

Call Flow Designer Guide

Yeastar P-Series Software Edition

Version: 2.0

Date: 2025-12-03



Contents

Overview.....	1
Call Flow Setup.....	4
Set up a Call Flow from Template.....	4
Set up a Call Flow from Scratch.....	11
Set up a Call Flow from Existing Flow.....	16
Call Flow Management.....	23
Export Call Flows.....	23
Enable or Disable a Call Flow.....	24
Components Basics.....	26
Initial Component.....	26
Initial Action.....	26
Call Control Components.....	35
Prompt.....	35
Business Hours.....	43
Menu.....	52
User Input.....	65
Language.....	75
Record.....	78
Dial by Number.....	88
Dial by Name.....	102
Transfer.....	104
Hang Up Call.....	113
Flow Components.....	114
Condition.....	114
Loop.....	119
Developer Component.....	125
Get Extension Presence Status.....	125
Set Extension Presence Status.....	130
Get Queue Agent.....	134
Get Queue Info.....	139

Get Agent Status.....	143
Set Agent Status.....	149
Email Sender.....	154
Database Access.....	159
HTTP Request.....	165
Expression Basics.....	171
Variables and Functions in Yeastar Expression.....	171
Firmware Dependency.....	201

Call Flow Designer Overview

Yeastar Call Flow Designer empowers you to build intelligent call routes with ease. Simply drag and drop ready-made components to create flexible and customizable call flows tailored to your business needs.

Requirements and restrictions

Requirements

- **Firmware:** 83.19.0.110  or later
- **Plan:** Ultimate Plan

Restrictions

- **Maximum number of call flows:** 200
- **Maximum number of components and branches per call flow:** 200
- **Maximum number of steps per a call flow:** 50

Call flow creation methods

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

Supported components

Component	Description
Initial Action	Initial Action component is the entry point of a call flow. It allows you to specify the phone number(s) that callers can dial to initiate the flow, and optionally configure an audio prompt to welcome callers or provide instructions for them. For more information, see Initial Action.
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name.

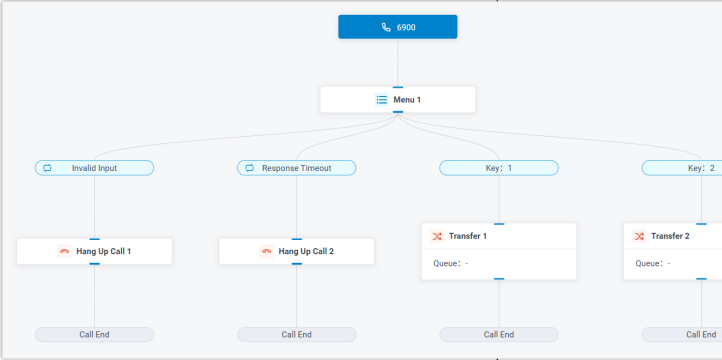
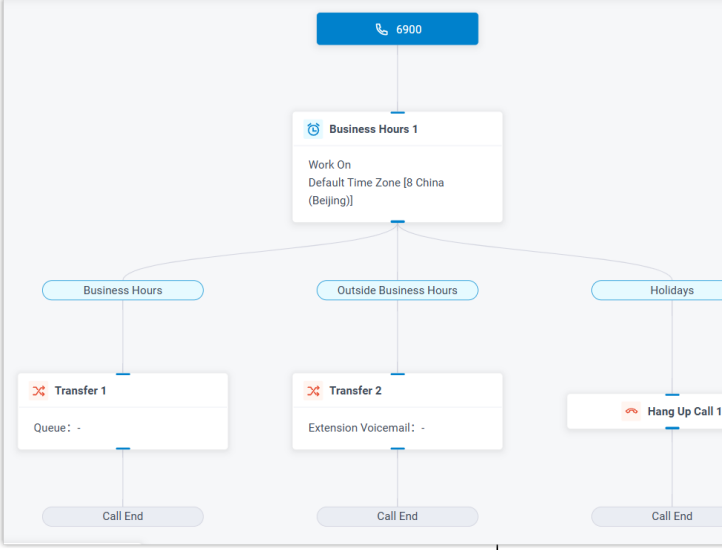
Component	Description
	For more information, see Dial by Name .
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

Call Flow Setup

Set up a Call Flow from Template

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

Template introduction

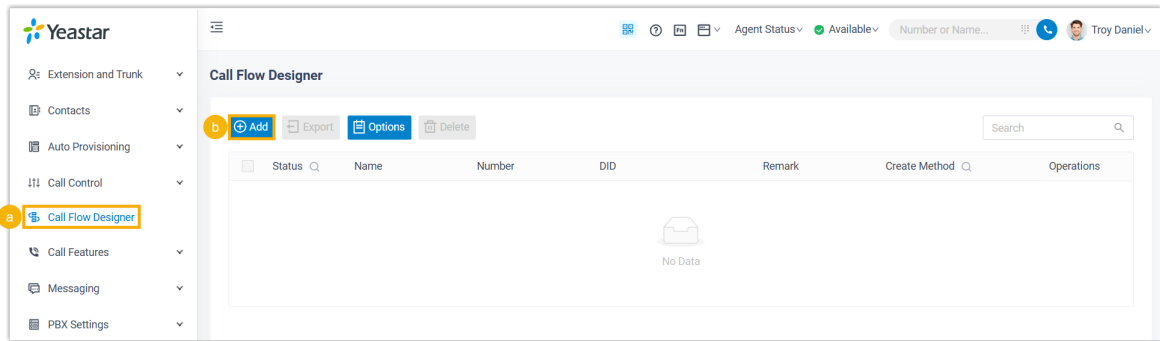
Template	Description	Screenshot
Linear Call Flow	A linear call flow allows you to route calls to different destinations based on the DTMF digit entered by callers.	
Time-based Call Flow	A time-based call flow allows you to route calls to different destinations based on the business hours configured for default time zone.	

Template	Description	Screenshot
Certified Call Flow	A certified call flow allows you route calls to different destinations by evaluating user input against pre-defined expressions.	<pre>graph TD; Start[6900] --> UI[User Input 1]; UI --> C1[Condition 1]; C1 --> E1(FX_EQUAL(\$UserInput1.u...)); C1 --> I1(Invalid Input); E1 --> T1[Transfer 1]; I1 --> H1[Hang Up Call 1]; T1 --> CE1(Call End); H1 --> CE2(Call End);</pre>

Template	Description	Screenshot
Database Authentication	A database authentication call flow allows you to route calls to different destinations by validating user input against database.	<pre> graph TD Start([6900]) --> UI[User Input 1] UI --> DA[Database Access 1] DA --> C1[Condition 1] C1 -- "FX_EQUAL(\$DatabaseAcc...)" --> T1[Transfer 1 Queue: -] C1 -- "Invalid Input" --> P1[Prompt 1] T1 --> CE1([Call End]) P1 --> H1[Hang Up Call 1] H1 --> CE2([Call End]) </pre>

Step 1. Create a call flow from template

1. Access the creation page of call flow.



- a. Log in to PBX web portal, go to **Call Flow Designer**.
 - b. Click **Add**.
2. In the pop-up window, complete the following settings, then click **Save**.

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

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

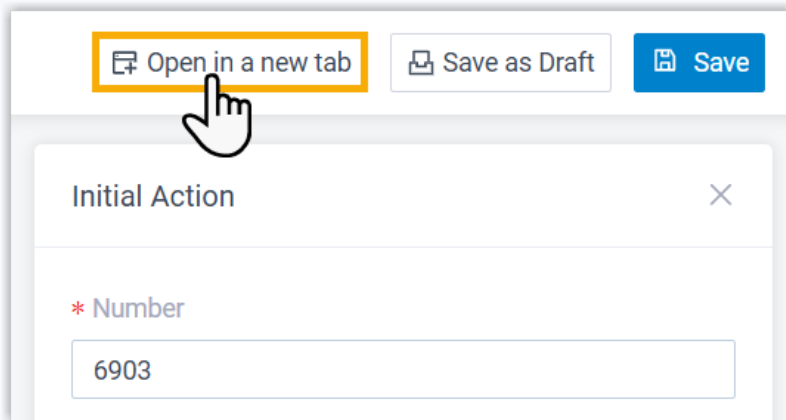
Step 2. Edit components



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.

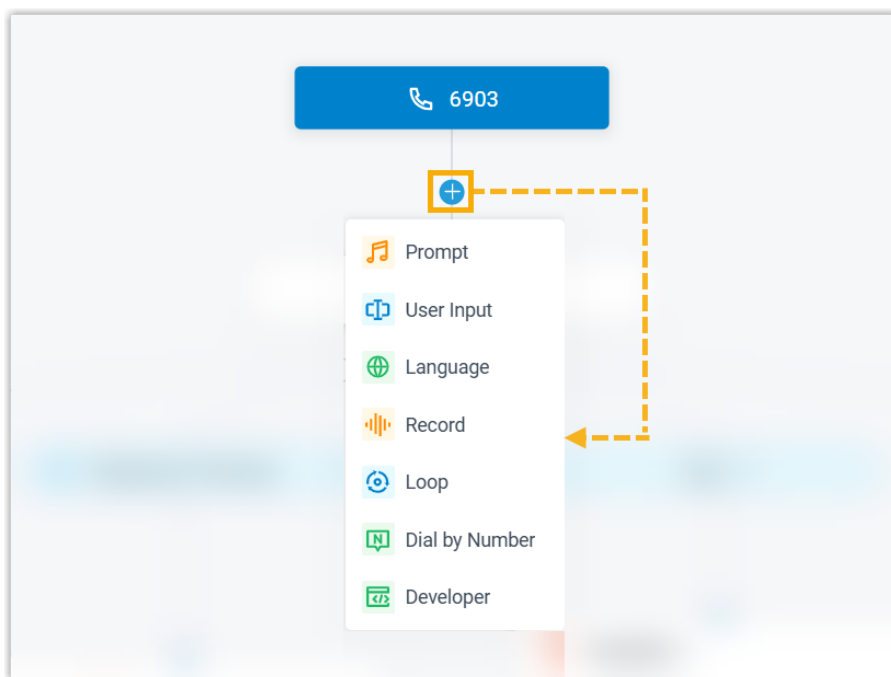
Refer to the table below for introduction and instruction on each component.

Component	Description
Initial Action	Initial Action component is the entry point of a call flow. It allows you to specify the phone number(s) that callers can dial to initiate the flow, and optionally configure an audio prompt to welcome callers or provide instructions for them. For more information, see Initial Action.
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.

Component	Description
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database.

Component	Description
	For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

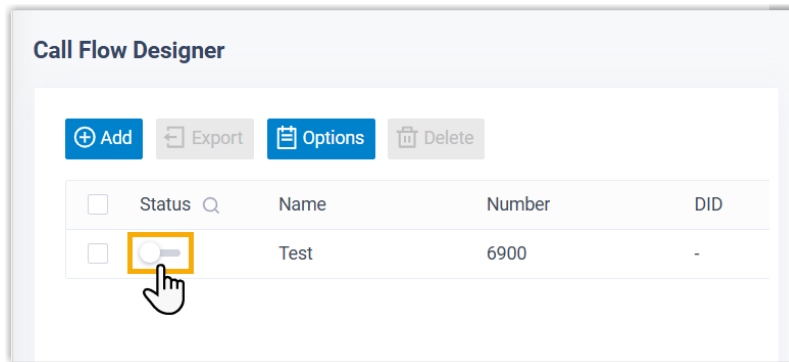
2. **Optional:** To add components, click  on the branch, then select a component.



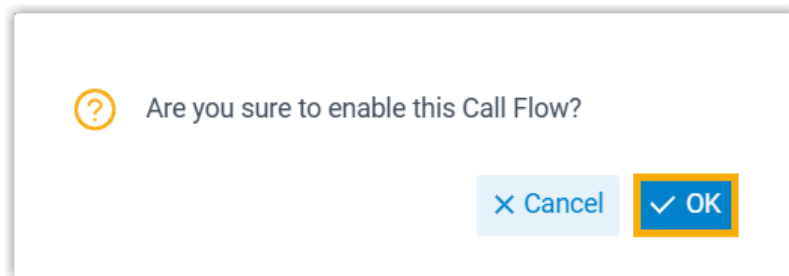
3. At the top-right corner, click **Save**.

Step 3. Enable the call flow

1. On the call flow list, turn on the switch of the desired call flow.



2. In the pop-up window, click **OK**.

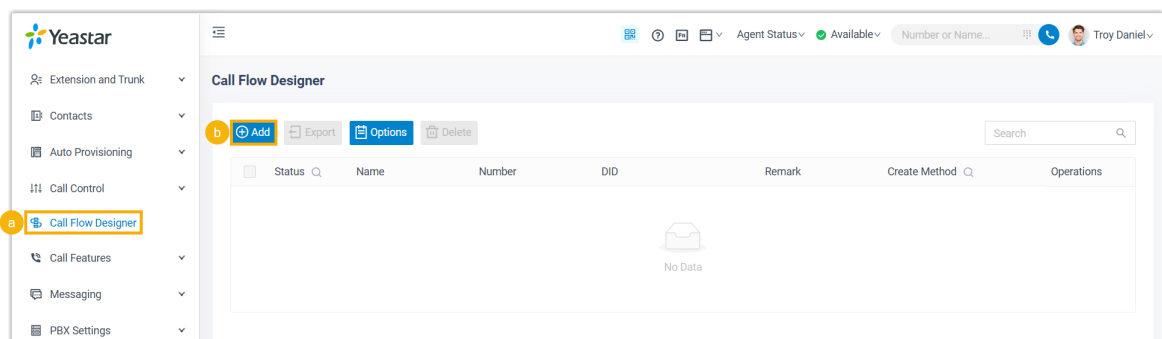


Set up a Call Flow from Scratch

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

Step 1. Create a call flow from scratch

1. Access the creation page of call flow.



a. Log in to PBX web portal, go to **Call Flow Designer**.

b. Click **Add**.

2. In the pop-up window, complete the following settings.

The screenshot shows a modal window titled "Add" with a close button (X) in the top right corner. The window contains three input fields: "Name" (with the text "Test"), "Remark" (empty), and "Create Method" (a dropdown menu showing "Start from Scratch"). A yellow box labeled "a" highlights the "Name", "Remark", and "Create Method" fields. At the bottom right, there are two buttons: "Cancel" (labeled "b") and "Save".

a. Fill in the following information.

- **Name:** Enter a name to help you identify the call flow.
- **Remark:** Add a brief description.
- **Create Method:** Select **Start from Scratch**.

b. Click **Save**.

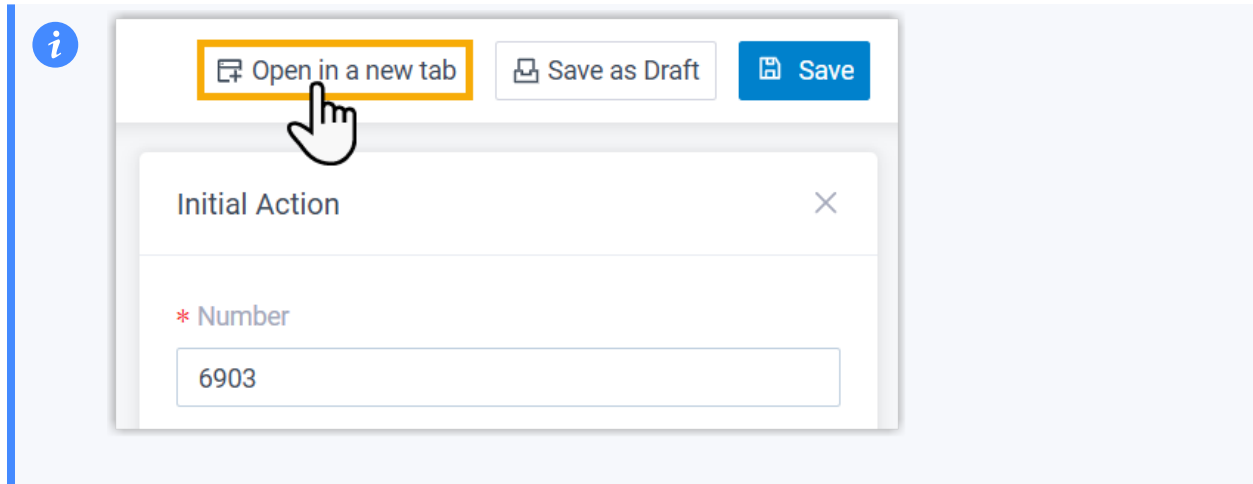
The call flow is created, and the webpage is redirected to the configuration page.

Step 2. Add and connect components



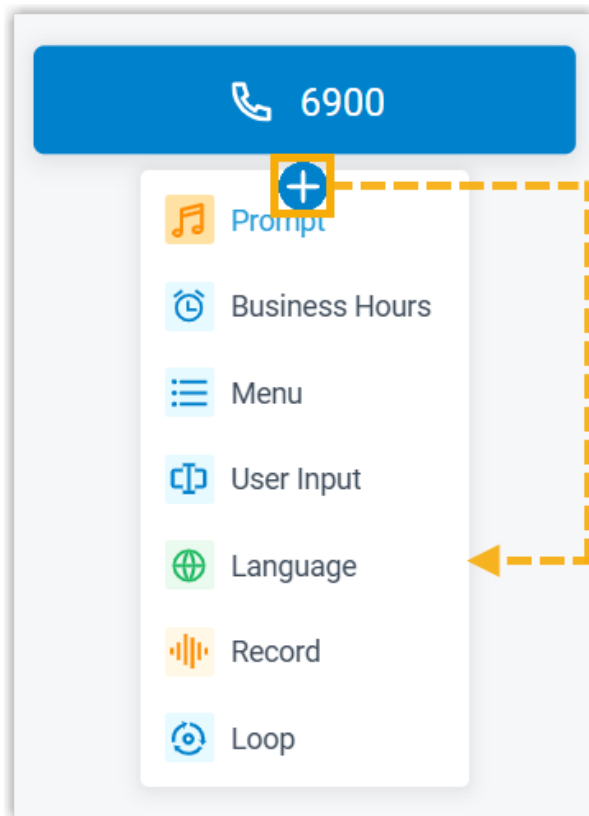
Tip:

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.

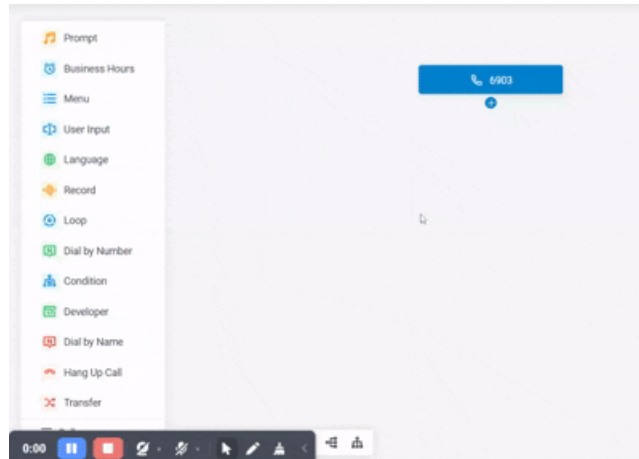


1. Add components using one of the following methods:

- Click  on an component, then select a component.



- From the left components 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
Initial Action	Initial Action component is the entry point of a call flow. It allows you to specify the phone number(s) that callers can dial to initiate the flow, and optionally configure an audio prompt to welcome callers or provide instructions for them. For more information, see Initial Action.
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.

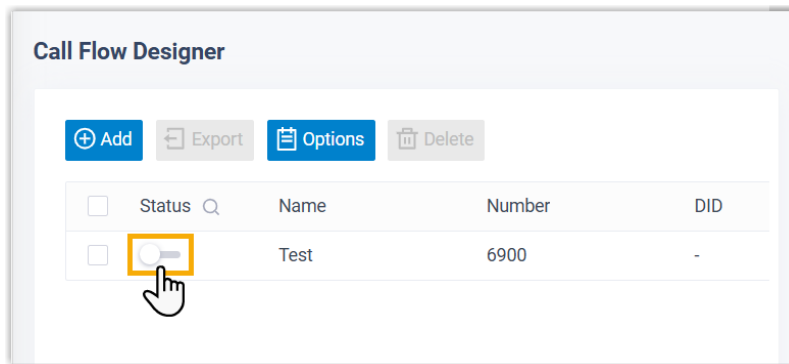
Component	Description
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent

Component	Description
	• Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

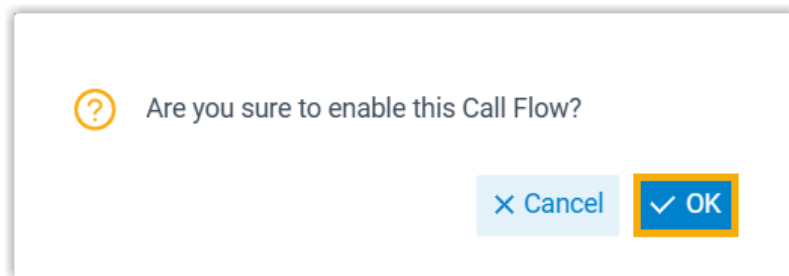
3. At the top-right corner, click **Save**.

Step 3. Enable the call flow

1. On the call flow list, turn on the switch of the desired call flow.



2. In the pop-up window, click **OK**.



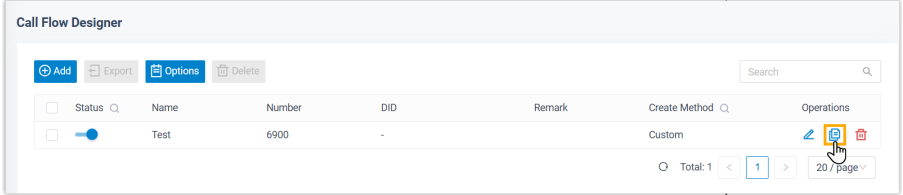
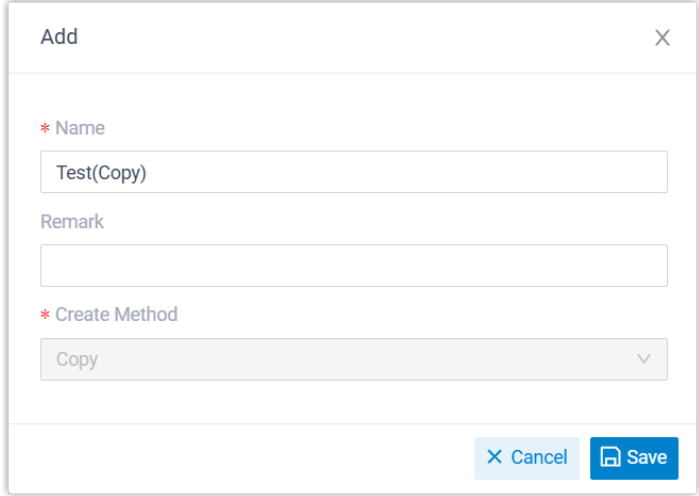
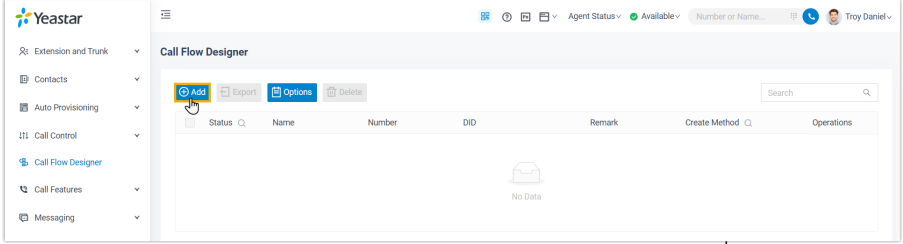
Set up a Call Flow from Existing Flow

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

Step 1. Create a call flow from an existing flow

1. Log in to PBX web portal, go to **Call Flow Designer**.

2. Choose one of the following methods to create a call flow.

Method	Description
Clone an existing call flow	a. On the right of the desired call flow, click .  b. In the pop-up window, edit name and remark for the call flow as needed, then click Save.  The call flow is created, and the webpage is redirected to the configuration page.
Import an existing call flow (JSON file)	a. At the top of the list, click Add.  b. In the pop-up window, complete the following settings, then click Save.

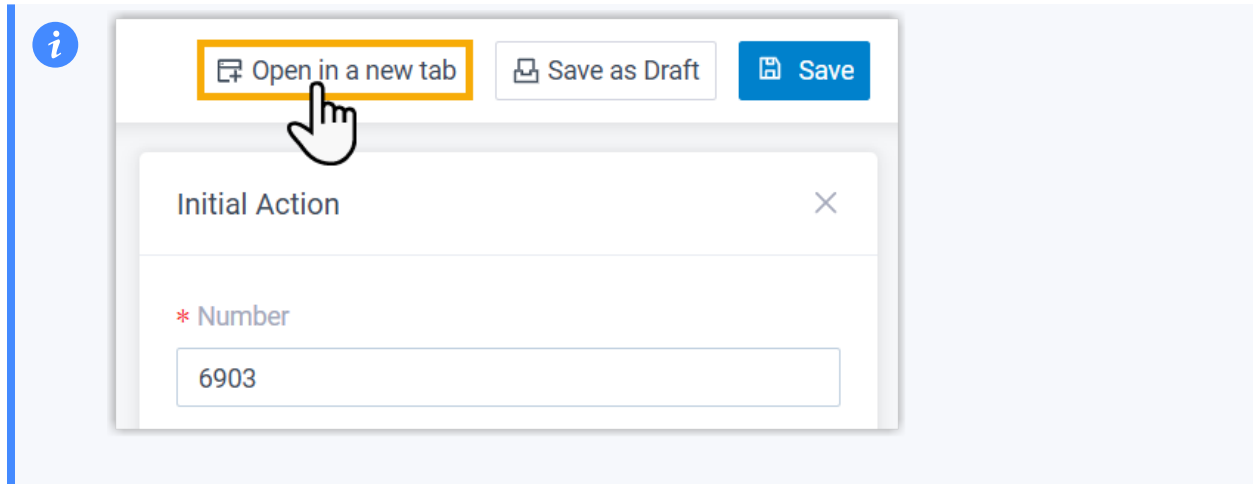
Method	Description
	Add * Name Demo Remark * Create Method Import from File * Upload Call Flow File -Test-37.19.0.90-export-20250615092259-JHpf Upload Cancel Save • Name: Enter a name to help you identify the call flow. • Remark: Add a brief description. • Create Method: Select Import from File. • Upload Call Flow File: Click Upload to select and upload a JSON file. The call flow is created, and the webpage is redirected to the configuration page.

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.

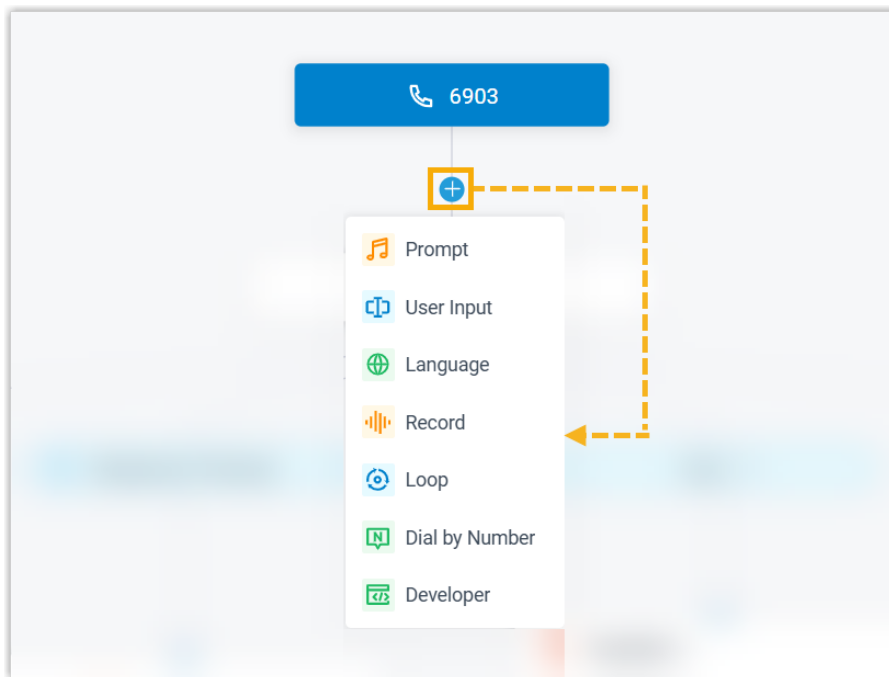
Refer to the table below for introduction and instruction on each component.

Component	Description
Initial Action	Initial Action component is the entry point of a call flow. It allows you to specify the phone number(s) that callers can dial to initiate the flow, and optionally configure an audio prompt to welcome callers or provide instructions for them. For more information, see Initial Action.
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow.

Component	Description
	For more information, see Language .
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status

Component	Description
	• Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

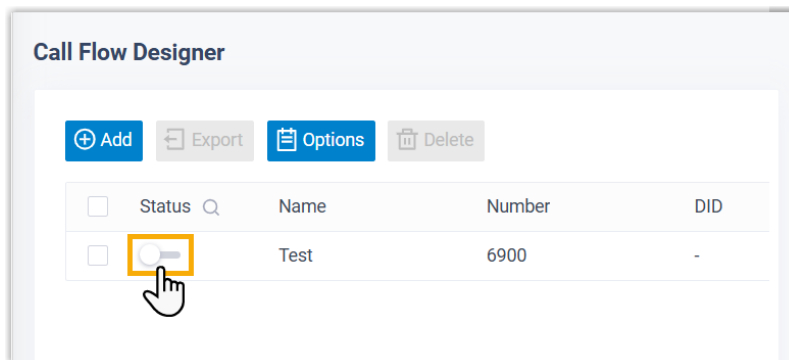
2. **Optional:** To add components, click  on the branch, then select a component.



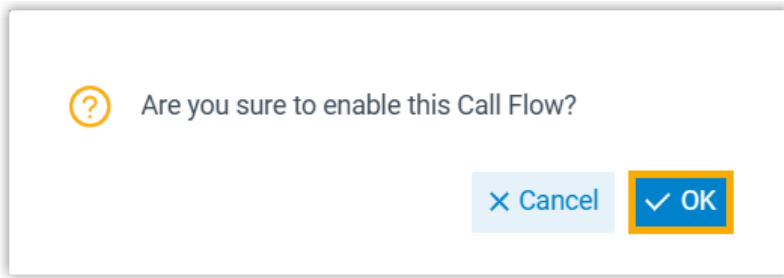
3. At the top-right corner, click **Save**.

Step 3. Enable the call flow

1. On the call flow list, turn on the switch of the desired call flow.



2. In the pop-up window, click **OK**.



Call Flow Management

Export Call Flows

Call flows configured on Yeastar P-Series Software Edition can be exported and imported to another P-Series PBX System. This topic describes how to export call flows.

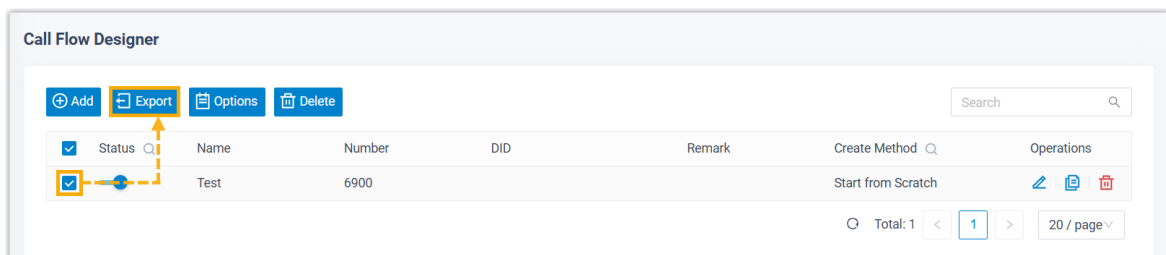


Important:

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

Procedure

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



Result

- If you export a single call flow, it is saved as a `.json` file.
- If you export multiple call 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 Call Flow from a JSON File](#).

Enable or Disable a Call Flow

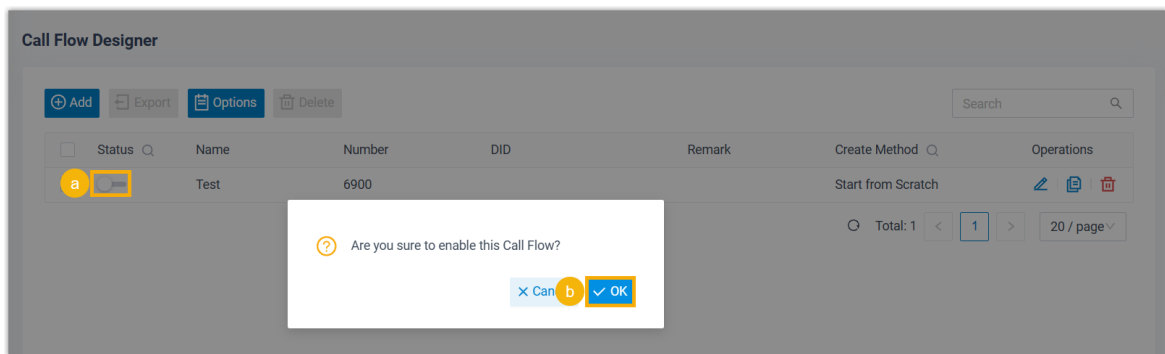
Enable or disable a call flow to determine how incoming calls to the associated number are handled.

Enable a call flow

When a call flow is enabled, incoming calls to the **call flow number** or the **associated DID number** will be routed to the call flow.

To enable a call flow, perform the following operations.

1. Log in to PBX web portal, go to **Call Flow Designer**.
2. Enable the desired call flow.



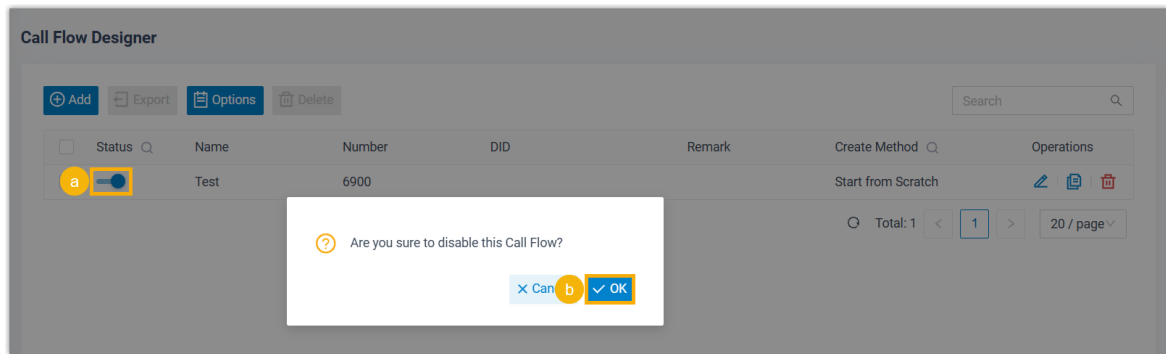
- a. On the call flow list, turn on the switch of the desired call flow.
- b. In the pop-up window, click **OK**.

Disable a call flow

When a call flow is disabled, incoming calls to the **call flow number** or the **associated DID number** will be automatically disconnected.

To disable a call flow, perform the following operations.

1. Log in to PBX web portal, go to **Call Flow Designer**.
2. Disable the desired call flow.



- a. On the call flow list, turn off the switch of the desired call flow.
- b. In the pop-up window, click **OK**.

Components Basics

Initial Component

Initial Action

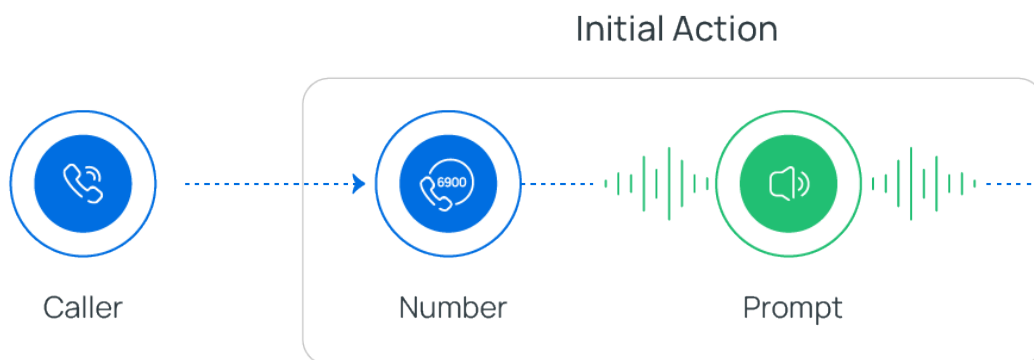
This topic provides an overview of the **Initial Action** component, and describes its restriction, configuration, as well as supported connections.

Component restriction

Each call flow contains exactly one **Initial Action** component. This component is automatically added upon call flow creation and can not be deleted.

Component introduction

Initial Action component is the entry point of a call flow. It allows you to specify the phone number(s) that callers can dial to initiate the flow, and optionally configure an audio prompt to welcome callers or provide instructions for them.



Supported number types

You can assign **extension number** or **trunk DID number(s)** to the component.

Number Type	Description
Extension Number	The system reserves a range of extension numbers for call flow - 6900 - 6999 (max. extensions ≤ 6000) or 50900 - 50999 (max. extensions > 6000).

Number Type	Description
	By default, the first available extension number within the range is assigned upon call flow creation, you can also assign a custom number outside the reserved range.  Note: To change the default extension range for call flow, go to PBX Settings > Preferences > Extension Preferences > Call Flow. Extension Preferences * User Extension 1000 - 5999 * Group Voicemail Extension 6100 - 6199 * Ring Group Extension 6300 - 6399 * Conference Extension 6500 - 6599

* Account Trunk

6700

-

6799

* Call Flow

6900

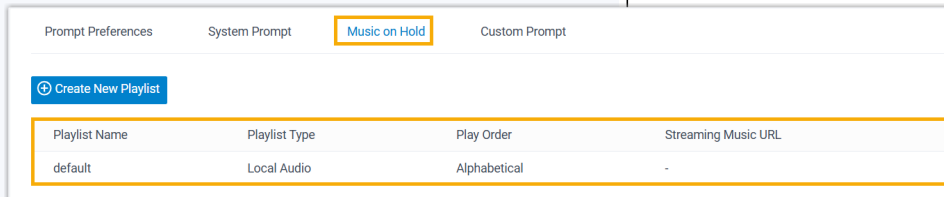
-

6999

Number Type	Description
	 If a DID number is assigned to both call flow and inbound route, the call flow takes precedence. Inbound calls to the DID number will be routed to call flow when it is ENABLED; otherwise, calls will be routed to the destination defined in inbound route.

Supported prompt types

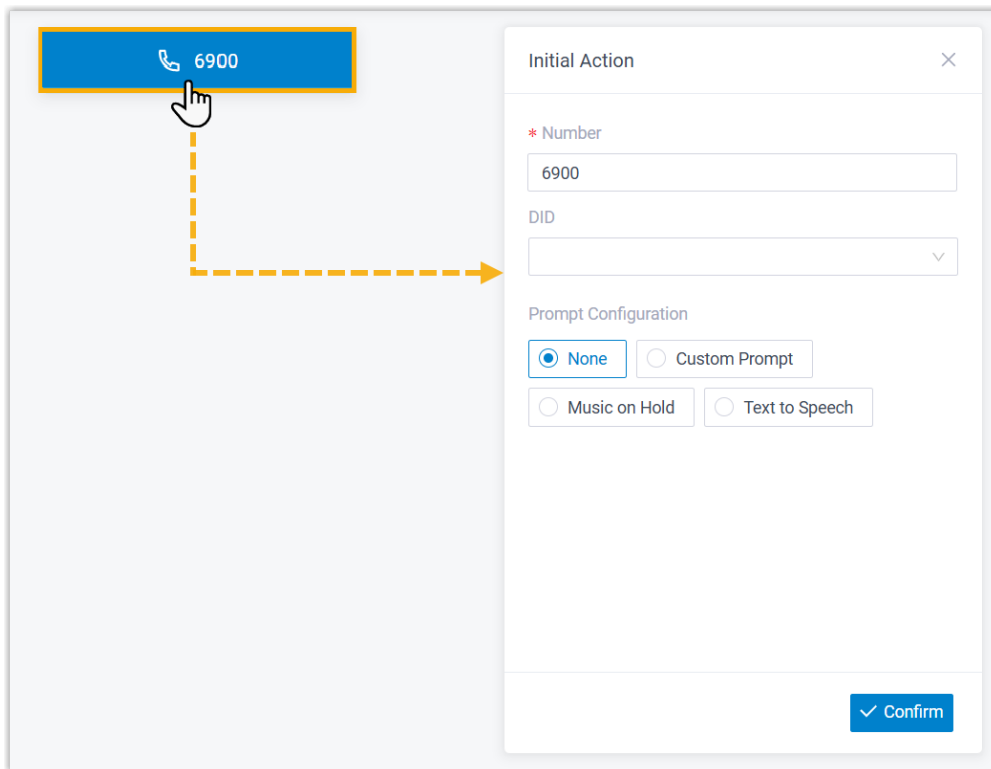
You can add audio prompt(s) or a text-to-speech message to interact with callers.

Prompt Type	Description
Custom Prompt	Choose from existing custom prompts, or create a new one by recording with an extension or uploading an audio file.
Music on Hold (MoH)	Choose from existing MoH.  Note: Before you begin, make sure the desired MoH is configured in PBX Settings > Voice Prompt > Music on Hold. 
Text to Speech (TTS)	Enter the desired text, which will be converted to speech in the selected language and voice style.  Note: Before you begin, make sure the Text-to-Speech feature is enabled in Integrations > AI.

Prompt Type	Description
	 Integrations / AI ☒ Text-to-Speech ☐ Voicemail Transcription

Component configuration

1. After creating a call flow, click **Initial Action** component to proceed with the configuration.



Initial Action

* Number

6900

DID

Prompt Configuration

☒ None ☐ Custom Prompt

☐ Music on Hold ☐ Text to Speech

✓ Confirm

2. Specify the number(s) that callers can dial to initiate the call flow.

Initial Action
✕

* Number

6900

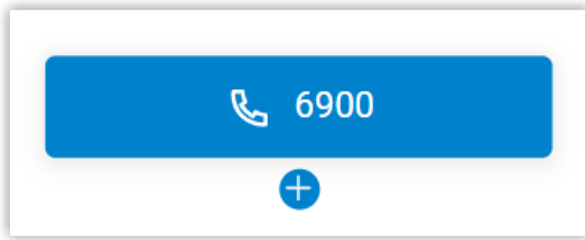
DID

5503301 (YS Tech Support) ✕

Item	Description
Number	Retain the default extension number, or customize a number.  Note: - The custom number can fall outside of the default range (6900 - 6999 / 50900 - 50999). - Once saved, the number can not be modified.
DID	Optional. Select one or more trunk DID number(s) from the drop-down list.  Note: - Up to 500 DID numbers are supported. Note that the more DID numbers assigned, the longer it takes to match inbound call with the call flow. - Inbound calls from DID numbers can be routed to the call flow only in the following cases: - If the received DID number is 7 digits or fewer, it exactly matches a configured DID number. - If the received DID number is longer than 7 digits, its last 7 digits exactly match a configured DID number.

3. **Optional:** Configure audio prompt(s) or a text-to-speech message to welcome callers or provide instructions for them.

Item	Description
Music on Hold (MoH)	Prompt Configuration ☐ None ☐ Custom Prompt a ☒ Music on Hold ☐ Text to Speech * Music on Hold b default a. Select Music on Hold. b. Select an existing MoH from the drop-down list of Music on Hold.  Note: The MoH will be overridden if the next connected component is configured with a prompt.



Component connections

Initial Action component can be connected to **one** component, which can be any of the components listed below.

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.

Component	Description
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

Call Control Components

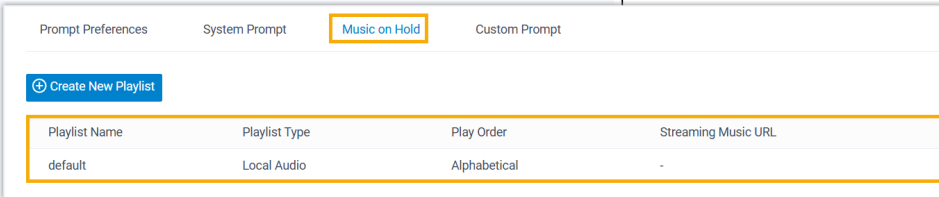
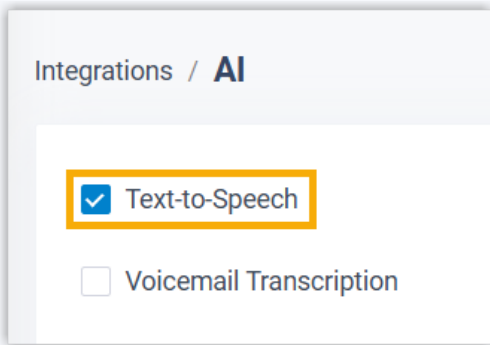
Prompt

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

Component introduction

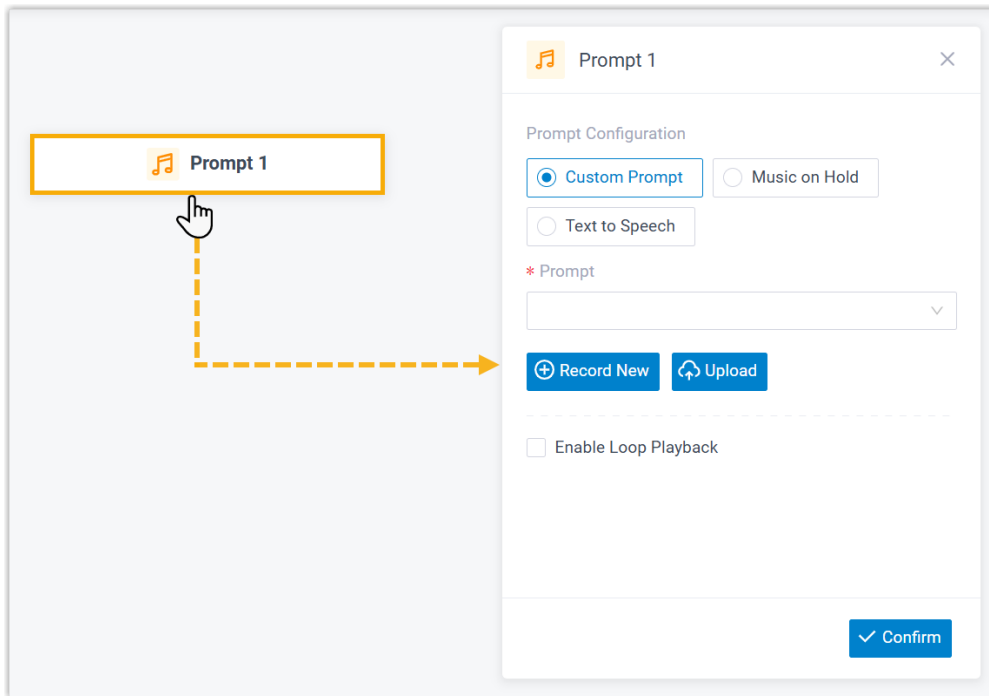
Prompt component allows you to play audio prompt(s) or a text-to-speech message, which can be used to welcome callers or provide them with instructions.

Supported prompt types

Prompt Type	Description
Custom Prompt	Choose from existing custom prompts, or create a new one by recording with an extension or uploading an audio file.
Music on Hold (MoH)	Choose from existing MoH.  Note: Before you begin, make sure the desired MoH is configured in PBX Settings > Voice Prompt > Music on Hold. 
Text to Speech (TTS)	Enter the desired text, which will be converted to speech in the selected language and voice style.  Note: Before you begin, make sure the Text-to-Speech feature is enabled in Integrations > AI. 

Component configuration

1. After adding **Prompt** component to a call flow, click **Prompt** component to proceed with the configuration.



2. Configure audio prompt(s) or a text-to-speech message.

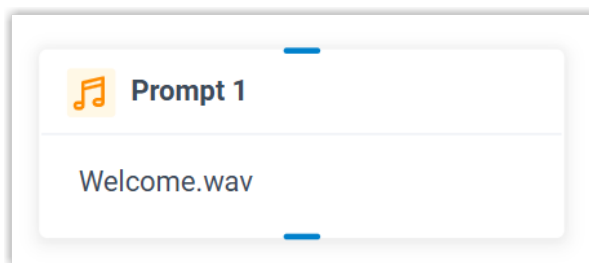
Item	Description
Custom Prompt	a. Click Custom Prompt.

Item	Description
	b. Specify custom prompt(s).  Note: Up to 5 custom prompts are supported. • To choose from an existing prompt, select it from the drop-down list of Prompt. • To use a new prompt, click Record New to record prompt with an extension, or click Upload to upload an audio file.  Note: The new prompt will be saved in PBX Settings > Voice Prompt > Custom Prompt. c. Optional: To play the custom prompt(s) in a loop, select the checkbox of Enable Loop Playback.  Note: The looped prompt will be overridden if the next connected component is configured with a prompt.
Music on Hold	a. Select Music on Hold. b. Select an existing MoH from the drop-down list of Music on Hold.  Note: The MoH will be overridden if the next connected component is configured with a prompt.

Item	Description
	 The new speech will be saved to the system and synchronized to PBX Settings > Voice Prompt > Custom Prompt.

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

The selected prompt will appear on the component.



Component variable

When a **Prompt** component with TTS configured is added to a call flow, the system stores TTS result in a variable. The variable can be referenced in expression-supported components, such as **Condition** or **Developer**, to retrieve the output value and flexibly route the call.

**Note:**

Since **Prompt** component can be added multiple times in a call flow, an index is appended to each component (e.g. Prompt 1, Prompt 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
\$Prompt{ <i>index</i> }.ttsResult	String	The Text-to-Speech (TTS) result of the Prompt component. • PromptTTSResult.Success: The text is successfully converted into speech, and the system plays it to caller. • PromptTTSResult.Timeout: The text is NOT converted into speech within the timeout period, and the system routes caller directly to the next component without playing a prompt when the timeout is reached.	\$Prompt1.ttsResult(STRING)=PromptTTSResult.Success

Component connections

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

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.

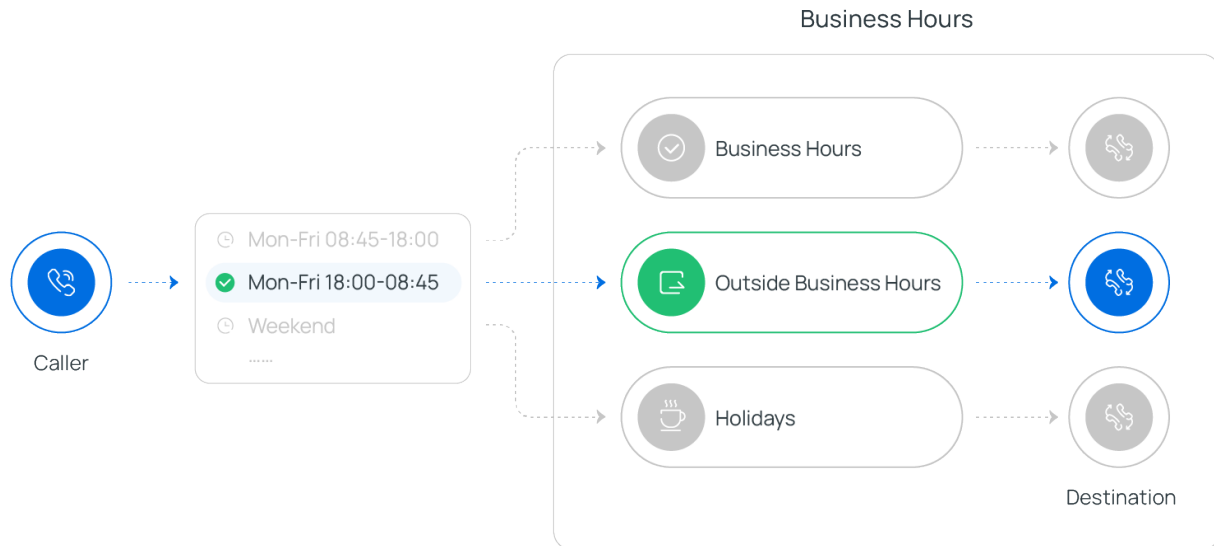
Component	Description
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

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 calls to different destinations based on the time of day. A variety of time-based modes are available to flexibly define office hours and implement time-based call routing.

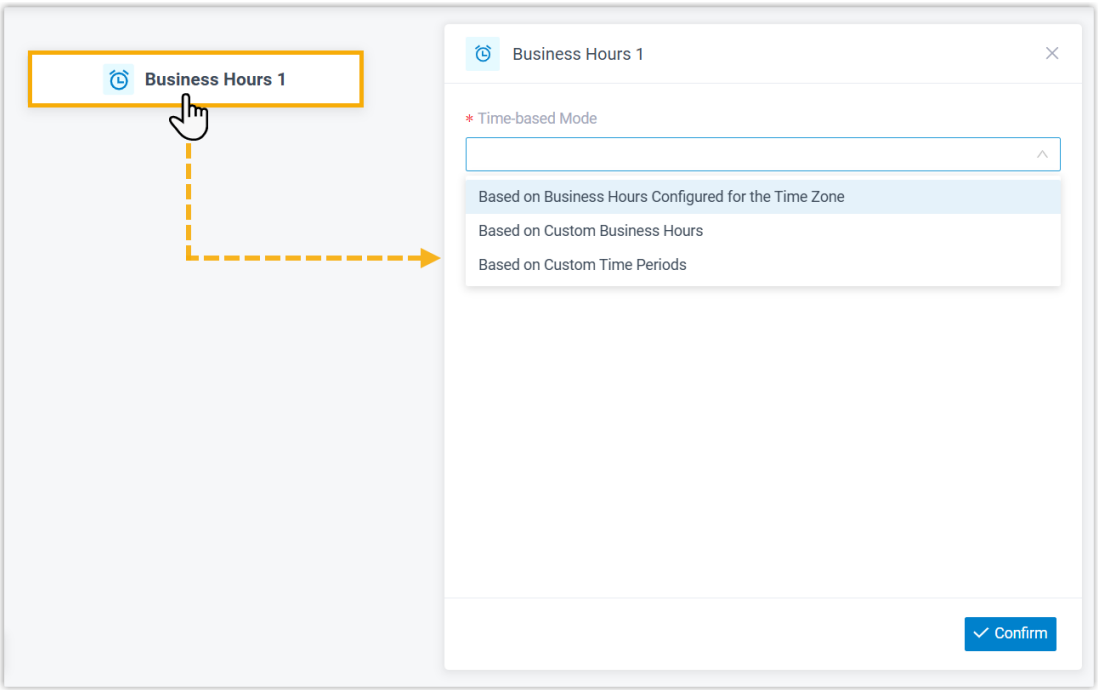


Supported time-based modes

Type	Description
Business Hours Configured for Time Zone	Use pre-configured business hours for 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 with a uniform or varied schedule through the week, or full time (24x7) schedule.
Custom Time Periods	Create custom time periods to accommodate granular scheduling needs.

Component configuration

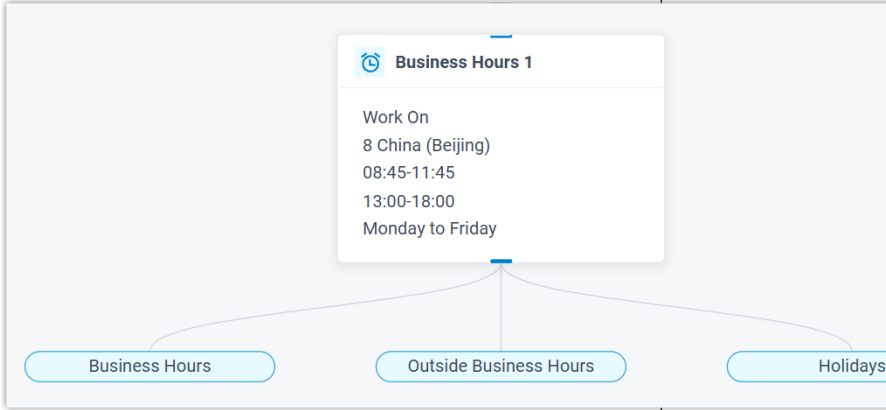
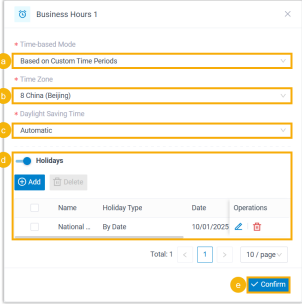
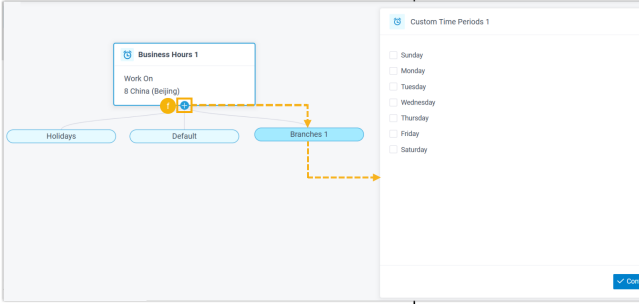
1. After adding **Business Hours** component to a call flow, click **Business Hours** component to proceed with the configuration.

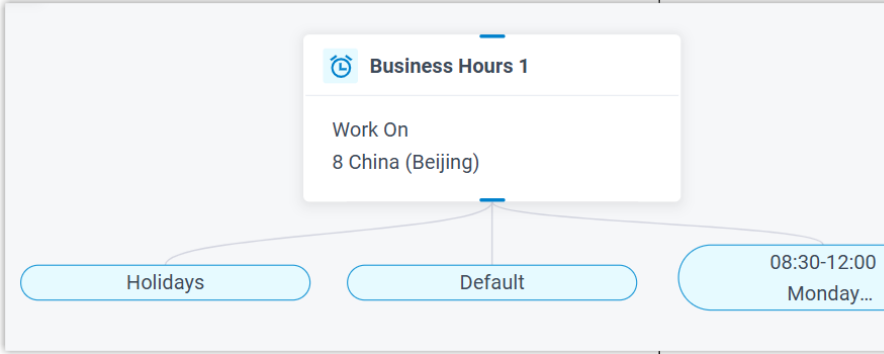


2. Select a time-based mode according to your business schedule, then configure its related settings.

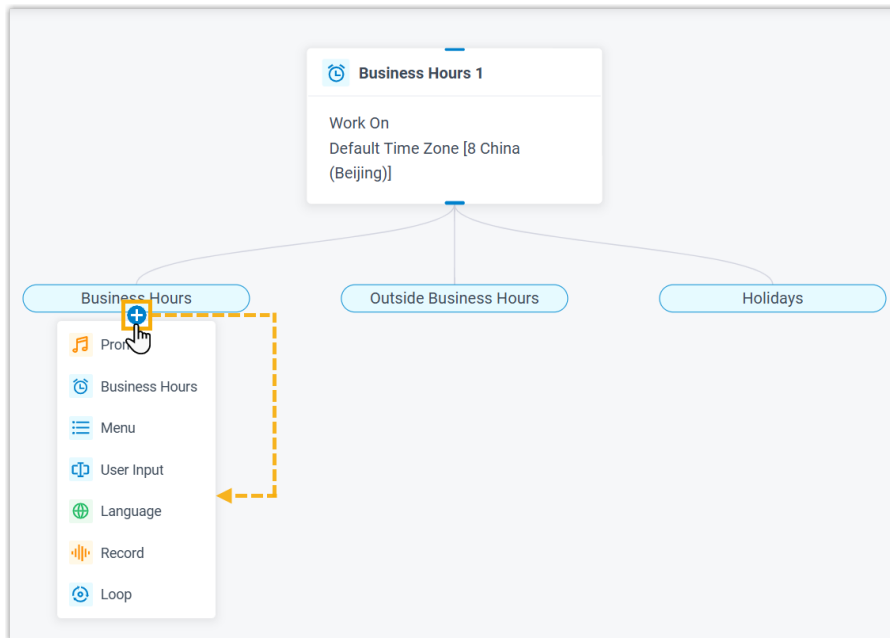
Scenario	Description
Use Business Hours Configured for Time Zone	a. In the Time-based Mode drop-down list, select Based on Business Hours Configured for the Time Zone. b. In the Time Zone drop-down list, select a time zone.

Scenario	Description
	c. Optional: To prevent incoming calls from being distributed to the holiday destination, select the checkbox of Ignore the Holiday Destination.  Note: If enabled, incoming calls during holidays will be routed to the destination defined for Business Hours or Outside Business Hours. d. At the bottom-right corner, click Confirm. The selected time zone and multiple time branches will appear on the component.  Business Hours 1 Work On Default Time Zone [8 China (Beijing)] Business Hours Outside Business Hours Holidays

Scenario	Description
	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 and multiple time branches will appear on the component. 
Use Custom Time Periods	  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.

Scenario	Description
	f. Click  on the component 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.  The diagram shows a configuration window titled "Business Hours 1" with a clock icon. Inside, it says "Work On" followed by "8 China (Beijing)". Below this, three branches are shown: "Holidays", "Default", and a time-based branch "08:30-12:00 Monday...".  Note: Default branch indicates the failover destination for incoming calls that don't match any configured time conditions.

3. Click  on the branch to add the next component to route the call or trigger other actions.



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 to route calls or trigger corresponding actions. Supported components are listed below.

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt .
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours .
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu .
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input .

Component	Description
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status

Component	Description
	• Set Agent Status • Email Sender • Database Access • HTTP Request

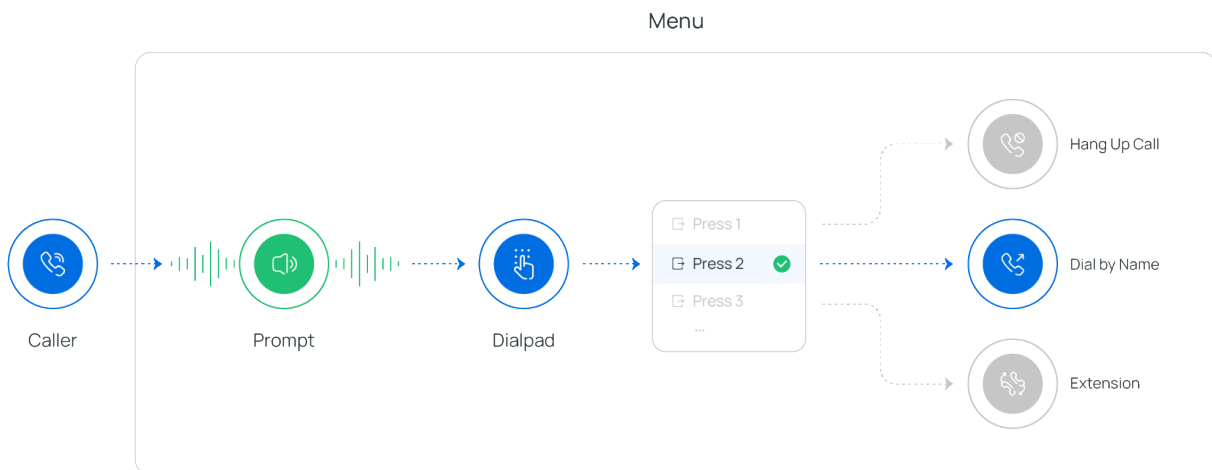
Menu

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 callers with a set of menu options, and route calls based on the DTMF digit that they press. The input value is stored in variables, which can be used as input for subsequent components or for condition evaluation.

You can configure when the system stops collecting DTMF digits - either when reaching digit or response timeout or when callers press an end key. Additionally, audio prompt(s) or a text-to-speech message can be set to prompt callers for input.



Supported prompt types

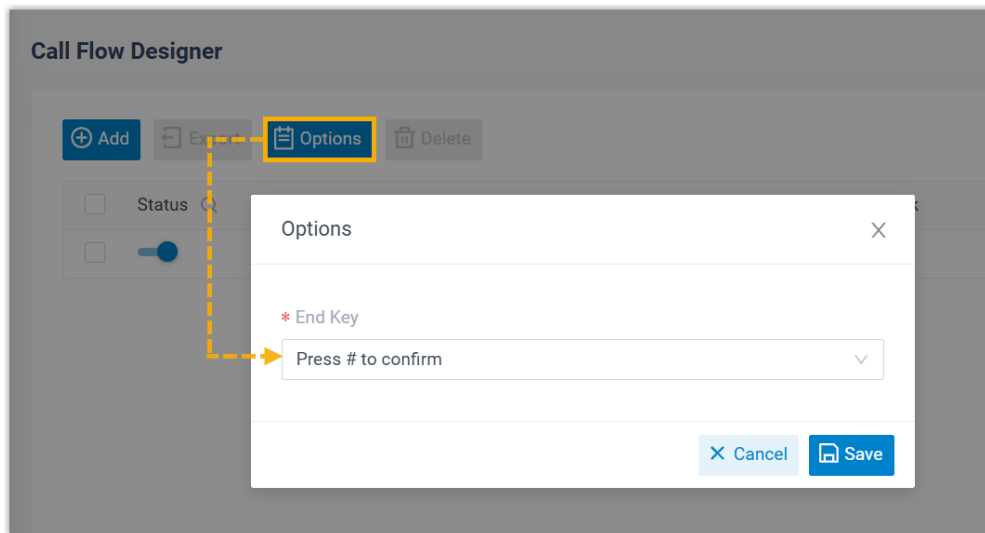
Prompt Type	Description
Custom Prompt	Choose from existing custom prompts, or create a new one by recording with an extension or uploading an audio file.
Music on Hold (MoH)	Choose from existing MoH.

Prompt Type	Description						
	Note: Before you begin, make sure the desired MoH is configured in PBX Settings > Voice Prompt > Music on Hold. Prompt PreferencesSystem PromptMusic on HoldCustom Prompt Create New Playlist <table><tr><th>Playlist Name</th><th>Playlist Type</th><th>Play Order</th></tr><tr><td>default</td><td>Local Audio</td><td>Alphabetical</td></tr></table>	Playlist Name	Playlist Type	Play Order	default	Local Audio	Alphabetical
Playlist Name	Playlist Type	Play Order					
default	Local Audio	Alphabetical					
Text to Speech (TTS)	Enter the desired text, which will be converted to speech in the selected language and voice style. Note: Before you begin, make sure the Text-to-Speech feature is enabled in Integrations > AI. Integrations / AI Text-to-Speech Voicemail Transcription						

Supported end keys

key or * key.

You can set the end key in **Call Flow Designer > Options**.



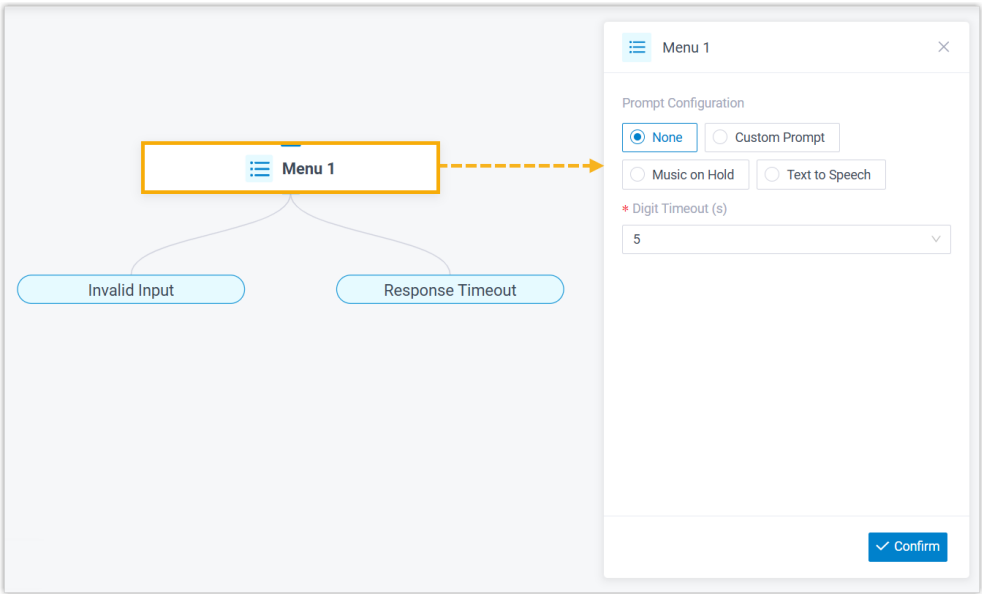
Component configuration

After adding **Menu** component to a call flow, you can configure keys for menu options, prompt for input instruction, timeout for input, and retry strategies for invalid or no input.

- [Set prompt and digit timeout](#)
- [Set retry strategy for invalid input](#)
- [Set response timeout and retry strategy](#)
- [Set keys for menu options](#)

Set prompt and digit timeout

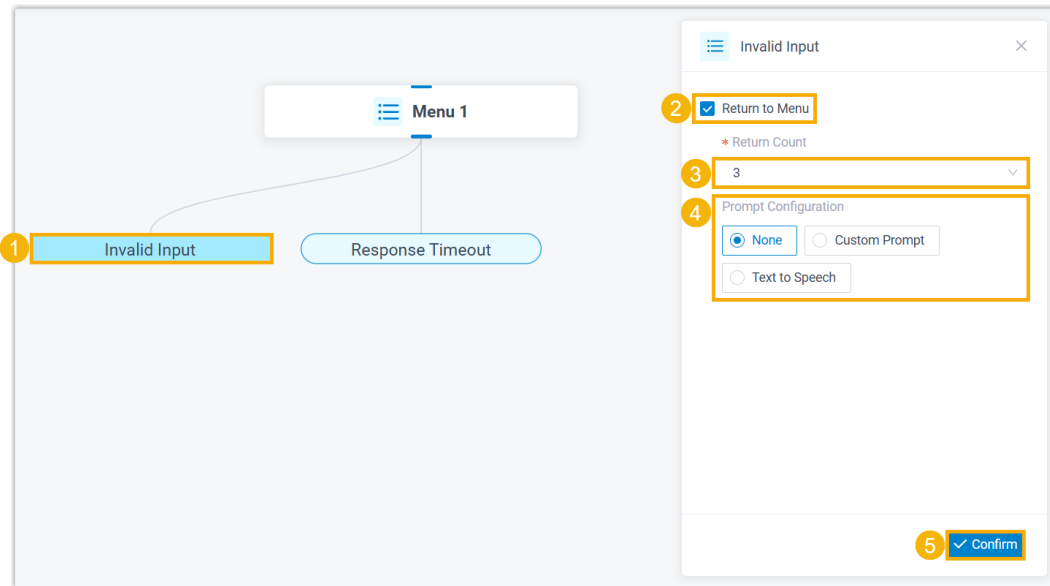
1. Click **Menu** component to proceed with the configuration.



2. Configure audio prompt(s) or a text-to-speech message to instruct callers.

Item	Description
Custom Prompt	Prompt Configuration ☐ None a ☒ Custom Prompt ☐ Music on Hold ☐ Text to Speech b * Prompt welcome.wav Record New Upload c ☐ Enable Loop Playback a. Click Custom Prompt. b. Specify custom prompt(s).  Note:

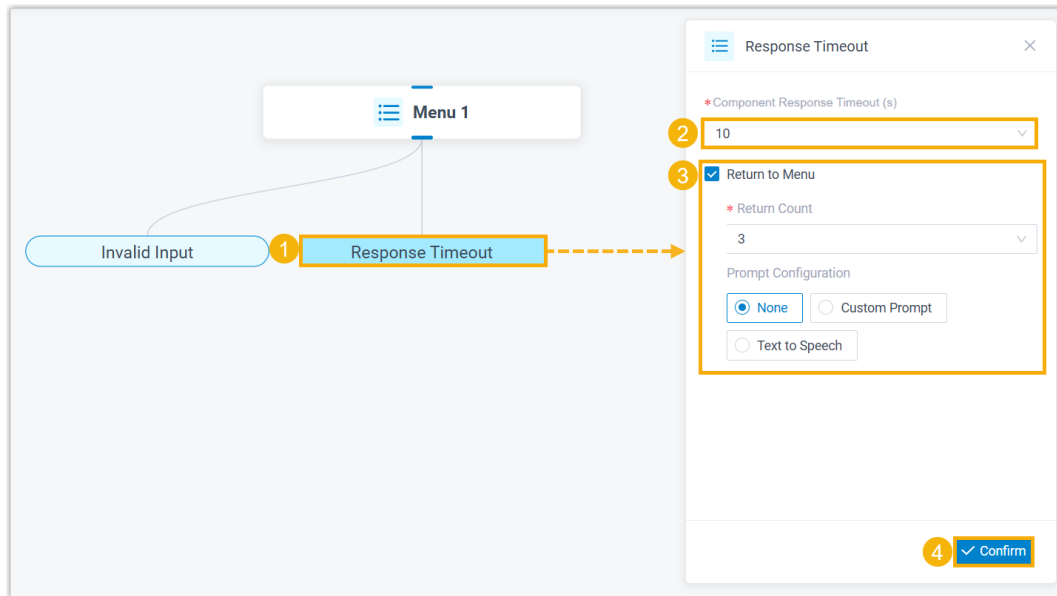
Item	Description
	 Up to 5 custom prompts are supported. • To choose from an existing prompt, select it from the drop-down list of Prompt. • To use a new prompt, click Record New to record prompt with an extension, or click Upload to upload an audio file.  Note: The new prompt will be saved in PBX Settings > Voice Prompt > Custom Prompt. c. Optional: To play the custom prompt(s) in a loop, select the checkbox of Enable Loop Playback.  Note: The looped prompt will be overridden if the next connected component is configured with a prompt.
Music on Hold (MoH)	Prompt Configuration ☐ None ☐ Custom Prompt ☒ Music on Hold ☐ Text to Speech * Music on Hold default a. Select Music on Hold. b. Select an existing MoH from the drop-down list of Music on Hold.  Note: The MoH will be overridden if the next connected component is configured with a prompt.



1. Click **Invalid Input** branch.
2. Select the checkbox of **Return to Menu**.
3. In the **Return Count** drop-down list, select how many times callers can return to the component upon invalid input.
4. **Optional:** Configure audio prompt(s) or a text-to-speech message to instruct callers to retry.
5. At the bottom-right corner, click **Confirm**.

Set response timeout and retry strategy

To change the default response timeout or allow callers to return to the component for a retry upon response timeout, complete the following settings.



1. Click **Response Timeout** branch.
2. In the **Component Response Timeout(s)** field, set how long to wait for the caller to operate.

You can select a value from the drop-down list, or enter a value between 1 and 9999.
3. Set the retry strategy upon response timeout.
 - a. Select the checkbox of **Return to Menu**.
 - b. In the **Return Count** drop-down list, select how many times callers can return to the component upon response timeout.
 - c. **Optional:** Configure audio prompt(s) or a text-to-speech message to instruct callers to retry.
4. At the bottom-right corner, click **Confirm**.

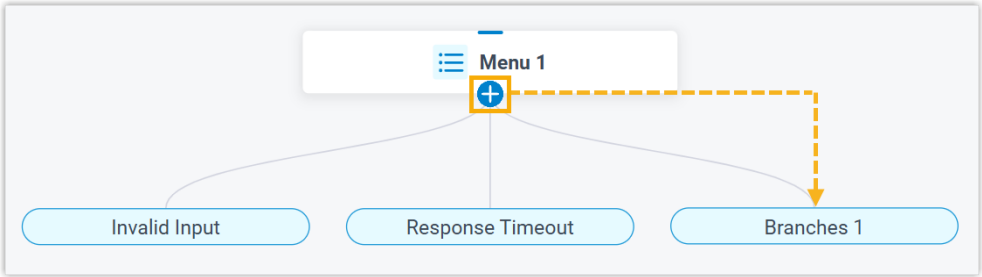
Set keys for menu options

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

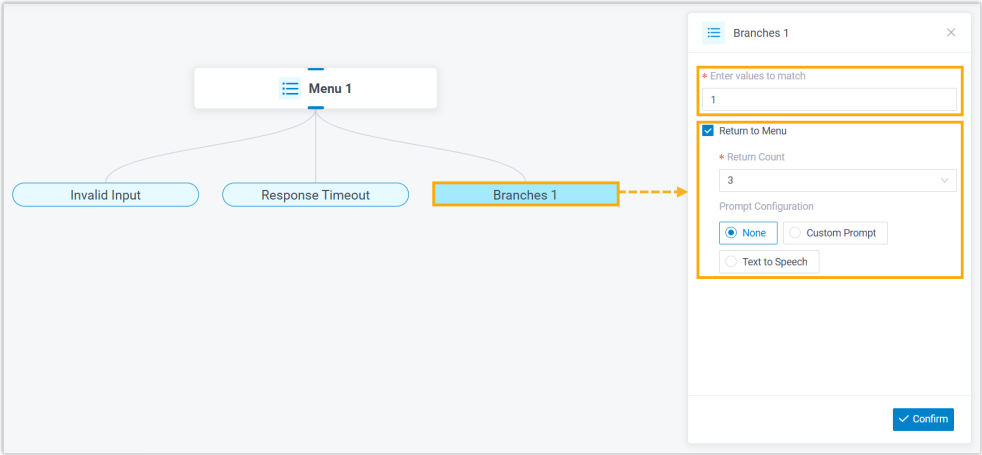


Note:

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



2. Click the branch to complete its settings.



Item	Description
Enter values to match	Enter a value.  Note: - Only numbers and characters * and # are allowed. - If the value contains * or #, ensure it does NOT conflict with the end key (configured in Call Flow Designer > Options). Otherwise, the input will be considered invalid and routed to the destination for invalid input.
Return to Menu	To allow callers to return to the component, proceed as follows.  Note: This is helpful for replaying the prompt and allowing a retry when the caller fails to make a selection.

Item	Description
	- Select the checkbox of Return to Menu. - In the Return Count drop-down list, select how many times callers can return to the component. Optional: Configure audio prompt(s) or a text-to-speech message to instruct callers to retry.

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

Component variables

When a **Menu** component is added to a call flow, the system stores caller's input in variables. These variables can be referenced in expression-supported components, such as **Condition** or **Developer**, to retrieve the input value and flexibly route the call.



Note:

Since **Menu** component can be added multiple times in a call flow, an index is appended to 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
<code>\$Menu{<i>index</i>}.result</code>	String	The result of the Menu component. <code>MenuResult.Timeout</code>: The caller didn't press any DTMF digit before the timeout, and was routed to the timeout destination. <code>MenuResult.ValidOption</code>: The caller pressed a valid key, and was routed to the corresponding destination. <code>MenuResult.InvalidOption</code>: The caller pressed an invalid key, and was routed to the invalid input destination.	<code>\$Menu1.result(STRING)=MenuResult.InvalidOption</code>
<code>\$Menu{<i>index</i>}.ttsResult</code>	String	The Text-to-Speech (TTS) result of the Menu component.	<code>\$Menu1.ttsResult(STRING)=MenuTTSResult.Success</code>

Variable	Type	Description	Example Value
		- MenuTTSResult.Success: The text is successfully converted into speech, and the system plays it to caller. - MenuTTSResult.Timeout: The text is NOT converted into speech within the timeout period, and the system routes caller directly to the next component without playing a prompt when the timeout is reached.	
\$Menu{index}.userInput	String	The DTMF digit that the caller pressed, excluding the end key (# or *).	"2"

Component connections

Menu component comes with two built-in branches - **Invalid Input** and **Response Timeout**, and supports up to 100 additional branches for DTMF keys. Each branch can be connected to **one** component, which can be any of the components listed below.

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow.

Component	Description
	For more information, see Language .
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access

Component	Description
	HTTP Request

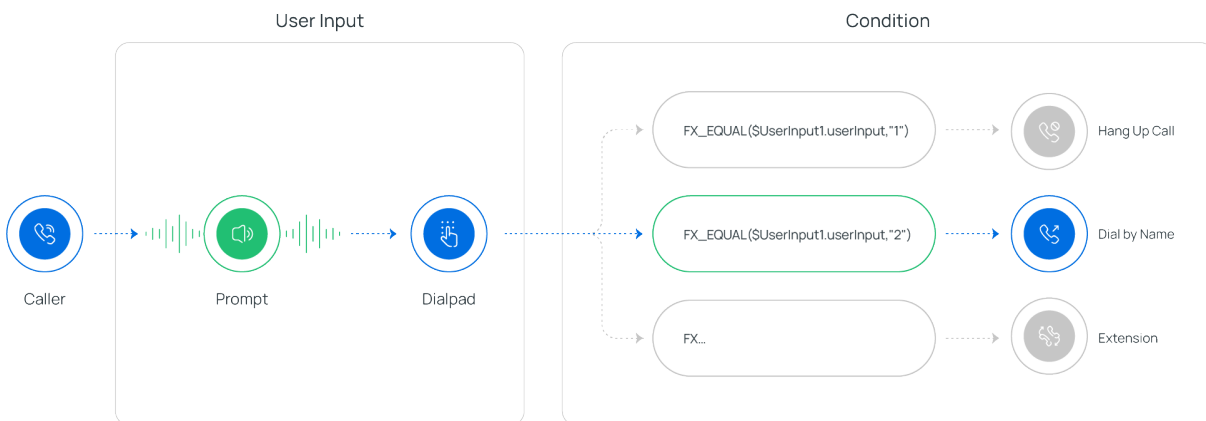
User Input

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

Component introduction

User Input component allows you to collect DTMF digits from callers, typically used with **Condition** component. The input value is stored in variables, which can be used as input for subsequent components or for condition evaluation.

You can configure when the system stops collecting DTMF digits - either when reaching digit or response timeout or when callers press an end key. Additionally, audio prompt(s) or a text-to-speech message can be set to prompt callers for input.



Supported prompt types

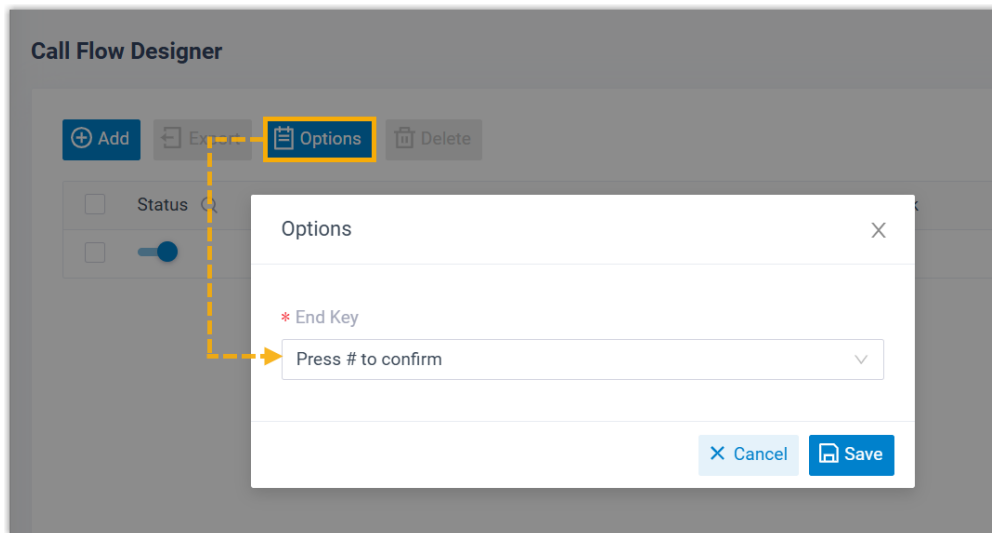
Prompt Type	Description
Custom Prompt	Choose from existing custom prompts, or create a new one by recording with an extension or uploading an audio file.
Music on Hold (MoH)	Choose from existing MoH. Note: Before you begin, make sure the desired MoH is configured in PBX Settings > Voice Prompt > Music on Hold.

Prompt Type	Description								
	Prompt Preferences System Prompt Music on Hold Custom Prompt Create New Playlist <table><tr><th>Playlist Name</th><th>Playlist Type</th><th>Play Order</th><th>Streaming Music URL</th></tr><tr><td>default</td><td>Local Audio</td><td>Alphabetical</td><td>-</td></tr></table>	Playlist Name	Playlist Type	Play Order	Streaming Music URL	default	Local Audio	Alphabetical	-
Playlist Name	Playlist Type	Play Order	Streaming Music URL						
default	Local Audio	Alphabetical	-						
Text to Speech (TTS)	Note: Before you begin, make sure the Text-to-Speech feature is enabled in Integrations > AI. Integrations / AI Text-to-Speech Voicemail Transcription								

Supported end keys

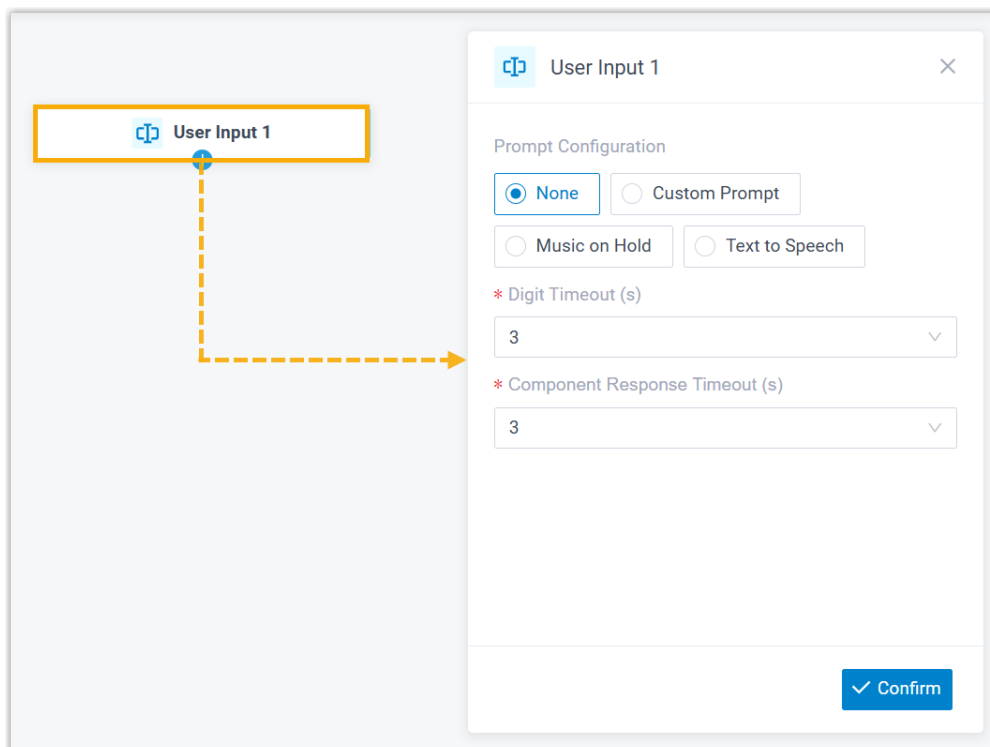
key or * key.

You can set the end key in **Call Flow Designer > Options**.



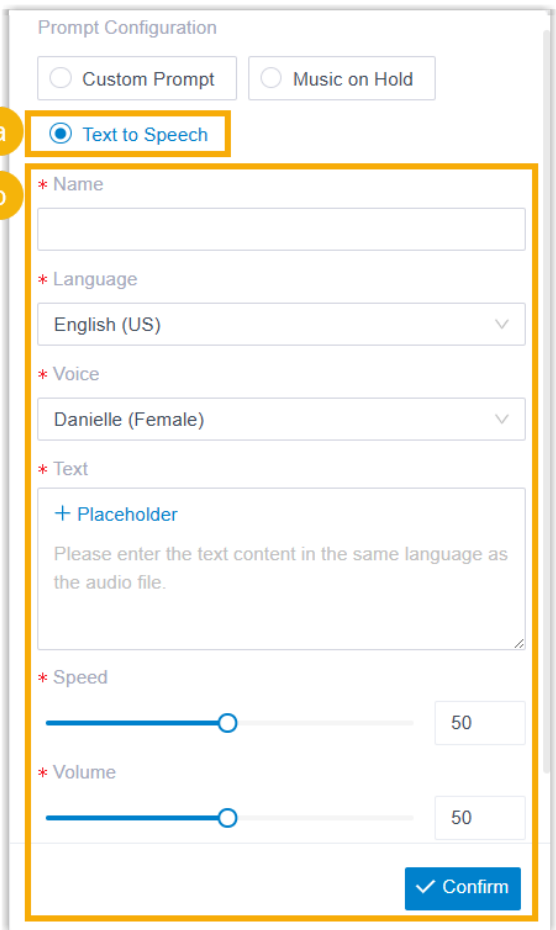
Component configuration

1. After adding **User Input** component to a call flow, click **User Input** component to proceed with the configuration.



2. **Optional:** Add audio prompt(s) or a text-to-speech message to instruct callers.

Item	Description
Custom Prompt	a. Click Custom Prompt. b. Specify custom prompt(s). Note: Up to 5 custom prompts are supported. • To choose from an existing prompt, select it from the drop-down list of Prompt. • To use a new prompt, click Record New to record prompt with an extension, or click Upload to upload an audio file. Note: The new prompt will be saved in PBX Settings > Voice Prompt > Custom Prompt. c. Optional: To play the custom prompt(s) in a loop, select the checkbox of Enable Loop Playback. Note: The looped prompt will be overridden if the next connected component is configured with a prompt.

Item	Description
Text to Speech (TTS)	 a. Select Text to Speech. b. Complete the rest of the configurations to convert text into speech.  Note: Both static text and dynamic text are supported to convert into speech. • Static text: Enter text directly in the Text field. • Dynamic text: Enter text and configure placeholders with corresponding variables to dynamically compose the text. Optionally, you can configure prompt generation timeout. If speech is not generated within the timeout period, caller will be routed to the next component without hearing a prompt. Example:

Item	Description
	The new speech will be saved to the system and synchronized to PBX Settings > Voice Prompt > Custom Prompt.

3. Configure the timeout for user input.

*** Digit Timeout (s)**

3 ▾

*** Component Response Timeout (s)**

3 ▾

- **Digit Timeout(s):** Set how long to wait for the caller to enter the next digit.

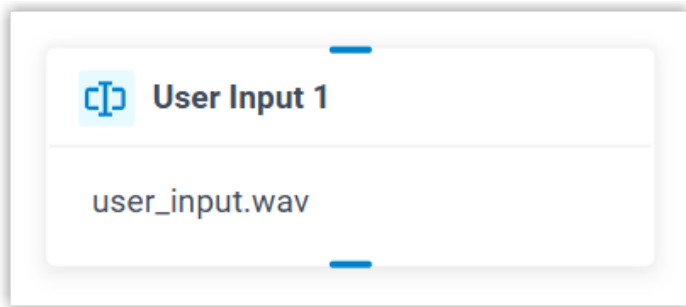
You can select a value from the drop-down list, or enter a value between 1 and 9999.

- **Component Response Timeout(s):** Set how long to wait for the caller to operate.

You can select a value from the drop-down list, or enter a value between 1 and 9999.

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

The selected prompt will appear on the component.



Component variables

When a **User Input** component is added to a call flow, the system stores caller's input in variables. These variables can be referenced in expression-supported components, such as **Condition** or **Developer**, to retrieve the input value and flexibly route the call.



Note:

Since **User Input** component can be added multiple times in a call flow, an index is appended to each component (e.g. User Input 1, User Input 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>\$UserInput{<i>index</i>}.result</code>	String	The result of User Input component. <code>UserInputResult.Timeout</code>: The caller didn't press any DTMF digit before the timeout, and was routed to the timeout destination. <code>UserInputResult.ValidOption</code>: The caller pressed a valid key, and was routed to the corresponding destination.	<code>\$UserInput1.result(STRING)=UserInputResult.Timeout</code>
<code>\$UserInput{<i>index</i>}.ttsResult</code>	String	The Text-to-Speech (TTS) result of the User Input component. <code>UserInputTTSResult.Success</code>: The text is successfully	<code>\$UserInput1.ttsResult(STRING)=UserInputTTSResult.Success</code>

Variable	Type	Description	Example Value
		converted into speech, and the system plays it to caller. <code>UserInputTTSResult.Timeout</code>: The text is NOT converted into speech within the timeout period, and the system routes caller directly to the next component without playing a prompt when the timeout is reached.	
<code>\$UserInput{index}.userInput</code>	String	The DTMF digit that the caller pressed, excluding the end key (# or *).	"2"

Component connections

User Input component can be connected to **one** component, which can be any of the components listed below.

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded.

Component	Description
	For more information, see Record .
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

Language

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

Component restriction

Component setting has lower priority than extension-specific setting.

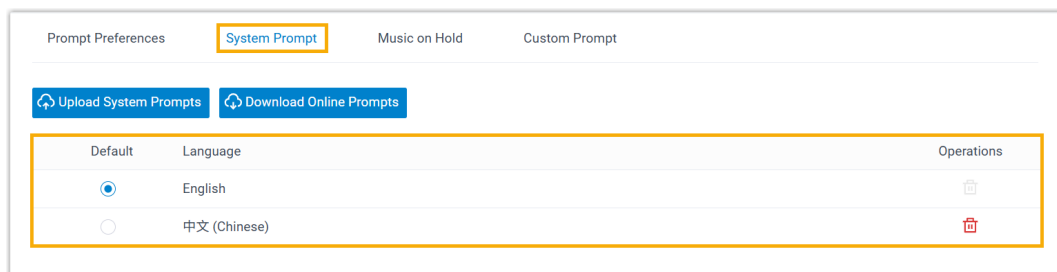
If the caller is an internal extension whose system prompt language (configured in **Extension and Trunk > Extension > Language > System Prompt Language**) differs from the language set in **Language** component, the extension's system prompt language will take precedence.

Component introduction

After setting up a call flow, the system plays system prompts in English by default. **Language** component allows you to change the system prompt language for subsequent components in the call flow, overriding the default language setting.

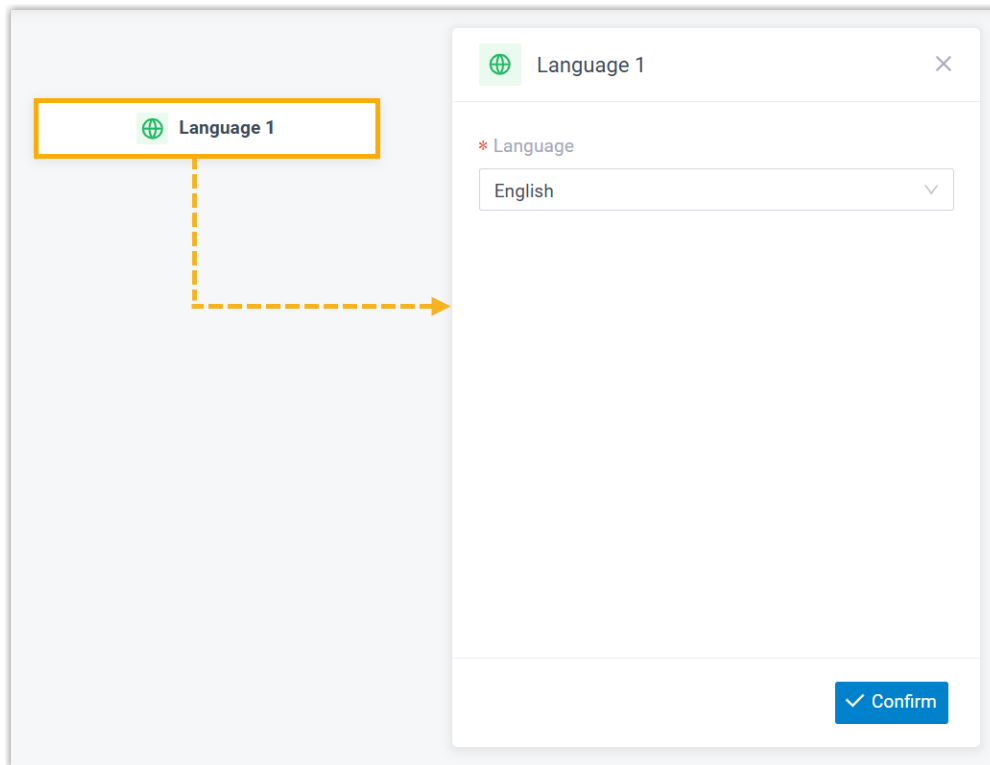
Supported languages

By default, only **English** is available. To use another language, download the desired language in **PBX Settings > Voice Prompt > System Prompt** before configuring the component.

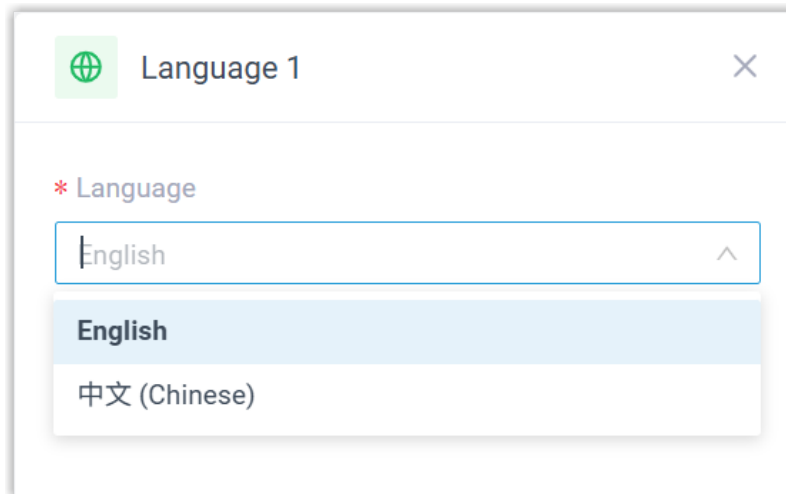


Component configuration

1. After adding **Language** component to a call flow, click **Language** component to proceed with the configuration.

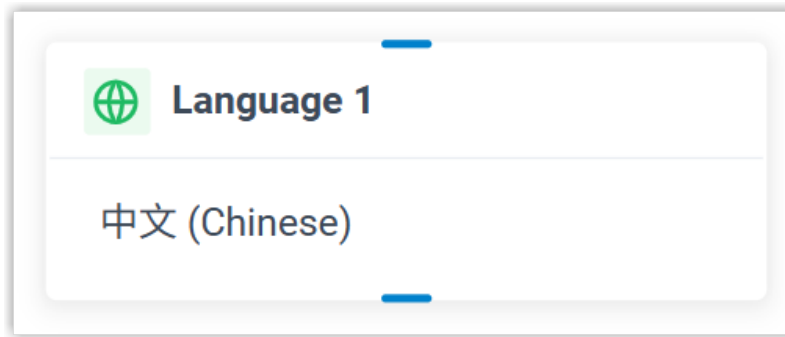


2. In the **Language** drop-down list, select a language.



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

The selected language will appear on the component.



Component connections

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

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt .
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours .
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu .
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input .
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language .
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record .
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination.

Component	Description
	For more information, see Dial by Number .
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

Record

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

Component requirement

Record component requires a storage location to save recording files.

You can store recording files on local storage, an external device, or a network drive. For centralized management or backup, you can also archive the files to external servers such as S3-compatible object storage, Google Cloud Storage, Microsoft SharePoint, FTP server, or SFTP server.

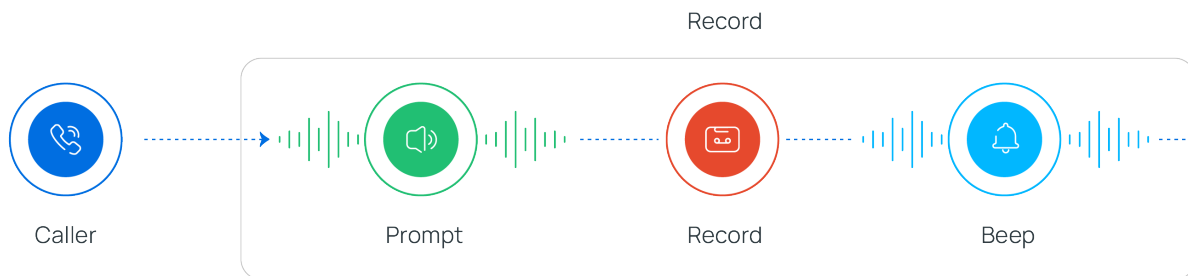
For more information, see the following topics:

- (Required) [Set Storage Location for Recording Files.](#)

- (Optional) [Archive Recording Files to External Server.](#)

Component introduction

Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording.



Note:

Once call recording is enabled, you can NOT use another **Record** component to pause or stop the recording. However, authorized extension users can change the



recording status according to their permissions (Path: **Extension and Trunk > Extension > Features > Call Recording**).

Extension and Trunk / Extension / **Edit (Phillip Huff)**

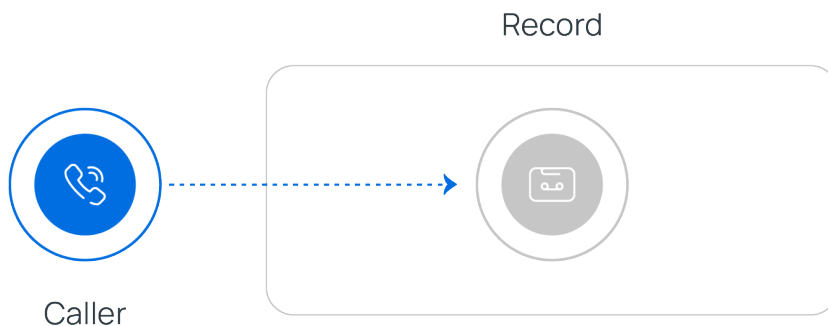
User Presence Voicemail **Features** Advanced Security Linkus Clients Phone Function Keys

Call Recording

Recording Operation (the operation takes effect by pressing the recording button or dialing the feature code *1 during a call)

☐ No Permission
☒ **Pause/Resume**
☐ Start/Pause/Resume

Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded.



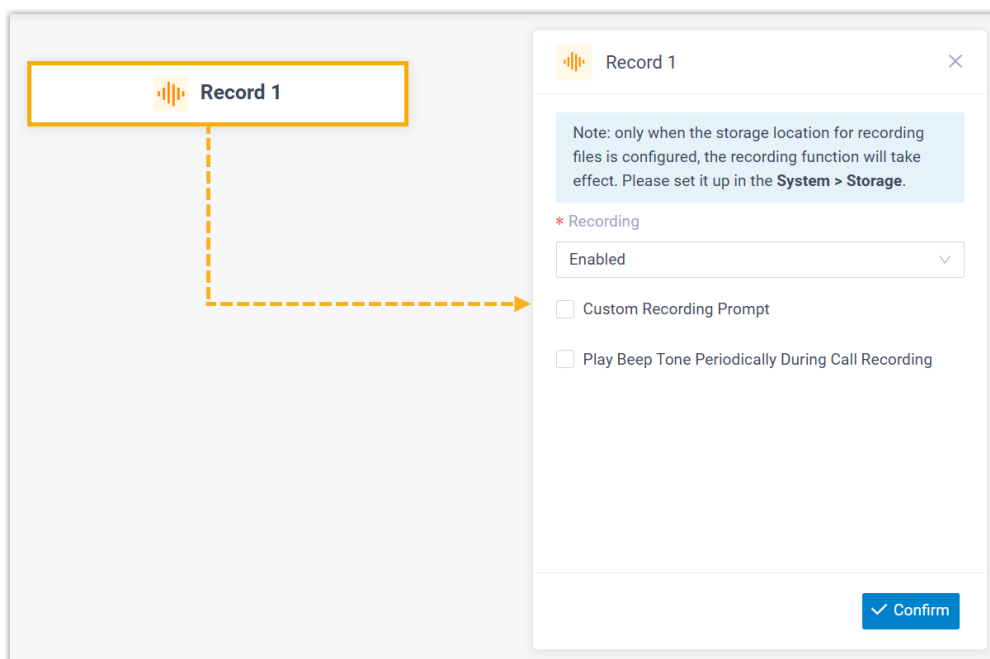
Supported prompt types

Prompt Type	Description
Custom Prompt	Choose from existing custom prompts, or create a new one by recording with an extension or uploading an audio file.
Text to Speech (TTS)	Enter the desired text, which will be converted to speech in the selected language and voice style.  Note: Before you begin, make sure the Text-to-Speech feature is enabled in Integrations > AI.

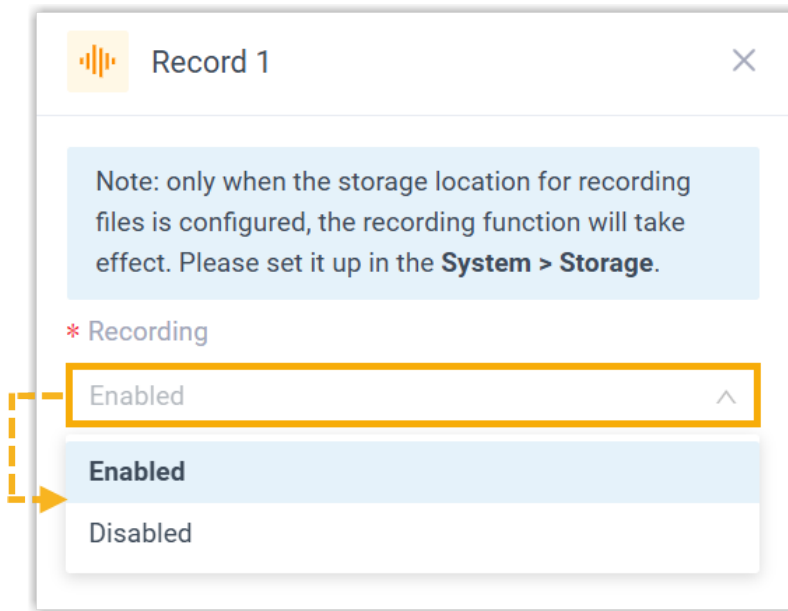
Prompt Type	Description
	 Integrations / AI ☒ Text-to-Speech ☐ Voicemail Transcription

Component configuration

1. After adding **Record** component to a call flow, click **Record** component to proceed with the configuration.



2. In the **Recording** drop-down list, select an option to enable or disable call recording.



3. If call recording is enabled, you can configure prompts to inform call participants of the recording.

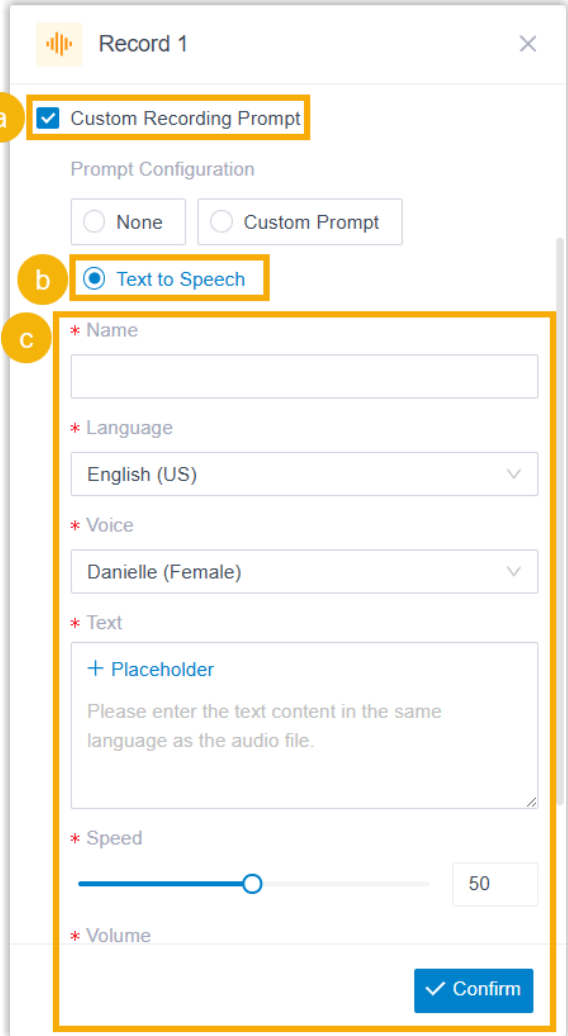


Note:

- For **Conference Call** and **Multi-party Call**, call participants will not hear the prompt.
- If no custom prompt is configured, the system will use the global prompt for recording (Path: **Call Features > Recording**), if available.

Play prompt at the start of call recording

You can play audio prompt(s) or a text-to-speech message to inform call participants when call recording starts.

Item	Description
Text to Speech (TTS)	 a. Select the checkbox of Custom Recording Prompt. b. Select Text to Speech. c. Complete the rest of the configurations to convert text into speech.  Note: Both static text and dynamic text are supported to convert into speech. • Static text: Enter text directly in the Text field. • Dynamic text: Enter text and configure placeholders with corresponding variables to dynamically compose the text.

Item	Description
	Optionally, you can configure prompt generation timeout. If speech is not generated within the timeout period, caller will be routed to the next component without hearing a prompt. Example: The new speech will be saved to the system and synchronized to PBX Settings > Voice Prompt > Custom Prompt.

Play prompt during call recording

You can play beep tones during call recording to inform call participants that the call is being recorded.

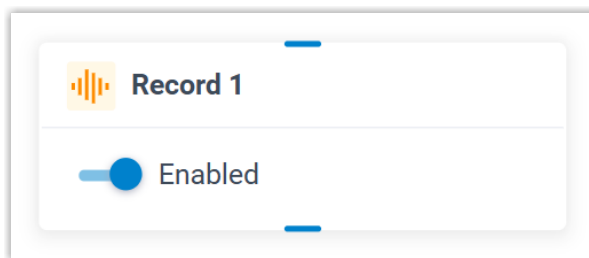
a ☒ Play Beep Tone Periodically During Call Recording

* Interval Duration (s)

b 15

- a. Select the checkbox of **Play Beep Tone Periodically During Call Recording**.
 - b. In the **Interval Duration(s)** field, enter the interval in seconds.
4. At the bottom-right corner, click **Confirm**.

The recording status will appear on the component.



Component variable

When a **Record** component with TTS configured is added to a call flow, the system stores TTS result in variable. The variable can be referenced in expression-supported components, such as **Condition** or **Developer**, to retrieve the output value and flexibly route the call.



Note:

Since **Record** component can be added multiple times in a call flow, an index is appended to each component (e.g. Record 1, Record 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>\$Record{<i>index</i>}.ttsResult</code>	String	The Text-to-Speech (TTS) result of the Record component. <code>RecordTTSResult.Success</code>: The text is successfully	<code>\$Record1.ttsResult(STRING)=RecordTTSResult.Timeout</code>

Variable	Type	Description	Example Value
		converted into speech, and the system plays it to caller. • <code>RecordTTSResult.Timeout</code>: The text is NOT converted into speech within the timeout period, and the system routes caller directly to the next component without playing a prompt when the timeout is reached.	

Component connections

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

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.

Component	Description
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

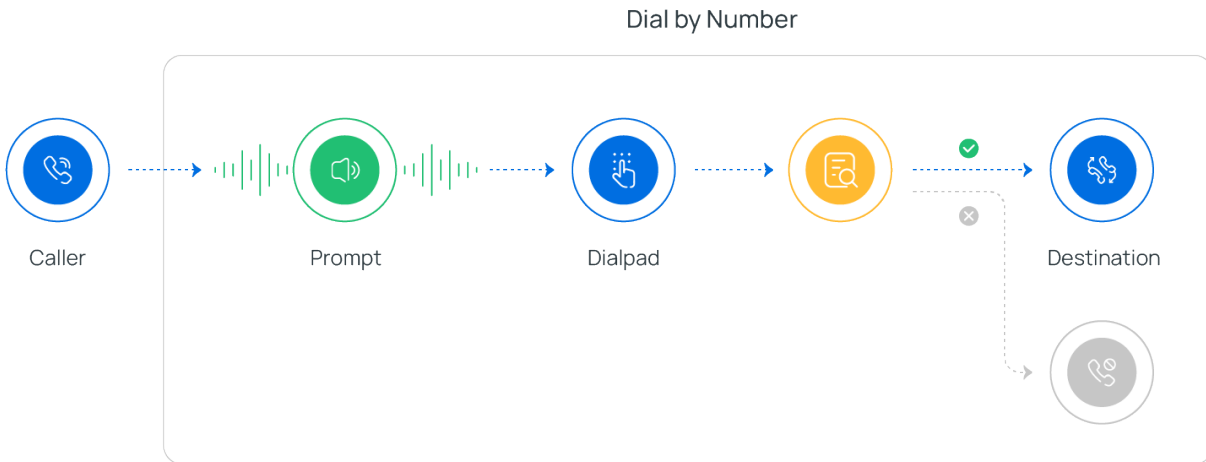
Dial by Number

This topic provides an overview of the **Dial by Number** component, and describes its configuration as well as supported connections.

Component introduction

Dial by Number component allows callers to directly dial a number to reach the destination.

You can configure when the system stops accepting caller input - either when reaching digit or response timeout or when callers press an end key. Additionally, audio prompt(s) or a text-to-speech message can be set to prompt callers for input.

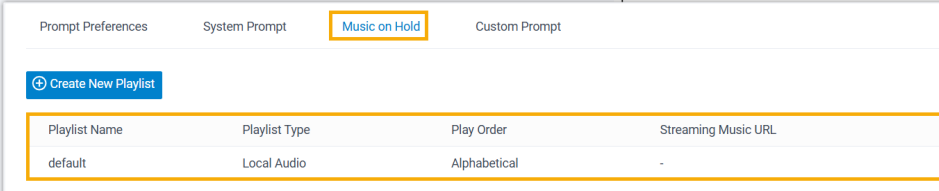
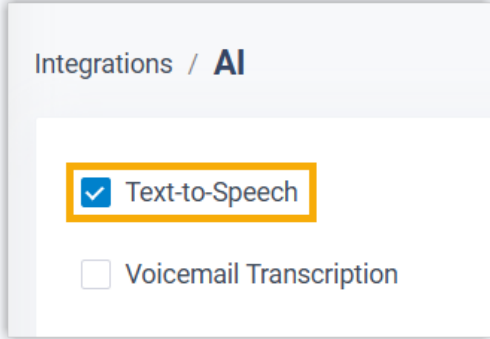


Supported destinations

- Extension
- Ring Group
- Queue
- Conference
- Call Flow
- Paging
- IVR
- External Number
- Check Voicemail (Extension Voicemail or Group Voicemail)

Supported prompt types

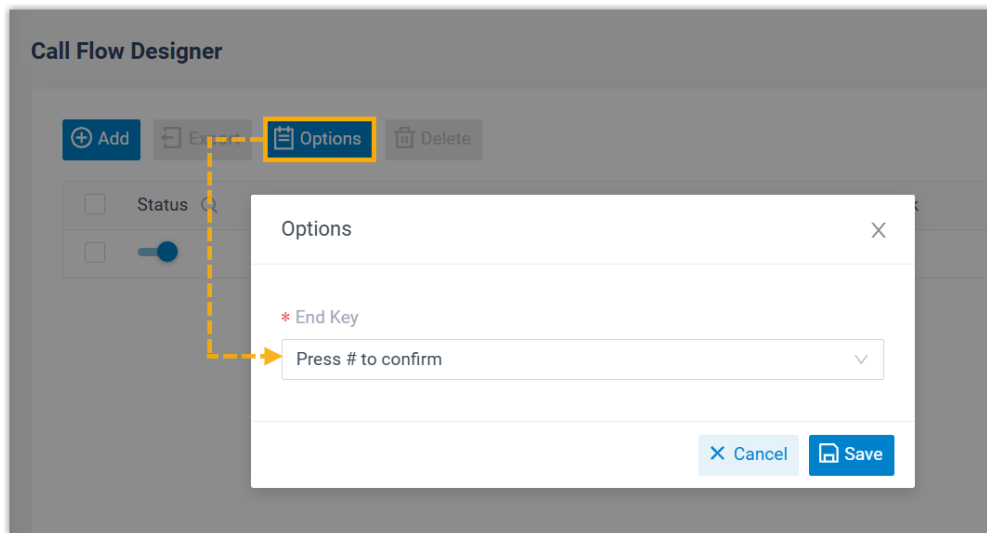
Prompt Type	Description
Custom Prompt	Choose from existing custom prompts, or create a new one by recording with an extension or uploading an audio file.
Music on Hold (MoH)	Choose from existing MoH.
 Note:	

Prompt Type	Description
	 Before you begin, make sure the desired MoH is configured in PBX Settings > Voice Prompt > Music on Hold. 
Text to Speech (TTS)	Enter the desired text, which will be converted to speech in the selected language and voice style.  Note: Before you begin, make sure the Text-to-Speech feature is enabled in Integrations > AI. 

Supported end keys

key or * key.

You can set the end key in **Call Flow Designer > Options**.



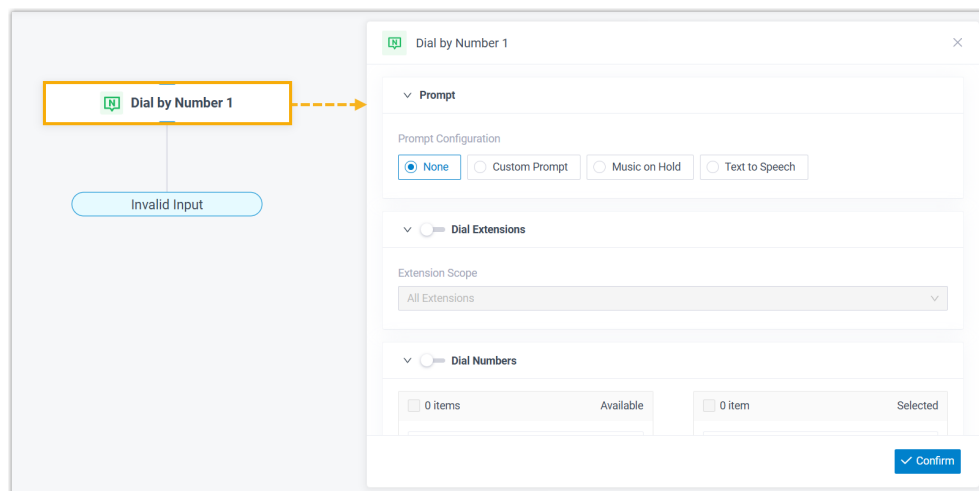
Component configuration

After adding **Dial by Number** component to a call flow, you can configure the numbers that callers are allowed to dial, as well as the retry strategy for invalid user input.

- [Specify the numbers that callers can dial](#)
- [Specify retry strategy for invalid input](#)

Specify the numbers that callers can dial

1. Click **Dial by Number** component to proceed with the configuration.



2. Configure audio prompt(s) or a text-to-speech message to instruct callers.

Item	Description
Custom Prompt	Prompt Configuration ☐ Non a ☒ Custom Prompt ☐ Music on Hold ☐ Text to Speech b * Prompt welcome.wav Record New Upload c ☐ Enable Loop Playback a. Click Custom Prompt. b. Specify custom prompt(s).  Note: Up to 5 custom prompts are supported. • To choose from an existing prompt, select it from the drop-down list of Prompt. • To use a new prompt, click Record New to record prompt with an extension, or click Upload to upload an audio file.  Note: The new prompt will be saved in PBX Settings > Voice Prompt > Custom Prompt. c. Optional: To play the custom prompt(s) in a loop, select the checkbox of Enable Loop Playback.  Note:

Item	Description
	 The looped prompt will be overridden if the next connected component is configured with a prompt.
Music on Hold (MoH)	Prompt Configuration ☐ None ☐ Custom Prompt ☒ Music on Hold ☐ Text to Speech * Music on Hold b default ▼ a. Select Music on Hold. b. Select an existing MoH from the drop-down list of Music on Hold.  Note: The MoH will be overridden if the next connected component is configured with a prompt.

Item	Description
Text to Speech (TTS)	Prompt Configuration ☐ Custom Prompt ☐ Music on Hold a ☒ Text to Speech b * Name * Language English (US) v * Voice Danielle (Female) v * Text + Placeholder Please enter the text content in the same language as the audio file. * Speed <input type="range" value="50"/> 50 * Volume <input type="range" value="50"/> 50 Confirm a. Select Text to Speech. b. Complete the rest of the configurations to convert text into speech.  Note: Both static text and dynamic text are supported to convert into speech. • Static text: Enter text directly in the Text field. • Dynamic text: Enter text and configure placeholders with corresponding variables to dynamically compose the text. Optionally, you can configure prompt generation timeout. If speech is not generated within the timeout period, caller will be routed to the

Item	Description
	 next component without hearing a prompt. Example: The new speech will be saved to the system and synchronized to PBX Settings > Voice Prompt > Custom Prompt.

3. Specify the numbers that callers can dial upon reaching the component.

Item	Description
Dial Extensions	a Dial Extensions * Extension Scope b All Extensions All Extensions Allowed Extensions Restricted Extensions a. Turn on the switch of Dial Extensions.

Item	Description
	b. In the Extension Scope drop-down list, select an option. • All Extensions: Allow callers to dial all extensions. • Allowed Extensions: Allow callers to dial only specific extensions. If you choose this option, select the allowed extensions from Available box to Selected box. • Restricted Extensions: Allow callers to dial all extensions EXCEPT the selected ones. If you choose this option, select the restricted extensions from Available box to Selected box.
Dial Numbers	a Dial Numbers 4 items Available Search here < Queue Conference > Number Name 6401 6401 3 items Selected Search here Number Name Type 6400 6400 Queue 6500 6500 Conference 6602 6602 Paging b > < a. Turn on the switch of Dial Numbers. b. Select the desired numbers from Available box to Selected box.

Item	Description
	 To change it, go to Call Features > Feature Code > Voicemail > Check Voicemail/Subscribe Voicemail Status. Voicemail * Check Voicemail/Subscribe Voicemail Status ☒ *2 • Extension Voicemail Access PIN: To change the PIN, go to Extension and Trunk > Extension > Voicemail > Voicemail Access PIN. User Presence Voicemail Features Advanced Security Linkus Clients Phone Function Keys • Enable Voicemail * Voicemail PIN Authentication Enabled * Voicemail Access PIN ***** • Group Voicemail Access PIN: To change the PIN, go to Call Features > Voicemail > Group Voicemail > Voicemail Access PIN. Basic * Type Custom * Number 6100 * Mode Shared by Members * Voicemail PIN Authentication Enabled * Name 6100 Group voicemail to Email * Voicemail Access PIN *****

4. Configure the timeout for user input.

▼

Timeout

* Digit Timeout (s)

5

▼

* Component Response Timeout (s)

10

▼

- **Digit Timeout(s)**: Set how long to wait for the caller to enter the next digit.

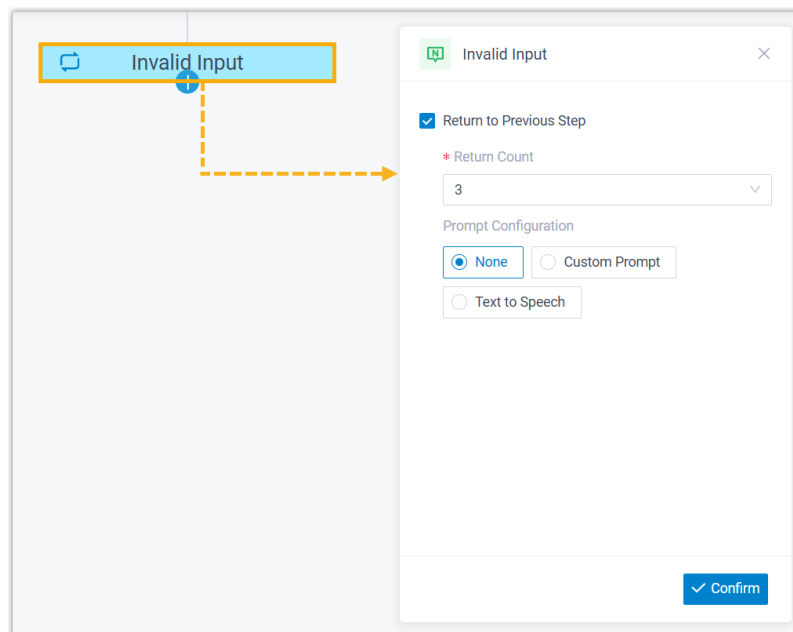
You can select a value from the drop-down list, or enter a value between 1 and 9999.

- **Component Response Timeout(s)**: Set how long to wait for the caller to operate.

You can select a value from the drop-down list, or enter a value between 1 and 9999.

Specify retry strategy for invalid input

1. Click **Invalid Input** branch.



2. Select the checkbox of **Return to Previous Step**.
3. In the **Return Count** drop-down list, select how many times callers can return to the component upon invalid input.
4. **Optional**: Configure audio prompt(s) or a text-to-speech message to instruct callers to retry dialing.
5. At the bottom-right corner, click **Confirm**.

Component variable

When a **Dial by Number** component with TTS configured is added to a call flow, the system stores TTS result in variable. The variable can be referenced in expression-supported

components, such as **Condition** or **Developer**, to retrieve the output value and flexibly route the call.



Note:

Since **Dial by Number** component can be added multiple times in a call flow, an index is appended to each component (e.g. Dial by Number 1, Dial by Number 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
\$DialByNumber{index}.ttsResult	String	The Text-to-Speech (TTS) result of the Dial by Number component. • <code>DialByNumberTTSResult.Success</code>: The text is successfully converted into speech, and the system plays it to caller. • <code>DialByNumberTTSResult.Timeout</code>: The text is NOT converted into speech within the timeout period, and the system routes caller directly to the next component without playing a prompt when the timeout is reached.	\$DialByNumber1.ttsResult(STRING)=DialByNumberTTSResult.Success

Component connections

Dial by Number component comes with a built-in **Invalid Input** branch that routes the call when no valid input is received. This branch can be connected to **one** component, which can be any of the components listed below.

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.

Component	Description
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.

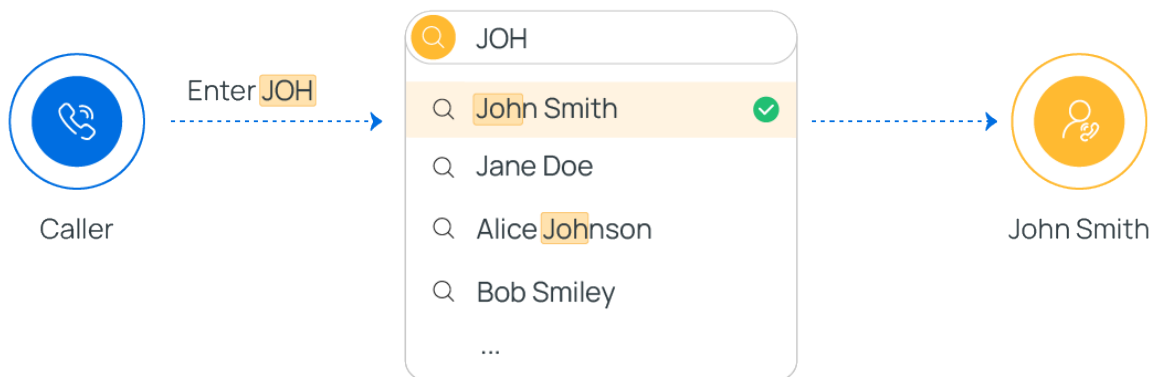
Component	Description
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

Dial by Name

This topic provides an overview of the **Dial by Name** component, and describes its configuration, usage example, as well as supported connections.

Component introduction

Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name.

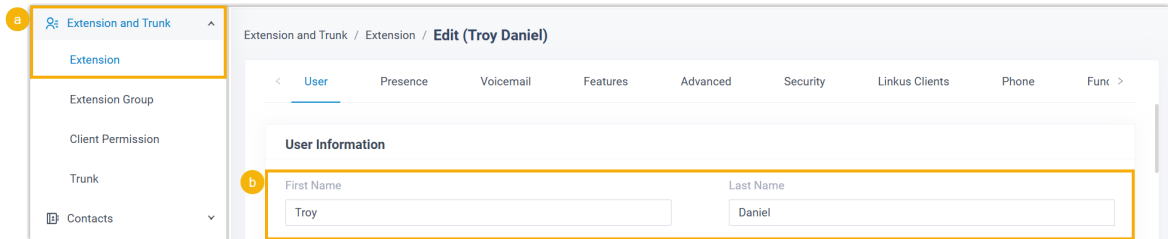


Component configuration

No configuration is needed for this component.

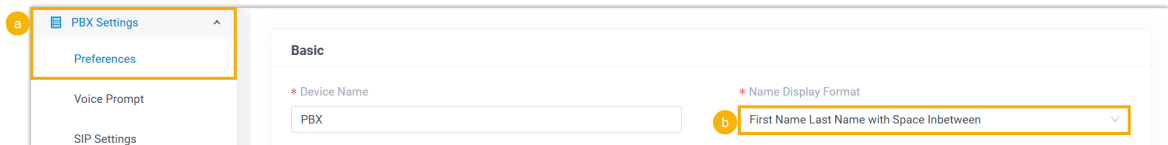
However, to ensure that callers can successfully reach extension users, make sure the following settings are properly configured.

1. Set Caller ID Name for extension.



- a. Go to **Extension and Trunk > Extension**, edit desired extension(s).
- b. In the **User Information** section, enter the user's first name and last name.

2. Set display format for extension's Caller ID Name.



- a. Go to **PBX Settings > Preferences**.
- b. In the **Name Display Format** drop-down list, select a display format.
 - To allow callers to reach extension users by first name, select **First Name Last Name with Space Inbetween**.
 - To allow callers to reach extension users by last name, select **Last Name First Name with Space Inbetween** or **Last Name First Name without Space Inbetween**.

When callers enter 3 letters, the system will search from the beginning of extensions' Caller ID Names based on the selected format.

Component usage example

When a caller reaches the **Dial by Name** component, the system guides the caller through the following process to search for an extension user.

1. The system plays the announcement "Welcome to the directory. Please enter the first three letters of your party's first or last name, using your touchtone keypad, use the 7 key for Q, and the 9 key for Z" to prompt caller for input.
2. The caller enters the first three letters of an extension user's first name.

**Tip:**

For example, to search for "Phillips Huff", the caller needs to enter 7-4-4 (corresponding to P H I).

3. The system looks for the best match and plays the corresponding announcement.
 - If no match is found, the system plays the announcement "No directory entries match your search".
 - If one or more matches are found, the system plays the announcement "[Name] extension [Number] If this is the person you are looking for, press 1 now, otherwise please press star now", then continue based on caller's input.
 - If the caller presses 1, the system will route the call to the extension user.
 - If the caller presses *, the system will continue the next match and make announcement. If no more matches, the system plays the announcement "There are no more compatible entries in the directory".

Component connections

Dial by Name component comes with a built-in **Call End** branch. This branch can be connected to one **Developer** component to interact with the PBX-native database or third-party database.

For more information about **Developer** component, see the following topics:

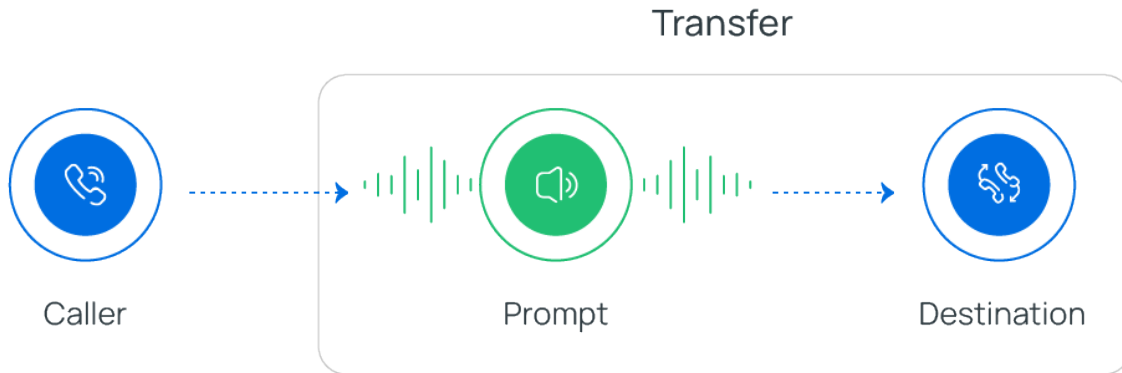
- [Get Extension Presence Status](#)
- [Set Extension Presence Status](#)
- [Get Queue Agent](#)
- [Get Queue Info](#)
- [Get Agent Status](#)
- [Set Agent Status](#)
- [Email Sender](#)
- [Database Access](#)
- [HTTP Request](#)

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 caller's connection to the current call flow. It allows you to blind transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer.



Supported transfer destinations

- Extension
- Extension Voicemail
- Ring Group
- Queue
- Group Voicemail
- Conference
- External Number
- Call Flow
- Custom

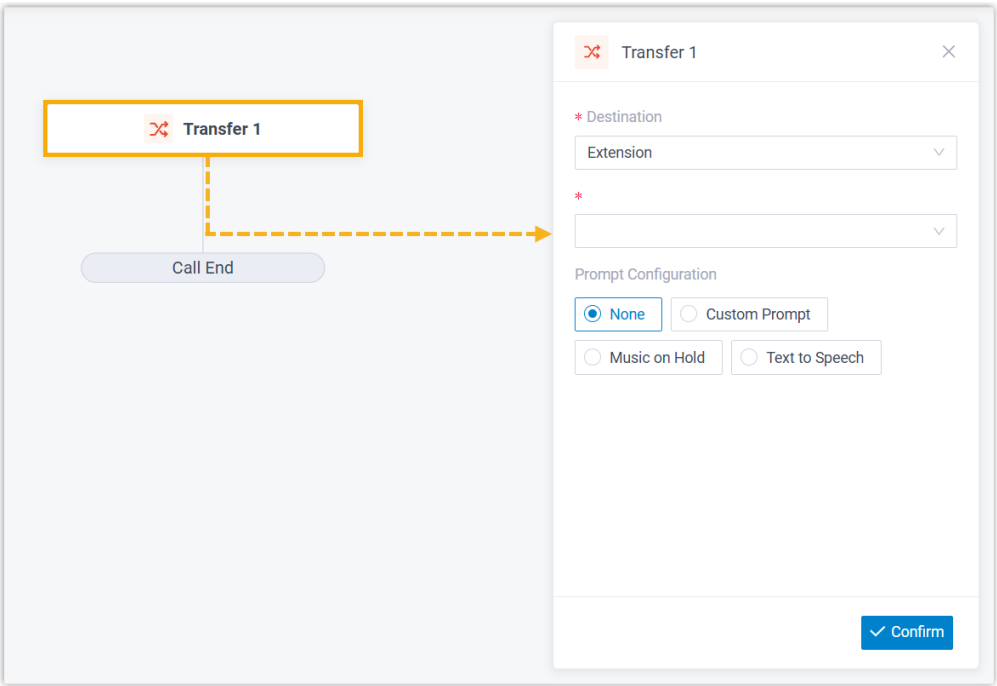
Supported prompt types

Prompt Type	Description
Custom Prompt	Choose from existing custom prompts, or create a new one by recording with an extension or uploading an audio file.
Music on Hold (MoH)	Choose from existing MoH.  Note: Before you begin, make sure the desired MoH is configured in PBX Settings > Voice Prompt > Music on Hold.

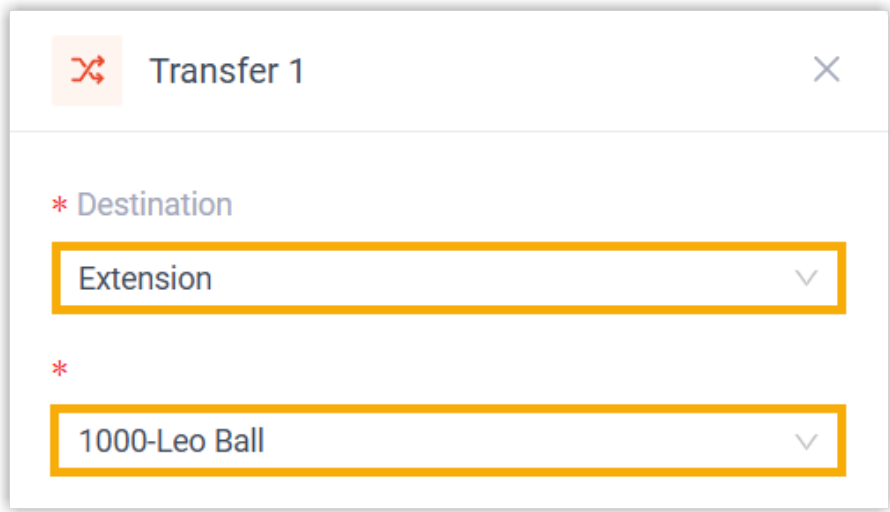
Prompt Type	Description								
	Prompt PreferencesSystem PromptMusic on HoldCustom Prompt Create New Playlist <table><tr><th>Playlist Name</th><th>Playlist Type</th><th>Play Order</th><th>Streaming Music URL</th></tr><tr><td>default</td><td>Local Audio</td><td>Alphabetical</td><td>-</td></tr></table>	Playlist Name	Playlist Type	Play Order	Streaming Music URL	default	Local Audio	Alphabetical	-
Playlist Name	Playlist Type	Play Order	Streaming Music URL						
default	Local Audio	Alphabetical	-						
Text to Speech (TTS)	Note: Before you begin, make sure the Text-to-Speech feature is enabled in Integrations > AI. Integrations / AI ☒ Text-to-Speech ☐ Voicemail Transcription								

Component configuration

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



2. In the **Destination** drop-down lists, specify the destination where callers will be transferred.



Option	Description
Extension	Transfer incoming calls to a specific extension.
Extension Voicemail	Transfer incoming calls to a specific extension's voicemail.
Ring Group	Transfer incoming calls to a specific ring group.
Queue	Transfer incoming calls to a specific queue.

Option	Description
Group Voicemail	Transfer incoming calls to a specific group voicemail.
Conference	Transfer incoming calls to a specific conference.
External Number	Transfer incoming calls to a specific external number.
Call Flow	Transfer incoming calls to a specific call flow.
Custom	Transfer incoming calls to a custom destination. You can specify the destination in either of the following ways: • Directly specify a specific number: In the Destination Number field, enter a specific number. • Dynamically specify a number: Click FX to configure an expression. Example: <code>\$HttpRequest1.responseContent</code> In this example, the system retrieves the on-duty phone number from a web-based calendar via the upstream HTTP Request 1 component, and routes incoming calls to that number.  Note: For more information about the expression, see Variables and Functions in Yeastar Expression. If the custom destination has a voicemail box (e.g. extension voicemail or group voicemail), you can additionally enable Transfer to Voicemail to allow callers to reach the mailbox directly.  Note: For External Number: If Filter Number is enabled (Path: PBX Settings > Preferences), the system will automatically remove all special characters (except digits, +, *, and #) from the custom number before sending to the trunk.

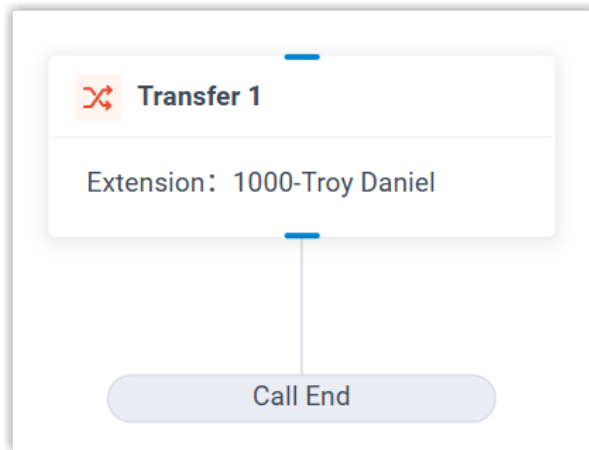
3. **Optional:** Configure audio prompt(s) or a text-to-speech message to inform callers of the call transfer.

Item	Description
	 The looped prompt will be overridden if the next connected component is configured with a prompt.
Music on Hold	Prompt Configuration ☐ None ☐ Custom Prompt ☒ Music on Hold ☐ Text to Speech * Music on Hold default a. Select Music on Hold. b. Select an existing MoH from the drop-down list of Music on Hold.
Text to Speech (TTS)	Prompt Configuration ☐ Custom Prompt ☐ Music on Hold ☒ Text to Speech * Name * Language English (US * Voice Danielle (Female) * Text + Placeholder Please enter the text content in the same language as the audio file. * Speed <input type="range"/> 50 * Volume <input type="range"/> 50 ✓ Confirm

Item	Description
	a. Select Text to Speech. b. Complete the rest of the configurations to convert text into speech.  Note: Both static text and dynamic text are supported to convert into speech. • Static text: Enter text directly in the Text field. • Dynamic text: Enter text and configure placeholders with corresponding variables to dynamically compose the text. Optionally, you can configure prompt generation timeout. If speech is not generated within the timeout period, caller will be routed to the next component without hearing a prompt. Example: The new speech will be saved to the system and synchronized to PBX Settings > Voice Prompt > Custom Prompt.

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

The selected destination will appear on the component.



Component variable

When a **Transfer** component with TTS configured is added to a call flow, the system stores TTS result in variable. The variable can be referenced in expression-supported components, such as **Condition** or **Developer**, to retrieve the output value and flexibly route the call.



Note:

Since **Transfer** component can be added multiple times in a call flow, an index is appended to each component (e.g. Transfer 1, Transfer 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>\$Transfer{index}.ttsResult</code>	String	The Text-to-Speech (TTS) result of the Transfer component. <code>TransferTTSResult.Success</code>: The text is successfully converted into speech, and the system plays it to caller. <code>TransferTTSResult.Timeout</code>: The text is NOT converted into	<code>\$Transfer1.ttsResult(STRING)=TransferTTSResult.Success</code>

Variable	Type	Description	Example Value
		speech within the timeout period, and the system routes caller directly to the next component without playing a prompt when the timeout is reached.	

Component connections

Transfer component comes with a built-in **Call End** branch. This branch can be connected to one **Developer** component to interact with the PBX-native database or third-party database.

For more information about **Developer** component, see the following topics:

- [Get Extension Presence Status](#)
- [Set Extension Presence Status](#)
- [Get Queue Agent](#)
- [Get Queue Info](#)
- [Get Agent Status](#)
- [Set Agent Status](#)
- [Email Sender](#)
- [Database Access](#)
- [HTTP Request](#)

Hang Up Call

This topic provides an overview of the **Hang Up Call** component, and describes its configuration as well as supported connections.

Component introduction

Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected.

Component configuration

No configuration is needed for this component.

Component connections

Hang Up Call component comes with a built-in **Call End** branch. This branch can be connected to one **Developer** component to interact with the PBX-native database or third-party database.

For more information about **Developer** component, see the following topics:

- [Get Extension Presence Status](#)
- [Set Extension Presence Status](#)
- [Get Queue Agent](#)
- [Get Queue Info](#)
- [Get Agent Status](#)
- [Set Agent Status](#)
- [Email Sender](#)
- [Database Access](#)
- [HTTP Request](#)

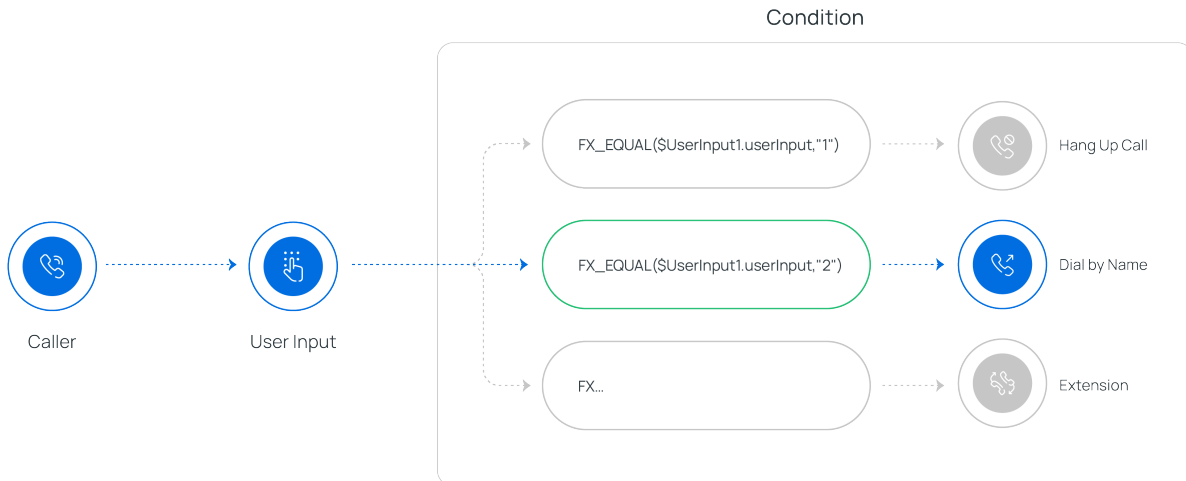
Flow Components

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



Component configuration

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

- [Add branch\(es\) for different conditions](#)
- [Specify destination for invalid input](#)

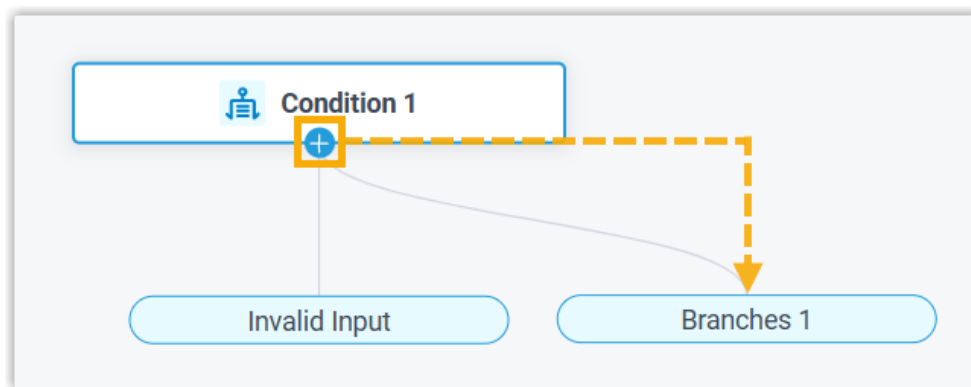
Add branch(es) for different conditions

1. Click  on the component to add a branch.

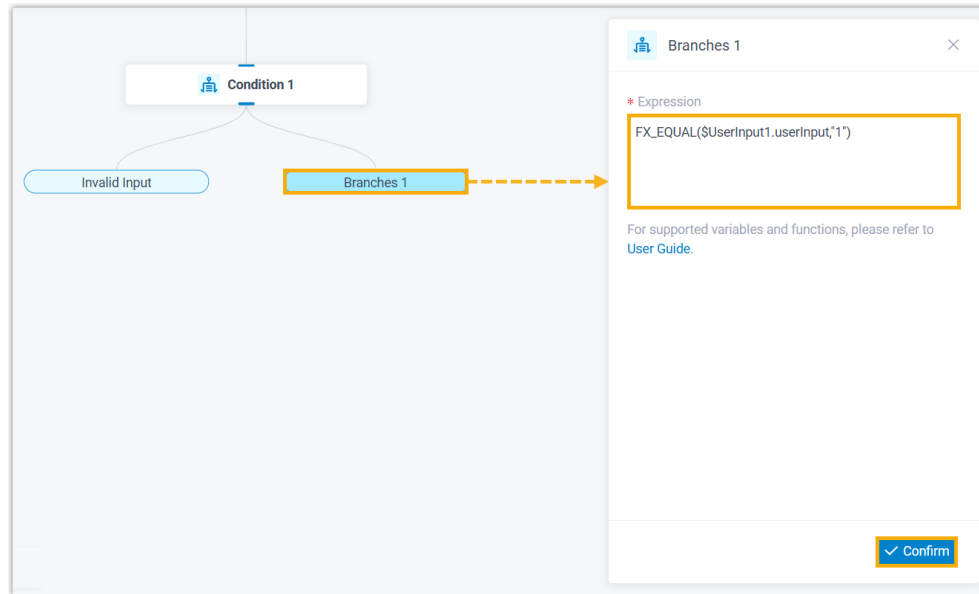


Note:

You can add up to **10 branches**, each associated with a unique expression.



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



For example, enter `FX_EQUAL($UserInput1.userInput, "1")`. In this way, if the caller presses 1, the call will be routed to the component connected to the branch.

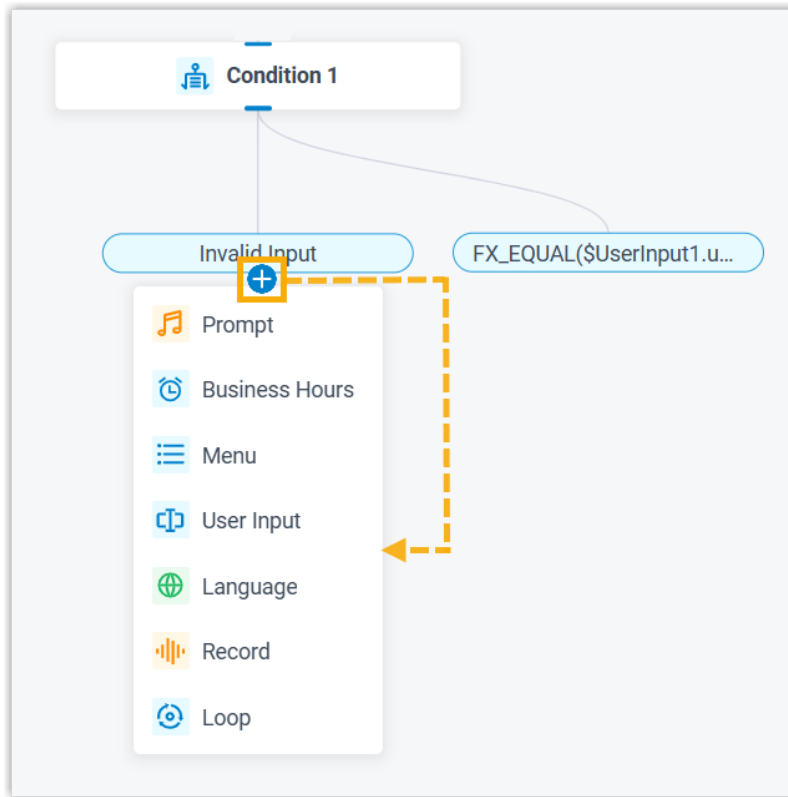


Note:

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

Specify destination for invalid input

Click  on **Invalid Input** to add a component, which will be executed if none of the branch conditions are met.



Component connections

Condition component comes with a built-in **Invalid Input** branch, and supports up to 10 additional branches for specific conditions.

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

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt .
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours .
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu .

Component	Description
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status

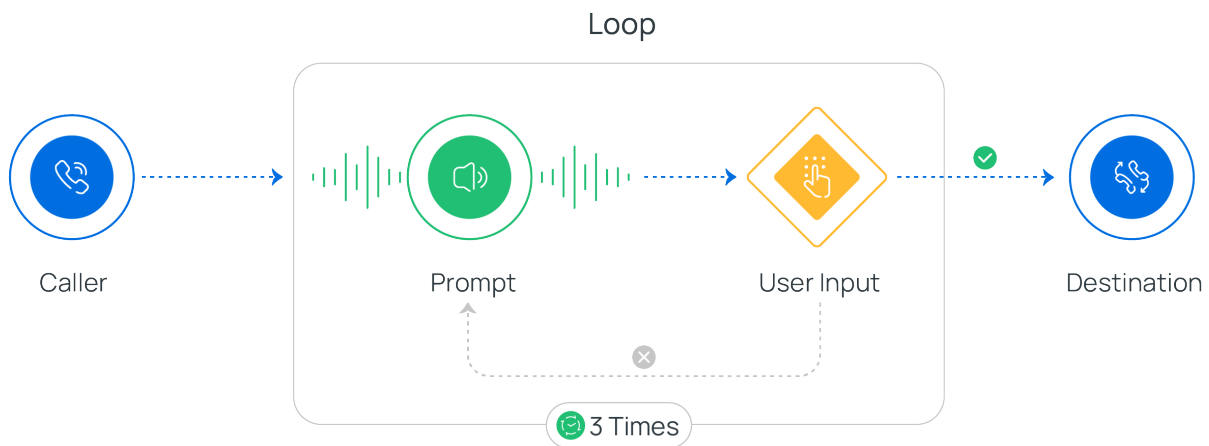
Component	Description
	• Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

Loop

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

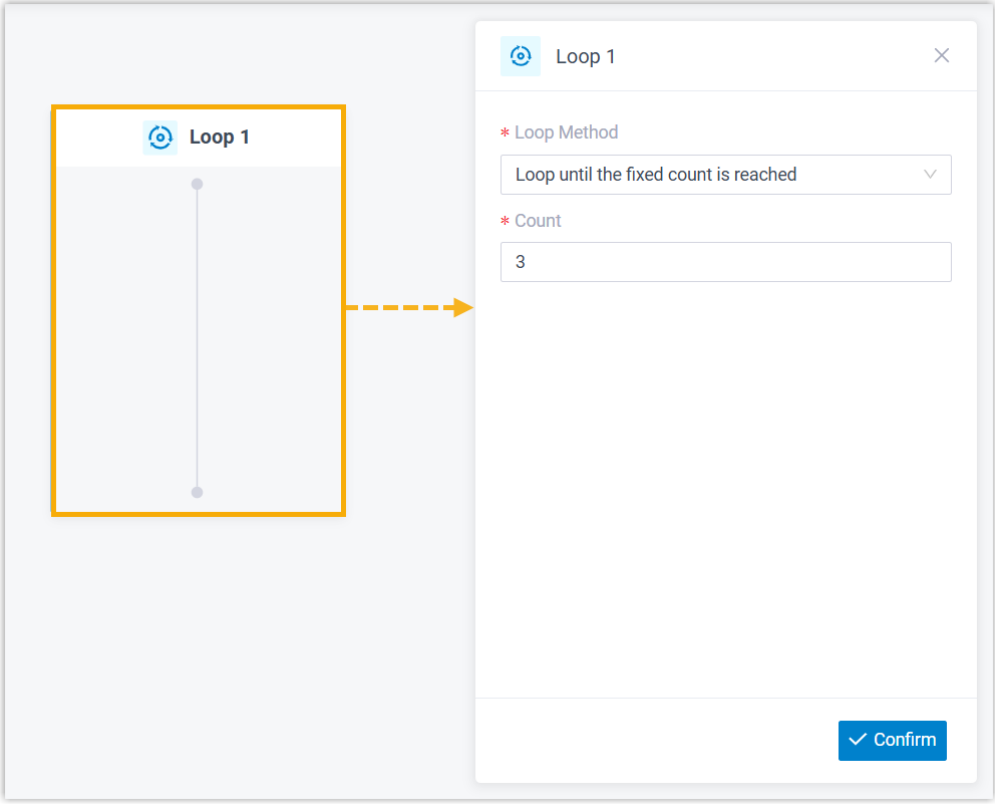
Component introduction

Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met, helpful in repeating menu options or returning to the previous menu.

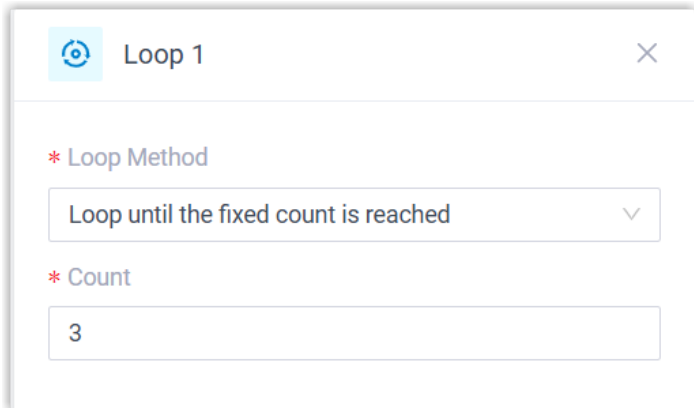


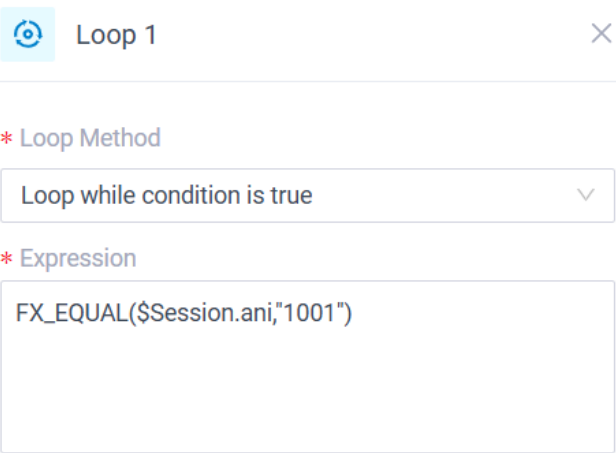
Component configuration

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



2. Set the loop method, then click **Confirm**.

Loop Method	Description
Loop until the fixed count is reached	 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:

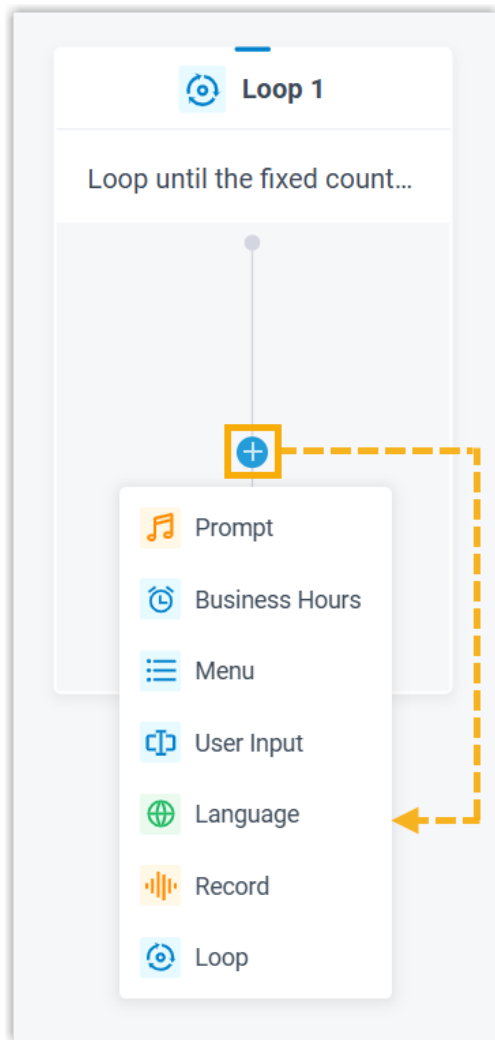
Loop Method	Description
Loop while condition is true	 You can enter a value between 1 and 99.  * Loop Method Loop while condition is true * Expression FX_EQUAL(\$Session.ani,"1001") For supported variables and functions, please refer to User Guide. a. In the Loop Method drop-down list, select Loop while condition is true. b. In the Expression field, enter a function. Example: FX_EQUAL(\$Session.ani,"1001") In this example, a call will enter the loop when the Caller ID is 1001. For more information about the function syntax, see Variables and Functions in Yeastar Expression.

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

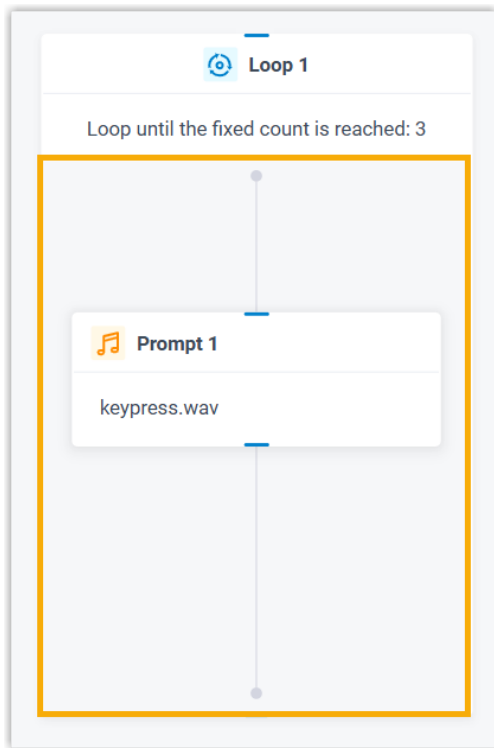


Note:

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



The components to be repeated will appear inside the **Loop** component.



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.

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.

Component	Description
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status

Component	Description
	• Set Agent Status • Email Sender • Database Access • HTTP Request

Developer 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.

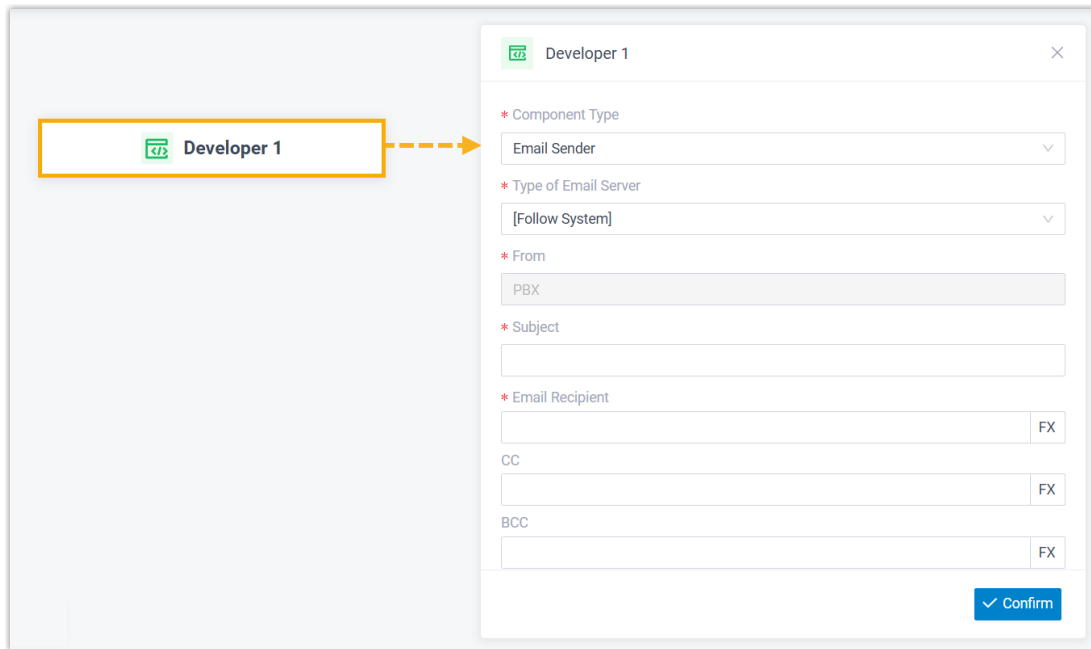
Component introduction

Get Extension Presence Status component allows you to query current status of a specific extension, including its presence status, call status, and ringing status.

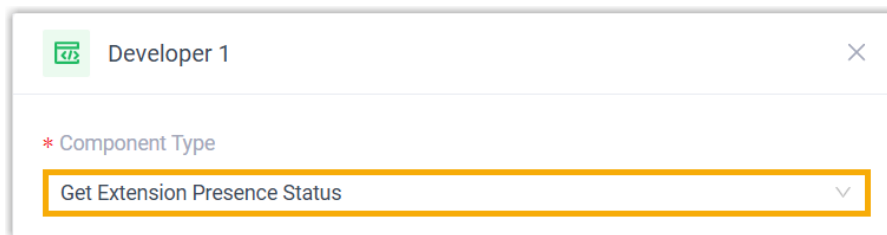
You can either select a specific extension directly, or configure an expression to dynamically specify the target extension. The query results are stored in variables, which can be used as input for subsequent components or for condition evaluation.

Component configuration

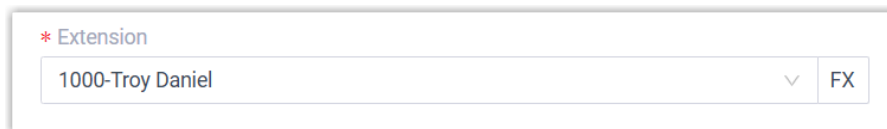
1. After adding **Developer** component to a call flow, click **Developer** component to proceed with the configuration.



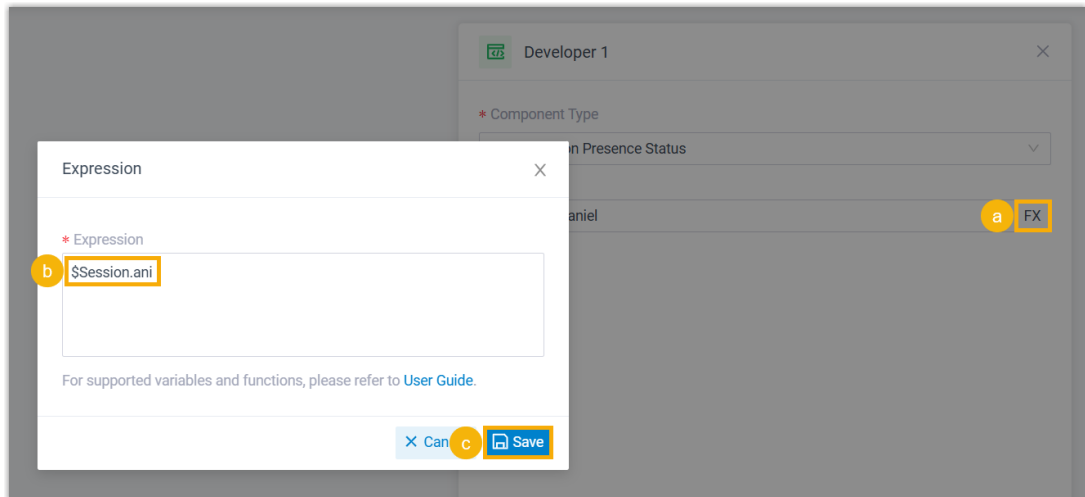
2. In the **Component Type** drop-down list, select **Get Extension Presence Status**.



3. Specify the target extension using one of the following methods.



- **Extension:** Select a specific extension from the drop-down list.
- **FX:** Click the icon to configure an expression.



For example, enter `$Session.ani` to use the caller's number (Caller ID) as the target extension.



Note:

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

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

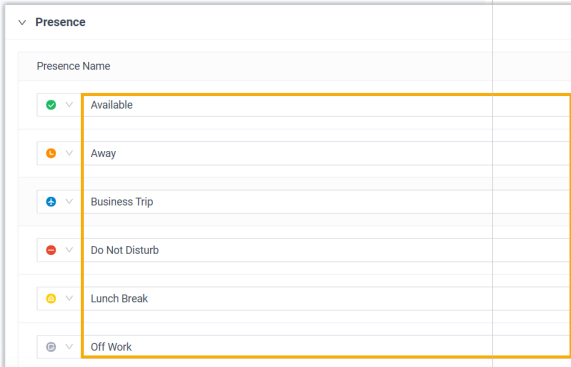
Component variables

When a **Get Extension Presence Status** component is added to a call flow, the system stores the extension's statuses in variables. These variables can be referenced in expression-supported components, such as **Condition** or **Developer**, 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 call flow, an index is appended to 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>}.isInCall</code>	Boolean	Whether the extension is currently on a call. • True: At least one endpoint registered to the extension is on a call. • False: None of the endpoints registered to the extension are on a call.	True
<code>\$GetExtensionStatus{ <i>index</i>}.isInRing</code>	Boolean	Whether the extension is currently ringing. • True: At least one endpoint registered to the extension is ringing. • False: None of the endpoints registered to the extension are ringing.	False

Component connections

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

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.

Component	Description
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

Set Extension Presence Status

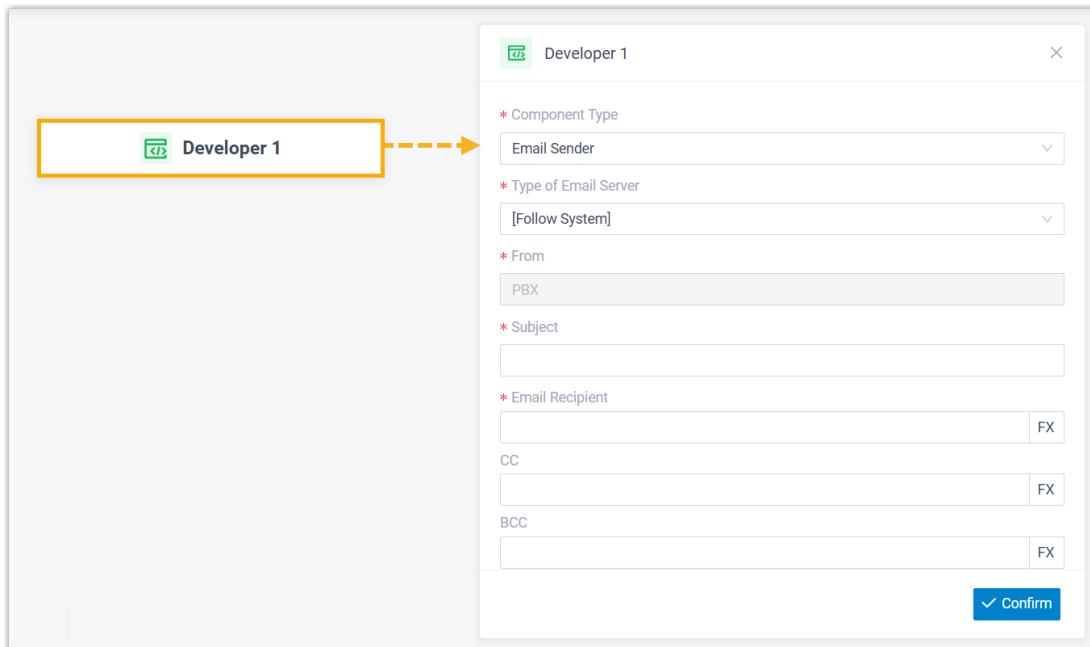
This topic provides an overview of the **Set Extension Presence Status** component, and describes its configuration as well as supported connections.

Component introduction

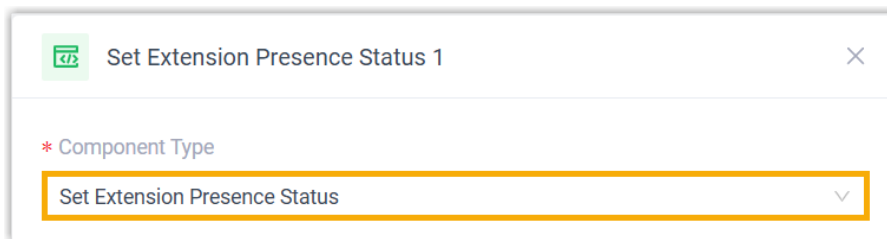
Set Extension Presence Status component allows you to change current presence status of a specific extension. You can either select a specific extension directly, or configure an expression to dynamically specify the target extension, and then define the target presence status.

Component configuration

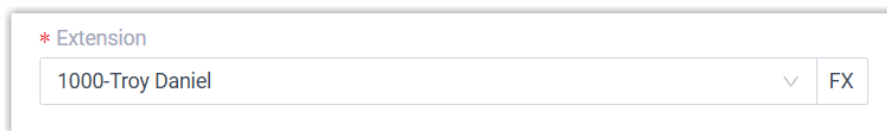
1. After adding **Developer** component to a call flow, click **Developer** component to proceed with the configuration.



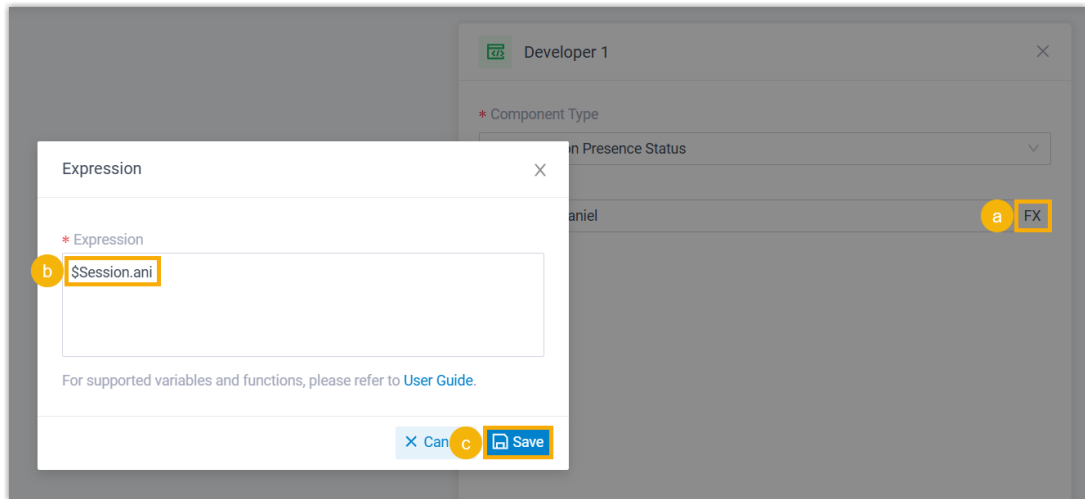
2. In the **Component Type** drop-down list, select **Set Extension Presence Status**.



3. Specify the target extension using one of the following methods.



- **Extension:** Select a specific extension from the drop-down list.
- **FX:** Click the icon to configure an expression.



For example, enter `$Session.ani` to use the caller's number (Caller ID) as the target extension.



Note:

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

4. In the **Presence** drop-down list, select a presence status.

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

Component connections

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

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.

Component	Description
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.

Component	Description
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

Get Queue Agent

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

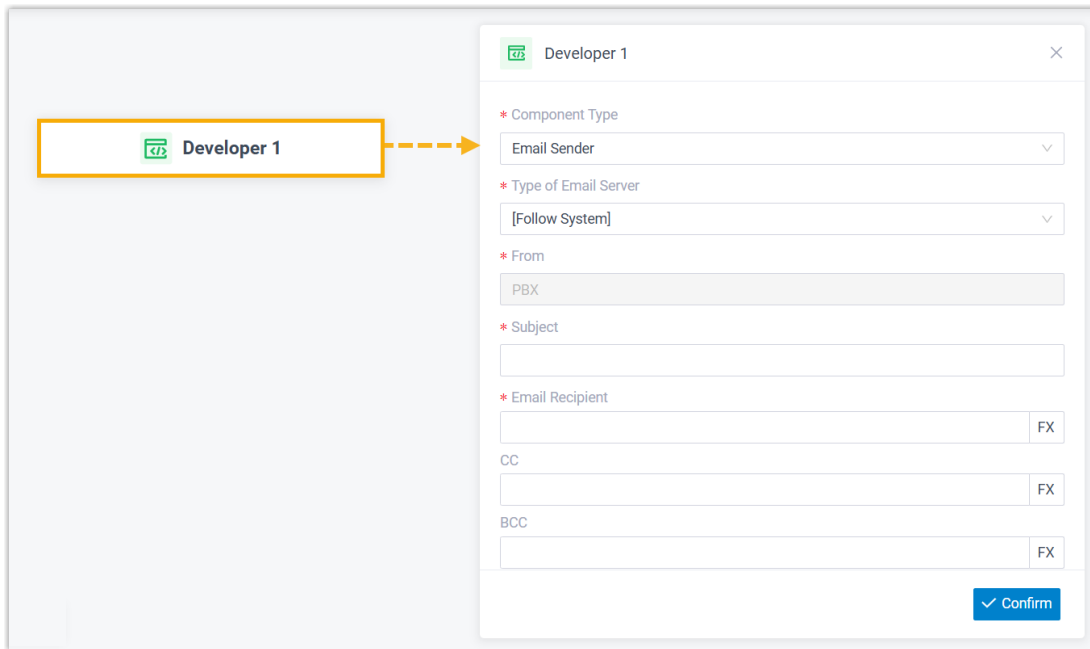
Component introduction

Get Queue Agent component allows you to query a list of agents with specific status(es) from a specific queue.

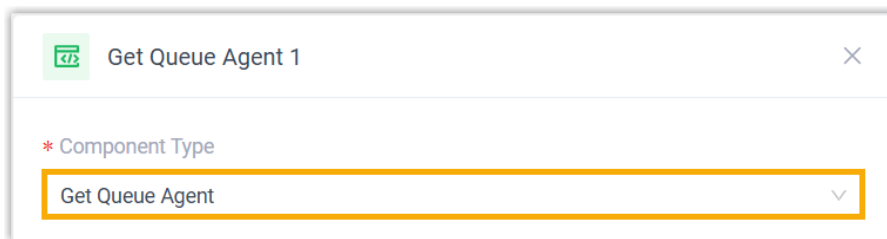
You can either select a queue directly, or configure an expression to dynamically specify the target queue, and then select agent status(es) to filter agents accordingly. The query results are stored in variables, which can be used as input for subsequent components or for condition evaluation.

Component configuration

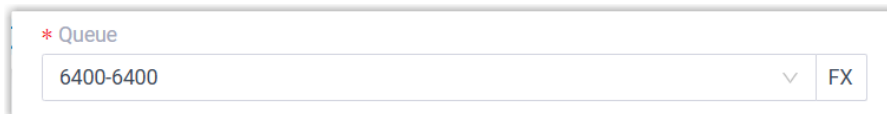
1. After adding **Developer** component to a call flow, click **Developer** component to proceed with the configuration.



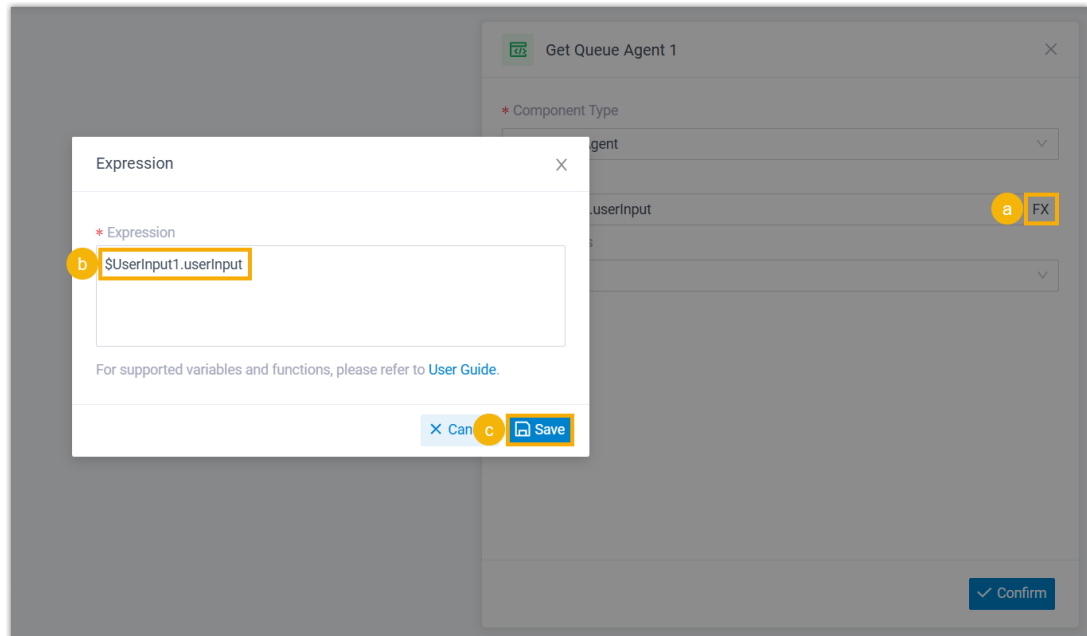
2. In the **Component Type** drop-down list, select **Get Queue Agent**.



3. Specify the target queue using one of the following methods.



- **Queue:** Select a specific queue from the drop-down list.
- **FX:** Click the icon to configure an expression.



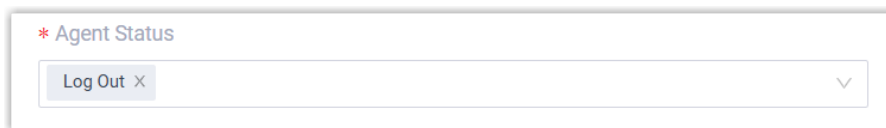
For example, enter `$UserInput1.userInput` to retrieve the queue number that the caller entered.



Note:

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

4. In the **Agent Status** drop-down list, select one or more agent statuses.



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

Component variables

When a **Get Queue Agent** component is added to a call flow, the system stores agent information in variables. These variables can be referenced in expression-supported components, such as **Condition** or **Developer**, to retrieve the agent data and respond accordingly based on the result.



Note:



Since **Get Queue Agent** component can be added multiple times in a call flow, an index is appended to each component (e.g. Get Queue Agent 1, Get Queue Agent 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
\$GetQueueAgent{ <i>index</i> }.agentList	Object	The list of agents with a specific status in a specific queue, returning each agent's extension number as well as their current status.	"1001,Log Out;1003,Log Out;1004,Log Out"
\$GetQueueAgent{ <i>index</i> }.agentNumberList	Object	The list of agents with a specific status in a specific queue, returning each agent's extension number.	"1001,1003,1004"

**Tip:**

You can use the following functions to get the values from the resulting list:

- `GET_LIST_ITEM_COUNT`: Return the number of items in the list.

For example, use `FX_GREATER_THAN(FX_GET_LIST_ITEM_COUNT($GetQueueAgent1.agentNumberList), 0)` to check if the number of logged-in agents is greater than 0.

- `GET_LIST_ITEM`: Retrieve the value at the specified index in the list.

For example, use `FX_GET_LIST_ITEM($GetQueueAgent1.agentNumberList, 2)` to get the third agent in the list.

Component connections

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

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.

Component	Description
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.

Component	Description
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

Get Queue Info

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

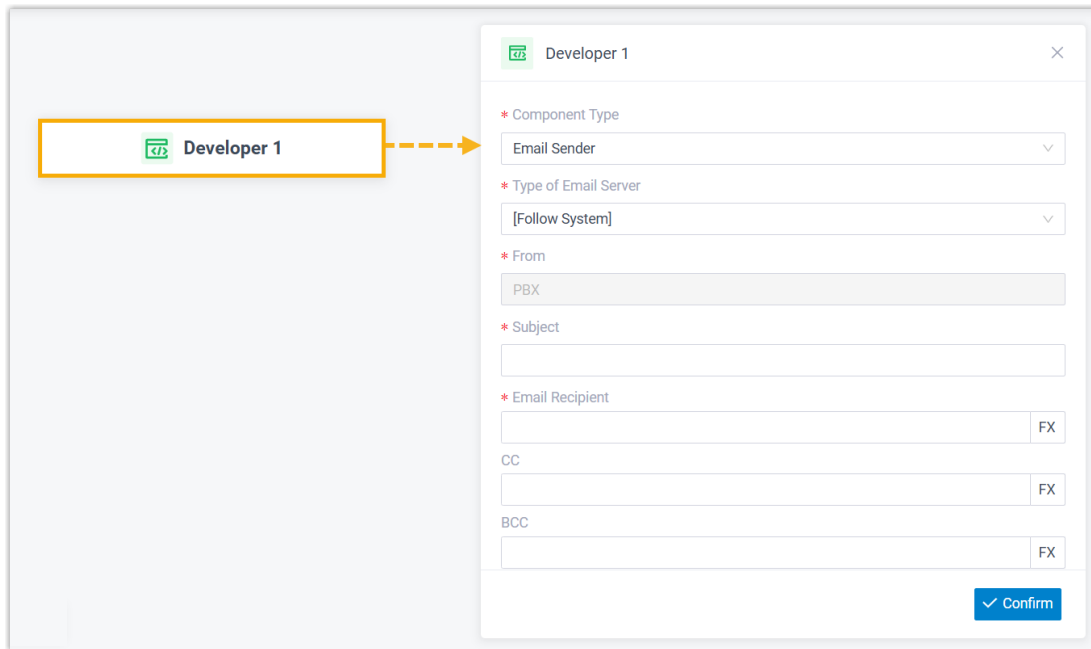
Component introduction

Get Queue Info component allows you to query real-time status of a specific queue, including the number of available agents, unavailable agents, active calls, and calls waiting to be connected.

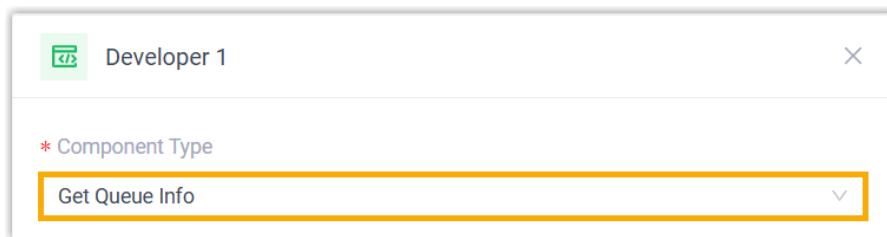
You can either select a queue directly, or configure an expression to dynamically specify the target queue. The query results are stored in variables, which can be used as input for subsequent components or for condition evaluation.

Component configuration

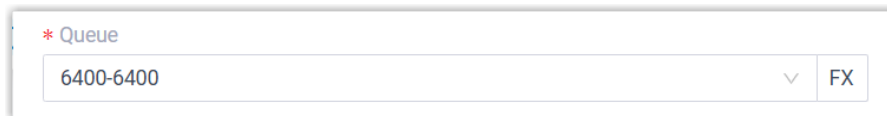
1. After adding **Developer** component to a call flow, click **Developer** component to proceed with the configuration.



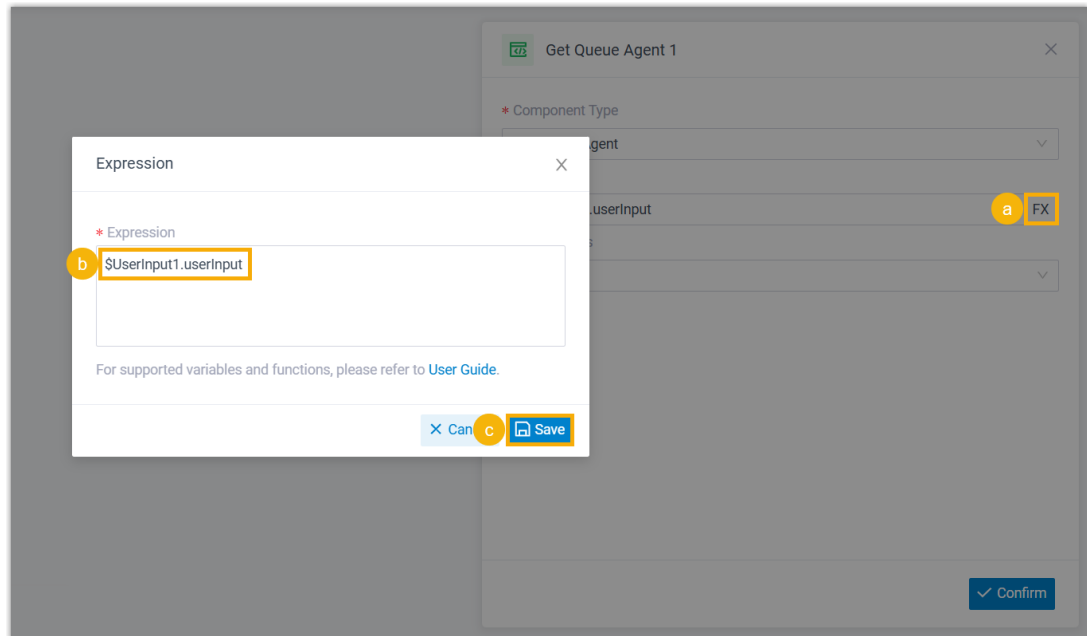
2. In the **Component Type** drop-down list, select **Get Queue Info**.



3. Specify the target queue using one of the following methods.



- **Queue:** Select a specific queue from the drop-down list.
- **FX:** Click the icon to configure an expression.



For example, enter `$UserInput1.userInput` to retrieve the queue number that the caller entered.



Note:

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

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

Component variables

When a **Get Queue Info** component is added to a call flow, the system stores queue information in variables. These variables can be referenced in expression-supported components, such as **Condition** or **Developer**, to retrieve the values and respond accordingly based on the result.



Note:

Since **Get Queue Info** component can be added multiple times in a call flow, an index is appended to each component (e.g. Get Queue Info 1, Get Queue 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
\$GetQueueInfo{ <i>index</i> }.availableAgents	Integer	The number of agents currently available to take calls.	5
\$GetQueueInfo{ <i>index</i> }.unavailableAgents	Integer	The number of agents currently unavailable to take calls.	1
\$GetQueueInfo{ <i>index</i> }.callsActive	Integer	The number of calls currently being answered by queue agents.	2
\$GetQueueInfo{ <i>index</i> }.callsWaiting	Integer	The number of calls currently waiting in the queue.	10

Component connections

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

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt .
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours .
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu .
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input .
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language .
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record .

Component	Description
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

Get Agent Status

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

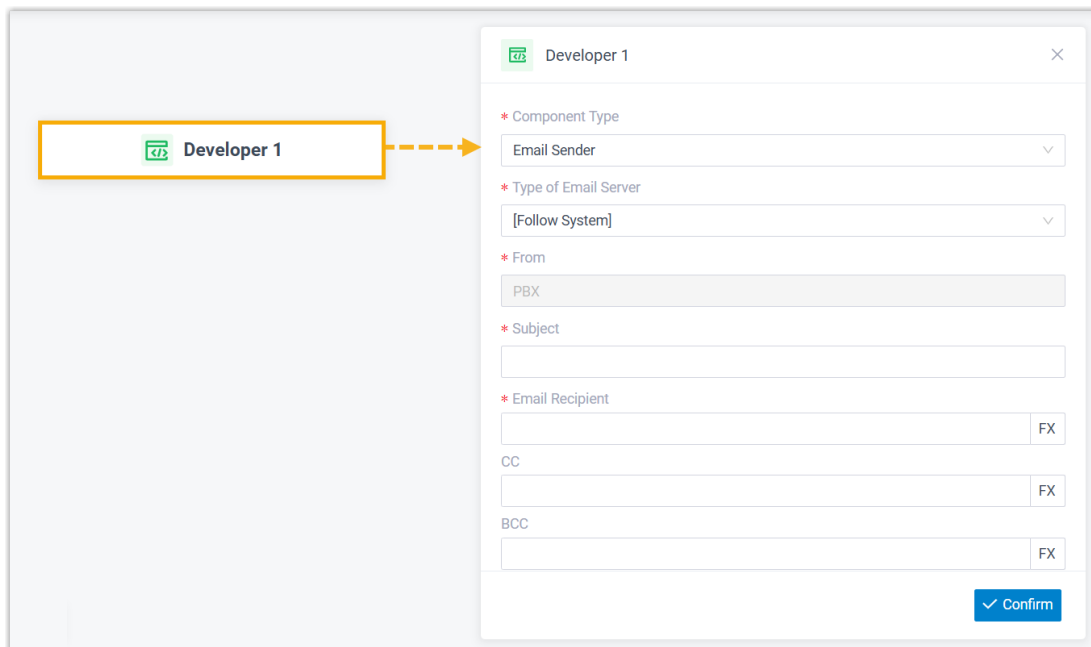
Component introduction

Get Agent Status component allows you to query current status of a specific agent in a specific queue, including the agent's login status, call status, and ringing status.

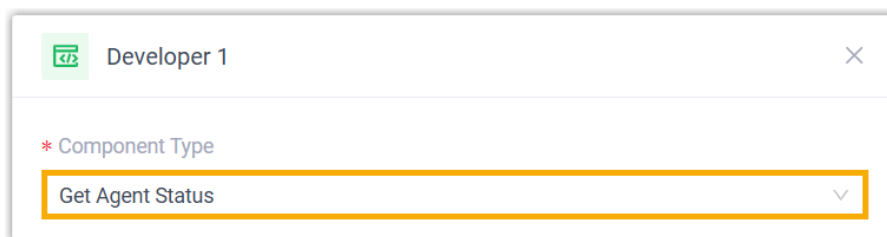
You can either select the queue and agent directly, or configure expressions to dynamically specify the target queue and agent. The query results are stored in variables, which can be used as input for subsequent components or for condition evaluation.

Component configuration

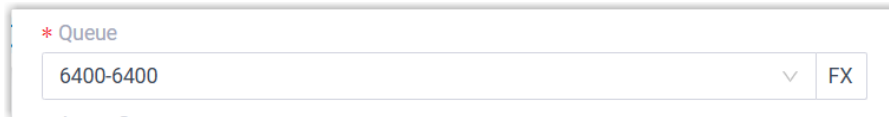
1. After adding **Developer** component to a call flow, click **Developer** component to proceed with the configuration.



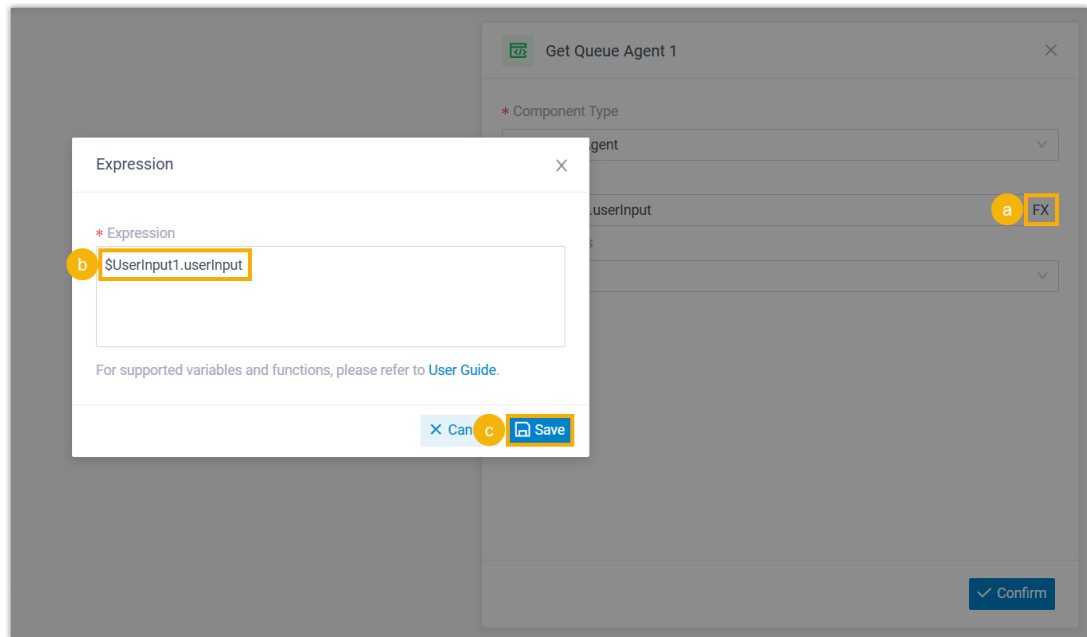
2. In the **Component Type** drop-down list, select **Get Agent Status**.



3. Specify the target queue using one of the following methods.



- **Queue:** Select a specific queue from the drop-down list.
- **FX:** Click the icon to configure an expression.



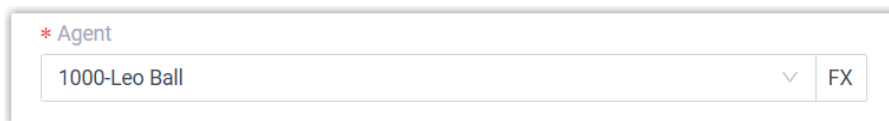
For example, enter `$UserInput1.userInput` to retrieve the queue number that the caller entered.



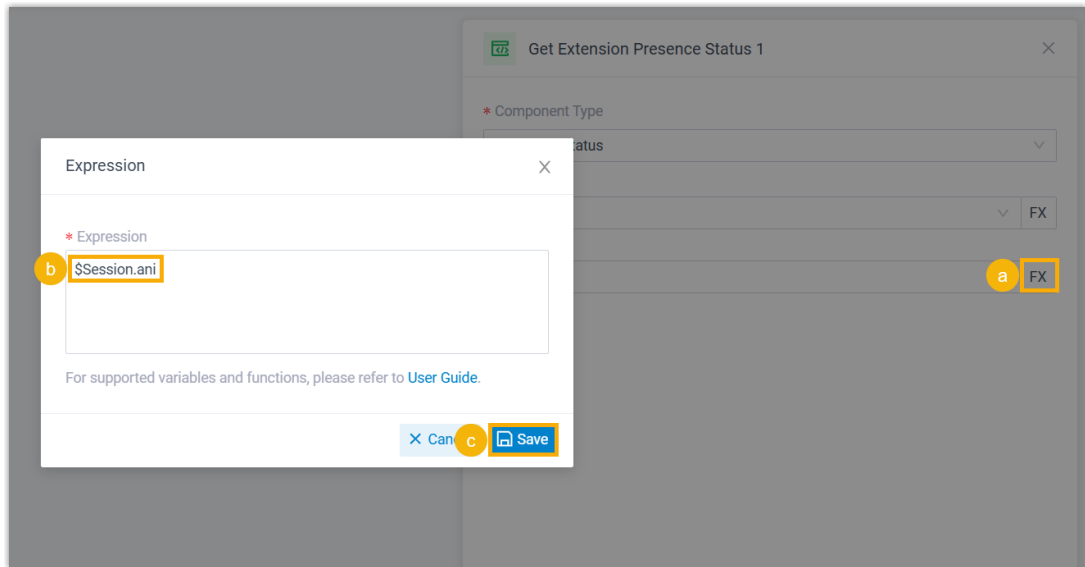
Note:

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

4. Specify the target agent using one of the following methods.



- **Agent:** Select a specific agent from the drop-down list.
- **FX:** Click the icon to configure an expression.



For example, enter `$Session.ani` to use the caller's number (Caller ID) as the target agent.



Note:

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

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

Component variables

When a **Get Agent Status** component is added to a call flow, the system stores agent status in variables. These variables can be referenced in expression-supported components, such as **Condition** or **Developer**, to retrieve the agent status and respond accordingly based on the result.



Note:

Since **Get Agent Status** component can be added multiple times in a call flow, an index is appended to each component (e.g. Get Agent Status 1, Get Agent 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
\$GetAgentStatus{ <i>index</i> }.currentProfile	String	Name and additional information of the agent's current status.  Note: When using functions to evaluate an agent's status, ensure that the status constant matches one of the supported values listed below. • Status for Static Agents: ◦ "Pause{<i>pause_reason</i>}" ◦ "Unpause" • Status for Dynamic Agents: ◦ "Log In" ◦ "Log Out" ◦ "Pause{<i>pause_reason</i>}"	"Log Out"
\$GetAgentStatus{ <i>index</i> }.isInCall	Boolean	Whether the agent is currently on a call. • True: The agent is on a call. • False: The agent is NOT on a call.	True
\$GetAgentStatus{ <i>index</i> }.isInRing	Boolean	Whether the agent's extension is currently ringing. • True: The agent's extension is ringing. • False: The agent's extension is NOT ringing.	True

Component connections

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

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers.

Component	Description
	For more information, see Prompt .
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.

Component	Description
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

Set Agent Status

This topic provides an overview of the **Set Agent Status** component, and describes its configuration as well as supported connections.

Component introduction

Set Agent Status component allows you to change current status of a specific agent in specific queue(s). You can either select queue(s) and agent directly, or configure an expression to dynamically specify the target queue(s) and agent.

Component configuration

1. After adding **Developer** component to a call flow, click **Developer** component to proceed with the configuration.

2. In the **Component Type** drop-down list, select **Set Agent Status**.

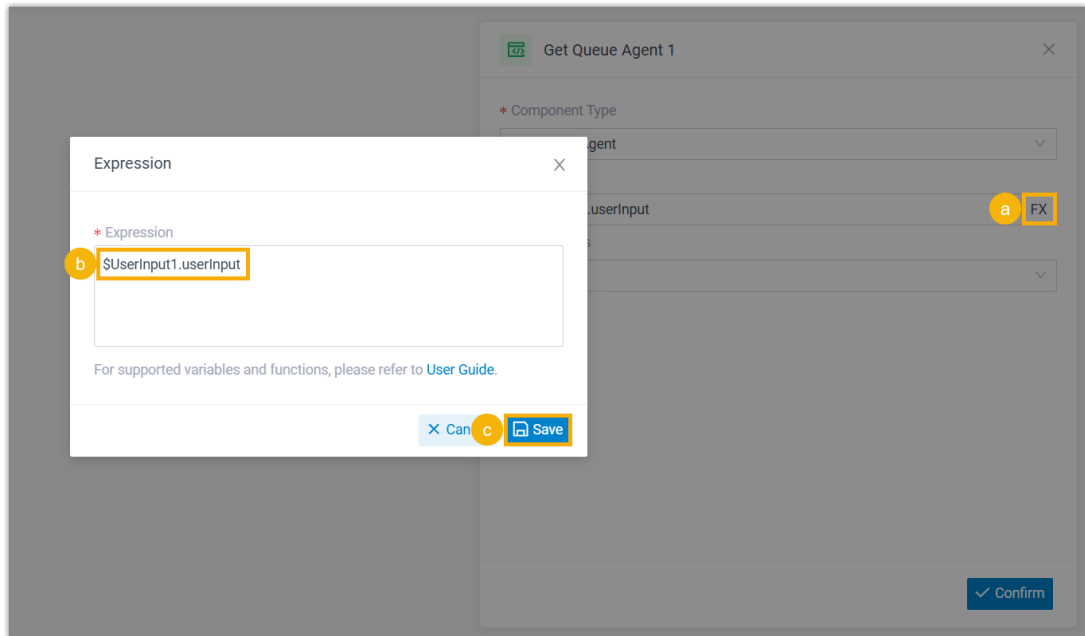
3. In the **Range** drop-down list, select the queue range.

- **Global:** Change the agent's status in all queues.
- **Specific:** Change the agent's status in a specific queue.

If you select this option, select a queue from the drop-down list.

4. Specify the target queue using one of the following methods.

- **Queue:** Select a specific queue from the drop-down list.
- **FX:** Click the icon to configure an expression.



For example, enter `$UserInput1.userInput` to retrieve the queue number that the caller entered.

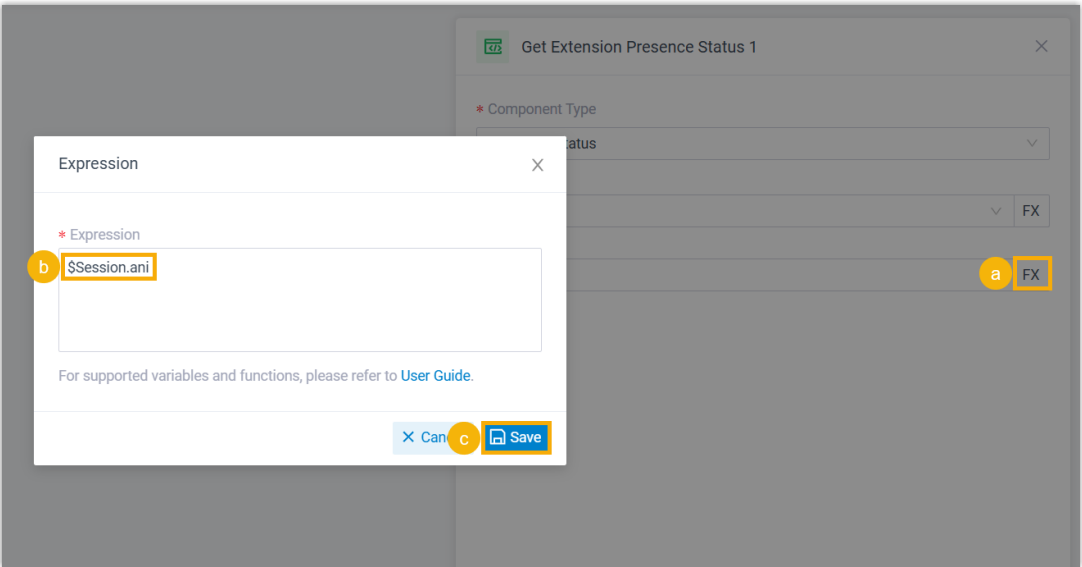


Note:

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

5. Specify the target agent using one of the following methods.

- **Agent:** Select a specific agent from the drop-down list.
- **FX:** Click the icon to configure an expression.



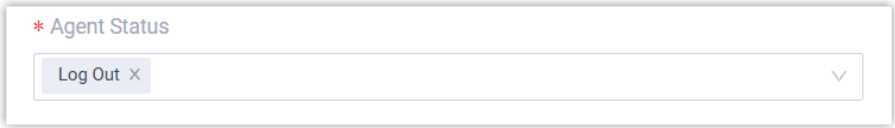
For example, enter `$Session.ani` to use the caller's number (Caller ID) as the target agent.



Note:

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

6. In the **Agent Status** drop-down list, select a status to which the agent will be switch.



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

Component connections

Set Agent Status component can be connected to **one** component, which can be any of the components listed below.

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt .
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day.

Component	Description
	For more information, see Business Hours .
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.

Component	Description
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

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 allows you to send emails, enabling real-time notifications, alerts, or delivery of user-specific information. You can specify the email server to use, and customize email fields such as recipients (To, CC, BCC), subject, message body, and file attachments.

Supported email servers

- Yeastar SMTP Server
- Custom Email Server



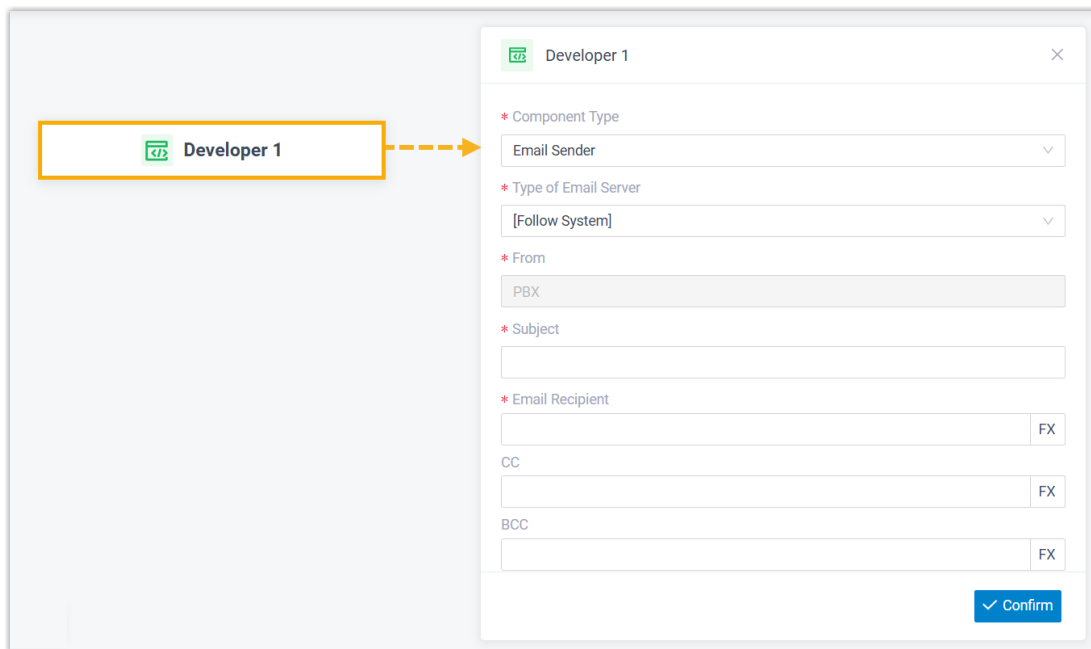
Note:

- Email servers from any provider are supported.
- To use a custom email server, you must configure it in **System > Email > Email Server** in advance.

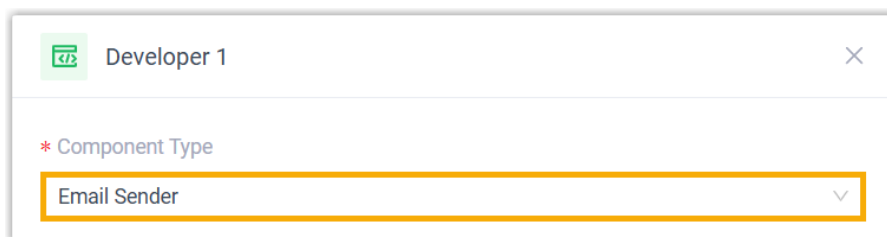
If you want to use **Gmail** or **Outlook**, refer to the guides for detailed instructions: [Set up Gmail as an Email Server](#) and [Set up Outlook as an Email Server](#).

Component configuration

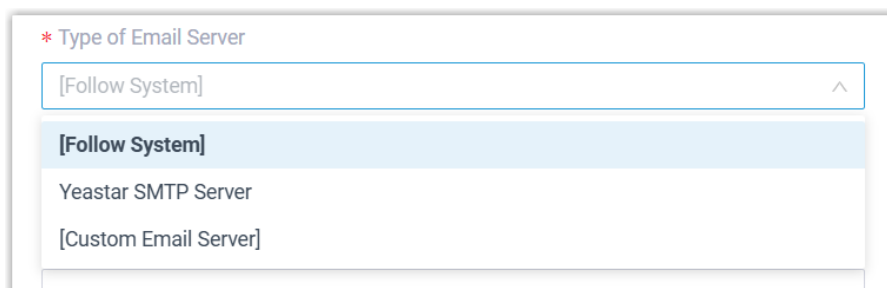
1. After adding **Developer** component to a call flow, click **Developer** component to proceed with the configuration.



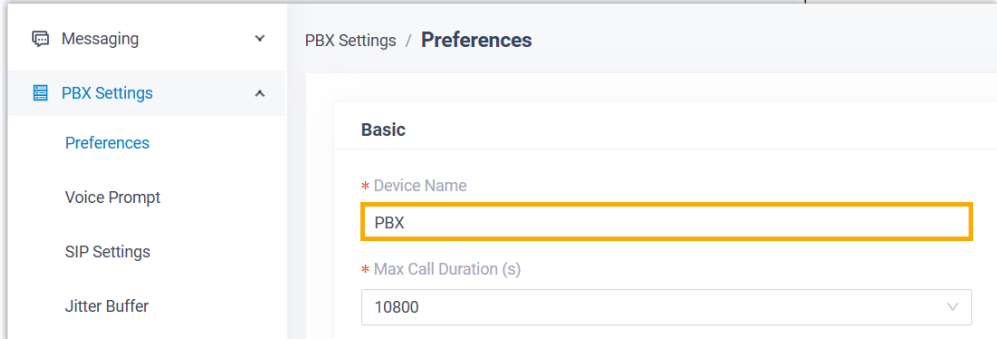
2. In the **Component Type** drop-down list, select **Email Sender**.



3. In the **Type of Email Server** drop-down list, select an option.



4. Customize email fields.

Item	Description
From	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. 
Subject	Enter the subject of the email.
Email Recipient	Recipient email address, which will appear as the "To" address for outgoing emails sent by the PBX. You can enter specific email addresses, or click FX to configure an expression.  Note: - Up to 5 email addresses are supported; Use commas , to separate multiple addresses. - For more information about the expression, see Variables and Functions in Yeastar Expression. Expression example: <pre data-bbox="519 1470 1096 1512">FX_TO_STRING(\$DatabaseAccess1.queryResult)</pre> In this example, the system retrieves the email address returned by the upstream Database Access component, which runs the SQL statement <code>SELECT email_addr FROM user WHERE number='\$Session.ani'</code>; to query email address based on the caller's number, and then convert the email address to a string.
CC	Carbon Copy (CC) email address, which will appear as the "Cc" address for outgoing calls sent by the PBX.

Item	Description
	You can enter specific email addresses, or click FX to configure an expression.  Note: - Up to 5 email addresses are supported; Use commas , to separate multiple addresses. - For more information about the expression, see Variables and Functions in Yeastar Expression.
BCC	Blind Carbon Copy (BCC) email address, which will appear as the "Bcc" address for outgoing calls sent by the PBX. You can enter specific email addresses, or click FX to configure an expression.  Note: - Up to 5 email addresses are supported; Use commas , to separate multiple addresses. - For more information about the expression, see Variables and Functions in Yeastar Expression.
Content	Enter the body content of the email.  Note: You can add system-related information using template variables. Template Variables {{.PBXName}}: the name of this PBX. It can be set up in the PBX Settings > Preferences. {{.PBXSN}}: PBX Serial Number. {{.PBXLANIP}}: PBX LAN IP Address. {{.PBXWANIP}}: PBX WAN IP Address. {{.EventTime}}: The time when the event happened.
Attachments	Click Add to add attachments. You can enter a file path or configure an expression, which must be an absolute path pointing to a file stored in the PBX's local storage. Example of file path: /ysdisk/syslog/pbxlog.log Example of expression: /ysdisk/email/'FX_CONCATENATE("yeastar","6602")'

Item	Description
	 Note: - Up to 10 attachments are supported. - For more information about the expression, see Variables and Functions in Yeastar Expression.
Do not send email if the attachment file does not exist	If selected, the system will NOT send email if any attachment file is missing.

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

Component connections

Email Sender component can be connected to **one** component, which can be any of the components listed below.

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.

Component	Description
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

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 call 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

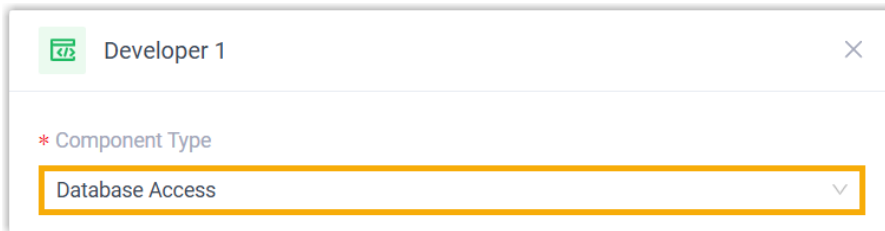
1. After adding **Developer** component to a call flow, click **Developer** component to proceed with the configuration.

The screenshot shows a call flow designer interface. On the left, a call flow canvas contains a component labeled 'Developer 1' with a green icon. A dashed yellow arrow points from this component to a configuration dialog box on the right. The dialog box is titled 'Developer 1' and contains the following fields:

- * Component Type**: A dropdown menu with 'Email Sender' selected.
- * Type of Email Server**: A dropdown menu with '[Follow System]' selected.
- * From**: A text field with 'PBX' entered.
- * Subject**: An empty text field.
- * Email Recipient**: A text field with a 'FX' button to its right.
- CC**: A text field with a 'FX' button to its right.
- BCC**: A text field with a 'FX' button to its right.

At the bottom right of the dialog is a blue button labeled '✓ Confirm'.

2. In the **Component Type** drop-down list, select **Database Access**.

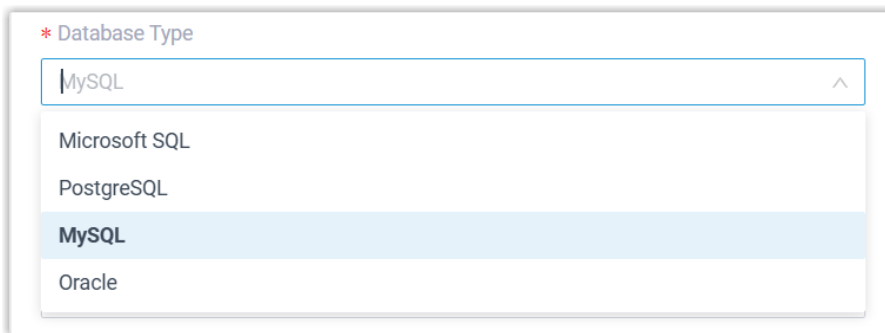


Developer 1

* Component Type

Database Access

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



* Database Type

MySQL

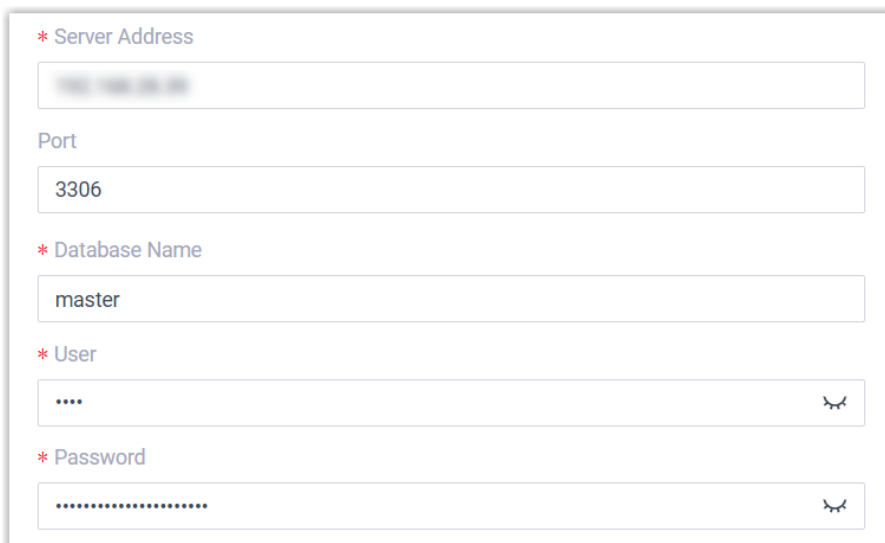
Microsoft SQL

PostgreSQL

MySQL

Oracle

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



* Server Address

192.168.0.10

Port

3306

* Database Name

master

* User

* Password

5. 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).

Item	Description
	• 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. Example: <code>SELECT email_addr FROM extension WHERE number='\$Session.ani';</code> In this example, the system queries the email address corresponding to the caller's number.  Note: For more information about the expression, see Variables and Functions in Yeastar Expression. • If multiple SQL statements are entered, only the first one will be executed.

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

Component variables

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



Note:

Since **Database Access** component can be added multiple times in a call flow, an index is appended to 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 can be connected to **one** component, which can be any of the components listed below.

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt.
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours.
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu.
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input.

Component	Description
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language.
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status

Component	Description
	• Set Agent Status • Email Sender • Database Access • HTTP Request

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

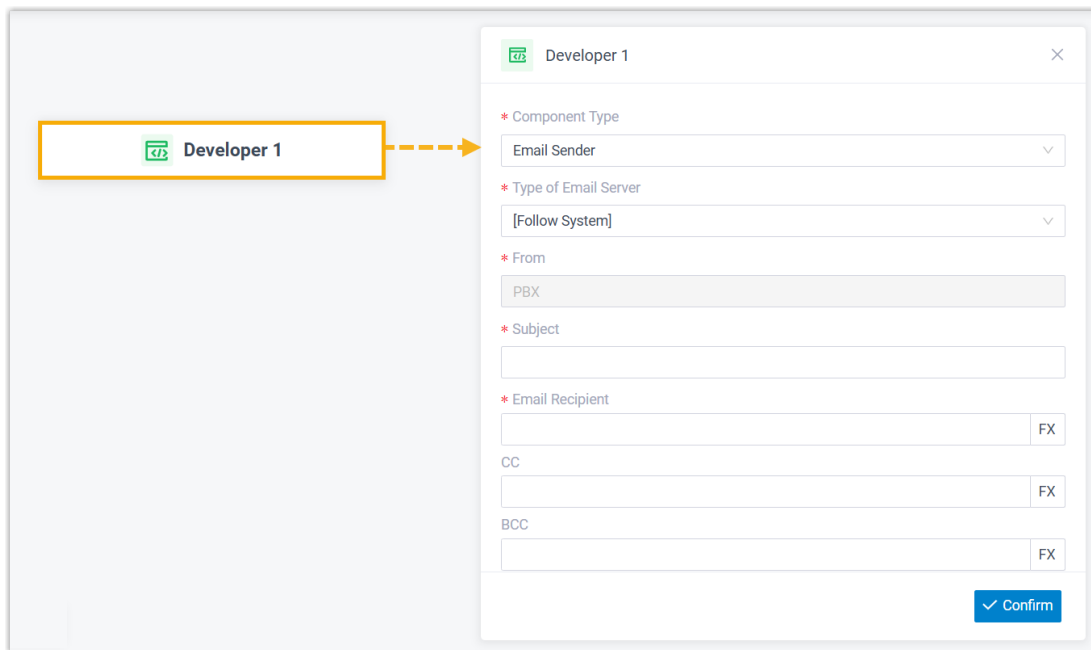
Supported content types

- 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

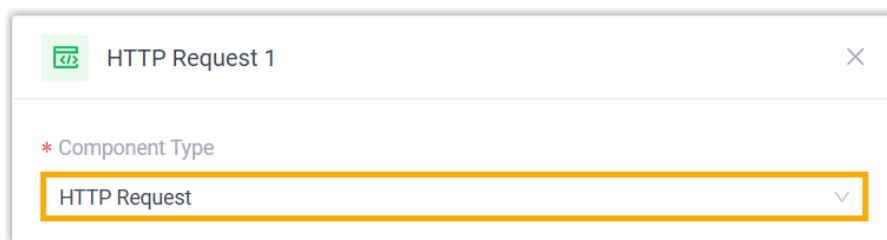
- image/gif

Component configuration

1. After adding **Developer** component to a call flow, click **Developer** component to proceed with the configuration.



2. In the **Component Type** drop-down list, select **HTTP Request**.



3. Configure the HTTP request.

Item	Description
URI	Enter the target URI to which the HTTP request will be sent. You can enter specific URI, or click FX to configure an expression. Expression example: <pre>FX_CONCATENATE("https://crm.example.com/check-caller?callernum=", \$Session.ani)</pre>

Item	Description
	In this example, the request is sent to a CRM system to verify the caller's number and identify the caller type.  Note: For more information about the expression, see Variables and Functions in Yeastar Expression.
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. You can enter specific query parameter, or click FX to configure an expression.  Note: • It is recommended not to exceed 30 parameters. • For more information about the expression, see Variables and Functions in Yeastar Expression. Expression example: • Key expression: <pre>FX_CONCATENATE("call_to_", \$Session.did)</pre>

Item	Description
	• Value expression: <pre>\$Session.ani</pre> In this example, the query parameter is dynamically generated to pass both the caller's number and the dialed number, allowing the CRM system to identify which trunk or service line the caller dialed into.
Content	Enter the request body to be sent to the server.
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. You can enter specific header, or click FX to configure an expression.  Note: • It's recommended not to exceed 30 headers. • For more information about the expression, see Variables and Functions in Yeastar Expression. Example: • Name: X-Caller-ID • Header: \$Session.ani In this example, the header name is fixed as X-Caller-ID, while the value is dynamically retrieved using the variable \$Session.ani, allowing the CRM system to obtain the caller's number from the request header and use it to look up customer records or match existing contact information.

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

Component variables

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



Note:

Since **HTTP Request** component can be added multiple times in a call flow, an index is appended to 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
\$HttpRequest{ <i>index</i> }.responseContent	String	The body content returned from HTTP response.	{"authenticated": true, "user": "10000"}
\$HttpRequest{ <i>index</i> }.responseStatusCode	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"

Component connections

HTTP Request component can be connected to **one** component, which can be any of the components listed below.

Component	Description
Prompt	Prompt component allows you to play audio prompt(s) or a text-to-speech message to callers. For more information, see Prompt .
Business Hours	Business Hours component allows you to route calls to different destinations based on the time of day. For more information, see Business Hours .
Menu	Menu component allows you to present callers with a set of menu options, and route calls based on the DTMF digit they press. For more information, see Menu .
User Input	User Input component allows you to collect DTMF digits from callers, typically used with Condition component to evaluate user input and route calls. For more information, see User Input .
Language	Language component allows you to change the system prompt language for subsequent components in a call flow. For more information, see Language .
Record	Record component allows you to start recording a call upon the caller being connected to another participant, and optionally configure prompts to inform call

Component	Description
	participants at the start and during the recording. Alternatively, you can use the component to disable call recording for calls that are supposed to be recorded. For more information, see Record.
Dial by Number	Dial by Number component allows callers to directly dial a number to reach the destination. For more information, see Dial by Number.
Dial by Name	Dial by Name component is one of the end components to terminate caller's connection to the current call flow. It allows callers to reach extension user by entering the first three letters of the user's name. For more information, see Dial by Name.
Transfer	Transfer component is one of the end components to terminate caller's connection to the current call flow. It allows you to transfer callers to a designated destination, and optionally configure prompts to inform them of the transfer. For more information, see Transfer.
Hang Up Call	Hang Up Call component is one of the end components to terminate caller's connection to the current call flow. When callers are routed to the component, the call will be disconnected. For more information, see Hang Up Call.
Condition	Condition component allows routing calls based on logical expressions. For more information, see Condition.
Loop	Loop component allows a group of components to be executed repeatedly, either for a specified number of times or until a condition is met. For more information, see Loop.
Developer	Developer component allows you to query and update data from PBX-native database or third-party database. For more information, see the following topics: • Get Extension Presence Status • Set Extension Presence Status • Get Queue Agent • Get Queue Info • Get Agent Status • Set Agent Status • Email Sender • Database Access • HTTP Request

Expression Basics

Variables and Functions in Yeastar Expression

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

Variable

Variables are used to dynamically retrieve and pass data throughout a call flow. There are two types of variables in Yeastar Call Flow Designer:

- **Session Variable:** Session variables store information about the current call session. These variables are globally accessible throughout a call 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 call flow.

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

Session Variables

Variable	Type	Description	Example Value
\$Session.ani	String	Caller's number.	"15812340987"
\$Session.callid	String	Unique ID for a call.	"1751336277.2"
\$Session.did	String	The DID number associated with the call flow that the caller reached.	"5503301"
\$Session.flowNum	String	The extension number associated with the call flow that the caller reached.	"6900"
\$Session.transferingNum	String	The number from which the call was forwarded.	"1000"

Component variables

**Note:**

Since the same component can be added multiple times in a call flow, an index is appended to 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.

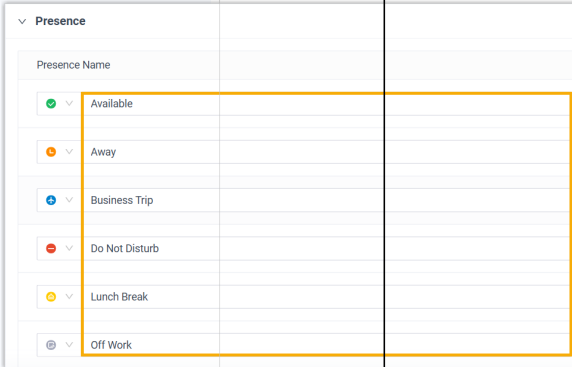
Variable	Type	Description	Example Value
Prompt component			
\$Prompt{ <i>index</i> }.ttsResult	String	The Text-to-Speech (TTS) result of the Prompt component. • PromptTTSResult.Success: The text is successfully converted into speech, and the system plays it to caller. • PromptTTSResult.Timeout: The text is NOT converted into speech within the timeout period, and the system routes caller directly to the next component without playing a prompt when the timeout is reached.	\$Prompt1.ttsResult(STRING)=PromptTTSResult.Success
Menu component			
\$Menu{ <i>index</i> }.result	String	The result of the Menu component. • MenuResult.Timeout: The caller didn't press any DTMF digit before	\$Menu1.result(STRING)=MenuResult.InvalidOption

Variable	Type	Description	Example Value
		the timeout, and was routed to the timeout destination. • <code>MenuResult.ValidOption</code>: The caller pressed a valid key, and was routed to the corresponding destination. • <code>MenuResult.InvalidOption</code>: The caller pressed an invalid key, and was routed to the invalid input destination.	
<code>\$Menu{index}.ttsResult</code>	String	The Text-to-Speech (TTS) result of the Menu component. • <code>MenuTTSResult.Success</code>: The text is successfully converted into speech, and the system plays it to caller. • <code>MenuTTSResult.Timeout</code>: The text is NOT converted into speech within the timeout period, and the system routes caller directly to the next component without playing a prompt when the timeout is reached.	<code>\$Menu1.ttsResult(STRING)=MenuTTSResult.Success</code>

Variable	Type	Description	Example Value
\$Menu{index}.userInput	String	The DTMF digit that the caller pressed, excluding the end key (# or *).	"2"
User Input component			
\$UserInput{index}.result	String	The result of User Input component. • UserInputResult.Timeout: The caller didn't press any DTMF digit before the timeout, and was routed to the timeout destination. • UserInputResult.ValidOption: The caller pressed a valid key, and was routed to the corresponding destination.	\$UserInput1.result(STRING)=UserInputResult.Timeout
\$UserInput{index}.ttsResult	String	The Text-to-Speech (TTS) result of the User Input component. • UserInputTTSResult.Success: The text is successfully converted into speech, and the system plays it to caller. • UserInputTTSResult.Timeout: The text is NOT converted into speech within the timeout period, and the system routes caller	\$UserInput1.ttsResult(STRING)=UserInputTTSResult.Success

Variable	Type	Description	Example Value
		directly to the next component without playing a prompt when the timeout is reached.	
\$UserInput{index}.userInput	String	The DTMF digit that the caller pressed, excluding the end key (# or *).	"2"
Record component			
\$Record{index}.ttsResult	String	The Text-to-Speech (TTS) result of the Record component. - RecordTTSResult.Success: The text is successfully converted into speech, and the system plays it to caller. - RecordTTSResult.Timeout: The text is NOT converted into speech within the timeout period, and the system routes caller directly to the next component without playing a prompt when the timeout is reached.	\$Record1.ttsResult(STRING)=RecordTTSResult.Timeout
Dial by Number component			
\$DialByNumber{index}.ttsResult	String	The Text-to-Speech (TTS) result of the Dial by Number component. DialByNumberTTSResult.Success:	\$DialByNumber1.ttsResult(STRING)=DialByNumberTTSResult.Success

Variable	Type	Description	Example Value
		The text is successfully converted into speech, and the system plays it to caller. • <code>DialByNumberTTSResult.Timeout</code>: The text is NOT converted into speech within the timeout period, and the system routes caller directly to the next component without playing a prompt when the timeout is reached.	
Transfer component			
<code>\$Transfer{index}.ttsResult</code>	String	The Text-to-Speech (TTS) result of the Transfer component. • <code>TransferTTSResult.Success</code>: The text is successfully converted into speech, and the system plays it to caller. • <code>TransferTTSResult.Timeout</code>: The text is NOT converted into speech within the timeout period, and the system routes caller directly to the next component without playing	<code>\$Transfer1.ttsResult(STRING)=TransferTTSResult.Success</code>

Variable	Type	Description	Example Value
		a prompt when the timeout is reached.	
Get Extension Presence Status component			
\$GetExtensionStatus{ <i>index</i> }.currentProfileName	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"
\$GetExtensionStatus{ <i>index</i> }.currentProfile	String	Name and additional information of the extension's current presence status.	"Available,work"
\$GetExtensionStatus{ <i>index</i> }.isInCall	Boolean	Whether the extension is currently on a call. • True: At least one endpoint	True

Variable	Type	Description	Example Value
		registered to the extension is on a call. • False: None of the endpoints registered to the extension are on a call.	
\$GetExtensionStatus{ <i>index</i> }.isInRinging	Boolean	Whether the extension is currently ringing. • True: At least one endpoint registered to the extension is ringing. • False: None of the endpoints registered to the extension are ringing.	False
Get Queue Agent component			
\$GetQueueAgent{ <i>index</i> }.agentList	Object	The list of agents with a specific status in a specific queue, returning each agent's extension number as well as their current status.	"1001,Log Out;1003,Log Out;1004,Log Out"
\$GetQueueAgent{ <i>index</i> }.agentNumberList	Object	The list of agents with a specific status in a specific queue, returning each agent's extension number.	"1001,1003,1004"
Get Queue Info component			
\$GetQueueInfo{ <i>index</i> }.availableAgents	Integer	The number of agents currently available to take calls.	5
\$GetQueueInfo{ <i>index</i> }.unavailableAgents	Integer	The number of agents currently unavailable to take calls.	1

Variable	Type	Description	Example Value
\$GetQueueInfo{ <i>index</i> }.callsActive	Integer	The number of calls currently being answered by queue agents.	2
\$GetQueueInfo{ <i>index</i> }.callsWaiting	Integer	The number of calls currently waiting in the queue.	10
Get Agent Status component			
\$GetAgentStatus{ <i>index</i> }.currentProfile	String	Name and additional information of the agent's current status.  Note: When using functions to evaluate an agent's status, ensure that the status constant matches one of the supported values listed below. • Status for Static Agents: ◦ "Pause{<i>pause_reason</i>}" ◦ "Unpause" • Status for Dynamic Agents: ◦ "Log In" ◦ "Log Out" ◦ "Pause{<i>pause_reason</i>}"	"Log Out"
\$GetAgentStatus{ <i>index</i> }.isInCall	Boolean	Whether the agent is currently on a call. • True: The agent is on a call.	True

Variable	Type	Description	Example Value
		• False: The agent is NOT on a call.	
\$GetAgentStatus{ <i>index</i> }.isInRing	Boolean	Whether the agent's extension is currently ringing. • True: The agent's extension is ringing. • False: The agent's extension is NOT ringing.	True
Database Access component			
\$DatabaseAccess{ <i>index</i> }.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, column): Get the value at the specified	"demo%40yeastar.com"

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

Function

Functions are used for condition evaluation and data processing, enabling flexible control of the call 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($UserInput1.userInput, "1"),
FX_EQUAL($Session.ani, "1001"))
```

In this example, the function returns `True` only when the user input is 1 and the caller's number is 1001.

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($Session.ani, "1001"), FX_EQUAL($GetAgentStatus1.isInCall, False))
```

In this example, the function returns `True` if either the caller's number is 1001 or the agent is not in a call.

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.isInCall)
```

In this example, the function returns `True` when the extension is NOT in a call.

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.ani, "1001")
```

In this example, the function returns `True` if the caller's number is a string and its value is exactly "1001".

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($GetAgentStatus1.isInRing, True)
```

In this example, the function returns **True** if the agent is **NOT** in the ringing state.

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($GetExtensionStatus1.currentProfile, "Available")
```

In this example, the function returns **True** if the extension's current presence status contains **Available**.

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($GetQueueInfo1.callsWaiting, 5)
```

In this example, the function returns **True** if the number of calls waiting in the queue exceeds 5.

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($GetQueueInfo1.callsWaiting, 5)
```

In this example, the function returns `True` if the number of calls waiting in the queue is 5 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($GetQueueInfo1.callsWaiting, 5)
```

In this example, the function returns `True` if the number of calls waiting in the queue is less than 5.

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($GetQueueInfo1.callsWaiting, 5)
```

In this example, the function returns `True` if the number of calls waiting in the queue is 5 or fewer.

TO_BOOLEAN

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

Syntax

```
FX_TO_BOOLEAN(value)
```

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

Example

```
FX_TO_BOOLEAN($GetExtensionStatus1.currentProfile)
```

In this example, the function converts the value returned by the **Get Extension Presence Status** component to a Boolean value. For example, if the value is "Available", the function returns `True`.

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("Caller",$Session.ani,"called",$Session.flowNum)
```

In this example, the function combines the caller's number and flow number into one string. E.g. "Caller1001called6900".

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, "status")
```

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

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 - "2025-07-01 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(1000000000, 2500000000)
```

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( -10000000000 )
```

In this example, the function returns 10000000000.

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.

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.

Firmware Dependency Reference

This topic outlines the key enhancements introduced to Call Flow Designer (CFD) feature and the corresponding minimum PBX firmware version required to use them.

Release Date	Description	Required PBX Firmware
October 13, 2025	• Component: For Prompt, Menu, User Input, Record, Dial by Number, and Transfer components, when Text-to-Speech (TTS) is configured as the prompt, these components support converting dynamic text into speech. You can enter static text along with placeholders that reference variables to dynamically compose the text. The system will convert it into speech and play to callers. • Variable: Added TTS-related variables. ◦ <code>\$Prompt{index}.ttsResult</code> ◦ <code>\$Menu{index}.ttsResult</code> ◦ <code>\$UserInput{index}.ttsResult</code> ◦ <code>\$Record{index}.ttsResult</code> ◦ <code>\$DialByNumber{index}.ttsResult</code> ◦ <code>\$Transfer{index}.ttsResult</code>	83.20.0.128 or later
August 5, 2025	• Transfer component: Added support for Custom destination. • Expression: Added JSON-related functions. ◦ <code>JSON_GET_STRING</code> ◦ <code>JSON_GET_INTEGER</code> ◦ <code>JSON_GET_OBJECT</code>	83.20.0.21 or later
July 7, 2025	Initial release.	83.19.0.110 or later