Xperience Routing

altitude

Contents

altitude

Introduction

This document describes the Xperience Routing concepts and provides a guide for the common tasks associated with automated scripts.

Xperience Routing

Xperience Routing is a web application that allows the user to design IVR and routing flows to use in campaigns.

Intended audience

This document is intended for everyone that needs to design graphical flows for campaigns using Xperience Routing.

What you should know

It is suggested, but not required, that the readers of this document be familiar with uSupervisor, Automated Agents, IVR and routing programming, Altitude Scripting Language expressions, and SQL.
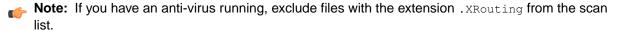
Document structure

Section	Description
Xperience Routing concepts (page 6)	Explains the concepts related to Xperience Routing.
How to use the Xperience Routing editor (page 13)	Contains the main tasks associated with the design of flows.
How to configure Xperience Routing (page 23)	Contains tasks to create and configure lists of numbers, and configure the folder for audio files.
Examples (page 28)	Shows examples of flows to solve common tasks in contact centers.
Details group reference (page 37)	Lists the available details groups with blocks.
Block reference (page 41)	Lists the available blocks and properties.
Expression reference (page 70)	Lists the predefined variables of Xperience Routing.

What do you need to run Xperience Routing

You need one of the following browsers:

- Internet Explorer 10 or above.
- Firefox 35 or above.
- Google Chrome.

☞ **Note:** If you have an anti-virus running, exclude files with the extension .XRouting from the scan list.

How to use this manual

If you want to learn the Xperience Routing concepts, read:

- Section *Xperience Routing concepts* (page 6) and all subsections.
- Section *How to use the Xperience Routing editor* (page 13) to familiarize yourself with the interface of Xperience Routing.

⟳ altitude

If you want to create a flow, read:

- Subsections of section *How to use the Xperience Routing editor* (page 13).
- Sections *Block reference* (page 41) and *Expression reference* (page 70) to get specific details of flow components.
- Section *Examples* (page 28) to view some example flows.
- Sections of the document `Altitude Scripting Language Reference` that explains expression building.

Xperience Routing concepts

Xperience Routing flows are IVR and routing scripts built with the Xperience Routing editor.
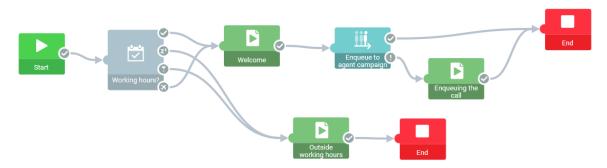
Figure 1: Xperience Routing flow.

Flows run as IVR or routing scripts in agent campaigns. You can use a flow to play audio files, ask for information, and enqueue interactions. For example, create a flow that enqueues interactions in the campaign that the customer selects.

Audio files are messages to play to customers. For example, "Welcome to the contact center. Your call will be answered as soon as possible.".

If you create a flow to transfer the call to a human agent campaign, ensure you enqueue the call before ending the script.

Components of a flow

Build flows using components.

Flows have the following components:

• Blocks
• Sequences.

A block is an action that the flow executes and has the following parts:

altitude
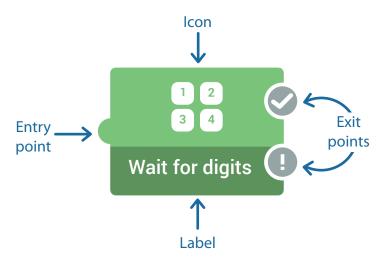
Figure 2: Parts of a block.

Entry point
> The entry point starts the execution of the block.

Label
> The label is the name of the block.

Exit points
> Exit points terminate the block execution and indicate if the block execution terminated successfully or with errors.

Icon
> The icon indicates the type of block.

A sequence is an arrow that connects two blocks.

Figure 3: Sequence.

Exit points

Blocks have exit points of the following types:

- On enqueued
- On transferred
- On true
- On digits
- Inside calendar
- On delivered

> Blocks that terminate execution with success exit through the exit point ✓ .

On false
> If the condition of the block **If** evaluates to false, or if the phone number is not on a list, the block exits through the exit point **On false**.

On error
> Blocks that terminate execution with errors exit through the exit point **On error**.

🕐 On timeout

Blocks that do not terminate execution before timeout exit through the exit point **On timeout**.

⓪ Not found

If the query to the database does not retrieve a record, database blocks exit through the exit point **Not found**.

① No error: Single row match

If the query to the database only retrieves one record, database blocks exit through the exit point **No error: Single row match** without error.

Ⓝ Error: Multiple rows match.

If the query to the database retrieves more than one record, database blocks exit through the exit point **Error: Multiple rows match** with error.

⓪①②③④⑤⑥⑦⑧⑨＊# Pressed

If the customer pressed one of the phone keys, the block **Menu** exits through the matching exit point.

💤 Non working hours

If the current time is outside the working hours, the block **Check calendar** exits through the exit point **Non working hours**.

🕐 Non working days

If the current date is outside the working days, the block **Check calendar** exits through the exit point **Non working days**.

✈ Exception days

If the current date is an exception day, the block **Check calendar** exits through the exit point **Exception days**.

Script blocks

Script blocks are blocks that start and end the script.

Figure 4: Blocks Start, End, and Assign.

Use a block **Start** to start a flow and one or more blocks **End** to end a flow. Use a block **Assign** to assign values to user or built-in variables.

Flows must have one block **Start** and one or more blocks **End**.

Related topics: *Assign* (page 41), *End* (page 46), *Start* (page 63), *Variables* (page 70).

⟳ altitude

Flow control blocks

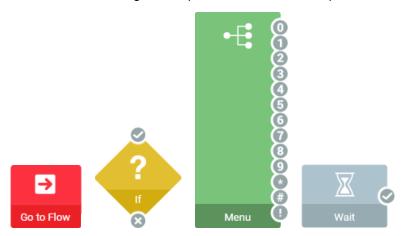Flow control blocks are blocks that change or suspend the flow of the script.

Figure 5: Blocks Go to flow, If, Menu, and Wait.

The block **Go to flow** continues the current flow in another flow, the block **If** executes one of two branches, the block **Menu** allows the user to select an option from a list of options, and the block **Wait** suspends the execution of the flow for a defined period of time.

Figure 6: Blocks Check agents available, Check calendar, Check emails, and Check phone numbers.

The block **Check agents available** checks if there are agents available in a campaign. The block **Check calendar** checks if the current date and time is inside a calendar, or outside the working days and times.

The blocks **Check emails** and **Check phone numbers** control the flow, checking if a phone number or email is in a list or in a directory.

Figure 7: Block Filter email.

The block **Filter email** defines a rule to filter the emails that the campaign receives. The rule can have multiple options to filter. For example, an email address, an email server, the subject, or if the email contains attachments.

Figure 8: Block Webservice.

The block **Webservice** configures a web service call. The return type of the call must be an object, where, through reflection, the return properties are fetched to assign the respective variables.

Database blocks

Database blocks are blocks that execute SQL queries to retrieve information from the database, or update information on the database.

Figure 9: Blocks SQL execute, SQL select, and Select contact profile.

The block **SQL execute** executes an SQL statement on the database. The block **SQL select** executes an SQL query to retrieve records from the database tables. The block **Select contact profile** retrieves records from the table of contact profiles.

Interaction blocks

Interaction blocks are blocks that handle interactions of type telephony and email.

Queue blocks

Queue blocks are blocks that enqueue, dequeue, and play messages while interactions wait in a queue.

Figure 10: Blocks Enqueue email and Enqueue phone.

The blocks **Enqueue email** and **Enqueue phone** enqueue interactions in a campaign. You can define the agent that must handle the interaction. The block **Enqueue phone** selects agents using skill profiles, priorities, and the contact profile unique id.

Figure 11: Block Transfer phone.

The block **Transfer phone** transfers phone calls to a campaign, a number, or an agent.

Figure 12: Block Wait while phone in queue.

altitude

The block **Wait while phone in queue** plays audio files to the customer while the interaction waits in queue.

Figure 13: Blocks Play expected waiting time and Play queue position.

The block **Play expected waiting time** plays the time that the customer is expected to wait before an agent can handle the phone call. The block **Play queue position** informs the customer about the position of the call in the campaign queue.

Related topics: *Enqueue email* (page 46), *Enqueue phone* (page 48), *Play expected waiting time* (page 53), *Play queue position* (page 55), *Transfer phone* (page 63), *Wait while phone in queue* (page 68).

Conversation blocks

Conversation blocks are blocks that can exchange messages with the customer.

Figure 14: Block Wait for digits.

The block **Wait for digits** can receive and store DTMF input from the customer.

Figure 15: Block Record voice message.

The block **Record voice message** can record a voice message from the customer, and save the recording in a file.

Figure 16: Block Send email.

The block **Send email** sends an email to the customer.

Related topics: *Record voice message* (page 56), *Send email* (page 59), *Wait for digits* (page 65).

Contact blocks

Contact blocks are blocks that schedule new interactions or create outcomes.

Figure 17: Block Callback.

The block **Callback** schedules a new call for the contact profile in a campaign.

Figure 18: Block Outcomes.

The block **Outcomes** adds an activity outcome in the current interaction. If you want to add more than one activity outcome per interaction, you can repeat the block.

Related topics: *Callback* (page 41), *Outcomes* (page 53).

Disconnect blocks

Disconnect blocks are blocks that handle ending phone calls.
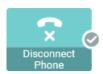
Figure 19: Block Disconnect phone.

👉 **Tip:** The block **Disconnect phone** always connects to a block **End** or to a block **If** for error handling. Connect the block **Disconnect phone** to a block **If** that evaluates variable `_Error`.

- If `_Error` is false, the block execution terminated with success and the IVR script must end.
- If `_Error` is true, the block execution terminated with an error and the IVR script can handle the error.

Figure 20: Block Customer disconnected phone.

The block **Customer disconnected phone** detects when a customer hangs up the phone call. Connect the block **Customer disconnected phone** to additional validations of the current flow. For example, update the customer information in the database, or create a callback for the customer.
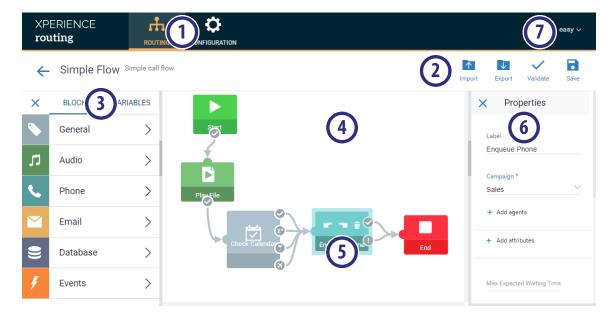
Related topics: *Customer disconnected phone* (page 45), *Disconnect phone* (page 46).

⟲ altitude

How to use the Xperience Routing editor

Use the Xperience Routing editor to create flows.

The Xperience Routing editor has the following parts:

1. Tabs
2. Toolbar
3. Block/Variables menu
4. Panel
5. Block actions
6. Property grid
7. User menu.

Tabs

Displays the following tabs:

Routing
Displays the Xperience Routing editor.

Configuration
Contains panels to allow you to configure the calendars, whitelists, blacklists, and audio resources to use in the flows.

Toolbar

The **Toolbar** of Xperience Routing editor displays the commands that you can use to import, export, validate, and save flows.

The Xperience Routing editor has the following commands in the toolbar:

Import
Loads a saved flow from the local disk.

Export
Saves the current flow to the local disk.

✓ Validate
Checks the current flow for errors.

🖫 Save
Saves the current flow.

Block/Variables menu

Displays the groups of blocks or variables.

Blocks
Selects the **Block menu**.

Variables
Selects the **Variable menu**.

👉 **Note:** You can click ✕ to hide the **Block menu** or the **Variables menu**.

The **Block menu** includes the following details groups:

General
General purpose blocks.

Audio
Blocks related to audio files.

Phone
Blocks related to phone call actions.

Email
Blocks related to email actions.

Database
Actions that can retrieve information from tables executing an SQL query, or execute an SQL query on the Altitude uCI Server database.

Events
Blocks related to telephony events.

The **Variables menu** includes the following details groups:

Predefined
List of built-in variables.

Campaign
List of campaign attributes, defined in the service.

Contact
List of activity attributes, defined in the service.

Transferred
List of activity attributes to use when transferring an interaction, defined in the service.

End user
List of user defined variables.

Panel

Displays the flow design.

◯ altitude

Block actions

Displays the actions that you can use on each block. Each block has the following actions:

  Rotate left
    Rotates the start and exit points of the selected block to the left.

  Rotate right
    Rotates the start and exit points of the selected block to the right.

  Delete
    Deletes the selected block.

Property grid

Displays the properties of a block that is selected in the **Panel.** To edit or select a value of a property, click the value of the property.

👉 **Note:** You can click ✕ to hide the property grid.

👉 **Note:** Use the browser zoom to zoom the IVR script.

User menu

Displays the following operations:

Operation	Description
About	Displays the current version of Xperience Routing.
Log out	Logs out the user.
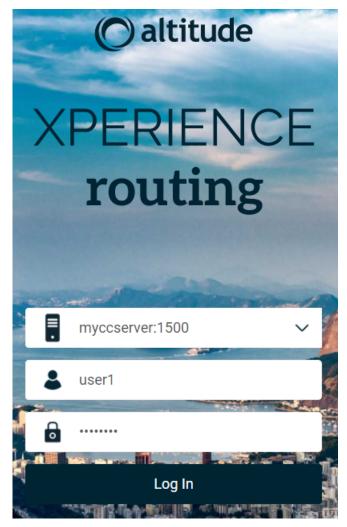
How to create a flow

Create flows to handle and route interactions. Publish the flow to use the generated script in a campaign.

1. Open Xperience Routing.
   In your Web browser, type the URL of Xperience Routing.

2. Login to the Altitude uCI Server.

    a) Define the login information.

        Domain
            Enter the hostname and port of the Altitude uCI Server. For example, `myccserver:1500`.

        Username
            Enter the name of the user. The user name must exist in the Altitude uCI Server.

        Password
            Enter the password of the user.

    b) Click **Log In**.

    c) Select a service.

3. Click ⊕ to add a new flow to the service.

4. Define the flow properties.

Name
> Enter the name of the flow.

Description
> Enter the description of the flow.
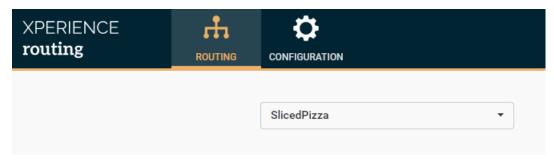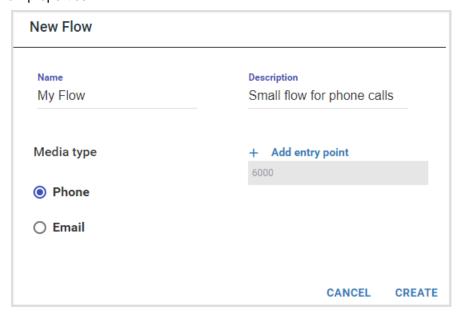
Media type
> Select the media type for the flow.

Add entry point
> Add entry points to the flow.
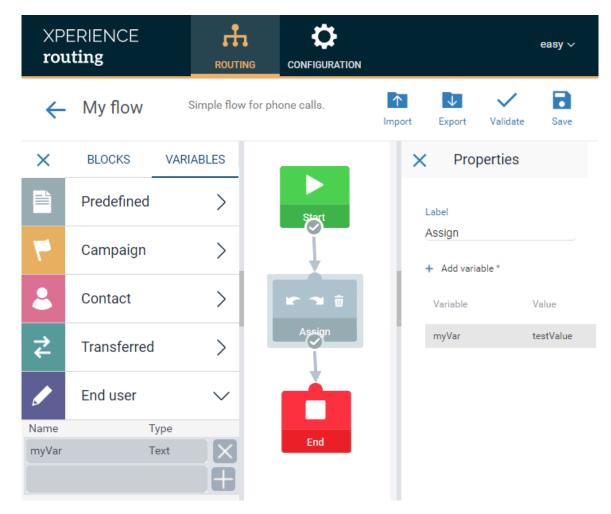
> 👉 **Note:** You must configure the entry points in uSupervisor and in the telephony switch or email server.

5. Click **Create**.

6. In the list of flows, click ✏ to edit the flow.

7. Add blocks.

a) In the **Block/Variables menu**, select a details group. For example, **Audio**.

b) Drag blocks to the **Panel**.

8. Delete blocks.

a) Select blocks.

b) In the **Block actions** of a block, click 🗑.

9. Connect blocks.

- To connect and sequence blocks, drag a ➝ from an end point of a block to the destination block.
- To delete a sequence, click the exit point of the origin block.

10. Configure blocks.

To configure a block, select the block and configure the properties of the block in the **Property grid**.

11. Use variables.

- To view predefined variables, in the **Block/Variables menu**, select the **Variables**, and the details group **Predefined**.
- To change read and write variables, configure a block **Assign**.

⚠️ **Important:** Remember to frequently validate the current flow. In the **Toolbar**, click ✓ to validate the current flow.

12. Save the flow.

In the **Toolbar**, click 💾.

13. Publish the flows to the Altitude uCI Server.

a) In the **Toolbar**, click ← to return to the list of flows.

b) Click ⬆ to publish the flows to the Altitude uCI Server.

When you publish the flows to the Altitude uCI Server, Xperience Routing creates:

- A new IVR script with a new timestamp.
- A new routing script with a new timestamp.

👉 **Note:** Xperience Routing automatically validates the IVR script before publishing.

🌜 altitude

How to add or delete variables to a flow

Define variables to share session information between the IVR script and the contact center.

1. Create a variable.

    a) In the **Block/Variables menu**, click **Variables.**
    b) Select the details group **End user**.

    c) Click ➕ to create a variable.

2. Define the properties of the variable.

    Name
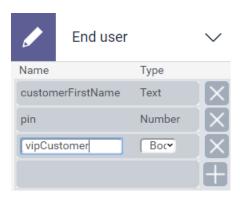   > Enter the name of the variable. The name of the variable must be a valid name in Altitude Scripting Language. A variable name:
    >
    > * Can contain letters, digits, and underscores. For example, `customer_email1`.
    > * Cannot start with a digit.
    > * Cannot match a built-in variable.

    Type
    > Select the type of the variable.

👉 **Note:** To delete a variable, click ✕.

How to configure blocks of type Enqueue

Use blocks of type Enqueue to enqueue interactions.

1. Add a block of type enqueue.

    a) In the **Block/Variables menu**, select the details group **Phone**.

    b) Drag the block **Enqueue Phone** to the **Panel**.

2. Configure the block.

    a) In the **Panel**, select the enqueue block.

    b) Define the enqueue attributes.

        Label

            Enter the block label. Must be unique in the flow.

        Campaign

            Enter the campaign to enqueue the phone call.

        Add agents

            Select **Add agents** to enqueue the phone call to a specific list of agents.

            Click ➕ to add an agent.

            👉 **Note:** To delete an agent, click 🗑.

        Add attributes

            Select **Add attributes** to add attributes to send to the campaign where the call is enqueued.

            Click ➕ to add an attribute.

            👉 **Note:** To delete an attribute, click 🗑.

        Max expected waiting time

            Enter the timeout, in seconds, for the call to wait in the interaction queue.

    c) Define the contact attributes.

        Directory

            Select the directory of the contact profile.

        Contact profile
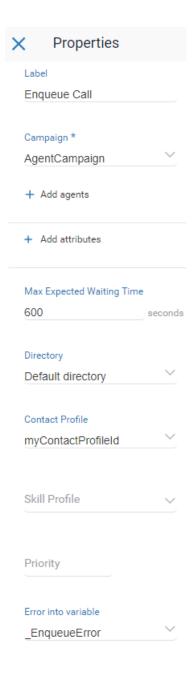
            Select the contact profile related to the customer.

        Skill Profile

            Select the skill profile required to handle the interaction.

        Priority

            Enter the priority of the call.

Properties panel:

Label: Enqueue Call

Campaign *: AgentCampaign

➕ Add agents

➕ Add attributes

Max Expected Waiting Time: 600 seconds

Directory: Default directory

Contact Profile: myContactProfileId

Skill Profile

Priority

Error into variable: _EnqueueError

altitude

How to configure block Wait for digits

Use block **Wait for digits** to collect input from customers.

1. Add a block of type **Wait for digits**.

   a) In the **Block/Variables menu**, select the details group **Audio**.
   b) Drag the block **Wait for digits** to the **Panel**.

2. Configure the block **Wait for digits.**

   In the **Panel**, select the block **Wait for digits**.

   Label
   : Enter the block label. Must be unique in the flow.

   Into variable
   : Select the variable to store the digits that the customer inputs.

3. Add audio files.

   Click **+** to add an audio message.

   👉 **Note:** To delete an announcement, click 🗑️ .

   Interruptible
   : Select **Interruptible** to allow the customer to interrupt the playing.

4. Configure the input size and timeouts.

   Input trailer digit
   : Enter the character that ends the input of digits. For example, #.

   Input size
   : Enter the number of digits that the customer can input.

   All digits timeout
   : Enter the maximum duration, in seconds, of the period to input the digits.

   First digit timeout
   : Enter the maximum duration, in seconds, of the period to input the first digit.

   Inter digit timeout
   : Enter the maximum duration, in seconds, of the period to input two consecutive digits.

   Number of tries
   : Enter the maximum number of tries to input the digits.
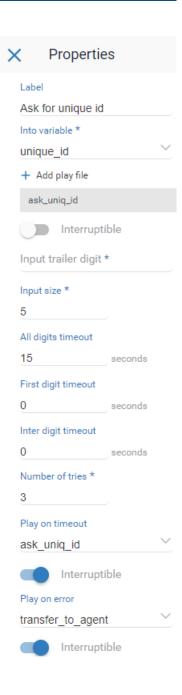
   Play on timeout
   : Select the name of the file with the announcement to play if timeout occurs.

   Play on error
   : Select the name of the file with the announcement to play if the customer reaches the maximum number of tries.

   Interruptible
   : Select **Interruptible** to allow the customer to interrupt the playing.

Properties

Label
Ask for unique id

Into variable *
unique_id

+ Add play file

ask_uniq_id

⬤ Interruptible

Input trailer digit *

Input size *
5

All digits timeout
15   seconds

First digit timeout
0   seconds

Inter digit timeout
0   seconds

Number of tries *
3

Play on timeout
ask_uniq_id

⬤ Interruptible

Play on error
transfer_to_agent

⬤ Interruptible

How to configure block Select contact profile

Use block **Select contact profile** to retrieve contact profile information.

1. Add a block of type **Select contact profile**.

   a) In the **Block/variables menu**, select the details group **Database**.

   b) Drag the block **Select contact profile** to the **Panel**.

2. Configure the block **Select contact profile.**

   In the **Panel**, select the block **Select contact profile**.

   Label
   > Enter the block label. Must be unique in the flow.

   Directory
   > Select the directory of the contact profile.

3. Define the filter.

   Field
   > Select the name of the contact profile attribute to filter the retrieved values.

   Operator
   > Select the operator.

   Text
   > Enter the value to match the retrieved value.

4. Add columns.

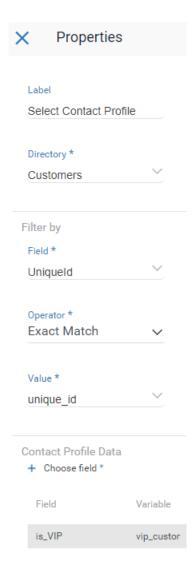   Click ✚ to add an column.

   👉 **Note:** To delete a column, click 🗑.

   Field
   > Select the name of the column to retrieve.

   Variable
   > Select the variable to store the retrieved value.

✕	Properties

Label

Select Contact Profile

Directory *

Customers ⌄

Filter by

Field *

UniqueId ⌄

Operator *

Exact Match ⌄

Value *

unique_id ⌄

Contact Profile Data
✚ Choose field *

Field	Variable
is_VIP	vip_custor

How to configure Xperience Routing

Create calendars, blacklists, whitelists, and configure the audio resources folder.

How to create a calendar

Create calendars to configure the working and non-working days of your campaign. Use blocks **Check calendar** to implement the logic regarding working and non-working days and times.

1. In Xperience Routing, select the tab **Configuration**.
2. Select **Calendars**.
3. Click ⊕ to add a calendar.
4. Define the calendar properties.

   Name
   > Enter the name of the calendar.

   Description
   > Enter the description of the calendar.

5. Add working times to the calendar.

a) In **Calendar**, click and drag a slot of the calendar from the beginning of the time slot to the end of the time slot.

For example, for Mondays until Fridays, from 9:00 until 5PM, select the following calendar slots:

⟳ altitude

6. Configure exceptions.

  a) Click **Exceptions**.
  b) Select a month.
  c) Click a day to add to the exceptions.

	CALENDAR							EXCEPTIONS	

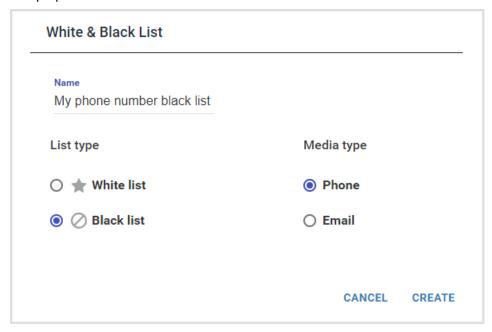|		December 2018						**Date**	**Non working days**
SUN	MON	TUE	WED	THU	FRI	SAT		01/01	New holiday
						1		25/04	New holiday
2	3	4	5	6	7	8		01/05	New holiday
9	10	11	12	13	14	15		10/06	New holiday
16	17	18	19	20	21	22		13/06	New holiday
23	24	25	26	27	28	29		15/08	New holiday
30	31							05/10	New holiday
								01/11	New holiday
								01/12	New holiday

7. Save the calendar.
Click **Save**.

How to create a list

Create white and black lists to allow or deny access to the flow options. Use blocks **Check phone numbers** and **Check emails** to check if a phone number or an email address is in a white or black list.

1. In Xperience Routing, select the tab **Configuration**.

2. Select **White & Black Lists**.

3. Click ⊕ to add a new list.

4. Define the list properties.

Name
> Enter the name of the list.

List type
> Select the type of list to create:

> > White list
> > > White lists allow all phone numbers or email addresses that are in the list.

> > Black list
> > > Black lists deny all phone numbers or email addresses that are in the list.

Media type
> Select the media type for the list.

5. Click **Create**.

6. Add items to the list.

The following example shows how to add phone numbers to a list. Follow the similar procedure for lists that of media type **Email**.

a) Add a phone number manually.

  1. Click **Add number to list**.
  2. Enter the phone number.

b) Import a file with phone numbers.

  1. Click **Import numbers**.
  2. Browse to the file that contains the phone numbers to import.

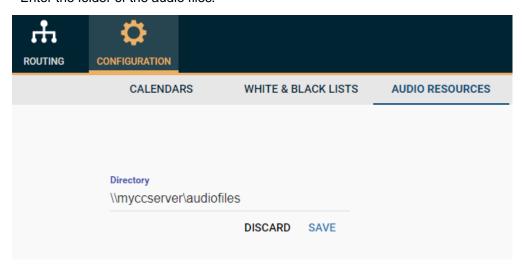 👉 **Note:** The file must contain one phone number per line.

altitude

How to configure the audio resources folder

Configure the audio resources folder that Xperience Routing uses for the audio files to play. The audio resources folder must be the same as defined in the telephony switch that plays the audio files. For example, if you are using Communication Server to play IVR and routing messages, ensure the Xperience Routing audio resources folder is the same as the **Sound files directory** in Communication Server.

1. In Xperience Routing, select the tab **Configuration**.

2. Select **Audio Resources**.

3. Define the audio resource folder.

    Directory
        Enter the folder of the audio files.

4. Save the information.

    Click **Save**.

Examples

The examples show flows to solve common problems of contact centers.

Play message

The example shows how to build a flow that plays a message.

Figure 21: Basic flow.

1. In the Xperience Routing editor, create a new flow.
2. Add blocks:

   a. Add blocks **Start** and **End**.
   b. Add block **Play File**.

3. Connect the blocks.

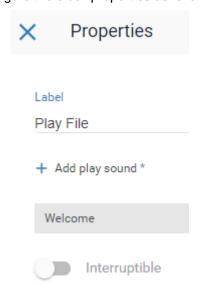Select the block **Play File** and configure the block properties as follows:

Figure 22: Properties of the block Play File.

👉 **Note:** The `Welcome` message must exist in the telephony switch.

1. Publish the flow in the Altitude uCI Server.
2. Associate the generated IVR script with a campaign.

How does the example work

1. The IVR campaign receives a call.
2. The IVR script runs and plays the `Welcome` message.

⬤ altitude

👉 **Note:** The IVR script does not hang up the phone call.

Enqueue call

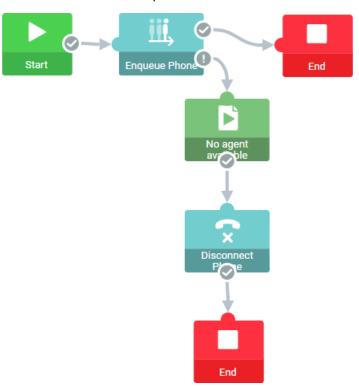The example shows how to build a flow that enqueues a call.

Figure 23: Enqueue flow.

1. In the Xperience Routing editor, create a new flow.
2. Add blocks and sequence to design the flow.

Select the block **Enqueue phone** and configure the block properties as follows:

Label
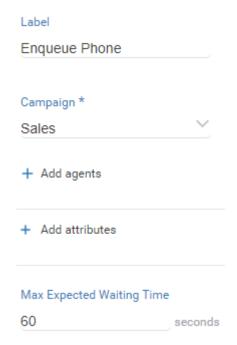
Enqueue Phone

Campaign *

Sales

+ Add agents

+ Add attributes

Max Expected Waiting Time

60   seconds

Figure 24: Properties of the block Enqueue phone.

1. Publish the flow in the Altitude uCI Server.
2. Associate the generated IVR script with a campaign.

How does the example work

1. The IVR campaign receives a call.
2. The IVR script runs and enqueues the call in the campaign `Sales`.
3. If the expected waiting time does not exceed 60 seconds, the call remains in the queue.
4. Otherwise, or if an error occurs, the IVR script plays a message informing the customer that there are no agents available, hangs up the call, and ends.

◯ altitude

Interactive queuing

The example shows how to build a flow that plays messages to the customer while the phone call waits in the campaign queue.
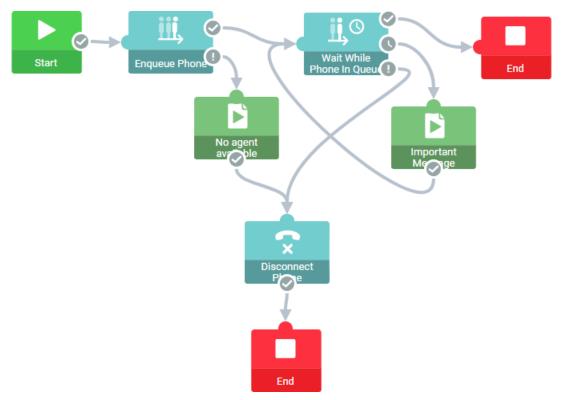

Figure 25: Interactive queuing.

1. In the Xperience Routing editor, create a new flow.
2. Add blocks:

   a. Start
   b. End
   c. Enqueue phone
   d. Wait while phone in queue
   e. Disconnect phone
   f. Play file.

3. Connect the blocks.

Block configuration

Select the block **Wait while phone in queue** and configure the block properties as follows:
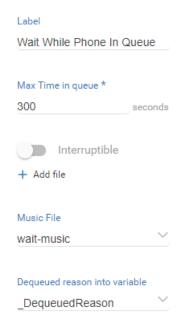
Figure 26: Properties of the block Wait while phone in queue.

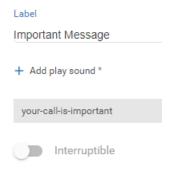Select the block **Play File** and configure the block properties as follows:

Figure 27: Properties of the block Important message.

1. Publish the flow in the Altitude uCI Server.
2. Associate the generated IVR script with a campaign.

How does the example work

1. The IVR campaign receives a call.
2. The IVR script runs and enqueues the call in the campaign `Sales`.
3. If the expected waiting time does not exceed 60 seconds, the call remains in the queue.
4. Otherwise, or if an error occurs, the IVR script plays a message informing the customer that there are no agents available, hangs up the call, and ends.
5. While the call is in the queue:

   a. Plays waiting music.
   b. If the phone call waits more than 300s:

      a. The flow exits the block `Wait while phone in queue` through the exit point **On timeout** and proceeds to the block `Important message`.
      b. The block `Important message` plays a message and exits.
      c. The flow executes the block `Wait while phone in queue`.

⟳ altitude

c. If the phone call is dequeued:

   a. The flow exits the block `Wait while phone in queue` through the exit point **On delivered**.
   b. The flow ends.

Customer identification

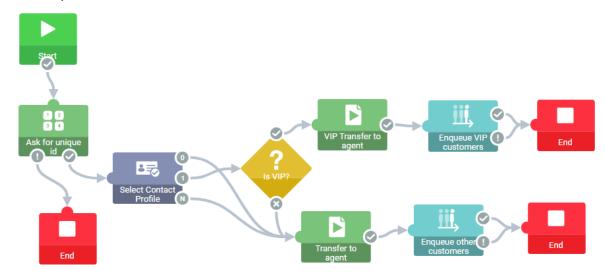The example shows how to build a flow that detects a VIP customer.

Figure 28: Customer identification.

1. In the Xperience Routing editor, create a new flow.
2. Add blocks:

   a. Start
   b. End
   c. Wait for digits
   d. Select contact profile
   e. If
   f. Enqueue phone
   g. Play File.

3. Connect the blocks.

Variable configuration

Add the following variables:

Figure 29: User defined variables.

Block configuration

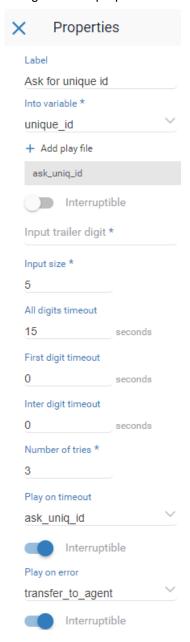Select the block **Wait for digits** and configure block properties as follows:

Figure 30: Properties of the block Ask for unique id.

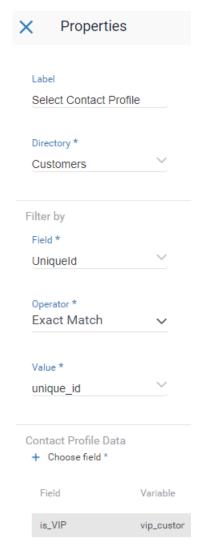Select the block **Select contact profile** and configure block properties as follows:

Figure 31: Properties of the block Select contact profile.

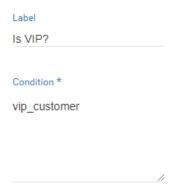Select the block **If** and configure block properties as follows:

Figure 32: Properties of the block Is VIP?

Select the block **Transfer to agent** and configure block properties as follows:
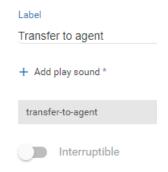
Label

Transfer to agent

+ Add play sound *

transfer-to-agent

⬤ Interruptible

Figure 33: Properties of the block Transfer to agent.

Select the block **Transfer to agent** and configure block properties as follows:

Label

VIP Transfer to agent

+ Add play sound *

vip-transfer-to-agent

⬤ Interruptible

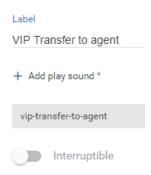**Figure 34: Properties of the block Transfer to agent.**

1. Publish the flow in the Altitude uCI Server.
2. Associate the generated IVR script with a campaign.

How does the example work

1. The IVR campaign receives a phone call.
2. The block `Ask for unique id`:

   a. Plays a message asking for the customer number.
   b. Waits for the customer input.
   c. Stores the customer input in variable `unique_id`.

3. The block `Select contact profile`:

   a. Uses `unique_id` to search for the contact profile of the customer.
   b. Retrieves the value of column **is_VIP** and stores the value in variable `vip_customer`.

4. The block `Is VIP?` evaluates the value of variable `vip_customer`:

   • If the variable `vip_customer` is false, the flow plays a message for non-VIP customers, enqueues the call to an agent campaign with priority 0, and ends.
   • If the variable `vip_customer` is true, the flow plays a message for VIP customers, enqueues the call to an agent campaign with priority 10, and ends.

⊙ altitude

Details group reference

Audio blocks

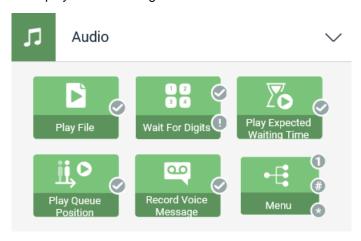Audio blocks are blocks that play audio messages.

Figure 35: Details group Audio.

Related topics: *Menu* (page 52), *Play expected waiting time* (page 53), *Play file* (page 54), *Play queue position* (page 55), *Record voice message* (page 56), *Wait for digits* (page 65).

Database blocks

Database blocks are blocks that execute select statements in the Altitude uCI Server database.

Figure 36: Details group Database blocks.

Related topics: *Select contact profile* (page 57), *SQL execute* (page 61), *SQL select* (page 61).

Email blocks

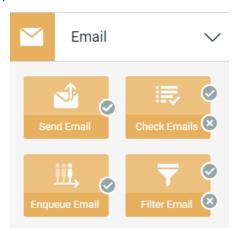Email blocks are blocks that enqueue, filter, send, and check if an email address is in a list.

Figure 37: Details group Email blocks.

Related topics: *Check emails* (page 44), *Enqueue email* (page 46), *Filter email* (page 50), *Send email* (page 59).

Event blocks

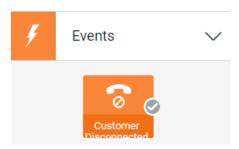Event blocks are blocks that handle events in an interaction.

Figure 38: Details group Event blocks.

Related topics: *Customer disconnected phone* (page 45).

altitude

General blocks

General blocks are blocks that perform general purpose actions.

Figure 39: Details group General.

Related topics: *Assign* (page 41), *Callback* (page 41), *Check agents available* (page 43), *Check calendar* (page 43), *End* (page 46), *Go to flow* (page 51), *If* (page 51), *Outcomes* (page 53), *Start* (page 63), *Wait* (page 64), *Webservice* (page 69).

Phone blocks

Phone blocks are blocks that handle phone calls.
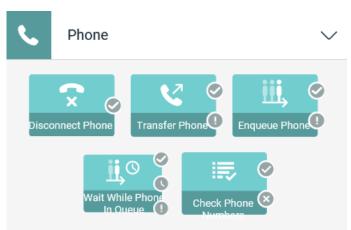

Figure 40: Details group Phone blocks.

Related topics: *Check phone numbers* (page 45), *Disconnect phone* (page 46), *Enqueue phone* (page 48), *Transfer phone* (page 63), *Wait while phone in queue* (page 68).

Block reference

Reference of the blocks that you can use in Xperience Routing.

Assign

Sets user defined variables and built-in variables that are read-write with user defined values.

Attributes

Label
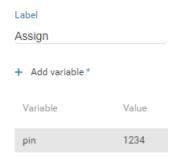> Name of the block with meaning in the flow. Must be unique in the flow.

Variable
> Name of the variable. For example, `count`. You must select a variable from the combo box.

Value
> If the value:

- Starts with the equal sign (`=`), the following text must be a valid expression of Altitude Scripting Language. For example, `=1 + count` or `="2015/18/11"`.
- Starts with any other character, the following text must be a literal of the type of variable. For example, if the variable `count` is of type `number`, the literal can be `24`.
- Is empty, resets the variable.

Label

Assign

+ Add variable *

| Variable | Value |
| pin | 1234 |

Exit points

Icon	Name	Exit condition
✓	Normal	None.

Callback

Schedules a new call for the customer in a campaign.

The block **Callback** schedules a new call for the contact profile in a campaign. If you do not specify a contact profile, the Altitude uCI Server creates a new contact profile. The phone number specified for the callback is the reschedule phone of the contact.

Attributes

Label
> Name of the block with meaning in the flow. Must be unique in the flow.

Campaign
> Campaign where to create the scheduled contact.

Phone
> The reschedule phone number.

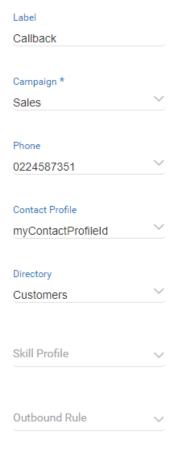Contact profile
> Identifier of the contact profile.

Directory
> Directory of the contact profile.

Skill profile
> Required skill profile to handle the call in the **Campaign**.

Outbound Rule
> Outbound rule to perform the callback.

Label

Callback

Campaign *

Sales

Phone

0224587351

Contact Profile

myContactProfileId

Directory

Customers

Skill Profile

Outbound Rule

Exit points

Icon	Name	Exit condition
✓	Normal	None.

⟲ altitude

Check agents available

Checks if there are agents available for handling an interaction in a campaign.

Attributes

Label
Name of the block with meaning in the flow. Must be unique in the flow.

Campaign
Name of the campaign to check if there are agents available for handling the interaction.

Label

Check Agents Available

Campaign *

Sales

Exit points

Icon	Name	Exit condition
✔	On true	When the campaign has agents available.
✖	On false	When the campaign does not have agents available.

Check calendar

Checks if the current date and time is a working day, and if the current time is inside the working hours.

Attributes

Label
Name of the block with meaning in the flow. Must be unique in the flow.

Calendar
Name of the calendar to check the date and time.

Label

Check Calendar

Calendar *

My Calendar

Exit points

Icon	Name	Exit condition
✓	Inside calendar	If the call date and time is inside the specified calendar.
Z²	Non working hours	If the call time is outside working hours.
▽	Non working days	If the call date is outside working days.
✕	Exception days	If the call date is an exception day.

Check emails

Checks whether the current email address belongs in a directory, a white list, or a black list.

The block **Check emails** evaluates whether an email address belongs to a white list, a black list, or is an email address that belongs to a contact profile in a specified directory. If the block evaluates to:

- True, the flow executes the block that connects to the exit point ✓.

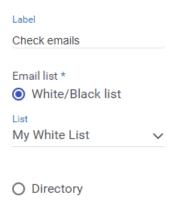- False, the flow executes the block connected to the exit point ✕.

Attributes

Label
> Name of the block with meaning in the flow. Must be unique in the flow.

Email list
> Select **White/Black list** and select a list to check the email address in the selected list.

Directory
> Select a directory to check the email address in the contact profiles of the directory.

Label

Check emails

Email list *

⦿ White/Black list

List

My White List   ⌄

○ Directory

Exit points

Icon	Name	Exit condition
✓	On true	When the block finds the email address.
✕	On false	When the does not find the email address.

Check phone numbers

Checks whether the current phone number belongs in a directory, a white list, or a black list.

The block **Check phone numbers** evaluates whether the phone number returned by the variable `_Ani` belongs to a white list, a black list, or is a phone number that belongs to a contact profile in a specified directory. If the block evaluates to:

- True, the flow executes the block that connects to the exit point ✓.

- False, the flow executes the block connected to the exit point ✗.

Attributes

Label
> Name of the block with meaning in the flow. Must be unique in the flow.

Phone number list
> Select **White/Black list** and select a list to check the phone number in the selected list.

Directory
> Select a directory to check the phone number in the contact profiles of the directory.

Label

Check phone numbers

Phone number list *
◉ White/Black list

List
My White List   ⌄

○ Directory

Exit points

Icon	Name	Exit condition
✓	On true	When the block finds the phone number.
✗	On false	When the does not find the phone number.

Customer disconnected phone

Detects when a customer hangs up a phone call.

The block **Customer disconnected phone** detects when a customer hangs up a phone call. You can specify the following steps after a customer hangs up the phone call.

Attributes

None.

Exit points

Icon	Name	Exit condition
✓	Normal	None.

Disconnect phone

Hangs up a phone call.

The block **Disconnect phone** disconnects the phone call.

Attributes

Label

Name of the block with meaning in the flow. Must be unique in the flow.

Exit points

Icon	Name	Exit condition
✓	Normal	None.

End

Script end.

The block **End** is the block that ends the flow.

Attributes

None.

Exit points

Icon	Name	Exit condition
None	-	-

Enqueue email

Enqueues an email in a campaign.

The block **Enqueue email** enqueues an email in a campaign. To enqueue the interaction, you can select an agent to handle the interaction, and specify customer data. For example, the contact profile unique id and the directory.

If you do not select an agent, the flow enqueues the interaction in the campaign, and any agent assigned to the campaign can handle the interaction.

Otherwise, the flow enqueues the interaction and the campaign delivers the interaction to the defined agent.

altitude

Attributes

Label
> Name of the block with meaning in the flow. Must be unique in the flow.

Campaign
> Campaign where to enqueue the email interaction.

Agent
> Agent to handle the email interaction.

Directory
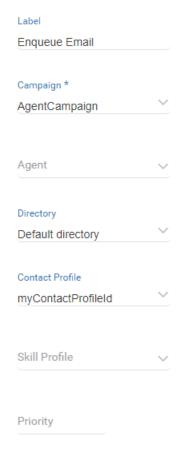> Directory of the contact profile to send with the enqueue operation.

Contact profile
> Identifier of the contact profile to send with the enqueue operation.

Skill profile
> Required skill profile to handle the email interaction in the **Campaign**.

Priority
> Priority of the contact in the **Campaign**.

Label

Enqueue Email

Campaign *

AgentCampaign

Agent

Directory

Default directory

Contact Profile

myContactProfileId

Skill Profile

Priority

Exit points

Icon	Name	Exit condition
✔	Normal	None.

Enqueue phone

Enqueues a phone call in a campaign.

The block **Enqueue phone** enqueues a phone call in a campaign. To enqueue the interaction, you can select a list of agents to handle the call, and specify customer data. For example, the contact profile unique id and the directory.

If you select a list of agents and do not select at least an agent, the flow enqueues the call in the campaign, and any agent assigned to the campaign can handle the call.

Otherwise, the flow enqueues the call and the campaign delivers the call to one of the defined agents.

☞ **Note:** Blocks **Enqueue phone** must always connect to a block **End** through the **On enqueued** (✓) exit point.

Attributes

Label
> Name of the block with meaning in the flow. Must be unique in the flow.

Campaign
> Campaign where to enqueue the call.

Add agents
> List of agents to handle the call.

Add attributes
> Additional attributes to send to the campaign script.

> ⚠ **Important:** To use the attributes in the agent campaign, you must declare the attributes in the script of the agent campaign, in the record `transferred`.

Max Expected Waiting Time
> Maximum time, in seconds, that the interaction can wait in the queue.

Directory
> Directory of the contact profile to send with the enqueue operation.

Contact profile
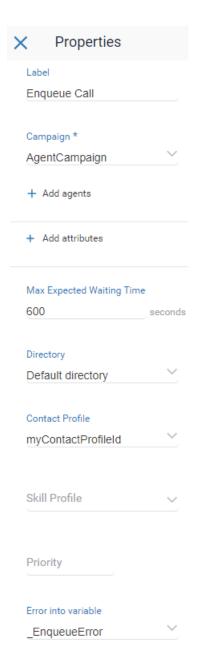> Identifier of the contact profile to send with the enqueue operation.

Skill profile
> Required skill profile to handle the call in the **Campaign**.

Priority
> Priority of the contact in the **Campaign**.

Error into variable
> Reason why the enqueue operation failed.

◐ altitude

Properties

Label

Enqueue Call

Campaign *

AgentCampaign

+ Add agents

+ Add attributes

Max Expected Waiting Time

600   seconds

Directory

Default directory

Contact Profile

myContactProfileId

Skill Profile

Priority

Error into variable

_EnqueueError

Exit points

Icon	Name	Exit condition
✓	On enqueued	Valid enqueue. Must always connect to a block **End**.
!	On error	Invalid enqueue. `_EnqueueError` has the failure reason.

Filter email

Defines a rule to filter the email contents.

The block **Filter email** defines a rule to filter the contents of the email that the campaign receives. The block evaluates to true if all options evaluate to true. If the block evaluates to:

- True, the flow executes the block that connects to the exit point ✓.

- False, the flow executes the block connected to the exit point ✕.

Attributes

Label
  Name of the block with meaning in the flow. Must be unique in the flow.

From
  If selected, filter the emails from a specific email address or an email server.

Subject
  If selected, filter the emails that contain a specific subject.

Body
  If selected, filter the emails that contain a specific text in the email body.

Has Attachments
  If selected, filters the emails that contain attachments.

altitude

Exit points

Icon	Name	Exit condition
✓	On true	When the block finds the email address.
✕	On false	When the does not find the email address.

Go to flow

Continues the current flow in a different flow.

The block **Go to flow** allows you to jump to a different flow to continue the execution of the interaction. For example, you can have a flow to check the schedule of a campaign, and if the campaign is currently working, you can go to a different flow that enqueues interactions in the campaign.

Attributes

Label
　　Name of the block with meaning in the flow. Must be unique in the flow.

Flow
　　Select a flow from the list.

Exit points

Icon	Name	Exit condition
None	-	-

If

Controls the flow according to a condition.

The block **If** evaluates a boolean expression. If the boolean expression evaluates to:

- True, the flow executes the block that connects to the exit point ✓.

- False, the flow executes the block connected to the exit point ✕.

Attributes
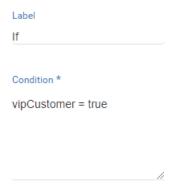
Label
　　Name of the block with meaning in the flow. Must be unique in the flow.

Condition
　　Valid boolean expression in Altitude Scripting Language.

Label

If

Condition *

vipCustomer = true

Exit points

Icon	Name	Exit condition
✓	On true	When the condition evaluates to true.
✗	On false	When the condition evaluates to false.

Menu

Displays a menu for selection of an option from a list of options.

The block **Menu** allow the customer to select an option from a list of available options. Each option must connect to another block.

Attributes

Label
> Name of the block with meaning in the flow. Must be unique in the flow.

Add play file
> List of sound files to play. Click **Add play file** to enter a new file name.

Digits
> Select the available options for the menu. You must select at least 2 options.

Label

Menu

+ Add play file

Digits

☑ 1   ☑ 2   ☑ 3
☑ 4   ☑ 5   ☑ 6
☑ 7   ☑ 8   ☑ 9
☑ *   ☑ 0   ☑ #

At least 2 digits must be selected

○ altitude

Exit points

Icon	Name	Exit condition
(0)(1)(2)(3) (4)(5)(6)(7) (8)(9)(*)(#)	Pressed	Valid user input.
(!)	On error	Invalid user input.

Outcomes

Adds an activity outcome to the current interaction.

The block **Outcomes** adds an activity outcome to the current interaction.

You can specify the activity outcome attributes and the values.

Attributes
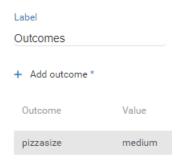
Label
> Name of the block with meaning in the flow. Must be unique in the flow.

Add outcome
> Add a new activity outcome attribute. Click **Add outcome** to select an activity outcome attribute and the value.

Label

Outcomes

+ Add outcome *

Outcome	Value
pizzasize	medium

Exit points

Icon	Name	Exit condition
✓	Normal	None.

Play expected waiting time

Plays the expected waiting time of the interaction.

The block **Play expected waiting time** plays a message with the value of the variable $\_QueueExpectedWaitingTime$. The announcement informs the customer of how much time the customer is expected to wait before an agent can handle the call.

The block uses the variable $\_QueueExpectedWaitingTime$ to play the message.

**Attributes**
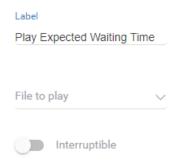
Label
Name of the block with meaning in the flow. Must be unique in the flow.

File to play
The message to play before playing the waiting time.

Interruptible
If selected, the caller can interrupt before the message ends.

Label

Play Expected Waiting Time

File to play ⌄

⬤▭ Interruptible

Exit points

Icon	Name	Exit condition
✔	Normal	None.

Play file

Plays a combination of sound files.

The block **Play file** plays a combination of sound files.

For each file, the flow can allow the customer to interrupt the message.

Attributes
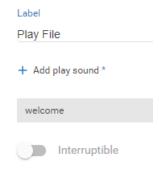
Label
Name of the block with meaning in the flow. Must be unique in the flow.

Add play sound
List of sound files to play. Click **Add play sound** to enter a new file name.

Interruptible
If selected, the caller can interrupt before the message ends.

◯ altitude

Label

Play File

＋ Add play sound *

welcome

Interruptible

Exit points

Icon	Name	Exit condition
✅	Normal	None.

Play queue position

Plays the position of the call in the campaign queue.

The block **Play queue position** plays a message with the value of the variable `_QueuePosition`. The message informs the caller of how many interactions are ahead of the call in the campaign queue.
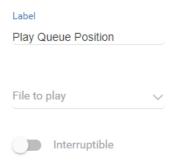
Attributes

Label
　　Name of the block with meaning in the flow. Must be unique in the flow.

File to play
　　The message to play before playing the queue position.

Label

Play Queue Position

File to play　　　　　　　　∨

Interruptible

Exit points

Icon	Name	Exit condition
✅	Normal	None.

Record voice message

Allows the customer to record a voice message.

The block **Record voice message** creates a file with the recorded message of the customer. You can configure the duration, size, and the folder of the recorded message.

Attributes

Label
Name of the block with meaning in the flow. Must be unique in the flow.

Working folder
Path to the folder where the file with the recording will be saved.

Recorded file path into variable
Variable that contains the file path of the recorded file.

Max file size
Maximum file size to record, in KB.

Input timeout
Maximum time to record, in seconds.

Keys to stop recording
List of keys that the customer can press to stop the recording.
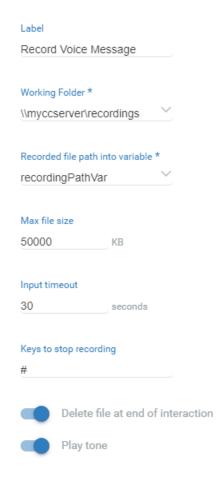
Delete file at end of interaction
If selected, the file is deleted at the end of the interaction. For example, you can select **Delete file at end of interaction** if you are sending the recording by email and do not need the recording for other uses.

Play tone
If selected, plays a tone before starting the recording.

Label

Record Voice Message

Working Folder *

\\myccserver\recordings

Recorded file path into variable *

recordingPathVar

Max file size

50000   KB

Input timeout

30   seconds

Keys to stop recording

#

Delete file at end of interaction

Play tone

Exit points

Icon	Name	Exit condition
✓	Normal	None.

Select contact profile

Executes a query to the contact profile table.

The block **Select contact profile** performs an SQL select to the contact profile table of **Directory** with the filter defined in attributes **Field**, **Operator**, and **Value**. If the query retrieves:

- 0 records, the block **Select contact profile** exits through exit point Not found

  (0).
- 1 record, the block **Select contact profile**:

  - Exits through exit point No error: single row match (1).
  - Stores the attributes of the record in the variables defined in the attribute **Variable.**

- More than 1 record, the block **Select contact profile**:

  - Exits through exit point Error: multiple rows match (N).
  - Sets the variable _Error to True.

- Logs the error.

Attributes

Label
Name of the block with meaning in the flow. Must be unique in the flow.

Directory
Directory name that contains the contact profile.

Filter by

Field
Name of the contact profile attribute to filter the retrieved values.

☞ **Note:** The name of the attribute must be defined as searchable in the directory.

Operator
Select one of the following operators:

- Exact match
- Any word
- All words.

Value
Text or variable to compare the retrieved value.
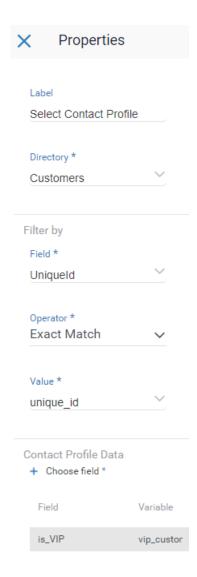
Field
Name of the contact profile attribute to retrieve.

Variable
Name of variable that receives the retrieved values.

◯ altitude

Properties

Label

Select Contact Profile

Directory *

Customers

Filter by

Field *

UniqueId

Operator *

Exact Match

Value *

unique_id

Contact Profile Data
+ Choose field *

Field	Variable
is_VIP	vip_custor

Exit points

Icon	Name	Exit condition
0	Not found	When the query does not find records.
1	No error: single row match	When the query finds one record.
N	Error: multiple rows match	When the query finds multiple records.

Send email

Sends an email to the customer.

The block **Send email** sends a new email to the customer, or replies to a customer email.

Send Email

Attributes

Label

Name of the block with meaning in the flow. Must be unique in the flow.

Send
Reply
Reply all
Forward

The action that the block performs.

To

For the actions **Send** and **Forward**, the email address where to send the email.

Subject

th subject of the email.

Body

The email body.

Add Attachment

Click + to add an attachment to the email. You have to specify the path to the attachment file.

Label

Send Email

- ● Send
- ○ Reply
- ○ Reply all
- ○ Forward

To *

cccustomer@customer.com ⌄

Subject

Feedback ⌄

Body

Thank you for your feedback.
We will reply to your message
as soon as possible.

+ Add attachment

Exit points

Icon	Name	Exit condition
✔	Normal	None.

altitude

Sequence

Sequences are arrows that connect blocks and define the sequence of execution of the blocks.

SQL execute

Executes an SQL command.

The block **SQL execute** executes a SQL statement.

Attributes

Label
> Name of the block with meaning in the flow. Must be unique in the flow.

Query
> The SQL statement to execute. You can use `INSERT`, `UPDATE`, and `DELETE` statements. Use the macro `VAR(varName)` to use variables in the query. For example, `UPDATE customers SET city = 'VAR(targetCityVar)' WHERE id = VAR(clientIdVar)`.

Exit points

Icon	Name	Exit condition
✓	Normal	None.

SQL select

Executes an SQL query.

The block **SQL select** performs a query to a table. If the query retrieves:

- `0` records, the block SQL select exits through exit point **Not found** (⓪).
- `1` record, the block SQL select:

  - Exits through exit point **Single row match** (①).

- Stores the attributes of the record in the variables defined in attribute **Into variable.** The name of the variables defined in attribute **Into variable** must match the name of attributes in the retrieved record.

- More than 1 record, block **SQL select**:

  - Exits through exit point **Multiple rows match** (**N**).
  - Sets variable `_Error` to `True`.
  - Logs the error.

Attributes

Label
> Name of the block with meaning in the flow. Must be unique in the flow.

Query
> The SQL query to execute. Must start with the `select` statement. Use macro `VAR(varName)` to use variables in the query. For example, `select firstname, lastname from clients where age > VAR(minAgeVar) AND city = 'VAR(targetCityVar)'`.

Variable
> Name of variable that receives the retrieved values.

Label

SQL Select

Query *

SELECT id, name, price,
stock FROM
business_db..product

+ Add variable *

productId

productName

productPrice

productStock

Exit points

Icon	Name	Exit condition
0	Not found	When the query does not find records.
1	No error: single row match	When the query finds one record.
N	Error: multiple rows match	When the query finds multiple records.

Start

Start of the flow.

The block **Start** is the block that starts the flow.

Attributes

None.

Exit points

Icon	Name	Exit condition
✓	Normal	None.

Transfer phone

Transfers a phone call to a number, an agent, or a campaign.

The block **Transfer phone** allows diverting the phone call to an agent or extension, on a routing script and transferring the call, on an IVR script.

The divert operation is called when the phone state is routing, otherwise, the block calls the transfer operation. The block performs a blind transfer if **Single step** is set to true. If the interaction is enqueued when the block runs, the interaction is dequeued automatically in order for the divert/transfer operation to succeed.

Attributes

Label
> Name of the block with meaning in the flow. Must be unique in the flow.

Transfer to
> Select whether to transfer the call to a number or to an agent.

Number
> Number where to transfer the call.

Agent
> Agent to transfer the call.

Campaign
> Campaign where to transfer the call.

Add attribute
> Additional attributes to send to the campaign script.
>
> ⚠️ **Important:** To use the attributes in the agent campaign, you must declare the attributes in the script of the agent campaign, in the record `transferred`.
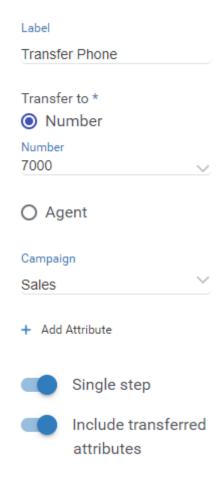
Single step
> If selected, the flow executes the transfer in a single step. For example, a blind transfer.

Include transferred attributes
> If selected, includes the transferred attributes to the destination script.

Exit points

Icon	Name	Exit condition
✓	On transferred	Valid transfer. Must always connect to a block **End**.
❗	On error	Invalid enqueue. `_EnqueueError` has the failure reason.

Wait

Pauses the flow execution.

Block **Wait** pauses the flow execution during **Seconds to pause** seconds and plays a voice message in loop. Playing stops after **Seconds to pause**.

Attributes

Label
Name of the block with meaning in the flow. Must be unique in the flow.
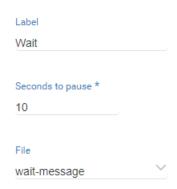
Seconds to pause
Duration of the waiting period in seconds.

altitude

File

> Name of the file with the message to play during the waiting period. If the duration of the announcement is smaller than the duration of the waiting period, the **Wait** block plays the announcement in loop.

> 👉 **Tip:** The duration of the message should be smaller than the duration of the waiting period.

Label

Wait

Seconds to pause *

10

File

wait-message ⌄

Exit points

Icon	Name	Exit condition
✅	Normal	None.

Wait for digits

Plays announcements and waits for user input.

The block **Wait for digits** plays a list of messages. For example, "Please, insert your customer number." After playing the initial messages, waits for the caller to input digits until one of the following occurs:

- The number of input digits equals the input size. For example, 6.
- The caller inputs the trailer digit. For example, #.
- The caller hangs up.
- A timeout occurs.

If the input digits are not valid and timeout occurs, the block **Wait for digits** repeats the following actions until the number of the caller inputs equals the **Number of tries**:

1. Plays the message **Play on timeout**.
2. Plays the initial messages.
3. Waits for the caller to input new digits.

If the input is valid, the block **Wait for digits** stores the input digits in the **Into variable**.

Playing and digit collecting stops with the trailer.

The block works as follows:

1. If the user input is not valid and a timeout occurs, the block plays the message **Play on timeout**.
2. Plays the list of messages and collects digits **Number of tries** times.
3. Collects digits.

4. If the user reaches **Number of tries**, the block plays **Play on error** and exits execution through exit point **On error**.

5. If the user input is valid, the block exits execution through exit point **On digits**. The block stores user input in **Into variable**.

Attributes

Label
Name of the block with meaning in the flow. Must be unique in the flow.

Into variable
Name of variable that receives the retrieved values.

Add play file
List of sound files to play. Click **Add play file** to enter a new file name.

Interruptible
If selected, the caller can interrupt before the message ends.

Input trailer digit
Character that ends the input of digits. For example, #.

Input size
Number of digits the customer can input.

Timeouts

All digits
Maximum duration, in seconds, of the period to input the digits.

First digit
Maximum duration, in seconds, of the period to input the first digit.

Inter digit
Maximum duration, in seconds, of the period to input two successively digits.

Number of tries
Maximum number of tries to input the digits.
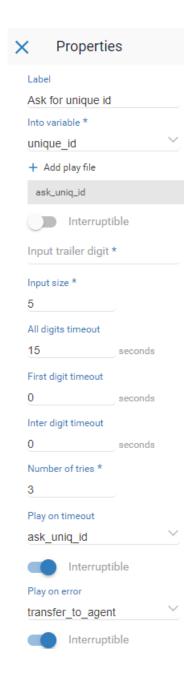
Play on timeout
Name of the file with the message to play if timeout occurs.

Play on error
Name of the file with the message to play if the customer reaches the maximum number of tries.

⟲ altitude

Exit points

Icon	Name	Exit condition
✔	On digits	Valid user input.
❗	On error	Invalid user input.
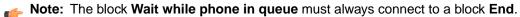
Wait while phone in queue

Waits until the routing center delivers the telephony interaction to a resource group.

Block **Wait while phone in queue** waits **Max time in queue** for the Altitude uCI Server to deliver the call to an agent.

If the campaign:

- Dequeues the interaction, the block stores the dequeue reason in the variable `_DequeueReason`.
- Delivers the interaction, the flow must end.

👉 **Note:** The block **Wait while phone in queue** must always connect to a block **End**.

While waiting, the block **Wait while phone in queue** plays a sequence of messages. After the messages, plays the **Music file** in loop.

Attributes

Label
Name of the block with meaning in the flow. Must be unique in the flow.

Max Time in Queue
Maximum time the call can wait in the queue. If a timeout occurs, the Altitude uCI Server dequeues the phone call.

Interruptible
If selected, the caller can interrupt before the message ends.

Add file
Click ＋ to add files to play.

Music file
Name of the file with the music to play after the messages.

Dequeued reason into variable
Reason why the call was dequeued.

Label

Wait While Phone In Queue

Max Time in queue *

300   seconds

◯ Interruptible

＋ Add file

Music File

wait-music   ⌄

Dequeued reason into variable

_DequeuedReason   ⌄

◯ altitude

Exit points

Icon	Name	Exit condition
✅	On delivered	The Altitude uCI Server delivers the call. Must always connect to a block **End**.
🕐	On timeout	The interaction waited in queue until a timeout occurs.
❗	On dequeue	The Altitude uCI Server dequeues the interaction and saves the dequeue reason in `_DequeueReason`.

Webservice

Configures a web service call.

The block **Webservice** configures a web service call. The return type of the call must be an object, where, through reflection, the return properties are fetched to assign the respective variables.

Attributes

Label
    Name of the block with meaning in the flow. Must be unique in the flow.

URL
    URL to the WSDL of the web service.

Label

Web Service

URL *   ⊕
http://myccserver/myWS.wsdl

ⓘ Url of the WSDL of the service

Exit points

Icon	Name	Exit condition
✅	Normal	None.

Expression reference

Variables

Reference of the flow built-in variables.

Flow variables can have one of the following types:

- Predefined
- User-defined.

Predefined variables

The predefined variables are the following:

_Ani

> If the variable _IsPhone is true, the variable _Ani contains the number that the ANI service provides.

_DequeuedReason

> The contact center sets the variable _DequeuedReason to the reason of the last dequeue. Possible values are the following:
>
> - overflow
> - abandon
> - reject
> - agentUnavailable
> - noAgentWithProfile
> - calculatedEwtIsAboveMaxEwt.

_Dnis

> The variable _Dnis contains the original DNIS of the interaction.

_EnqueueError

> If the block **Enqueue phone** cannot enqueue an interaction, the block **Enqueue phone** sets the variable _EnqueueError to the error that occurred during the action enqueue. Possible values are the following:
>
> - UnknownContactProfile
> - UnknownDirectory
> - UnknownCampaign
> - InvalidOperation.

_Error

> If an error occurs during the execution of the the block, the flow sets the variable _Error to true. Blocks initialize the variable _Error to false except for the block **If**.

_IsEnqueued

> The flow sets the variable _IsEnqueued to true if the interaction is enqueued in a campaign queue. Otherwise, the flow sets the variable to false.

_IsIm

> The variable _IsIm contains true if the interaction is an instant messaging message.

altitude

`_IsPhone`

The variable `_IsPhone` contains `true` if the interaction is a phone call.

`_QueueExpectedWaitingTime`

The variable `_QueueExpectedWaitingTime` contains the expected waiting time of the interaction in the queue after the execution of a successful enqueue.

`_QueuePosition`

The variable `_QueuePosition` contains the position of the interaction in the queue after the execution of a successful enqueue.

Copyright

altitude

Index

altitude

altitude