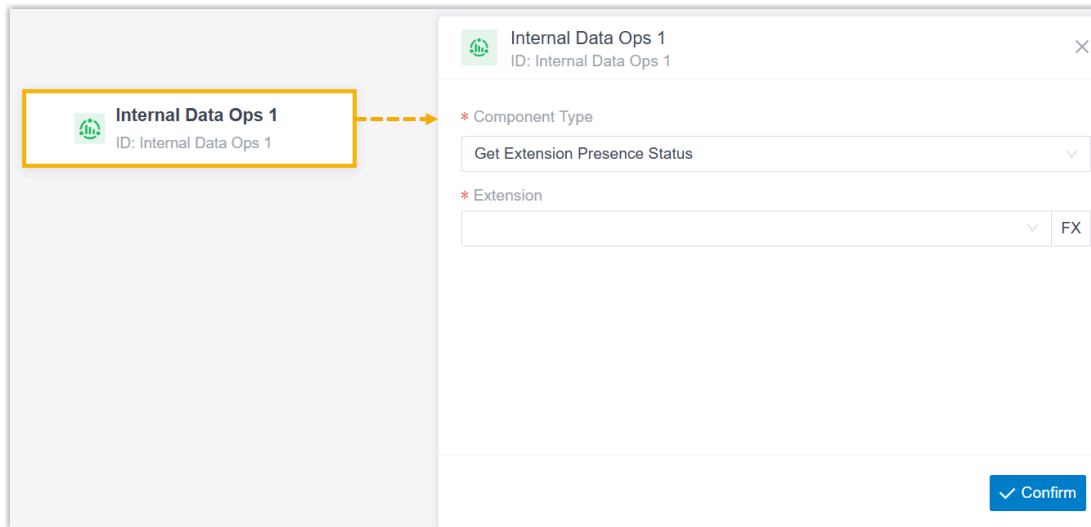


Note:

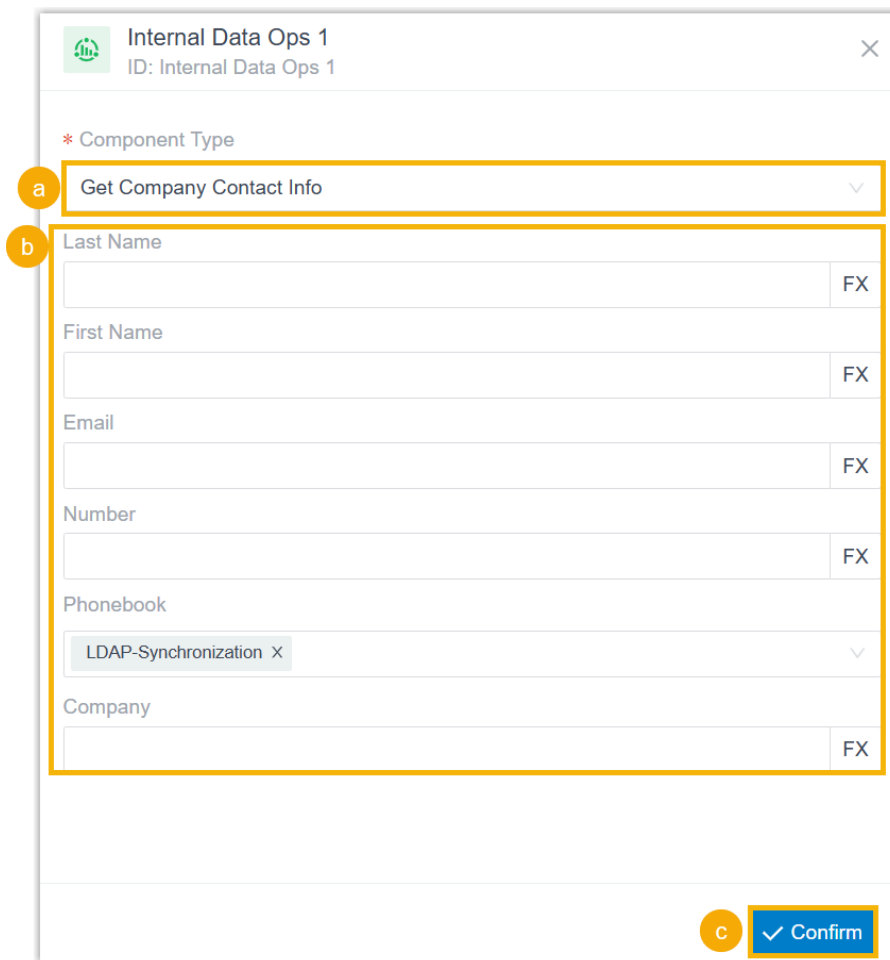
If multiple matches are found, the system returns the information of the most recently created company contact.

Component configuration

1. After adding **Internal Data Ops** component to a message flow, click **Internal Data Ops** component to proceed with the configuration.



2. In the pop-up right panel, complete the following settings.



- a. In the **Component Type** drop-down list, select **Get Company Contact Info**.
- b. Specify the target company contact using the following filters.

Filter	Description
Last Name	Enter contact's last name.
First Name	Enter contact's first name.
Email	Enter contact's email address.
Number	Enter contact's phone number.  Note: The system follows the number matching rule (set in Contacts > Company Contacts > Options > Number Match) when searching for company contacts. If Do Not Match is selected, the system performs an exact number match.
Phonebook	Select phonebook(s) to search contacts in these directories.  Note: If no phonebooks are configured, the system searches across all directories.
Company	Enter keywords from company information.


Note:

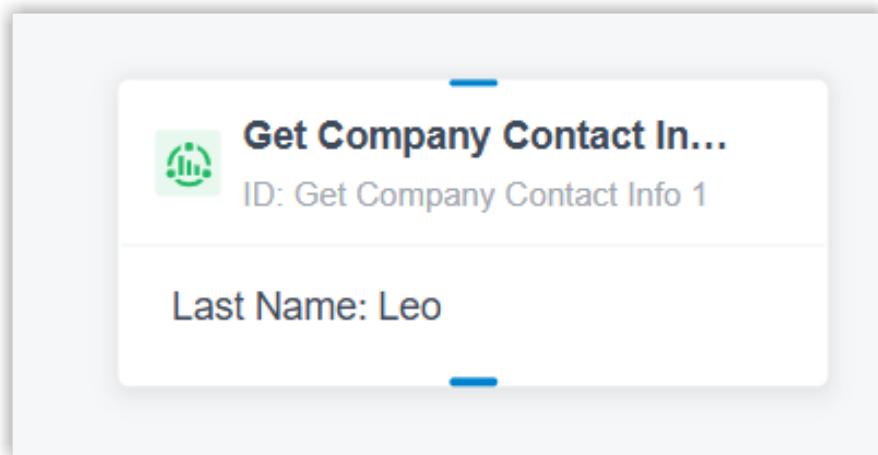
For **Last Name**, **First Name**, **Email**, **Number** and **Company** filters, you can click **FX** to configure expressions to dynamically filter the contacts.

An expression may include variables, constants, and functions. You can enter **FX** to view and select system-supported functions, and enter **\$** to view and select variables already stored in the current message flow.

For more information about functions and variables, see [Variables and Functions in Yeastar Expression](#).

c. At the bottom-right corner, click **Confirm**.

The configured information appears on the component.



3. **Optional:** To route the message session or trigger other actions, hover over the component and click  to add the next component.

For more information about components that support to be connected, see [Component connections](#).

Component variables

When a **Get Company Contact Info** component is added to a message flow, the system stores company contact information in variables. These variables can be referenced in expression-supported components (such as **Condition**) to retrieve the contact information and respond accordingly based on the result.



Note:

Since **Get Company Contact Info** component can be added multiple times in a message flow, an index is appended to each component (e.g. Get Company Contact Info 1, Get Company Contact Info 2) based on the order in which the components are added. To retrieve data from the right component, the component variable must contain the corresponding index.

Variable	Type	Description	Example Value
\$GetCompanyContactInfo{ <i>index</i> }.contactName	String	Contact name.	"Leo Poll"
\$GetCompanyContactInfo{ <i>index</i> }.company	String	Company name.	"YS"
\$GetCompanyContactInfo{ <i>index</i> }.email	String	Email address.	"leo@ys.com"

Variable	Type	Description	Example Value
\$GetCompanyContactInfo{index}.phonebookList	String	The phonebook(s) to which the contact belongs.	"All"
\$GetCompanyContactInfo{index}.remark	String	Remark about the contact.	"remark"
\$GetCompanyContactInfo{index}.business	String	Business number.	"01082345678"
\$GetCompanyContactInfo{index}.business2	String	Business number.	"02160123456"
\$GetCompanyContactInfo{index}.mobile	String	Mobile number.	"13800001234"
\$GetCompanyContactInfo{index}.mobile2	String	Mobile number.	"15900005678"
\$GetCompanyContactInfo{index}.home	String	Home number.	"01087654321"
\$GetCompanyContactInfo{index}.home2	String	Home number.	"075523456789"
\$GetCompanyContactInfo{index}.businessFax	String	Business fax number.	"01082345679"
\$GetCompanyContactInfo{index}.homeFax	String	Home fax number.	"02160123457"
\$GetCompanyContactInfo{index}.other	String	Other phone number.	"4008880000"

Component connections

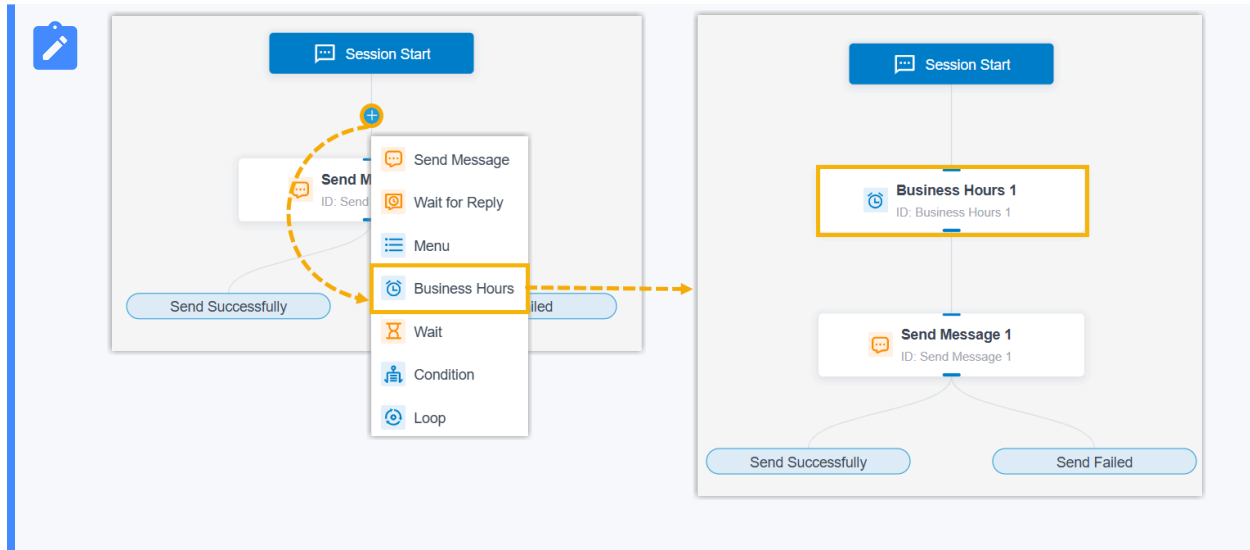
Get Company Contact Info component can be connected to **one** component, which can be any of the components listed below.



Note:

If a component is already connected to the component, and you need to insert a new one between them, hover over the line between them, click  and select the target component. The selected one is inserted between them with connections auto-established on both ends, without the need to delete the connection, add the component and reconnect manually.

End components **Transfer** and **Close** are unavailable for this insertion method.



Component	Description
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message.
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates. Important: This component is available only when the message flow is only applicable to WhatsApp channel type. Edit * Name MF Remark * Applicable to ☐ SMS ☒ WhatsApp ☐ Facebook ☐ Live Chat Cancel Save

Component	Description
	For more information, see Send WhatsApp Template Message .
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services.

Component	Description
	For more information, see HTTP Request .
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

Email Sender

This topic provides an overview of the Email Sender component, and describes its configuration as well as supported connections.

Component introduction

Email Sender component enables system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. You can customize email fields such as recipients (To, CC, BCC), subject, message body, and file attachments on the component.

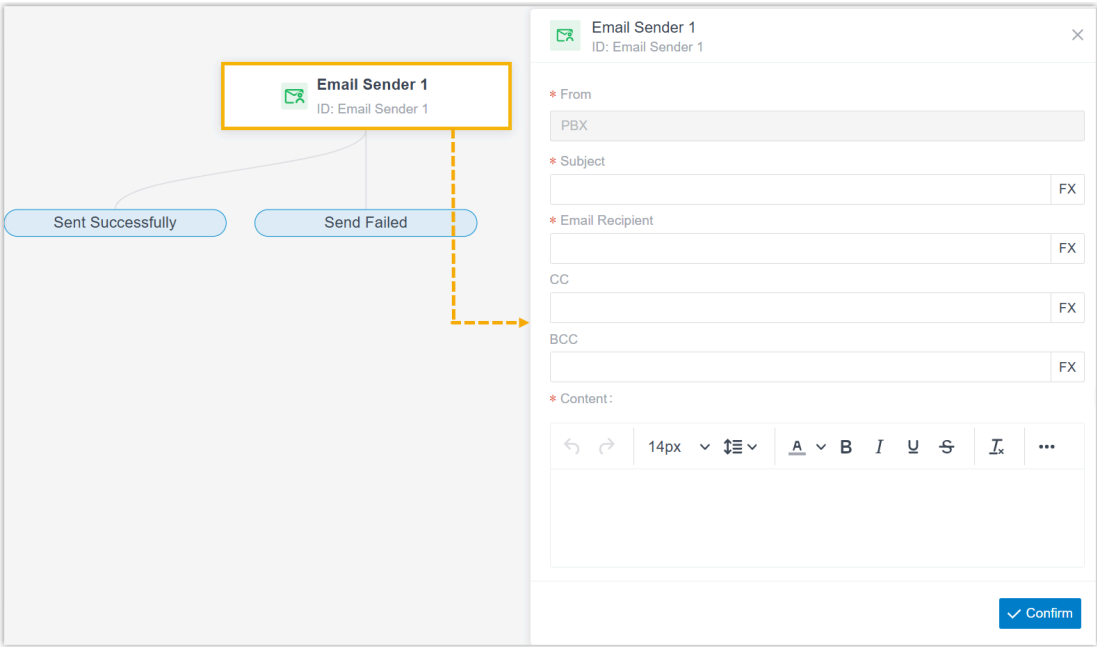


Note:

You can check configured email server at **System > Email > Email Server** and check email send logs at **System > Email > Email Send Logs**.

Component configuration

1. After adding **Email Sender** component to a message flow, click **Email Sender** component to proceed with the configuration.



2. In the pop-up right panel, customize email fields and click **Confirm**.



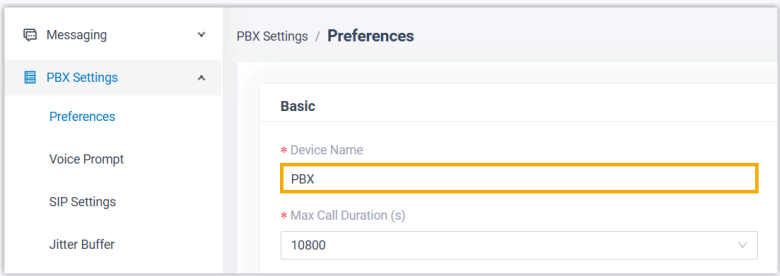
Note:

The following email fields can be defined via expressions that may include variables, constants, and functions.

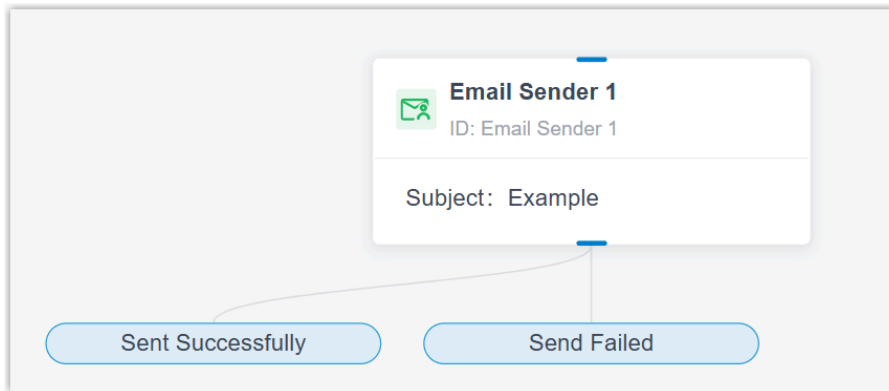
You can click **FX** to configure expressions. In expression editor, enter **FX** to view and select system-supported functions, and enter **\$** to view and select variables already stored in the current message flow.

For more information about functions and variables, see [Variables and Functions in Yeastar Expression](#).

Item	Description
From	Display the sender name, which will appear as the "From" address for outgoing emails sent by the PBX. Note: You can modify the default value in PBX Settings > Preferences > Device Name.

Item	Description
	 
Subject	Enter the subject of the email.
Email Recipient	Enter the recipient email address, which will appear as the "To" address for outgoing emails sent by the PBX.  Note: Up to 5 email addresses are supported; Use commas ; to separate multiple addresses.
CC	Enter the Carbon Copy (CC) email address, which will appear as the "Cc" address for outgoing calls sent by the PBX.  Note: Up to 5 email addresses are supported; Use commas ; to separate multiple addresses.
BCC	Enter the Blind Carbon Copy (BCC) email address, which will appear as the "Bcc" address for outgoing calls sent by the PBX.  Note: Up to 5 email addresses are supported; Use commas ; to separate multiple addresses.
Content	Enter the body content of the email.  Note: To insert system-related information, copy the template variable and paste it into the content body. <code>{{.PBXName}}</code> the name of this PBX. It can be set up in the PBX Settings > Preferences. <code>{{.PBXSN}}</code> PBX Serial Number. <code>{{.EventTime}}</code> The time when the event happened.

The configured information appears on the component.



3. **Optional:** To route the message session or trigger other actions when the component's branch conditions are met, hover over the desired branch and click  to add the next component.

For more information about components that support to be connected, see [Component connections](#).

Component connections

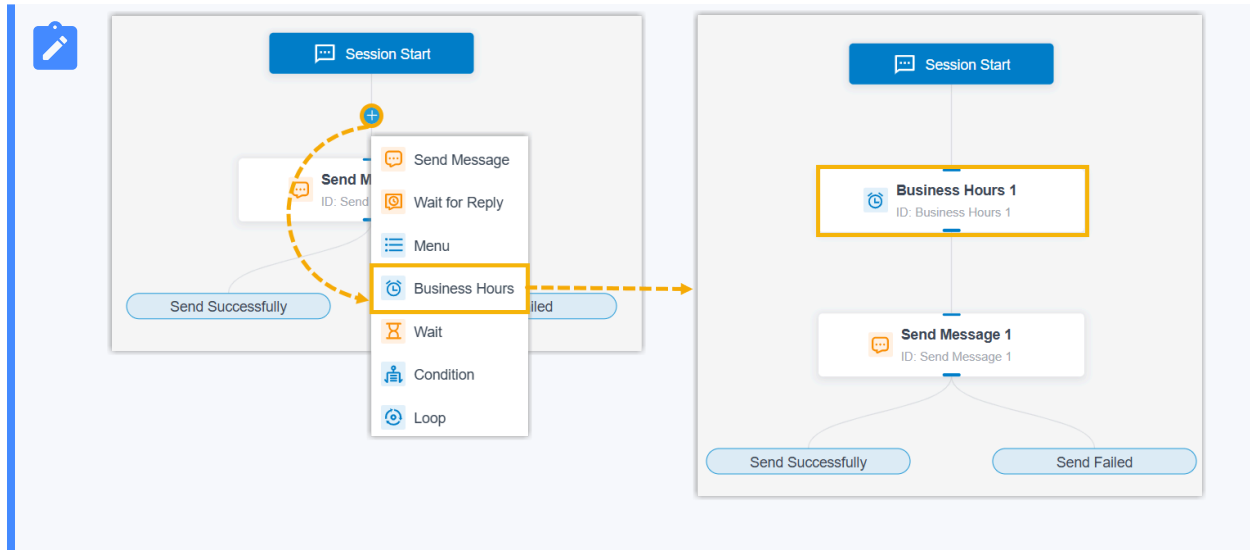
Email Sender component comes with two built-in branches **Send Successfully** and **Send Failed**. Each branch can be connected on **one** component, which can be any of the components listed below.



Note:

If a component is already connected to the branch, and you need to insert a new component between them, hover over the line, click  and select the target component. The selected component is inserted between them with connections auto-established on both ends, without the need to delete the connection, add the component and reconnect manually.

End components **Transfer** and **Close** are unavailable for this insertion method.



Component	Description
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message.
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates. Important: This component is available only when the message flow is only applicable to WhatsApp channel type. Edit * Name MF Remark * Applicable to ☐ SMS ☒ WhatsApp ☐ Facebook ☐ Live Chat Cancel Save

Component	Description
	For more information, see Send WhatsApp Template Message .
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services.

Component	Description
	For more information, see HTTP Request .
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

Database Access

This topic provides an overview of the Database Access component, and describes its configuration, variables, as well as supported connections.

Component introduction

Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. The results are stored in variables, which can be used as input for subsequent components or for condition evaluation.

Supported database types

- Microsoft SQL
- PostgreSQL
- MySQL
- Oracle

Supported SQL statements

- Query
- NonQuery
- Scalar

Component configuration

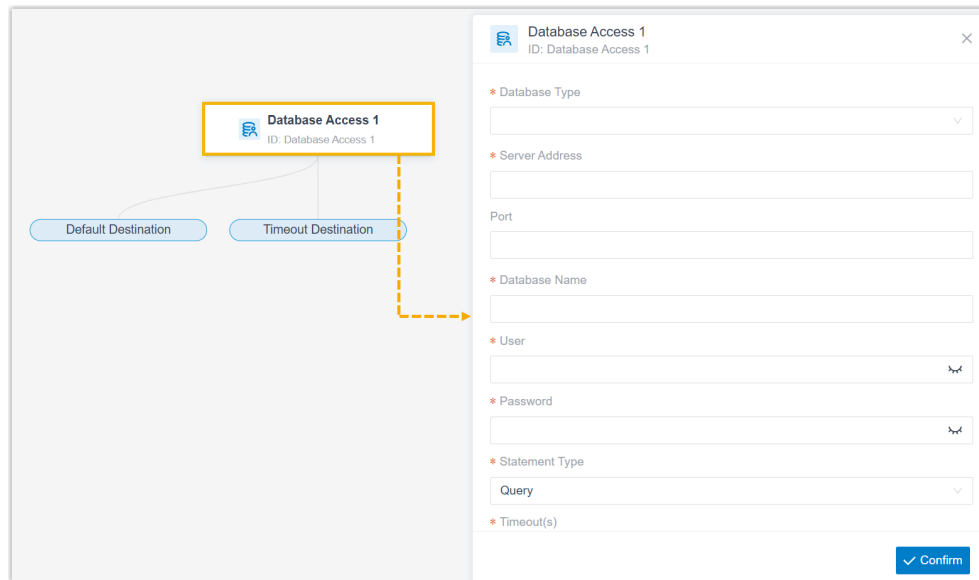
Prerequisites

- Obtain the basic information of the external database, including server address, port, database name, username and password.

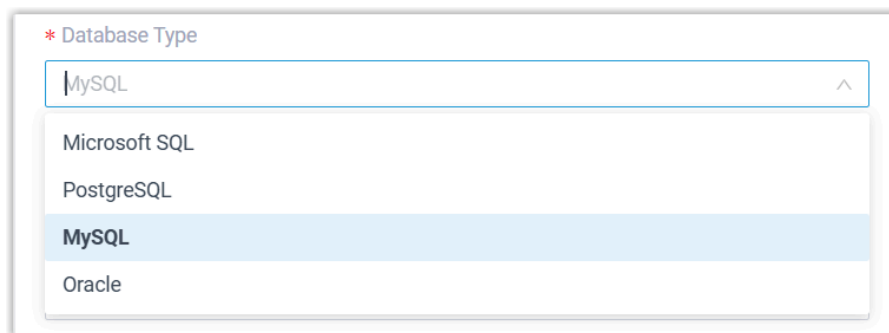
- Make sure the PBX can access the external database.

Procedure

1. After adding **Database Access** component to a message flow, click **Database Access** component to proceed with the configuration.



2. In the **Database Type** drop-down list, select a database type.



3. Fill in the following information to connect to a database.

* Server Address

Port

3306

* Database Name

master

* User

....

* Password

.....

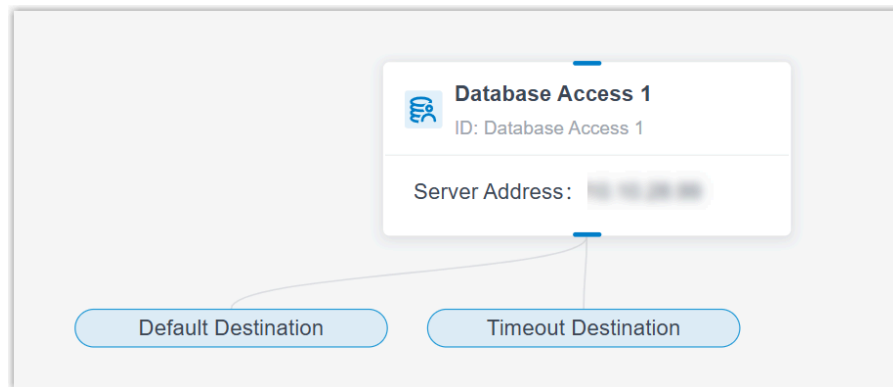
4. Configure the SQL operation and timeout settings.

Item	Description
Statement Type	Select the type of SQL operation. • Query: Execute a SQL statement that returns one or more rows of data (e.g. SELECT). • NonQuery: Execute a SQL statement that modifies data but doesn't return any rows (e.g. INSERT, DELETE, UPDATE). • Scalar: Execute a SQL statement that returns a single value (e.g. SUM, COUNT).
Timeout(s)	Set the time to wait in seconds before terminating the attempt to execute a SQL statement.
SQL Statement	Enter the SQL statement. Note: • For Microsoft SQL Server and PostgreSQL, the SQL statement must include the schema name. For example, if the schema is pbx, you should enter <code>SELECT * FROM pbx.user;</code>. • You can use expressions in the SQL statement for dynamic values. For example, insert <code>SELECT email_addr FROM contact WHERE number='\$Session.contactNumber';</code> to return the email address corresponding to the customer's number.

Item	Description
	For more information about functions and variables, see Variables and Functions in Yeastar Expression. If multiple SQL statements are entered, only the first one will be executed.

5. At the bottom-right corner, click **Confirm**.

The configured information appears on the component.



6. **Optional:** To route the message session or trigger other actions when the component's branch conditions are met, hover over the desired branch and click  to add the next component.

For more information about components that support to be connected, see [Component connections](#).

Component variables

When a **Database Access** component is added to a message flow, the system stores database information in variables. These variables can be referenced in expression-supported components (such as **Condition**) to retrieve the database information and respond accordingly based on the result.



Note:

Since **Database Access** component can be added multiple times in a message flow, an index is appended to the unique ID of each component (e.g. Database Access 1, Database Access 2) based on the order in which the components are added. To retrieve data from the right component, the component variable must contain the corresponding index.

Variable	Type	Description	Example Value
<code>\$DatabaseAccess{index}.queryResult</code>	Object	The table returned by a SQL query.  Tip: You can use the variable with the following functions to retrieve data from the result. • <code>GET_TABLE_ROW_COUNT(table)</code>: Get the total number of rows. • <code>GET_TABLE_CELL_VALUE(table, row, column)</code>: Get the value at the specified row and column.	"demo%40yeastar.com"
<code>\$DatabaseAccess{index}.nonQueryResult</code>	Integer	The number of rows affected by a non-query SQL statement.	99
<code>\$DatabaseAccess{index}.scalarResult</code>	String	The single value returned by a scalar SQL query.	"10"

Component connections

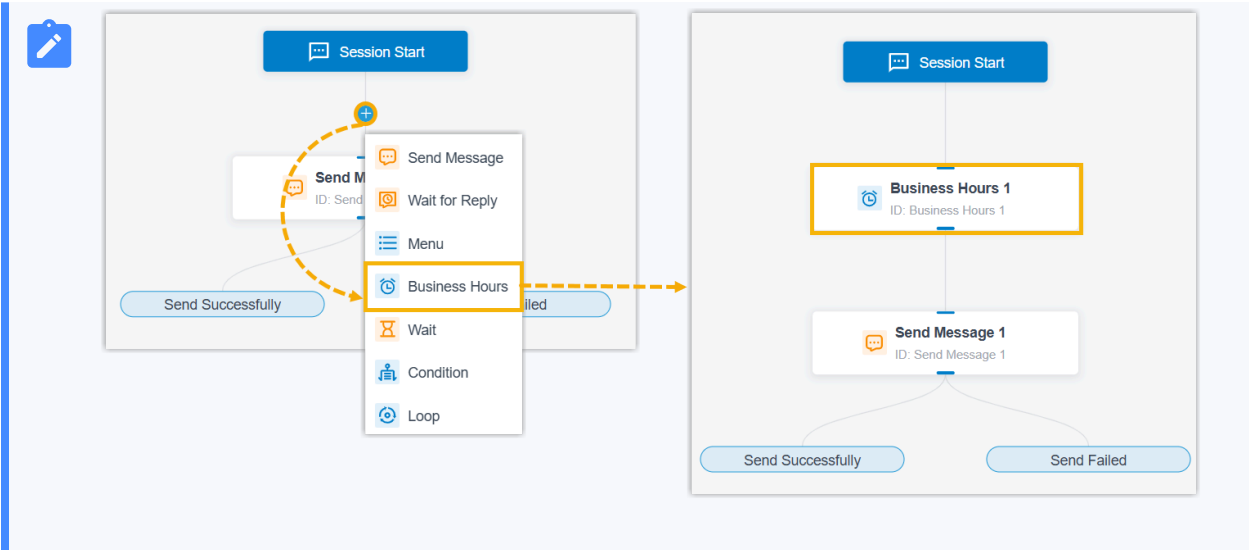
Database Access component comes with two built-in branches **Default Destination** and **Timeout Destination**. Each branch can be connected on **one** component, which can be any of the components listed below.



Note:

If a component is already connected to the branch, and you need to insert a new component between them, hover over the line, click  and select the target component. The selected component is inserted between them with connections auto-established on both ends, without the need to delete the connection, add the component and reconnect manually.

End components **Transfer** and **Close** are unavailable for this insertion method.



Component	Description
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message.
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates. Important: This component is available only when the message flow is only applicable to WhatsApp channel type. Edit * Name MF Remark * Applicable to ☐ SMS ☒ WhatsApp ☐ Facebook ☐ Live Chat Cancel Save

Component	Description
	For more information, see Send WhatsApp Template Message .
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services.

Component	Description
	For more information, see HTTP Request .
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

HTTP Request

This topic provides an overview of the HTTP Request component, and describes its configuration, variables, as well as supported connections.

Component introduction

HTTP Request component allows you to send HTTP requests to external Web servers, enabling data exchange with third-party services. The response is stored in variables, which can be used as input for subsequent components or for condition evaluation.

Supported request types

- GET
- HEAD
- OPTIONS
- POST
- PUT
- TRACE
- DELETE

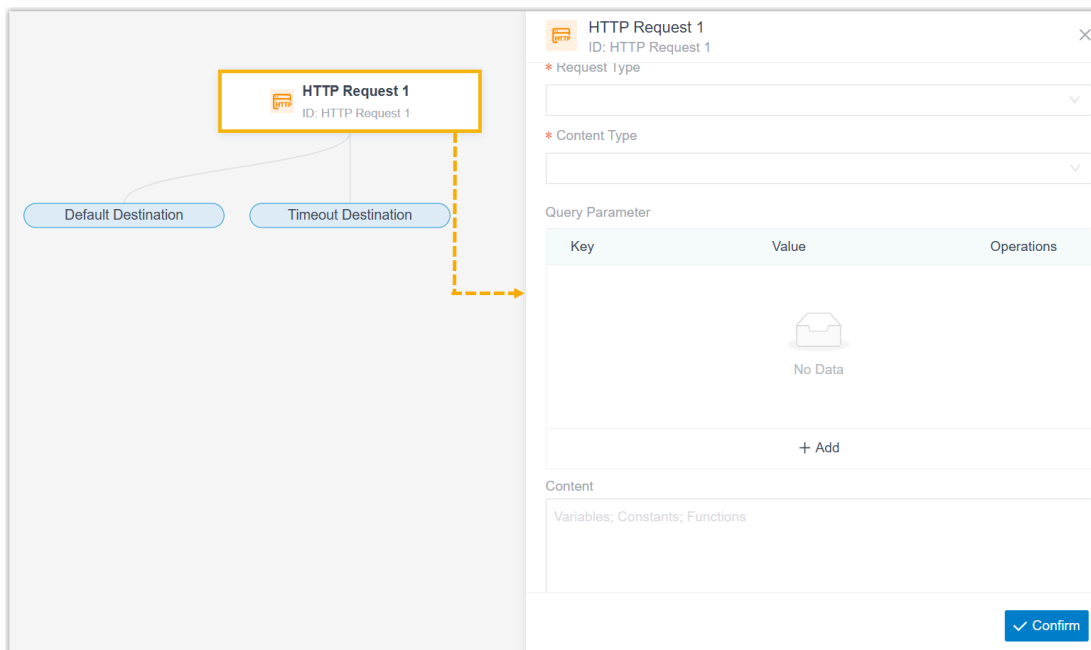
Supported content types

- application/javascript
- application/json
- application/x-www-form-urlencoded
- application/pdf
- application/xml
- application/zip
- multipart/form-data
- text/css

- text/html
- text/plain
- image/png
- image/jpeg
- image/gif

Component configuration

1. After adding **HTTP Request** component to a message flow, click **HTTP Request** component to proceed with the configuration.



2. In the pop-up right panel, configure the HTTP request and click **Confirm**.



Note:

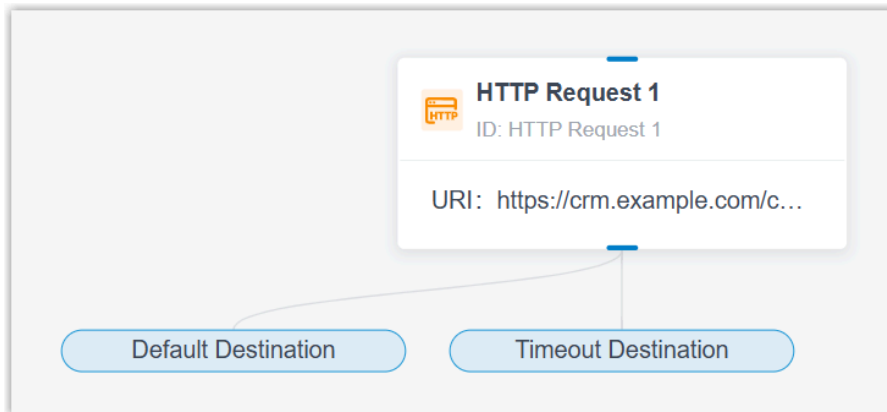
The **URI**, **Query Parameter**, **Content** and **Header** can be defined via expressions that may include variables, constants, and functions.

You can click **FX** to configure expressions. In expression editor, enter **FX** to view and select system-supported functions, and enter **\$** to view and select variables already stored in the current message flow.

For more information about functions and variables, see [Variables and Functions in Yeastar Expression](#).

Item	Description
URI	Specify the target URI to which the HTTP request will be sent.
Request Type	Select the HTTP method to use. • GET • HEAD • OPTIONS • POST • PUT • TRACE • DELETE
Content Type	Select a content type. • application/javascript • application/json • application/x-www-form-urlencoded • application/pdf • application/xml • application/zip • multipart/form-data • text/css • text/html • text/plain • image/png • image/jpeg • image/gif
Query Parameter	Click Add to add query parameters.  Note: It is recommended not to exceed 30 parameters.
Content	Enter the request body to be sent to the server.  Note: If variables or functions are inserted in the content box, enclose them in single quotes <code>' '</code>.
Timeout(s)	Set the time in seconds to wait for the HTTP request to complete.
Header	Click Add to specify HTTP headers for the request.  Note: It's recommended not to exceed 30 headers.

The configured information appears on the component.



3. **Optional:** To route the message session or trigger other actions when the component's branch conditions are met, hover over the desired branch and click  to add the next component.

For more information about components that support to be connected, see [Component connections](#).

Component variables

When a **HTTP Request** component is added to a message flow, the system stores response information in variables. These variables can be referenced in expression-supported components (such as **Condition**) to retrieve the response information and respond accordingly based on the result.



Note:

Since **HTTP Request** component can be added multiple times in a message flow, an index is appended to the unique ID of each component (e.g. HTTP Request 1, HTTP Request 2) based on the order in which the components are added. To retrieve data from the right component, the component variable must contain the corresponding index.

Variable	Type	Description	Example Value
<code>\$HttpRequest{<i>index</i>}.responseContent</code>	String	The body content returned from HTTP response.	<code>{"authenticated": true, "user": "10000"}</code>
<code>\$HttpRequest{<i>index</i>}.responseStatusCode</code>	String	The HTTP status code returned from the request.	<code>"200"</code>

Variable	Type	Description	Example Value
		 Note: When using functions to evaluate HTTP status code, ensure that the constant is a numeric code, without any accompanying text.	

Component connections

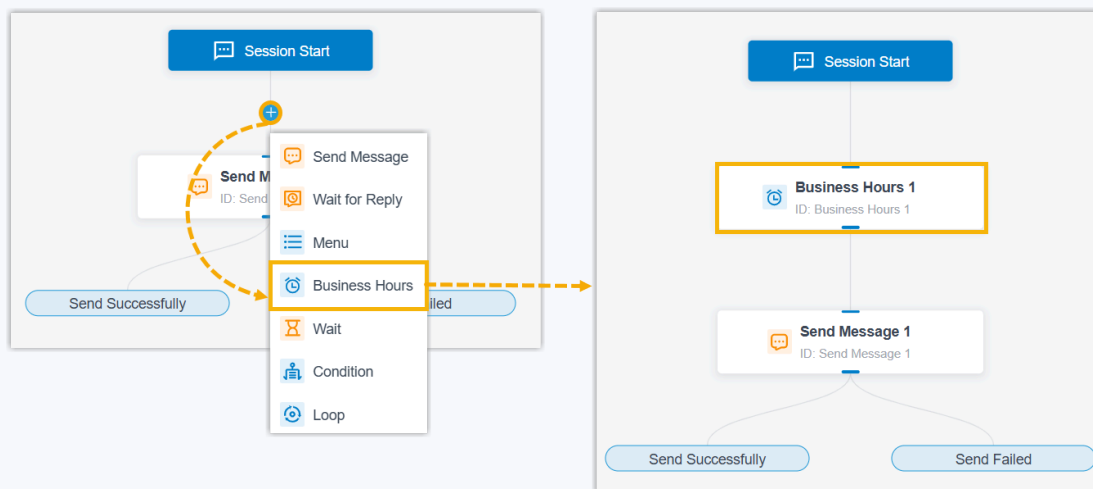
HTTP Request component comes with two built-in branches **Default Destination** and **Timeout Destination**. Each branch can be connected on **one** component, which can be any of the components listed below.

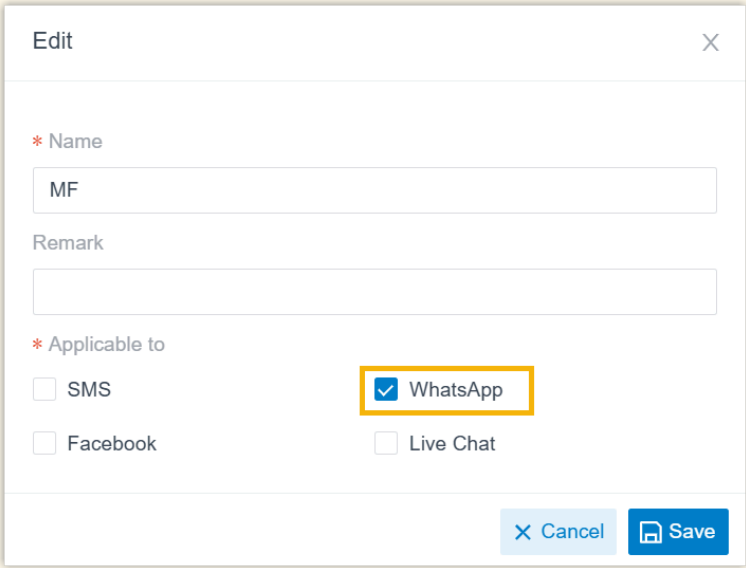


Note:

If a component is already connected to the branch, and you need to insert a new component between them, hover over the line, click  and select the target component. The selected component is inserted between them with connections auto-established on both ends, without the need to delete the connection, add the component and reconnect manually.

End components **Transfer** and **Close** are unavailable for this insertion method.



Component	Description
Send Message	Send Message component enables the system to automatically send messages to customers. For more information, see Send Message.
Send WhatsApp Template Message	Send WhatsApp Template Message component enables the system to automatically send messages to customers using specified WhatsApp message templates.  Important: This component is available only when the message flow is only applicable to WhatsApp channel type.  For more information, see Send WhatsApp Template Message.
Wait for Reply	Wait for Reply component allows you to set the reply timeout and route the message session depending on whether a reply is received. For more information, see Wait for Reply.
Menu	Menu component allows you to present the customer with a set of menu options, and route the message session based on the value that they reply. For more information, see Menu.
Business Hours	Business Hours component allows you to route the message session to different destinations based on the time of day. For more information, see Business Hours.

Component	Description
Wait	Wait component enables the system to automatically route the message session to the connected component upon reply timeout. For more information, see Wait.
Condition	Condition component allows routing the message session based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly based on the loop strategy. For more information, see Loop.
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services. For more information, see HTTP Request.
Transfer	Transfer component allows you to transfer the message session to a specific destination to terminate the current message flow. For more information, see Transfer.
Close	Close component enables the system to automatically disconnect the message session to terminate the current message flow. For more information, see Close.

End Components

Transfer

This topic provides an overview of the Transfer component, and describes its configuration as well as supported connections.

Component introduction

Transfer component is one of the end components to terminate the current message flow. It allows you to transfer the message session to one of the following destinations when the session is routed to the component.

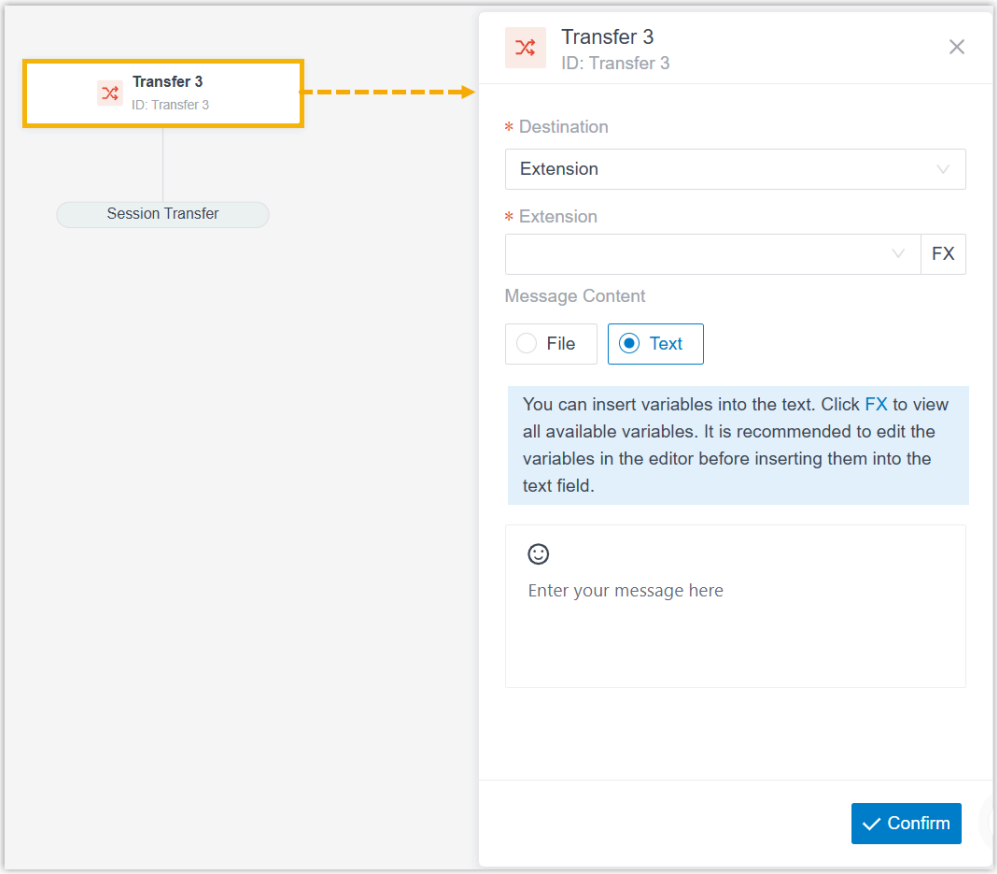
- Extension
- Message Queue
- Third-Party Message Analytics Platform(Transmitted via API)

**Note:**

This destination type is available only when the PBX server has been integrated with the third-party message analytics platform [via API](#).

Component configuration

1. After adding **Transfer** component to a message flow, click the component to proceed with the configuration.



2. In the pop-up right panel, complete the following settings.
- a. Specify the destination where the session will be transferred as needed.

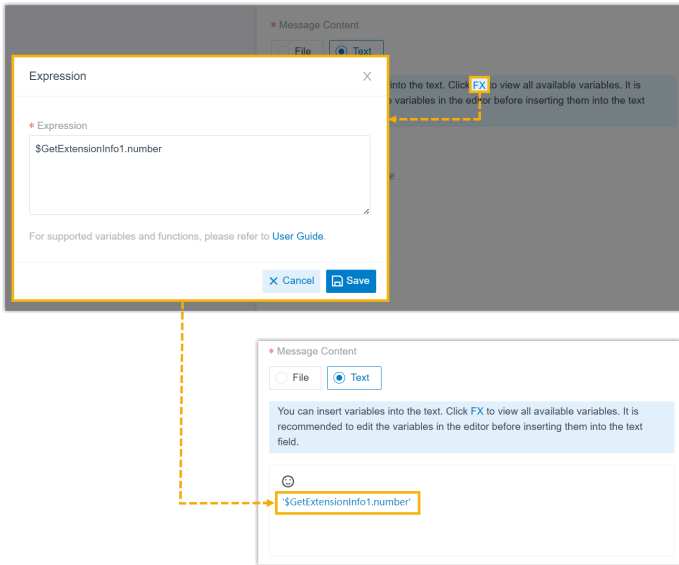
Destination	Instruction
Extension	To transfer the message session to an extension, select Extension from the Destination drop-down list and specify the target extension using one of the following methods. • Select an extension from the drop-down list. • Click FX beside the drop-down list, enter an expression and click Save.  Note: An expression may include variables, constants, and functions. You can enter FX to view and select system-supported functions, and enter \$ to view and select variables already stored in the current message flow.

Destination	Instruction
	For more information about functions and variables, see Variables and Functions in Yeastar Expression. * Extension FX Expression X * Expression \$GetExtensionInfo1.number For supported variables and functions, please refer to User Guide. X Cancel Save

Destination	Instruction
	* Message Queue FX Expression * Expression FX_GET_LIST_ITEM(\$DatabaseAccess1.queryResult,2) For supported variables and functions, please refer to User Guide. Cancel Save
Third-Party Message Analytics Platform (Transmitted via API)	To automatically transfer the message session to a third-party message analytics platform for advanced processing, select Third-Party Message Analytics Platform (Transmitted via API) from the Destination drop-down list.  Note: This option is available only when the PBX server has been integrated with the third-party message analytics platform via API.

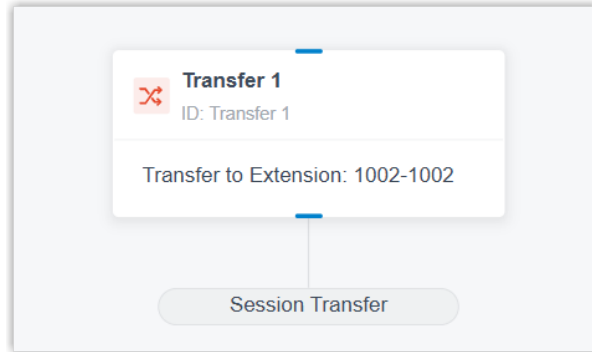
b. Fill in the message content using one of the following methods to inform customer of the message transfer.

Method	Instruction
File upload	Click File and upload a file that includes the content to be sent.  Note: The file must not exceed 100 MB.

Method	Instruction
	
Text input	Click Text and fill in the content to be sent in the text field. • To insert fixed text, enter plain text and emojis directly. • To insert expression, click FX, enter an expression and click Save.  Note: An expression may include variables, constants, and functions. You can enter FX to view and select system-supported functions, and enter \$ to view and select variables already stored in the current message flow. For more information about functions and variables, see Variables and Functions in Yeastar Expression.  The expression added in the editor is automatically filled in the text field.

c. At the bottom-right corner, click **Confirm**.

The configured destination appears on the component.



Component connections

Transfer component comes with a built-in **Session Transfer** branch. This branch can be connected on **one** developer component, which can be any of the components listed below.

Component	Description
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services. For more information, see HTTP Request.

Close

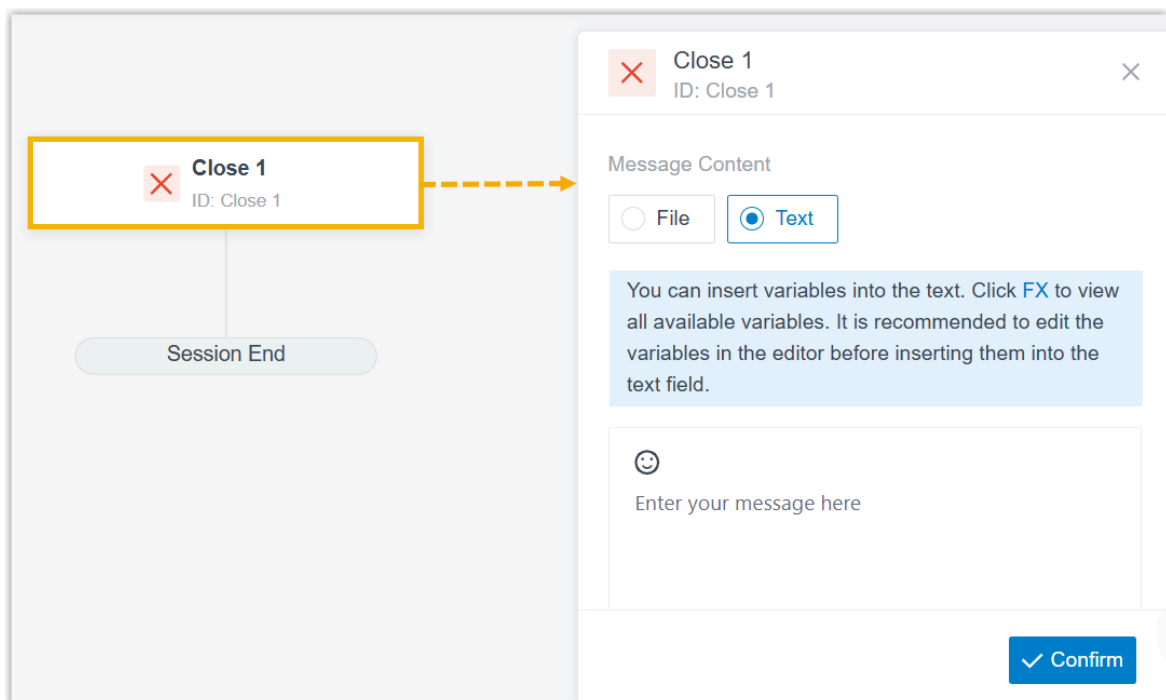
This topic provides an overview of the Close component, and describes its configuration as well as supported connections.

Component introduction

Close component is one of the end components to terminate the current message flow. When the session are routed to the component, the session will be disconnected.

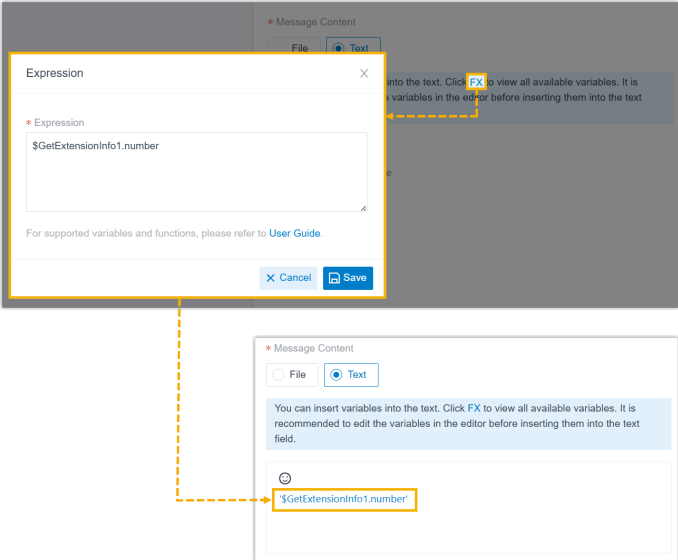
Component configuration

1. After adding **Close** component to a message flow, click the component to proceed with the configuration.



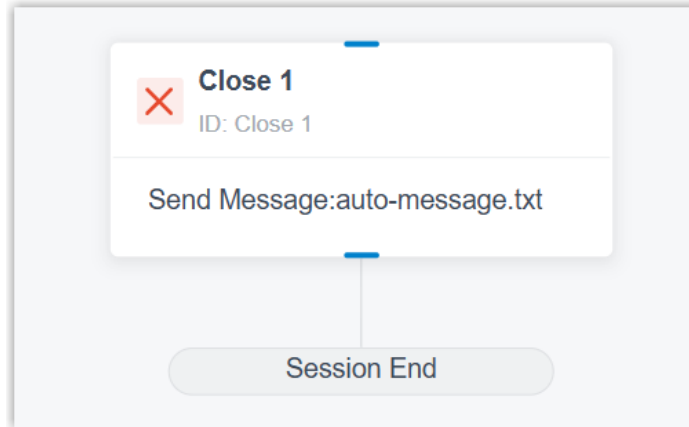
2. In the pop-up right panel, complete the following settings.
 - a. Fill in the message content using one of the following methods to inform customer of the message session close.

Method	Instruction
File upload	Click File and upload a file that includes the content to be sent.  Note: The file must not exceed 100 MB.

Method	Instruction
	
Text input	Click Text and fill in the content to be sent in the text field. • To insert fixed text, enter plain text and emojis directly. • To insert expression, click FX, enter an expression and click Save.  Note: An expression may include variables, constants, and functions. You can enter FX to view and select system-supported functions, and enter \$ to view and select variables already stored in the current message flow. For more information about functions and variables, see Variables and Functions in Yeastar Expression.  The expression added in the editor is automatically filled in the text filed.

b. At the bottom-right corner, click **Confirm**.

The configured content appears on the component.



Component connections

Close component comes with a built-in **Session End** branch. This branch can be connected on **one** developer component, which can be any of the components listed below.

Component	Description
Internal Data Ops	Internal Data Ops component allows you to query data from PBX-native database. For more information, see the following topics: • Get Extension Presence Status • Get Extension Info • Get Company Contact Info
Email Sender	Email Sender component enables the system to automatically send emails to a recipient via configured email server and record the send logs, enabling real-time notifications. For more information, see Email Sender.
Database Access	Database Access component allows you to interact with database during a message flow. You can execute SQL operations to retrieve or update data as needed. For more information, see Database Access.
HTTP Request	HTTP Request component allows you to send HTTP requests to external web servers, enabling data exchange with third-party services. For more information, see HTTP Request.

Expression Basics

Variables and Functions in Yeastar Expression

This topic introduces the supported variables and functions in Yeastar Message Flow Designer.

Variable

Variables can be referenced in expression-supported components (such as **Condition**) to dynamically retrieve data stored in a message flow. There are two types of variables in Yeastar Message Flow Designer:

- **Session Variable:** Session variables store information about the current message session. These variables are globally accessible throughout a message flow and do not depend on any specific components.

For more information, see [Session Variables](#).

- **Component Variable:** Component variables hold the values returned by specific components. These variables become available after the corresponding components are added to a message flow.

For more information, see [Component variables](#).

Session Variables

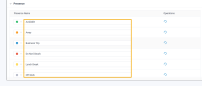
Variable	Type	Description	Example Value
\$Session.contact Number	String	The customer's number.	"15812340987"
\$Session.channelNumber	String	The number of the message channel.	"LC00113"
\$Session.contact Name	String	The full name of the contact that matches the customer's number.	"Leo Bell"
\$Session.contact FirstName	String	The first name of the contact that matches the customer's number.	"Leo"
\$Session.contact LastName	String	The last name of the contact that matches the customer's number.	"Bell"

Component variables

**Note:**

- After the components listed in the table below are added to a message flow, the system stores their response information in corresponding variables, which can thereafter be accessed by expression-supported components throughout the message flow.
- Since the same component can be added multiple times in a message flow, an index is appended to the unique ID of each component (e.g. Menu 1, Menu 2) based on the order in which the components are added. To retrieve data from the right component, the component variable must contain the corresponding index.

Variable	Type	Description	Example Value
Menu component			
\$Menu{ <i>index</i> }.receivedMessage	String	The message content that the customer sends in response to the menu.	"1"
Wait for Reply component			
\$WaitForReply{ <i>index</i> }.receivedMessage	String	The message content that the customer replies.	"Transfer to a live agent."
Get Extension Presence Status component			
\$GetExtensionStatus{ <i>index</i> }.currentProfileName	String	Name of the extension's current presence status.	"Available"
 Note: When using functions to evaluate an extension's presence status, ensure that the status constant matches one of the values			

Variable	Type	Description	Example Value
		 defined in PBX Settings > Preferences > Presence . 	
\$GetExtensionStatus{ <i>index</i> }.currentProfile	String	Name and additional information of the extension's current presence status.	"Available, work"
\$GetExtensionStatus{ <i>index</i> }.isUnavailable	Boolean	Whether the extension is unregistered.	False
Get Extension Info component			
\$GetExtensionInfo{ <i>index</i> }.number	String	Extension number.	"1001"
\$GetExtensionInfo{ <i>index</i> }.name	String	Extension name.	"Phillip Huff"
\$GetExtensionInfo{ <i>index</i> }.email	String	Extension's email address.	"phillip@sample.com"
\$GetExtensionInfo{ <i>index</i> }.mobileNumber	String	Extension's mobile number.	"15880123456"
Get Company Contact Info component			
\$GetCompanyContactInfo{ <i>index</i> }.contactName	String	Contact name.	"Leo Poll"
\$GetCompanyContactInfo{ <i>index</i> }.company	String	Company name.	"YS"
\$GetCompanyContactInfo{ <i>index</i> }.email	String	Email address.	"leo@ys.com"
\$GetCompanyContactInfo{ <i>index</i> }.phonebookList	String	The phonebook(s) to which the contact belongs.	"All"
\$GetCompanyContactInfo{ <i>index</i> }.remark	String	Remark about the contact.	"remark"
\$GetCompanyContactInfo{ <i>index</i> }.business	String	Business number.	"01082345678"

Variable	Type	Description	Example Value
\$GetCompanyContactInfo{index}.business2	String	Business number.	"02160123456"
\$GetCompanyContactInfo{index}.mobile	String	Mobile number.	"13800001234"
\$GetCompanyContactInfo{index}.mobile2	String	Mobile number.	"15900005678"
\$GetCompanyContactInfo{index}.home	String	Home number.	"01087654321"
\$GetCompanyContactInfo{index}.home2	String	Home number.	"075523456789"
\$GetCompanyContactInfo{index}.businessFax	String	Business fax number.	"01082345679"
\$GetCompanyContactInfo{index}.homeFax	String	Home fax number.	"02160123457"
\$GetCompanyContactInfo{index}.other	String	Other phone number.	"4008880000"
Database Access component			
\$DatabaseAccess{index}.queryResult	Object	The table returned by a SQL query.  Tip: You can use the variable with the following functions to retrieve data from the result. • GET_TABLE_ROW_COUNT(table): Get the total number of rows. • GET_TABLE_CELL_VALUE(table, row, c)	"demo%40yeastar.com"

Variable	Type	Description	Example Value
		 column): Get the value at the specified row and column.	
\$DatabaseAccess{ <i>index</i> }.nonQueryResult	Integer	The number of rows affected by a non-query SQL statement.	99
\$DatabaseAccess{ <i>index</i> }.scalarResult	String	The single value returned by a scalar SQL query.	"10"
HTTP Request component			
\$HttpRequest{ <i>index</i> }.responseContent	String	The body content returned from HTTP response.	{"authenticated": true, "user": "10000"}
\$HttpRequest{ <i>index</i> }.statusCode	String	The HTTP status code returned from the request.  Note: When using functions to evaluate HTTP status code, ensure that the constant is a numeric code, without any accompanying text.	"200"

Function

Functions are used for condition evaluation and data processing, enabling flexible control of the message flow. The supported functions are as follows.

AND

Perform a logical operation across two or more conditions, returning `True` only if all conditions are met at the same time.

Syntax

```
FX_AND(condition1, condition2, ...)
```

- **Request Type:** Boolean
- **Response Type:** Boolean

Example

```
FX_AND(FX_EQUAL($Menu1.receivedMessage, "1"),
FX_EQUAL($Session.contactNumber, "15812340987"))
```

In this example, the function returns `True` only when the message content that the customer sends in response to the menu1 is "1" and the customer's number is "15812340987".

OR

Perform a logical operation across two or more conditions, returning `True` if any of the conditions is met.

Syntax

```
FX_OR(condition1, condition2, ...)
```

- **Request Type:** Boolean
- **Response Type:** Boolean

Example

```
FX_OR(FX_EQUAL($Session.contactNumber, "15812340987"),
FX_NOT($GetExtensionStatus1.isUnavailable))
```

In this example, the function returns `True` if at least one of the following conditions are met:

- The customer's phone number is "15812340987".
- The extension is registered.

NOT

Perform a logical operation on a single condition to get the opposite of its Boolean result.

Syntax

```
FX_NOT(condition)
```

- **Request Type:** Boolean
- **Response Type:** Boolean

Example

```
FX_NOT($GetExtensionStatus1.isUnavailable)
```

In this example, the function returns `True` when the extension is not registered.

EQUAL

Perform a logical check to determine whether two values are equal (both in value and data type).

Syntax

```
FX_EQUAL(value1, value2)
```

- **Request Type:** Any
- **Response Type:** Boolean

Example

```
FX_EQUAL($Session.contactNumber, "15812340987")
```

In this example, the function returns `True` if the customer's number is exactly "15812340987".

NOT_EQUAL

Perform a logical check to determine whether two values are not equal.

Syntax

```
FX_NOT_EQUAL(value1, value2)
```

- **Request Type:** Any
- **Response Type:** Boolean

Example

```
FX_NOT_EQUAL($GetExtensionStatus1.isUnavailable, True
)
```

In this example, the function returns `True` if the extension is registered.

CONTAINS

Perform a logical operation to check if the first string contains the second string.

Syntax

```
FX_CONTAINS(value1, value2)
```

- **Request Type:** String
- **Response Type:** Boolean

Example

```
FX_CONTAINS($GetCompanyContactInfo1.company, "Yeastar
")
```

In this example, the function returns `True` if the company name contains "Yeastar".

GREATER_THAN

Perform a logical check to determine if the first value is greater than the second value.

Syntax

```
FX_GREATER_THAN(value1, value2)
```

- **Request Type:** Integer
- **Response Type:** Boolean

Example

```
FX_GREATER_THAN($DatabaseAccess1.nonQueryResult, 20)
```

In this example, the function returns `True` if the number of rows affected by a non-query SQL statement exceeds 20.

GREATER_THAN_OR_EQUAL

Perform a logical check to determine if the first value is greater than or equal to the second value.

Syntax

```
FX_GREATER_THAN_OR_EQUAL(value1, value2)
```

- **Request Type:** Integer
- **Response Type:** Boolean

Example

```
FX_GREATER_THAN_OR_EQUAL($DatabaseAccess1.nonQueryResult, 20)
```

In this example, the function returns `True` if the number of rows affected by a non-query SQL statement is 20 or more.

LESS_THAN

Perform a logical check to determine if the first value is less than the second value.

Syntax

```
FX_LESS_THAN(value1, value2)
```

- **Request Type:** Integer
- **Response Type:** Boolean

Example

```
FX_LESS_THAN($DatabaseAccess1.nonQueryResult, 20)
```

In this example, the function returns `True` if the number of rows affected by a non-query SQL statement is less than 20.

LESS_THAN_OR_EQUAL

Perform a logical check to determine if the first value is less than or equal to the second value.

Syntax

```
FX_LESS_THAN_OR_EQUAL(value1, value2)
```

- **Request Type:** Integer
- **Response Type:** Boolean

Example

```
FX_LESS_THAN_OR_EQUAL($DatabaseAccess1.nonQueryResult, 20)
```

In this example, the function returns `True` if the number of rows affected by a non-query SQL statement is 20 or fewer.

TO_BOOLEAN

Perform a logical operation to convert a value to a Boolean.



Note:

This function returns `False` if the parameter is the string `"0"`, `"False"` or `null`.

Syntax

```
FX_TO_BOOLEAN(value)
```

- **Request Type:** Any
- **Response Type:** Boolean

Example

```
FX_TO_BOOLEAN($GetExtensionStatus1.currentProfileName)
```

In this example, the function converts the value retrieved from the **Get Extension Presence Status** component into a Boolean

value. The function returns `True` whenever the component returns a valid status value.

CONCATENATE

Perform a string operation to concatenate every string parameter and return the resulting string.

Syntax

```
FX_CONCATENATE(string1, string2, ...)
```

- **Request Type:** String
- **Response Type:** String

Example

```
FX_CONCATENATE("Customer Name:",  
               "$Session.contactName")
```

In this example, the function combines the string "Customer Name: " and the full name of the contact corresponding to the customer. E.g. "Customer Name: Leo Bell".

TRIM

Perform a string operation to remove leading and trailing invisible characters (e.g. spaces, new lines, etc.) from the given value.

Syntax

```
FX_TRIM(value)
```

- **Request Type:** String
- **Response Type:** String

Example

```
FX_TRIM(" Hello ")
```

In this example, the function removes spaces from the word and returns "Hello".

LEFT

Perform a string operation to extract a specified number of characters from the beginning of the given text.

Syntax

```
FX_LEFT(text, count)
```

- **Request Type:**
 - `text`: String
 - `count`: Integer
- **Response Type:** String

Example

```
FX_LEFT("13800138000", 3)
```

In this example, the function extracts the first three characters from the given text and returns "138".

MID

Perform a string operation to extract a substring from a given text starting at a specified position with a specified length.

Syntax

```
FX_MID(text, start_position, length)
```

- **Request Type:**
 - `text`: String
 - `start_position`: Integer
 - `length`: Integer
- **Response Type:** String

Example

```
FX_MID("13800138000", 4, 4)
```

In this example, the function extracts 4 characters starting from the 4th character of the text, and returns "0013".

RIGHT

Perform a string operation to extract a specified number of characters from the end (right side) of a given text.

Syntax

```
FX_RIGHT( text, count )
```

- **Request Type:**
 - text: String
 - count: Integer
- **Response Type:** String

Example

```
FX_RIGHT( "13800138000", 4 )
```

In this example, the function extracts the last 4 characters from the given text, and returns "8000".

UPPER

Perform a string operation to convert all characters in the given text to upper-case.

Syntax

```
FX_UPPER( text )
```

- **Request Type:** String
- **Response Type:** String

Example

```
FX_UPPER( "Yeastar" )
```

In this example, the function converts all letters in the text to uppercase, and returns "YEASTAR".

LOWER

Perform a string operation to convert all characters in the given text to lower-case.

Syntax

```
FX_LOWER(text)
```

- **Request Type:** String
- **Response Type:** String

Example

```
FX_LOWER("Yeastar")
```

In this example, the function converts all letters in the text to lowercase, and returns "yeastar".

REPLACE

Perform a string operation to replace a specified substring with another substring in the given text.

Syntax

```
FX_REPLACE(text, target, replacement)
```

- **Request Type:**
 - text: String
 - target: String
 - replacement: String
- **Response Type:** String

Example

```
FX_REPLACE("+8613800138000", "+86", "")
```

In this example, the function removes the country code +86 from the phone number, and returns "13800138000".

REPLACE_REG_EXP

Perform a string operation to replace substrings that match a regular expression pattern with a specified replacement string.

Syntax

```
FX_REPLACE_REG_EXP(text, expression, replacement)
```

- **Request Type:**
 - `text`: String
 - `expression`: String
 - `replacement`: String
- **Response Type:** String

Example

```
FX_REPLACE_REG_EXP("Call123", "[0-9]", "X")
```

In this example, the function matches all numeric characters using the regular expression `[0-9]` and replaces them with `x`, and returns `"CallXXX"`.

JSON_GET_STRING

Perform an operation to retrieve a string value from a JSON object based on the specified key.

Syntax

```
FX_JSON_GET_STRING(JSONObject, key)
```

- **Request Type:**
 - `JSONObject`: String
 - `key`: String
- **Response Type:** String

Example

```
FX_JSON_GET_STRING($HttpRequest1.responseContent, "user")
```

In this example, the body content in JSON format returned by the **HTTP Request 1** component is shown below.

```
{
  "authenticated": true,
  "user": "10000"
}
```

The function returns the value for the `"user"` key - `"10000"`.

TO_STRING

Perform an operation to convert a value to a string.

Syntax

```
FX_TO_STRING(value)
```

- **Request Type:** Any
- **Response Type:** String

Example

```
FX_TO_STRING(10086)
```

In this example, the function converts 10086 into a string parameter, and returns "10086".

NOW

Return the current date and time as a DateTime object.

Syntax

```
FX_NOW( )
```

- **Request Type:** Null. This function takes no parameters.
- **Response Type:** dateTime

Example

```
FX_NOW( )
```

In this example, the function returns the current system date and time - "2026-08-12 17:29:08".

LEN

Perform a string operation to return the number of characters in the given text.

Syntax

```
FX_LEN(text)
```

- **Request Type:** String
- **Response Type:** Integer

Example

```
FX_LEN("13800138000")
```

In this example, the function returns 11, which indicates the length of the text.

SUM

Perform an operation to sum two or more 32-bit integer values.

Syntax

```
FX_SUM(value1, value2, ...)
```

- **Request Type:** Integer
- **Response Type:** Integer

Example

```
FX_SUM(100, 200, 300)
```

In this example, the function sums the three values and returns 600.

SUM_LONG

Perform an operation to sum two or more 64-bit integer values.

Syntax

```
FX_SUM_LONG(value1, value2, ...)
```

- **Request Type:** Integer
- **Response Type:** Integer

Example

```
FX_SUM_LONG(10000000000, 25000000000)
```

In this example, the function sums the two values and returns 3500000000.

NEGATIVE

Perform an operation to return the negative value of a 32-bit integer.

Syntax

```
FX_NEGATIVE(value)
```

- **Request Type:** Integer
- **Response Type:** Integer

Example

```
FX_NEGATIVE(100)
```

In this example, the function returns -100.

NEGATIVE_LONG

Perform an operation to return the negative value of a 64-bit integer.

Syntax

```
FX_NEGATIVE_LONG(value)
```

- **Request Type:** Integer
- **Response Type:** Integer

Example

```
FX_NEGATIVE_LONG(1000000000)
```

In this example, the function returns -1000000000.

MULTIPLY

Perform an operation to multiply two or more 32-bit integer values.

Syntax

```
FX_MULTIPLY(value1, value2, ...)
```

- **Request Type:** Integer
- **Response Type:** Integer

Example

```
FX_MULTIPLY(10, 20, 3)
```

In this example, this function returns 600.

MULTIPLY_LONG

Perform an operation to multiply two or more 64-bit integer values.

Syntax

```
FX_MULTIPLY_LONG(value1, value2, ...)
```

- **Request Type:** Integer
- **Response Type:** Integer

Example

```
FX_MULTIPLY_LONG(100000, 20000)
```

In this example, the function returns 2000000000.

DIVIDE

Perform an operation to divide the first 32-bit integer value by the second 32-bit integer value.

Syntax

```
FX_DIVIDE(value1, value2)
```

- **Request Type:** Integer
- **Response Type:** Integer

Example

```
FX_DIVIDE(100, 20)
```

In this example, the function divides 100 by 20 and returns 5.

DIVIDE_LONG

Perform an operation to divide the first 64-bit integer value by the second 64-bit integer value.

Syntax

```
FX_DIVIDE_LONG(value1, value2)
```

- **Request Type:** Integer
- **Response Type:** Integer

Example

```
FX_DIVIDE_LONG(10000000000, 2000000000)
```

In this example, the function divides 10000000000 by 2000000000 and returns 5.

ABS

Perform an operation to return the absolute (non-negative) value of a 32-bit integer.

Syntax

```
FX_ABS(value)
```

- **Request Type:** Integer
- **Response Type:** Integer

Example

```
FX_ABS(-100)
```

In this example, the functions returns 100.

ABS_LONG

Perform an operation to return the absolute (non-negative) value of a 64-bit integer.

Syntax

```
FX_ABS_LONG(value)
```

- **Request Type:** Integer
- **Response Type:** Integer

Example

```
FX_ABS_LONG( -100000000000)
```

In this example, the function returns 100000000000.

GET_TABLE_ROW_COUNT

Perform an operation to return the number of rows in a table.

Syntax

```
FX_GET_TABLE_ROW_COUNT(table)
```

- **Request Type:** Any
- **Response Type:** Integer

Example

```
FX_GET_TABLE_ROW_COUNT($DatabaseAccess1.queryResult)
```

In this example, the function returns the number of rows from the result returned by the **Database Access 1** component.

GET_LIST_ITEM_COUNT

Perform an operation to return the number of items in a given list.

Syntax

```
FX_GET_LIST_ITEM_COUNT(list)
```

- **Request Type:** Any
- **Response Type:** Integer

Example

```
FX_GET_LIST_ITEM_COUNT($DatabaseAccess1.queryResult)
```

In this example, the function returns the number of items from the result returned by the **Database Access 1** component.

JSON_GET_INTEGER

Perform an operation to retrieve an integer value from a JSON object based on the specified key.

Syntax

```
FX_JSON_GET_INTEGER(JSONObject, key)
```

- **Request Type:**
 - `JSONObject`: String
 - `key`: String
- **Response Type:** Integer

Example

```
FX_JSON_GET_INTEGER($HttpRequest1.responseContent, "httpStatus")
```

In this example, the function returns the value of the `httpStatus` key from the JSON content returned by the **HTTP Request 1** component.

TO_INTEGER

Perform an operation to convert a value to a 32-bit integer.

Syntax

```
FX_TO_INTEGER(value)
```

- **Request Type:** String
- **Response Type:** Integer

Example

```
FX_TO_INTEGER("12345")
```

In this example, the function converts the string "12345" into the integer 12345.

TO_LONG

Perform an operation to convert a value to a 64-bit integer.

Syntax

```
FX_TO_LONG(value)
```

- **Request Type:** String
- **Response Type:** Integer

Example

```
FX_TO_LONG("1234567890123")
```

In this example, the function converts the string "1234567890123" into the integer 1234567890123.

GET_TABLE_CELL_VALUE

Perform an operation to retrieve the value from a table cell at the specified row and column.

Syntax

```
FX_GET_TABLE_CELL_VALUE(table, row, column)
```

- **Request Type:**
 - `table`: Any
 - `row`: Integer
 - `column`: Integer
- **Response Type:** Object

Example

```
FX_GET_TABLE_CELL_VALUE($DatabaseAccess1.queryResult, 0, 1)
```

In this example, the function returns the value from the first row and second column in the result returned by the **Database Access 1** component.

GET_LIST_ITEM

Perform an operation to retrieve the value from a list at the specified index position.

Syntax

```
FX_GET_LIST_ITEM(list, index)
```

- **Request Type:**

- `list`: Any
- `index`: Integer

- **Response Type:** Object

Example

```
FX_GET_LIST_ITEM($DatabaseAccess1.queryResult, 2)
```

In this example, the function returns the third item in the result returned by the **Database Access 1** component.

GET_JSON_ARRAY_ITEM

Perform an operation to retrieve the value from an object at the specified index in a JSON array.

Syntax

```
FX_GET_JSON_ARRAY_ITEM(jsonData, arrayKey, index, fieldName)
```

- **Request Type:**

- `jsonData`: String
- `arrayKey`: String
- `index`: Integer
- `fieldName`: String

- **Response Type:** Object

Example

```
FX_GET_JSON_ARRAY_ITEM($HttpRequest1.responseContent, "data", 0, "phoneNumber")
```

In this example, the function returns the value of `phoneNumber` from the first object in the `data` array.

JSON_GET_OBJECT

Perform an operation to retrieve a nested JSON object from a parent JSON object based on the specified key.

Syntax

```
FX_JSON_GET_OBJECT(JSONObject, key)
```

- **Request Type:**

- `JSONObject`: String
- `key`: String

- **Response Type:** Object

Example

```
FX_JSON_GET_OBJECT($HttpRequest1.responseContent, "data")
```

In this example, the function returns the JSON object associated with the `data` key from the JSON content returned by the **HTTP Request 1** component.