

# CX Cloud, Digital and AI for Salesforce Service Cloud

Genesys Innovations

v. 1.10

# Contents

CX Cloud, Digital and AI for Salesforce Service Cloud Installation	1
Prerequisites	1
Install the package	2
Post Install steps	3
Create the Service Channels	3
Assign Field-Level Security	4
Create the Service Presence Statuses	7
Assign the Presence Statuses to User Profiles	8
Create the Presence User Configuration	9
Create a Routing Configuration	10
Create the flows	11
Create a new Record Page for the Experience	22
Create a new Record Page for the Email Experience	23
Configure a Lightning App for the Experience	24
Einstein Next Best Action Integration in Salesforce	30
Genesys Cloud for Real Time Voice Transcription	35
Tips for your Genesys Cloud Architect Flow	37
Multi-Org Setup	37
Single Salesforce Org with multiple Genesys Cloud Orgs	37



# CX Cloud, Digital and AI for Salesforce Service Cloud Installation

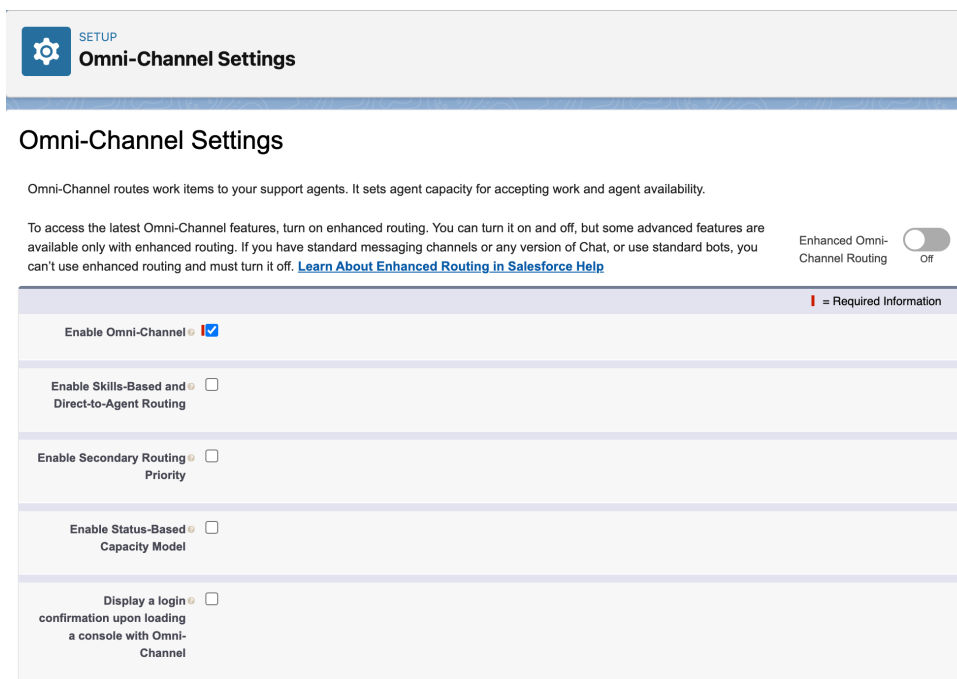
## Prerequisites

Before installing the package, you need to make sure that you have the following features enabled in your Salesforce org:

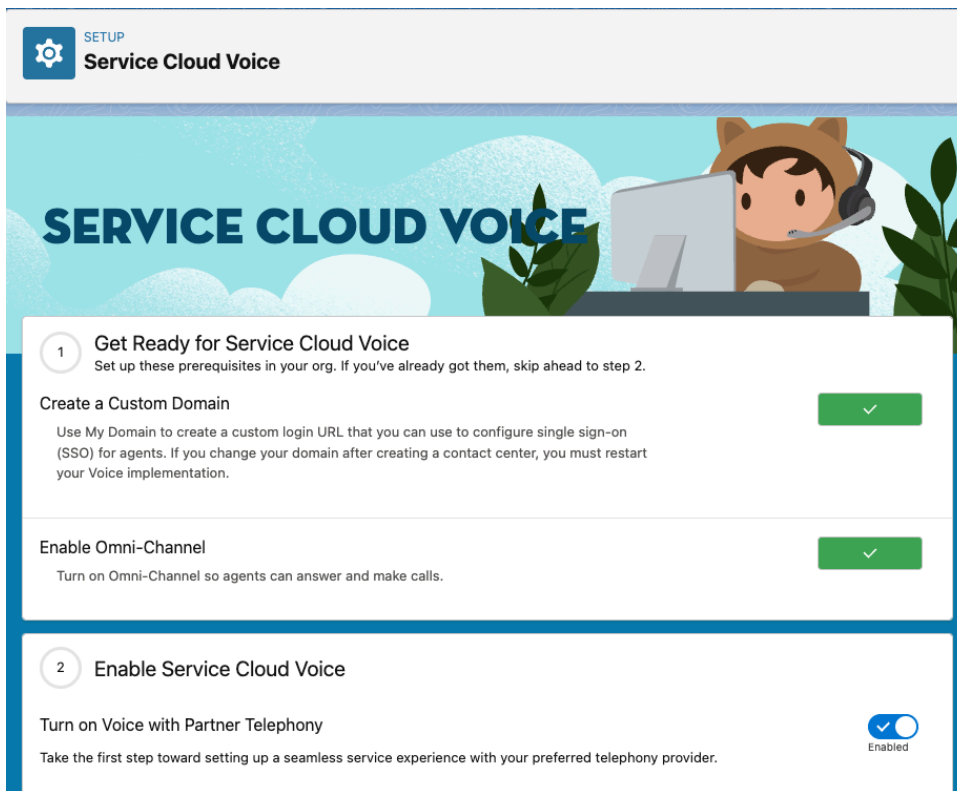
- Omni-Channel
- Service Cloud Voice

The Genesys AppFoundry [CX Cloud from Genesys and Salesforce](#) must also be installed. This package is also available on the [Salesforce AppExchange](#).

On the setup screen of your Salesforce org, search for **Omni-Channel Settings**, and make sure that the **Enable Omni-Channel** checkbox is checked:

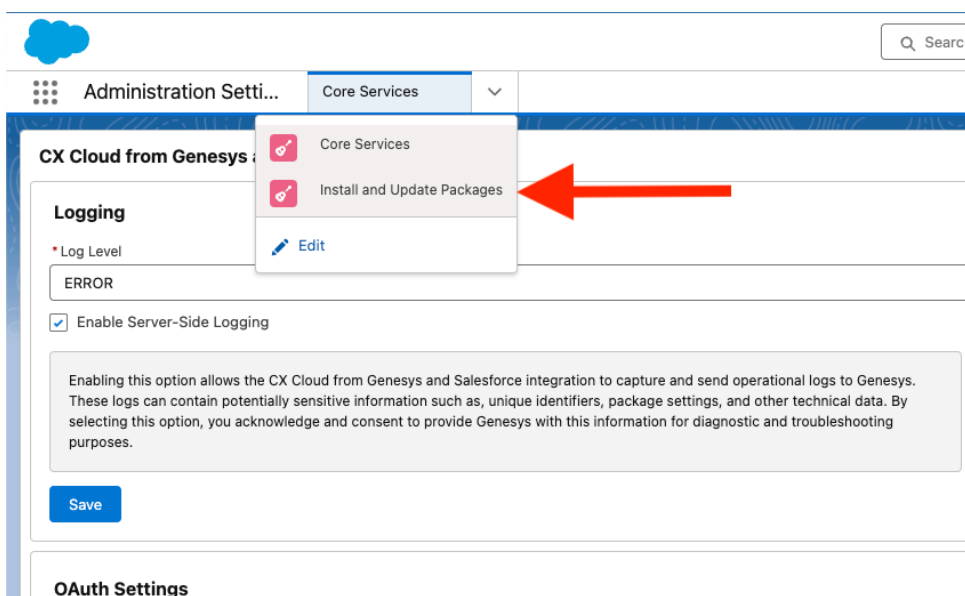


Search for **Partner Telephony Setup**, and make sure that the **Turn on Voice with Partner Telephony** checkbox is checked:

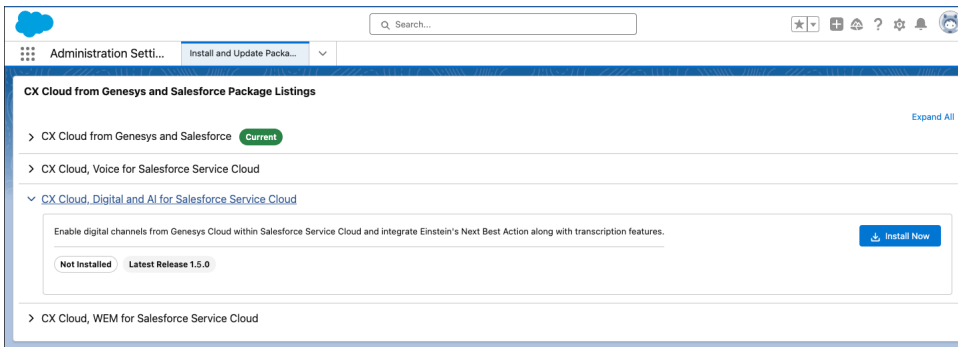


## Install the package

Open the lightning application **Administration Settings** (you can find it in the App Launcher) that was installed during this [process](#) and open the tab **Install and Update Packages** :



In the list of available packages, open the **CX Cloud, Digital and AI for Salesforce Service Cloud** package, and click on the **Install now** button:



## Post Install steps

Note: This is a work in progress. The following steps are not yet automated. The goal is to automate them in the future.

### Create the Service Channels

Create two Service Channels:

- One for the Genesys Cloud Chat Messaging, called `Genesys Cloud Messaging` and related to `genesysps__Experience__c`,
- One for the Genesys Cloud Email Messaging, called `Genesys Cloud Email` and related to `genesysps__EmailExperience__c`.

On the Setup screen, search for `Service Channels`, click on `New`, and fill in the form:

**Service Channels**

Route work from a Salesforce object, such as cases, chats, leads, or even custom objects, to support agents.

Save Cancel

**Basic Information**

Service Channel Name: `Genesys Cloud Messagi`

Developer Name: `Genesys_Cloud_Messa`

Salesforce Object: `Experience`

Custom Console Footer Component:

Minimize the Omni-Channel widget when work is accepted:

Automatically accept work requests:

**Audio Settings**

Override agents' audio settings:

Save Cancel

### Service Channels

Route work from a Salesforce object, such as cases, chats, leads, or even custom objects, to support agents.

Save Cancel

---

**Basic Information**

Service Channel Name

Developer Name

Salesforce Object

Custom Console Footer Component

Minimize the Omni-Channel widget when work is accepted

Automatically accept work requests

Is Interruptible

---

**Audio Settings**

Override agents' audio settings

Save Cancel

You can choose **Automatically accept work requests** if you want to have your agents answering chat conversations automatically. This will configure all agents that use this Service Channel. See the [Presence Configuration](#) section below for a more granular configuration.

## Assign Field-Level Security

On the setup screen, search for **Profiles** and click on **Profiles**. Click on the profile the agents will use (do not click on the 'Edit' link).

SETUP
Profiles

---

### Profiles

**All Profiles** | Edit | Delete | Create New View

	Action	Profile Name ↑	User License
<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Clone</a>	<a href="#">Salesforce API Only System Integrations</a>	Salesforce Integration
<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Clone</a>	<a href="#">Silver Partner User</a>	Silver Partner
<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Clone</a>	<a href="#">Solution Manager</a>	Salesforce
<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Clone</a>	<a href="#">Standard Platform User</a>	Salesforce Platform
<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Clone</a>	<a href="#">Standard User</a>	Salesforce
<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Clone</a>	<a href="#">System Administrator</a>	Salesforce

Scroll down to the **Field-Level Security** and **Custom Field-Level Security**, click on the **View** link of the **Experience** object. Click on the **Edit** button and set the accesses as shown here:

<a href="#">Edit</a> <a href="#">Back to Profile</a>			
Field Name	Field Type	Read Access	Edit Access
Account	Lookup	✓	✓
Agent Id	Text	✓	<input type="checkbox"/>
Authenticated	Checkbox	✓	<input type="checkbox"/>
Case	Lookup	✓	✓
Completed	Checkbox	✓	✓
Contact	Lookup	✓	✓
Created By	Lookup	✓	<input type="checkbox"/>
Customer Id	Text	✓	<input type="checkbox"/>
Detail Analytics	URL	✓	<input type="checkbox"/>
Ended	Date/Time	✓	<input type="checkbox"/>
Experience Name	Auto Number	✓	<input type="checkbox"/>
Genesys Cloud Interaction Transcript	Lookup	✓	<input type="checkbox"/>
Intent	Text	✓	<input type="checkbox"/>
Interaction Id	Text	✓	✓
Last Modified By	Lookup	✓	<input type="checkbox"/>
Last utterance	Text	✓	<input type="checkbox"/>
Media Type	Picklist	✓	<input type="checkbox"/>
Next Experience	Lookup	✓	<input type="checkbox"/>
Owner	Lookup	✓	✓
Previous Experience	Lookup	✓	<input type="checkbox"/>
Queue Id	Text	✓	<input type="checkbox"/>
Queue Name	Text	✓	<input type="checkbox"/>
Related Object	Picklist	✓	<input type="checkbox"/>
Started	Date/Time	✓	<input type="checkbox"/>
Wrapped	Date/Time	✓	<input type="checkbox"/>
Wrap-up Code Id	Text	✓	<input type="checkbox"/>
Wrap-up Code Name	Text	✓	<input type="checkbox"/>
Wrap-up Code Notes	Text Area	✓	<input type="checkbox"/>
Wrap-up Code Timeout	Number	✓	<input type="checkbox"/>
Wrap-up Code Type	Text	✓	<input type="checkbox"/>

Do the same thing with the GCX Chat Transcript :

<a href="#">Edit</a> <a href="#">Back to Profile</a>			
Field Name	Field Type	Read Access	Edit Access
Body	Long Text Area	✓	<input type="checkbox"/>
Chat Transcript Name	Text	✓	✓
Created By	Lookup	✓	<input type="checkbox"/>
Last Modified By	Lookup	✓	<input type="checkbox"/>
Owner	Lookup	✓	✓

Similarly, do the same thing with the Email Experience object:



<a href="#">Edit</a> <a href="#">Back to Profile</a>			
Field Name	Field Type	Read Access	Edit Access
Account	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Agent Id	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Case	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Completed	Checkbox	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Contact	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Created By	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Customer Id	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Ended	Date/Time	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Experience Name	Auto Number	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Forwarded	Date/Time	<input checked="" type="checkbox"/>	<input type="checkbox"/>
From	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
From Address	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Interaction Id	Text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Last Modified By	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Next Email Experience	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Owner	Lookup	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Previous Experience	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Queue Id	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Queue Name	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Related Object	Picklist	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Responded	Date/Time	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Started	Date/Time	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Subject	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
To Address	Text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Wrapped	Date/Time	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Wrap-up Code Id	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Wrap-up Code Name	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Wrap-up Code Notes	Text Area	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Wrap-up Code Timeout	Number	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Wrap-up Code Type	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Email Experience Message object:

<a href="#">Edit</a> <a href="#">Back to Profile</a>			
Field Name	Field Type	Read Access	Edit Access
BCC Address	Text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CC Address	Text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Created By	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
EmailMessageId	Text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
From Address	Email	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
From Name	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
HTML Body	Long Text Area	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Is Incoming	Checkbox	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Last Modified By	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Message Date	Date/Time	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Message Identifier	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Name	Text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Owner	Lookup	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
RelatedToId	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Subject	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Text Body	Long Text Area	<input checked="" type="checkbox"/>	<input type="checkbox"/>
To Address	Text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

And finally the Email Experience Message Attachment object:

<span>Edit</span> <span>Back to Profile</span>			
Field Name	Field Type	Read Access	Edit Access
AttachmentId	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
AttachmentName	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Content Length	Number	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Content Type	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Content URI	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Content Version Id	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Created By	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Is Inline	Checkbox	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Is Uploaded	Checkbox	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Last Modified By	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Name	Text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Owner	Lookup	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
RelatedToId	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>

## Create the Service Presence Statuses

Note: In this section, if you use Genesys Cloud for Service Cloud Voice, you don't need to create new Service Presence Statuses, you just need to add the newly created Service Channel to the existing available statuses you want your agents to use when they process Genesys Cloud Chats.

Create, at least, the following Service Presence Statuses connected to the **Genesys Cloud Messaging** Service Channel you created in the previous step:

- **Available**
- **Available on Queue**
- **Busy**

On the Setup screen, search for **Presence Statuses**, click on **New**, and fill in the form:

### Presence Statuses

Let agents indicate when they're online and available to receive work items from a specific service channel, or whether they're away or offline.

Save Cancel

**Basic Information** | = Required Information

Status Name

Developer Name

**▼ Status Options**

Choose whether agents are online or busy when they use this status. Online statuses let agents receive new work items. Busy statuses make your agents appear away and indicate that they're unavailable to receive work items.

Online

Busy

**▼ Service Channels**

Select one or more service channels to assign to this presence status. Agents logged into this presence status can receive work from the channels you select.

Available Channels

Add

▶

Remove

◀

Selected Channels

Genesys Cloud Email

Genesys Cloud Messaging

Messaging

Phone

Repeat the process for the **Busy** status:

## Presence Statuses

Let agents indicate when they're online and available to receive work items from a s

---

**Basic Information**

Status Name

Developer Name

▼ **Status Options**

Choose whether agents are online or busy when they use this status. Online statuses let agents indicate that they're unavailable to receive work items.

Online  
 Busy

And finally the **Available** status:

## Presence Statuses

Let agents indicate when they're online and available to receive work items from a specific service channel, or whether they're away or offline.

---

**Basic Information** ! = Required Information

Status Name

Developer Name

▼ **Status Options**

Choose whether agents are online or busy when they use this status. Online statuses let agents receive new work items. Busy statuses make your agents appear away and indicate that they're unavailable to receive work items.

Online  
 Busy

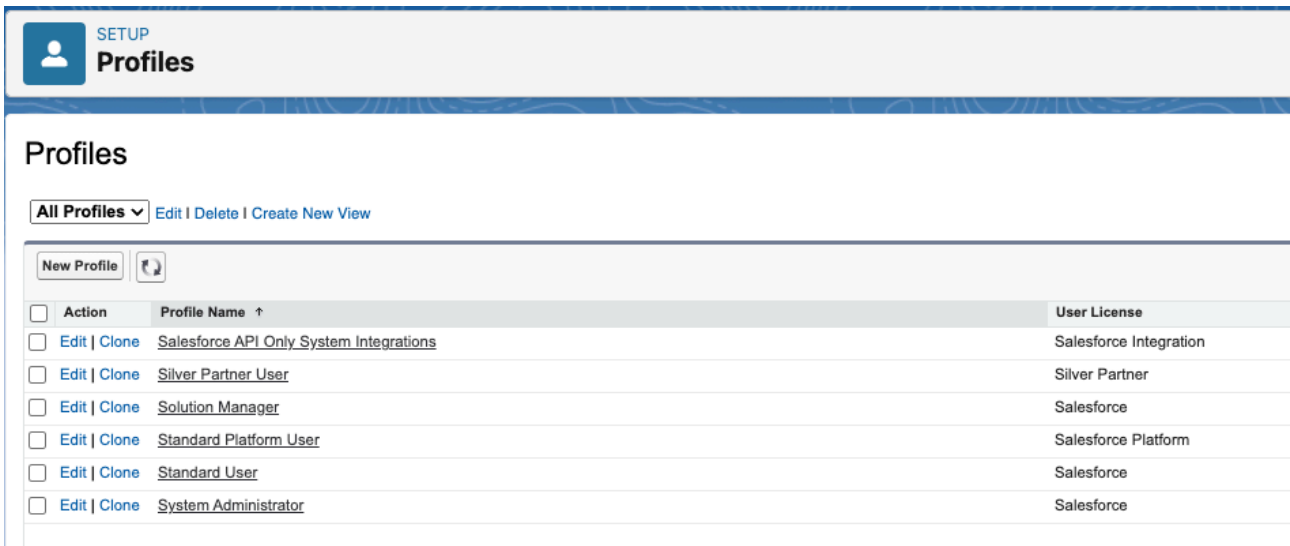
▼ **Service Channels**

Select one or more service channels to assign to this presence status. Agents logged into this presence status can receive work from the channels you select.

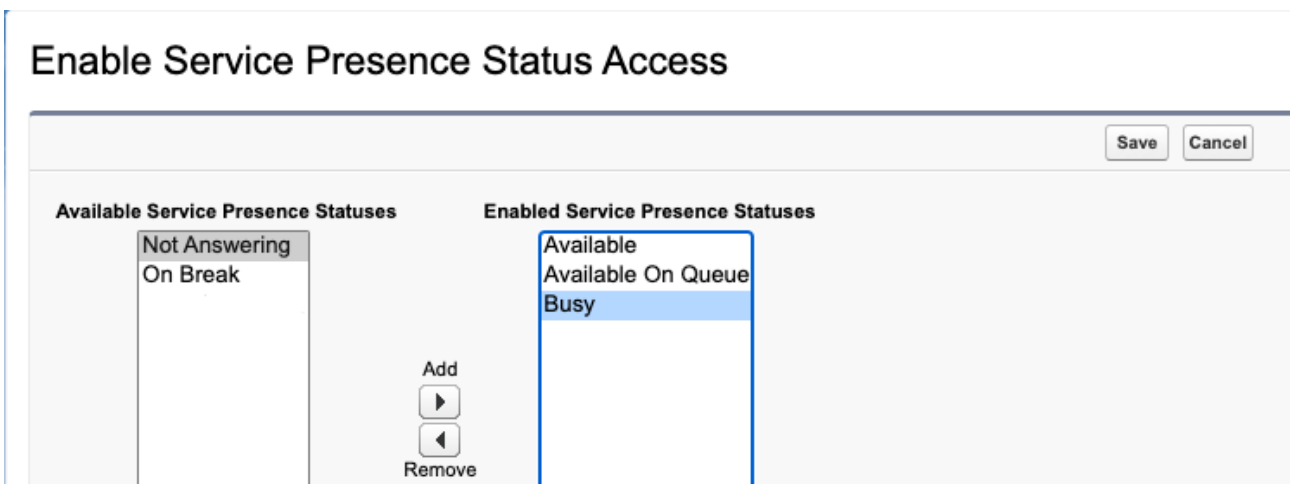
<p><b>Available Channels</b></p> <div style="border: 1px solid #ccc; height: 40px; width: 100%;"></div>	<input type="button" value="Add"/> <input type="button" value="Remove"/>	<p><b>Selected Channels</b></p> <div style="border: 1px solid #ccc; padding: 5px;">           Genesys Cloud Email            Genesys Cloud Messaging            Messaging            Phone         </div>
---	---	---

## Assign the Presence Statuses to User Profiles

On the setup screen, search for **Profiles** and click on **Profiles**. Click on the profile the agents will use (do not click on the 'Edit' link).



Click on the `Enabled Service Presence Status Access[0]` link, and then on the `Edit` button, and add all the statuses you created in the previous step:



### Create the Presence User Configuration

Create a Presence User Configuration for agents that will be handling Genesys Cloud chats, emails. You should set the capacity of your agents. For example, if you have 10 agents, and each agent can handle 5 chats at the same time, then you should set the `Capacity` to 50.

On the Setup screen, search for `Presence Configurations`, click on `New`, and fill in the form:

Save Cancel

**Basic Information** | = Required Information

Presence Configuration Name

Developer Name

Capacity

Interruptible Capacity

Automatically accept work requests

Allow agents to decline work requests

Update Status on Decline

Allow agents to choose a decline reason

Update Status on Push Timeout

**Audio Settings**

Play a notification sound for work requests

Notification Sound  Default  Custom sound

Sound Length (Seconds)

*Maximum: 30*

Play a notification sound if Omni Channel loses connection

**After Conversation Work Time**

Give agents wrap-up time after conversations

Assign the new Presence Configuration to agents and/or user profiles.

Notes:

- If you want to have your agents answering conversations automatically, you should configure the **Presence Configuration** in Salesforce and not set the queue to auto-answer in Genesys Cloud. Only agents that are mentioned in this conversation or member of the profile that is mentioned in the **Presence Configuration** will be able to automatically answer.
- You can also use difference configurations for Chat, Voice and Email, etc. Some configurations would allow auto-answer for chats, but not for emails, for example. They could also use different alerts, capacities, etc.

## Create a Routing Configuration

On the setup screen, search for **Routing Configurations**, and click on the **New** button.

Create a Routing Configuration called **Experiences** with a priority of 1 and **External Routing** model.

## Routing Configurations

The routing priority determines the order in which work items are pushed to agents. Higher priority work items are pushed first.

The routing model determines how to distribute work items to your agents. It acts as a tiebreaker if two work items have the same priority. It also acts as a tiebreaker between work item capacity and open work items.

**Basic Information**

Routing Configuration Name

Developer Name

Overflow Assignee ⚠ If you don't give the overflow assignee, the overflow work items will be assigned to the user specified in the overflow assignee dropdown.

User

**Routing Settings**

The routing priority determines the order in which work items across your Omni-Channel queues get processed. Lower priority work items are processed first.

The routing model determines how to evenly distribute work items to your agents. It acts as a tiebreaker if two work items have the same priority.

Routing Priority

Routing Model

Push Time-Out (seconds)

**Work Item Size**

Specify the size of the work items in the queues associated with this configuration. You can size items in units or as a percentage of capacity.

Units of Capacity

Percentage of Capacity

### Create the flows

Create Backup Queues for the flows

On the setup screen, search for **Queues**, and click on the **New** button.

Create a Queue called **Experiences** that will deal with the **Experience** object and add the users/groups you desire as queue members. Also attach the Routing Configuration previously created.

### Queue Edit

#### Queue Name and Email Address

Enter the name of the queue and the email address to use when sending notifications (for example

Label

Queue Name  i

Queue Email

Send Email to Members

#### Configuration with Omni-Channel Routing

If your organization uses Omni-Channel, you can link queues to a routing configuration. This will pu

Routing Configuration  🔍

#### Supported Objects

Select the objects you want to assign to this queue. Individual records for those objects can then be

Available Objects		Selected Objects
Agent Work		Experience
Alternative Payment Method		
Appointment Bundle Config		
Appointment Bundle Policy		
Attribute Definition		
Attribute Picklist		
Authorization Form		
Authorization Form Consent		
Authorization Form Data Use		
Business Brand		
Case		
Change Request		
GCX Chat Transcript		
Communication Subscription		

Add

Remove

#### Queue Members

To add members to this queue, select a type of member, then choose the group, role, or user from those objects.

Search:  for:

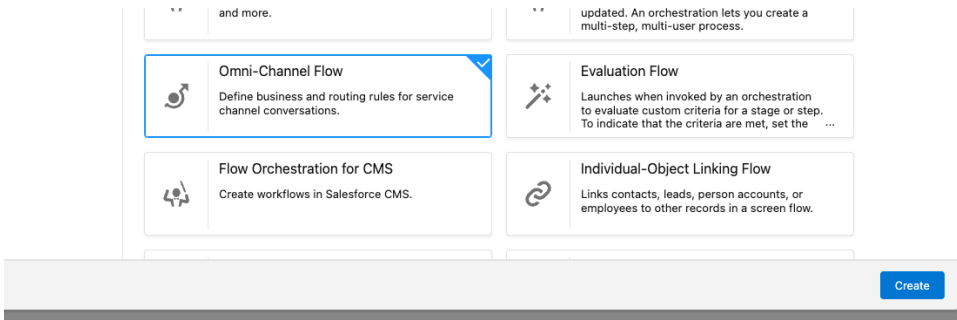
Available Members	Selected Members
User: Integration User	User User
User: Security User	

Add

Create another queue called **Email Experiences** that will deal with the **Email Experience** object and add the users/groups you desire as queue members. Also attach the Routing Configuration previously created.

Create the Flow to pop an Experience to an agent

On the setup screen, search for **Flows** and click on **New Flow**. Select **All + Templates**, **Omni-Channel Flow** and click on **Create** :



Add a new Resource of type Variable called input\_record of type genesysps\_\_Experience\_\_c :

The 'New Resource' dialog box is shown with the following configuration:
 

- Resource Type:** Variable
- API Name:** input\_record
- Description:** (empty text area)
- Data Type:** Record
- Allow multiple values (collection):**
- Object:** Experience
- Availability Outside the Flow:**
  - Available for input
  - Available for output

 Buttons for 'Cancel' and 'Done' are at the bottom right.

Note: Make sure to check the Available for input checkbox.

Add a new Resource of type Variable called recordId if type Text :


The 'New Resource' dialog box is shown with the following configuration:
 

- Resource Type:** Variable
- API Name:** recordId
- Description:** (empty text area)
- Data Type:** Text
- Allow multiple values (collection):**
- Default Value:** Enter value or search resources... (with search icon)
- Availability Outside the Flow:**
  - Available for input
  - Available for output

 Buttons for 'Cancel' and 'Done' are at the bottom right.

Add a Route Work step called Route to Agent after the Start step. Configure that step as follows:



**Route to Agent** (Route\_to\_Agent) 

## Set Input Values

**\* How Many Work Records to Route?** 

- Single  
 Multiple

**\* Record ID Variable****\* Service Channel****\* Route To****Agent**

- Select Agent  
 Use Variable

**\* Agent ID**

Required Agent 

**Backup Queue**

- Select Queue  
 Use Variable

**\* Queue ID**

The flow should look like this:



Save it as **Pop Experience to Agent** and activate the flow.

Create the Flow to route the Experience to an agent

On the setup screen, search for **Flows** and click on **New Flow**. Select **Record-Triggered Flow** and click on **Create**. Then, search for **Experience** as the triggering object. The Entry Conditions should be set to **All Conditions are Met (AND)** the condition set as follows:

**Select Object**  
 Select the object whose records trigger the flow when they're created, updated, or deleted.

• Object

Experience

---

**Configure Trigger**

• Trigger the Flow When:

A record is created

A record is updated

A record is created or updated

A record is deleted

---

**Set Entry Conditions**  
 Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

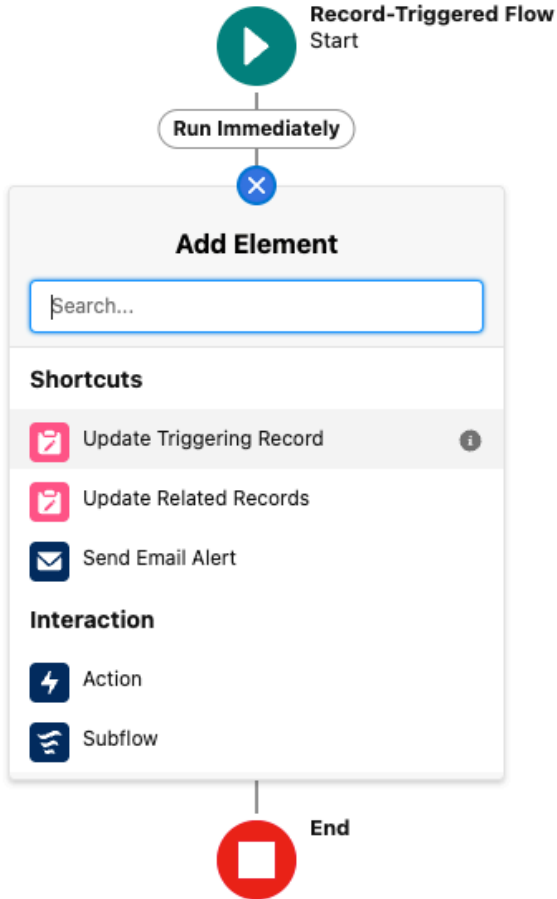
If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements

All Conditions Are Met (AND)

Field	Operator	Value
genesysps__Interaction_Id__c	Is Null	False

Add a **Subflow** step after the **Start** step:



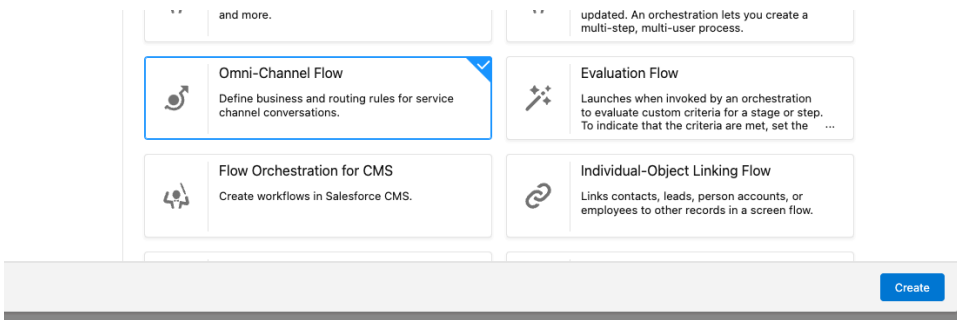
And configure the subflow as follows:

The screenshot shows the "New Subflow" configuration interface. At the top, it says "New Subflow". Below that is a "Referenced Flow" field containing the text "Pop Experience to Agent". A paragraph of text explains: "Use values from the parent flow to set the inputs for the 'Pop Experience to Agent' flow. By default, the parent flow stores all outputs. You can either reference outputs via the API name of the Subflow element or manually assign variables in the parent flow to store individual outputs from the 'Pop Experience to Agent' flow." Below this are two input fields: "Label" with the value "Pop Experience" and "API Name" with the value "Pop\_Experience". There is a "Description" text area below. Another "Referenced Flow" section shows a selection of "Pop Experience to Agent" with an "Open Referenced Flow" button. The "Set Input Values" section has two rows: the first row has "input\_record" with a value of "{!\$Record}" and an "Include" toggle that is turned on; the second row has "recordId" with a value of "{!\$Record.Id}" and an "Include" toggle that is also turned on. At the bottom right, there are "Cancel" and "Done" buttons.

Save it as **Route Experience to Agent** and activate the flow.

Create the Flow to pop an Email Experience to an agent

On the setup screen, search for **Flows** and click on **New Flow** . Select **All + Templates** , **Omni-Channel Flow** and click on **Create** :



Add a new **Resource** of type **Variable** called **input\_record** of type **genesysps\_\_EmailExperience\_\_c** :

### New Resource

---

\* Resource Type

Variable

\* API Name

input\_record

Description

\* Data Type

Record  Allow multiple values (collection) ⓘ

\* Object

Email Experience

**Availability Outside the Flow**

Available for input

Available for output

Cancel Done

Note: Make sure to check the **Available for input** checkbox.

Add a new **Resource** of type **Variable** called **recordId** if type **Text** :

### New Resource

**\*Resource Type**  
Variable

**\*API Name**  
recordId

Description

**\*Data Type**  
Text  Allow multiple values (collection) ⓘ

Default Value  
Enter value or search resources...

**Availability Outside the Flow**  
 Available for input  
 Available for output

Add a **Route Work** step called **Route to Agent** after the **Start** step. Configure that step as follows:

## Set Input Values

**\* How Many Work Records to Route?** ⓘ

- Single
- Multiple

**\* Record ID Variable**

**\* Service Channel**

**\* Route To**

---

### Agent

- Select Agent
- Use Variable

**\* Agent ID**

Required Agent ⓘ

---

### Backup Queue

- Select Queue
- Use Variable

**\* Queue ID**

---

## Set Additional Input Values

Screen Pop Collection Variable

The flow should look like this:



Save it as `Pop Email Experience to Agent` and activate the flow.

Create the Flow to route the Email Experience to an agent

On the setup screen, search for `Flows` and click on `New Flow`. Select `Record-Triggered Flow` and click on `Create`. Then, search for `Email Experience` as the triggering object. The Entry Conditions should be set to `All Conditions are Met (AND)` the condition set as follows:

### Select Object

Select the object whose records trigger the flow when they're created, updated, or deleted.

\* Object

### Configure Trigger

\* Trigger the Flow When:

- A record is created
- A record is updated
- A record is created or updated
- A record is deleted

### Set Entry Conditions

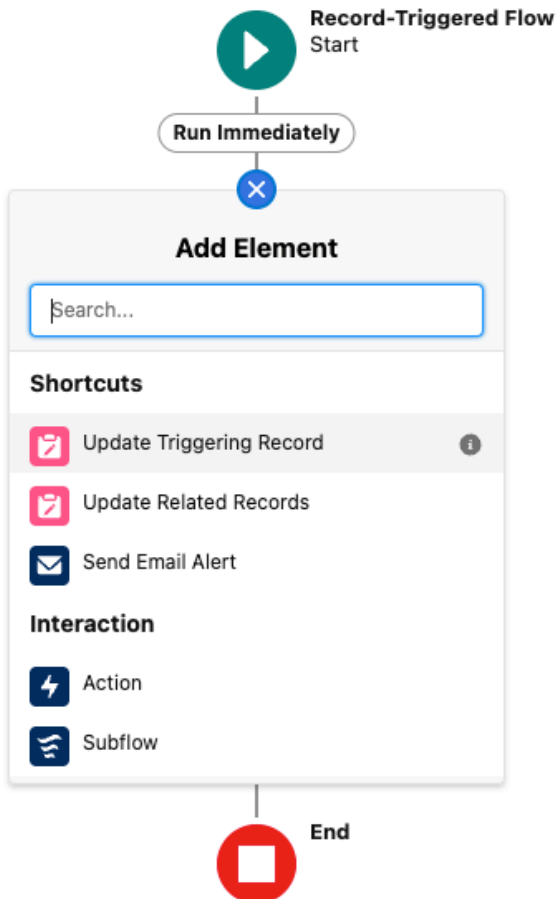
Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements

Field	Operator	Value
genesysps__Interaction_Id__c	Is Null	False

Add a **Subflow** step after the **Start** step:





And configure the subflow as follows:

### New Subflow

---

Referenced Flow

Pop Email Experience to Agent

Use values from the parent flow to set the inputs for the "Pop Email Experience to Agent" flow. By default, the parent flow stores all outputs. You can either reference outputs via the API name of the Subflow element or manually assign variables in the parent flow to store individual outputs from the "Pop Email Experience to Agent" flow.

\* Label \* API Name

Pop Email Experience

Pop\_Email\_Experience

Description

Referenced Flow

**Pop Email Experience to Agent**

[Open Referenced Flow](#)

Set Input Values

input\_record
?

Include

{\$Record}

---

recordId
?

Include

{\$Record.Id}

Cancel
Done

Save it as **Route Email Experience to Agent** and activate the flow.

## Create a new Record Page for the Experience

On the setup screen, search for **Lightning App Builder** and click on **New** to create a new **Record Page** for the **Experience** object called "Experience Record Page". And add the various Components you desire.

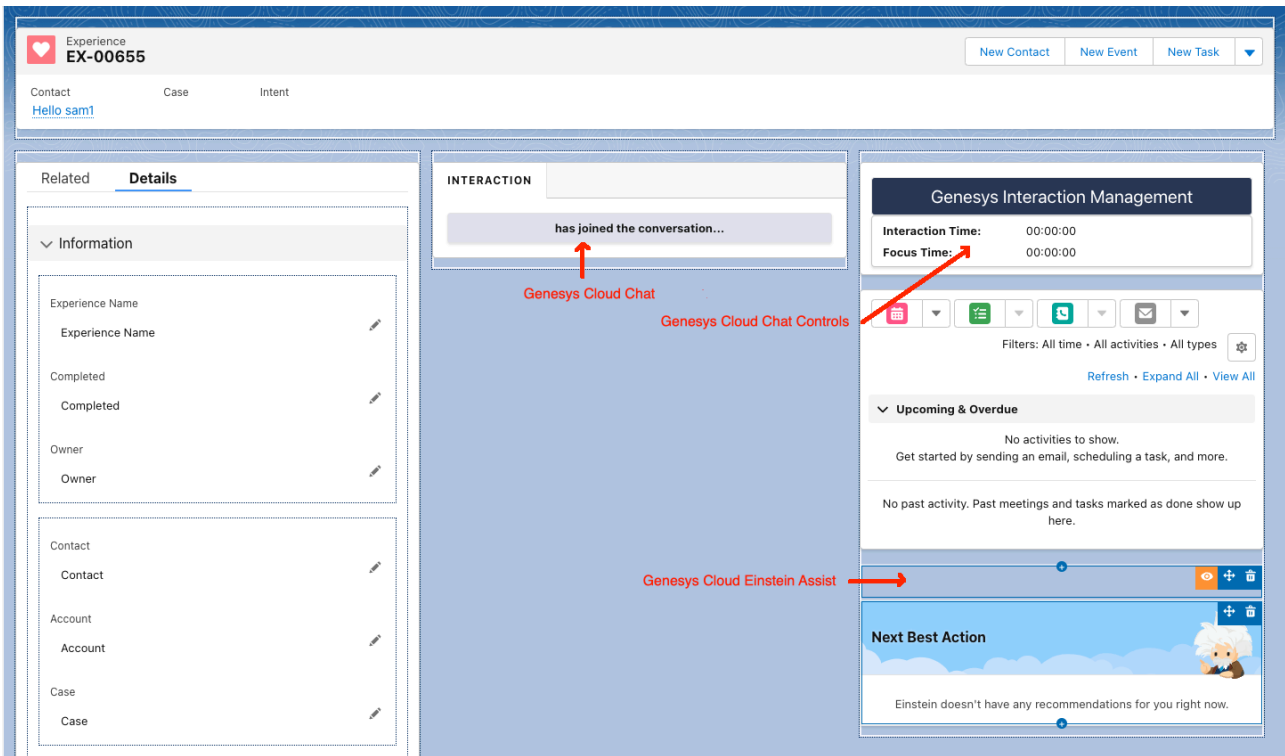
For an optimal experience, we recommend the following components:

- **Genesys Cloud Chat**  
This component shows the active chat conversation as well as the transcript once the chat has been disconnected.
- **Genesys Cloud Chat Controls**  
This component is used to control the active chat (disconnect, transfer, etc.)
- **Genesys Cloud Einstein Assist**  
This component transmits the chat text to Einstein's Next Best Action. It can be added to the Experience Record Page or to the Utility Bar (not both).

3 more components are available:

- **Genesys Cloud Chat Bar Utility**  
This component is used in the Utility Bar (where Omni-Channel shows) and should not be added to the Record page.
- **Genesys Cloud Transcript**  
This component is used to see the transcript of a voice or chat conversation. It is not used on the Experience Record page, it can rather be used on a Voice Record page, for example.
- **Genesys Cloud Responses**  
This component is used to show the canned responses available in Genesys Cloud to the agent. Canned responses will be available when the agent is logged in. It works for both Chat and Email.

Here is an example of what the page could look like:



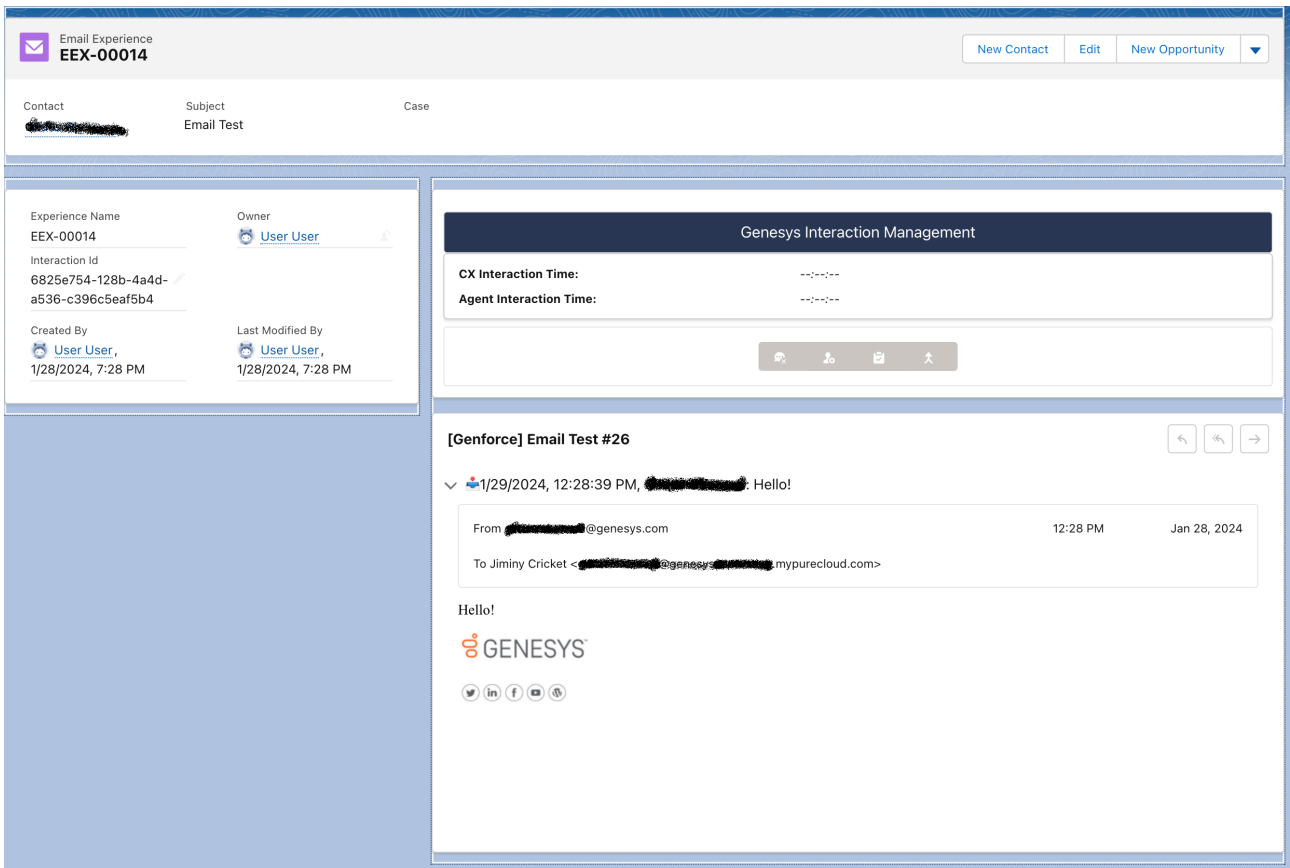
### Create a new Record Page for the Email Experience

On the setup screen, search for **Lightning App Builder** and click on **New** to create a new **Record Page** for the **Email Experience** object called "Email Experience Record Page". And add the various Components you desire.

For an optimal experience, we recommend the following components:

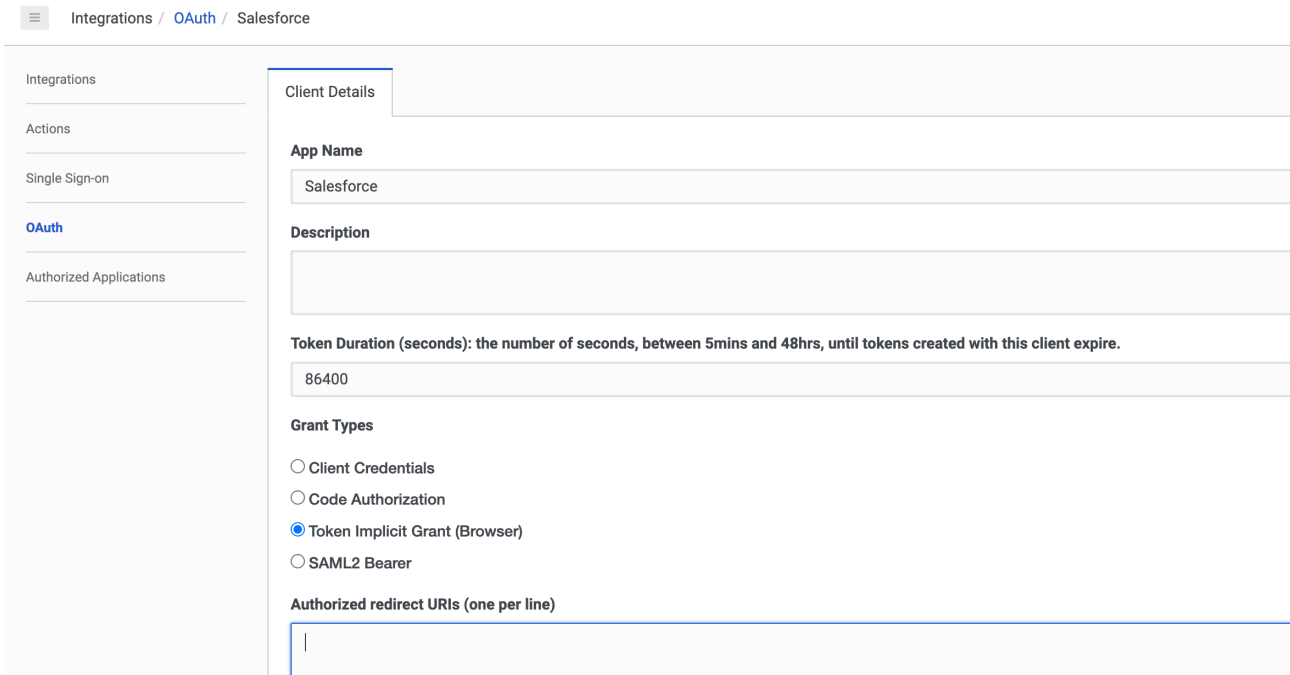
- **Genesys Cloud Email**  
This component shows the active email conversation.
- **Genesys Cloud Email Controls**  
This component is used to control the active email (disconnect, transfer, etc.)

Here is an example of what the page could look like:



### Configure a Lightning App for the Experience

Log on your Genesys Cloud console with an Administrator account and create a **Token Implicit Grant** :



**Notes:**

- The Token Duration must be longer than the shift of your agents.
- The scopes for the OAUTH implicit grant should be:
  - **conversations** ,

- notifications ,
- presence ,
- response-management:readonly (if you plan to use Canned Responses),
- routing:readonly ,
- upload .
- users

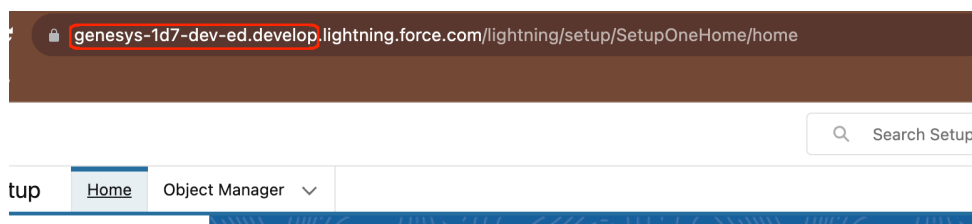
- The permissions for the agents should be:

- conversation:communication:blindTransfer ,
- conversation:message:create ,
- conversation:message:view ,
- conversation:participant:wrapup ,
- conversation:webmessaging:create ,
- conversation:webmessaging:view ,
- directory:user:view ,
- externalContacts:contact:view ,
- responses:library:view , (if you plan to use Canned Responses)
- responses:response:view , (if you plan to use Canned Responses)
- routing:queue:search ,

- Do not forget to add your Salesforce redirect URI to the list of allowed redirect URIs in the Genesys Cloud OAUTH configuration. That URL should be something like this:

`https://xxxx.lightning.force.com/lightning/page/home`

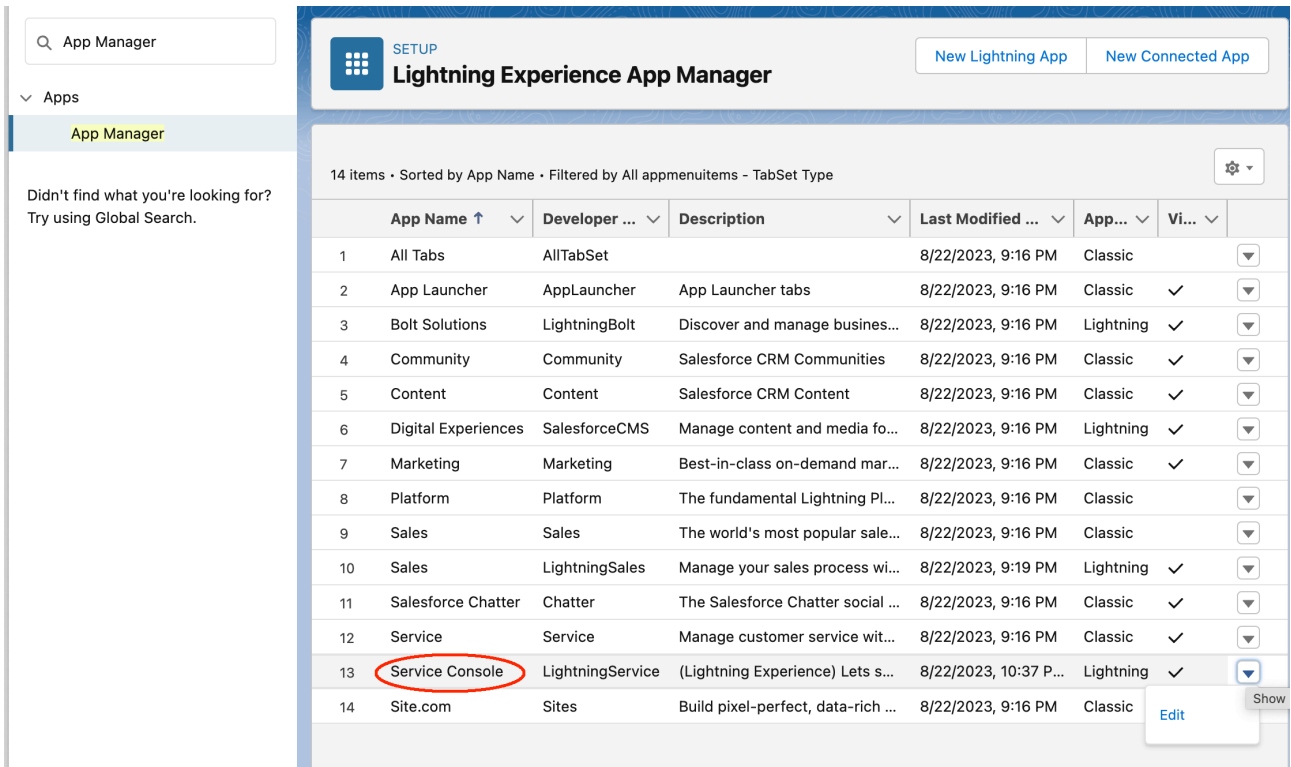
where `xxxx` is the base name of your org. You can find that name in the URL of your Salesforce org:



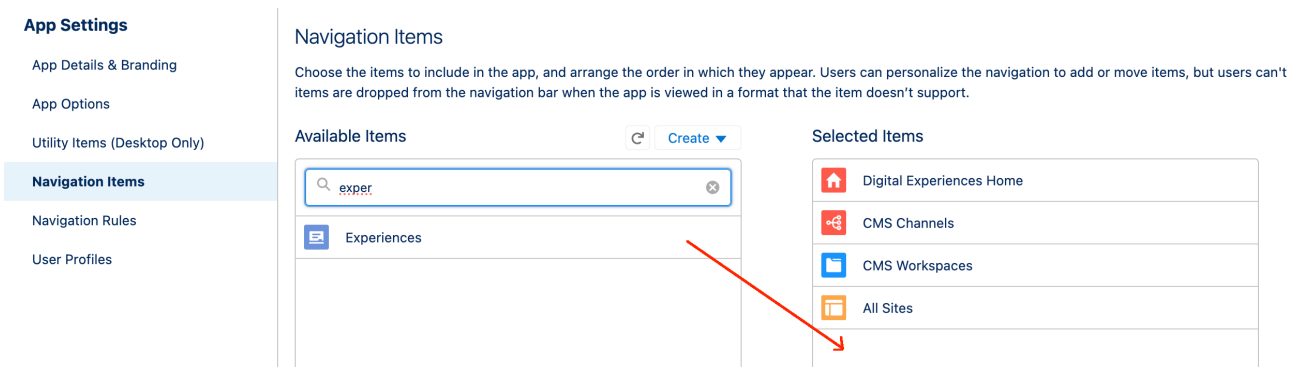
In this example, the basename is `genesys-1d7-dev-ed.develop` , so the redirect URI in Genesys Cloud should be set to:

`https://genesys-1d7-dev-ed.develop.lightning.force.com/lightning/page/home`

Back on the Salesforce pages, on the setup screen, search for the `App Manager` and choose one of the Lightning apps (you can also create your own), here we will use the `Service Console` app. Click on the `Edit` button:



In the Lightning App Builder, click on the **Navigation Items** tab, and add the **Experiences** to the right column (Selected Items):



In the Lightning App Builder, click on the **Utility Items** tab, and add the **CX Cloud, Digital and AI** component:

**App Settings**

App Details & Branding

App Options

**Utility Items (Desktop Only)**

Navigation Items

Navigation Rules

User Profiles

Utility Items (Desktop O

Give your users quick access to p

Add Utility Item

Search...

- Visualforce
- Custom (0)**  
No components available.
- Custom - Managed (4)**
  - CX Cloud, Digital and AI
  - Genesys Cloud Einstein Assist
  - Genesys Cloud for Salesforce Pa...
  - Genesys Cloud for Salesforce Se...

Fill in the Component Properties with the region of your Genesys Cloud Organization and the client ID of your Genesys Cloud OAuth Implicit Grant. Also pick what Salesforce Presence Status should be used for Available and Available on Queue:

Omni-Channel

**CX Cloud, Digital and AI**

History

Notes

PROPERTIES  
CX Cloud, Digital and AI

↑ ↓ Remove

Utility Item Properties

\* Label ⓘ  
CX Cloud, Digital and AI

Icon ⓘ  
⚡ fallback ×

Panel Width ⓘ  
340

Panel Height ⓘ  
480

Start automatically ⓘ

Component Properties

\* Is CX Cloud, Voice for Salesforce Cloud Voice installed?  
No 🔍

\* Genesys Cloud Region  
Americas (US East) 🔍

\* Genesys Cloud Implicit Grant Client Id  
  
This field is required.

Available Presence Status  
Available 🔍

Available On Queue Presence Status  
Available on Queue 🔍

Tab name format  
Firstname Lastname 🔍

You can also change the label and icon to your liking.

Notes:

- Do not forget to check the **Start Automatically** box.
- If you have installed CX Cloud, Voice for Salesforce Cloud Voice, select Yes. You can choose any status in the latter fields, they will not be used by CX Cloud, Digital and AI for Salesforce Cloud Voice.
- The 'Tab name format' will be used to name the tabs containing Experiences. **Firstname** and **Lastname** come from the Contact associated with the Experience.

If you didn't add **Genesys Cloud Einstein Assist** on your Experience Record Page, you should add it here in the **Utility Items** (remember, don't add it in both places):

## App Settings

App Details & Branding

App Options

**Utility Items (Desktop Only)**

Navigation Items

Navigation Rules

User Profiles

## Utility Items (Desktop Only)

Give your users quick access to p

Add Utility Item


Search...

 Visualforce


▼ **Custom (0)**


No components available.

▼ **Custom - Managed (4)**

 CX Cloud, Digital and AI

 Genesys Cloud Einstein Assist

 Genesys Cloud for Salesforce Pa...

 Genesys Cloud for Salesforce Se...



---

Page > Genesys Cloud Einstein As...

---

Trained ModelId 

Email ID 

Private Key 

Contact Center Name

Use Last Utterance for NBA? 

#### Component Properties:

- **Trained Model ID (deprecated):** Specifies the trained model for Einstein Assistant’s Knowledge Articles. May not be needed for other GCEA features, such as Real-Time Voice Transcription and NBA strategy activation. If a value is obligatory, placeholder text is acceptable.
- **Email ID (deprecated):** Associates an email with your Einstein Assistant for Knowledge Articles. It’s not always needed for other features; a placeholder may be used if necessary.
- **Private Key :** Provides a secure means of authenticating and ensuring data integrity between Salesforce and Genesys. The RSA private key, usually shared with Genesys for SCV setup, should be pasted here. It’s a lengthy alphanumeric code that typically starts with “--BEGIN RSA PRIVATE KEY--” and ends with a corresponding “--END RSA PRIVATE KEY--” tag.
- **Contact Center Name :** Designates which contact center within Genesys the Salesforce integration should point to. Select the interaction name from the dropdown.
- **Use Last Utterance for NBA? :** When the option “Yes” is selected for Next Best Action, the last customer utterance will be utilized. Conversely, if the option is not selected, the last utterance will not be used.

## Einstein Next Best Action Integration in Salesforce

### Key Components:

- **Strategy Development for Recommendations:** Crafting an effective strategy is crucial to tailor recommendations that align with business goals and customer needs. This involves analyzing customer data and behaviors to generate personalized and contextually relevant suggestions.
- **Implementation of Suggested Actions:** After developing a strategy, the next step is to implement actionable suggestions. These actions are designed to enhance customer engagement and decision-making processes, leveraging the predictive power of Einstein to deliver optimal recommendations at the right moment.

### Create Recommendations

Recommendations in Salesforce are treated as standard records, much like accounts and contacts. These recommendation records are processed by strategies and linked to flows. The role of strategies is to decide which recommendation records should be surfaced. This is achieved through the use of business rules, predictive

models, and various data sources. The outcome of this process yields recommendations tailored to specific contexts, enabling you to present them to users.

Follow the steps to [Create Recommendations](#).

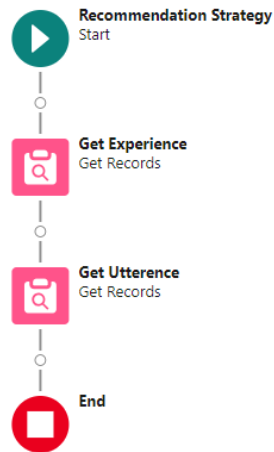
## Create a Recommendation Strategy for Einstein's Next Best Action

On the setup screen, search for **Process Automation**. Click on **Next Best Action** and create a new strategy. For Voice and Messaging, two different strategies should be created.

### Using **Flow Builder**

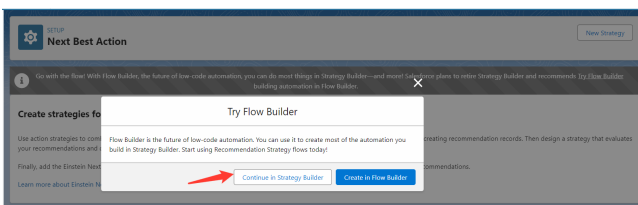
- Go to Setup and use the Quick Find box to search for "Flows". Select 'Flows' and then click 'New Flow'.
- In the 'Templates' tab, choose 'Recommendation Strategy' as the flow type and click 'Create'.
- To load the desired records for your recommendation, add a 'Get Records' element to the flow. Set a label and an API name.

- Choose the Experience object.
- In the 'Filter' section, define conditions to filter the records from your chosen object for the strategy.
- Add another 'Get Records' element to bring a specific recommendation into your strategy. Again, set a label and an API name.
- Select the 'Recommendations' object.
- Use conditions in the 'Filter' section to specify the particular recommendation you wish to use.
- Incorporate additional flow elements as necessary to complete your strategy.
- To make the recommendation available in an Einstein Next Best Action component, add an 'Assignment' element.
- For the 'Variable', choose 'outputRecommendations'.
- Set the 'Operator' to 'Equals' and the 'Value' to your predefined recommendation.
- Save the flow.



- Finally, activate your flow.

Select **Continue in Strategy Builder**



For VoiceCall: Select the **VoiceCall** Object.

New Strategy

\* Name: Voice Strategy      \* API Name: Voice

Description:

Object Where Recommendations Display: Voice Call

Template

Cancel Done

For Messaging: Select the **Experience** Object.

New Strategy

\* Name: Messaging\_Strategy      \* API Name: Messaging\_Strategy

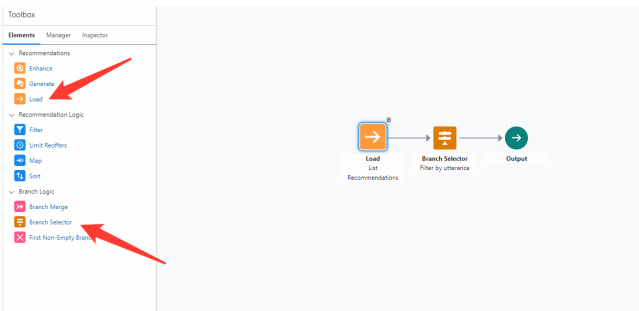
Description:

Object Where Recommendations Display: Experience

Template

Cancel Done

Workflow Example :



**Load List Recommendations:** Begin by importing or inputting a list of recommendations into the system.

**Object:** Defines the source from which recommendations will be pulled. Value: "Recommendation"

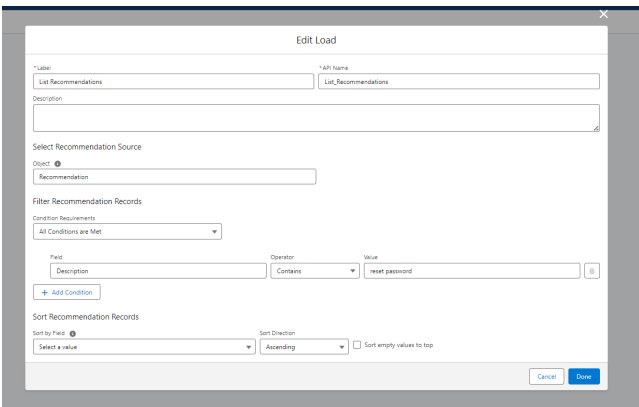
**Field:**

**Purpose:** Select the specific field from the recommendation records that will be evaluated. Input Type: Drop-down Value: "Description" or "Name"

**Operator:** Purpose: Specifies the relationship between the Field and Value. Input Type: Dropdown Example Value: "Contains"

**Value:** Purpose: The content against which the Field will be evaluated based on the Operator. Input Type: Text Example Value: "reset password" (preferable lower case)

**Add Condition:** Button to add more filter conditions.



**Branch Selector (Filter by utterance):** Evaluate user input (e.g., Last utterance) and determine the appropriate branch or recommendation logic to follow.

**Resource:**

**Purpose:** Identifies the specific data point or resource that will be evaluated.

**Input Type:** Dropdown

**Example Value:** "\$Record.genesysps\_Last\_utterance"

**Operator:**

**Purpose:** Specifies the relationship between the Resource and Value.

**Input Type:** Dropdown

**Example Value:** "Contains"

**Value:**

**Purpose:** The content against which the Resource will be evaluated based on the Operator.

**Input Type:** Text

**Example Value:** "password"

**Add Condition:** Button to incorporate additional filter conditions.

**Edit Branch Selector**

Add conditions to branches. When the condition is true, the recommendations in the branch move forward.

\* Label:  \* API Name:

Description:

**Settings**

Evaluate each branch in order, starting at the top of the canvas and moving down. Branches that meet the conditions are allowed.

Only allow recommendations from the first branch that meets the conditions:

**Branch 1: List Recommendations**

Recommendations from branch 1 are allowed when this condition is true.

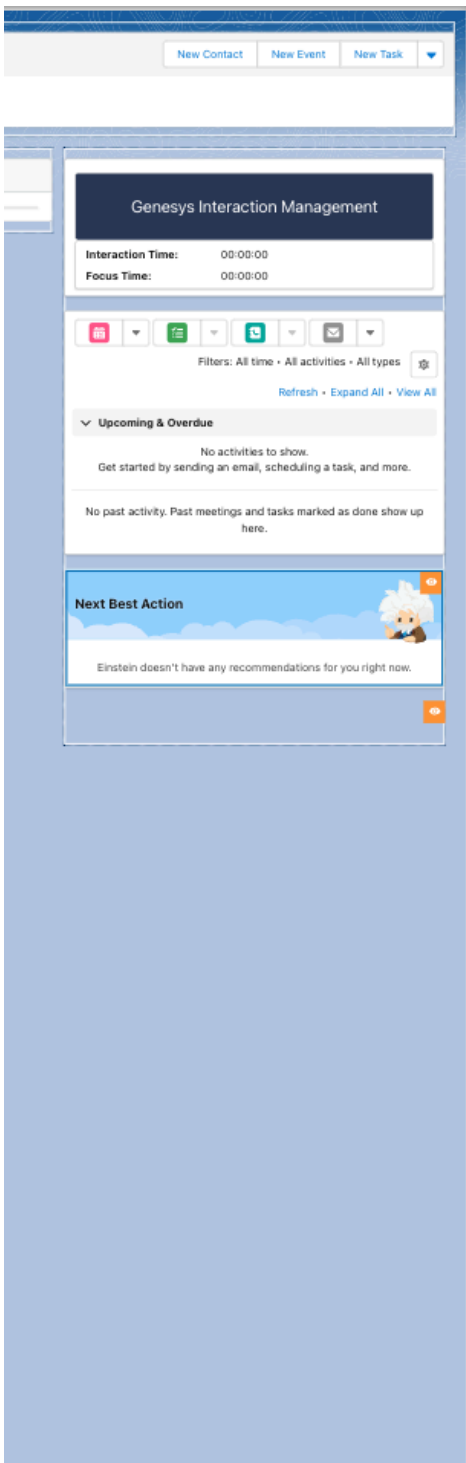
Standard | Advanced

When to Execute Filter:

Resource	Operator	Value
<input type="text" value="\$Record.genesysps__Last_utterance"/>	<input type="text" value="Contains"/>	<input type="text" value="password"/>

## Save the Strategy

In the **VoiceCall** or **Experience** page, add an Einstein **Next Best Action** component and set its Action Strategy to the **Recommendation Strategy** you just created:



Page > Einstein Next Best Action

Show users the best action or offer after creating recommendations and action strategies. [Learn more.](#)

Title ⓘ

Next Best Action

Hide Einstein Header

\* Strategy Source ⓘ

Strategy Builder X

\* Action Strategy ⓘ

Message strategy X

Maximum Recommendations Displayed

1 X

Hide Empty Component ⓘ

Launch Recommendation Action In

Dialog Window X

Show Title

Show Image

Show Description

Show Reject Option

Launch Flow on Rejection

Set Component Visibility

Filters

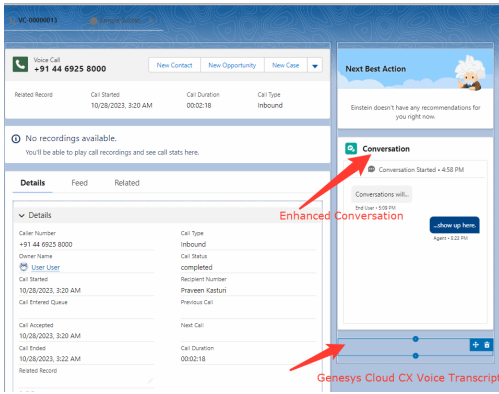
1 Record > Completed Equal false X

+ Add Filter

### Genesys Cloud for Real Time Voice Transcription

To add a Real Time Voice Transcription Component to a voice call record page in Salesforce, you would typically use Lightning App Builder. Here are the steps to do so:

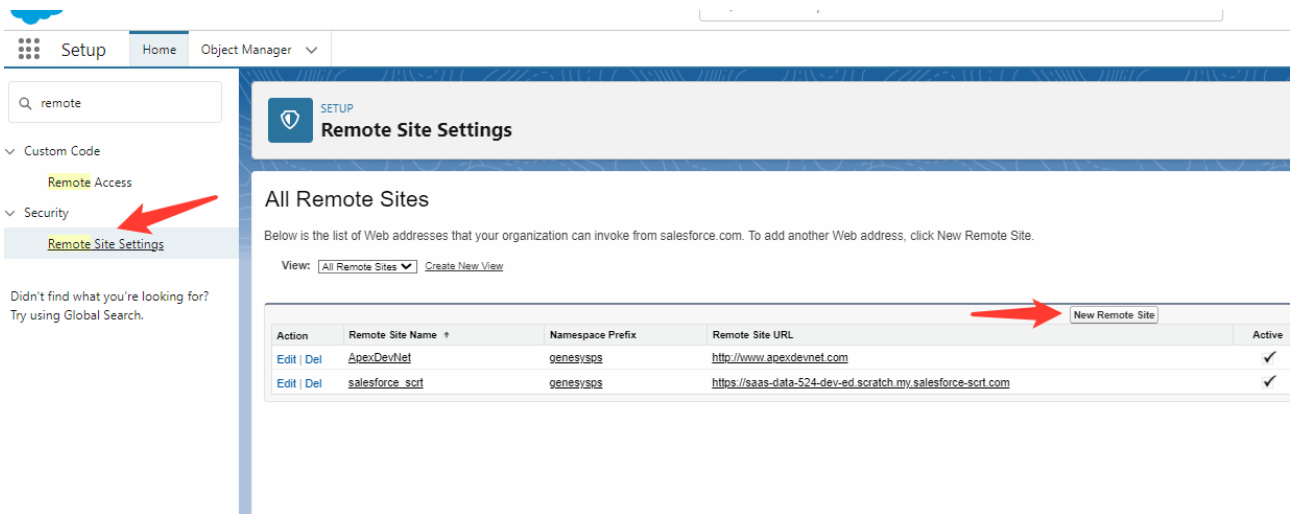
- Go to Setup in Salesforce.
- In the Quick Find box, enter "Lightning App Builder."
- Click "Lightning App Builder" under User Interface.
- In the Lightning App Builder, select the VoiceCall record page
- In the App Builder, add the following components:
  - Enhanced Conversation (Standard Salesforce Component)
  - Genesys Cloud CX Voice Transcript
  - Genesys Cloud Einstein Assist (Ignore if it's added in omni-channel utility)



Verify that voice transcription is enabled at both the organization level and the queue level in [Configure voice transcription](#)

The real-time voice transcription functions through the Voice Call object, sending the transcript to Salesforce for display on the user interface. It is essential to configure the correct private key for this operation; note that this is not the Einstein private key, but rather the one specifically for setting up the Service Cloud Voice integration. Additionally, ensure you select the appropriate contact center. For access to the remote site settings, proceed with the following steps.

Navigate to Setup -> Quick Find -> Security -> Remote Site Settings -> New Remote Site



To modify your Salesforce domain URL and replace “lightning.force.com” with “.my.salesforce-scr1.com”, follow these steps:

Identify Your Current Domain URL: Make sure you know your current Salesforce domain URL. It will look something like this: https://saas-data-524-dev-ed.scratch.lightning.force.com.

Extract Domain Prefix: From the URL, extract the domain prefix. For the example URL provided, the domain prefix is saas-data-524-dev-ed.scratch.

Modify The Domain URL: Replace “lightning.force.com” with “.my.salesforce-scr1.com”. Using the extracted domain prefix, your new URL will look like this: https://saas-data-524-dev-ed.scratch.my.salesforce-scr1.com.

Save the setting.

## Tips for your Genesys Cloud Architect Flow

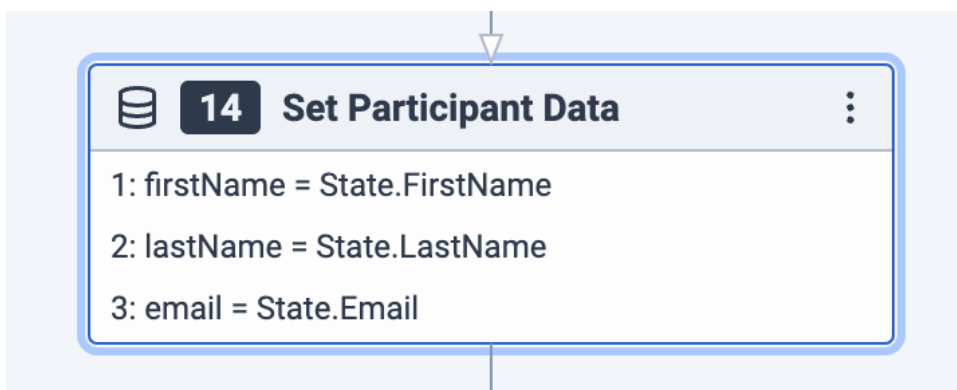
It is possible to make CX Cloud, Digital and AI for Salesforce Service Cloud stitch Salesforce Contact and Case automatically.

To do so, you need to add the following attributes to the customer participant data to the Genesys Cloud Architect Flow:

- firstName
- lastName
- email
- caseNumber

When the Experience gets created, CX Cloud, Digital and AI for Salesforce Service Cloud will search for a Salesforce Contact with the same email. If not found, it will create a new Contact. If found, it will also try to connect to the given Case Number (if provided).

In Architect, you can use the Set Participant Data step to set the attributes:



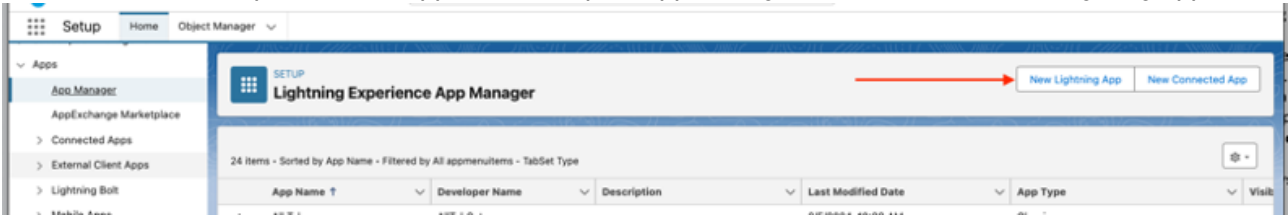
## Multi-Org Setup

### Single Salesforce Org with multiple Genesys Cloud Orgs

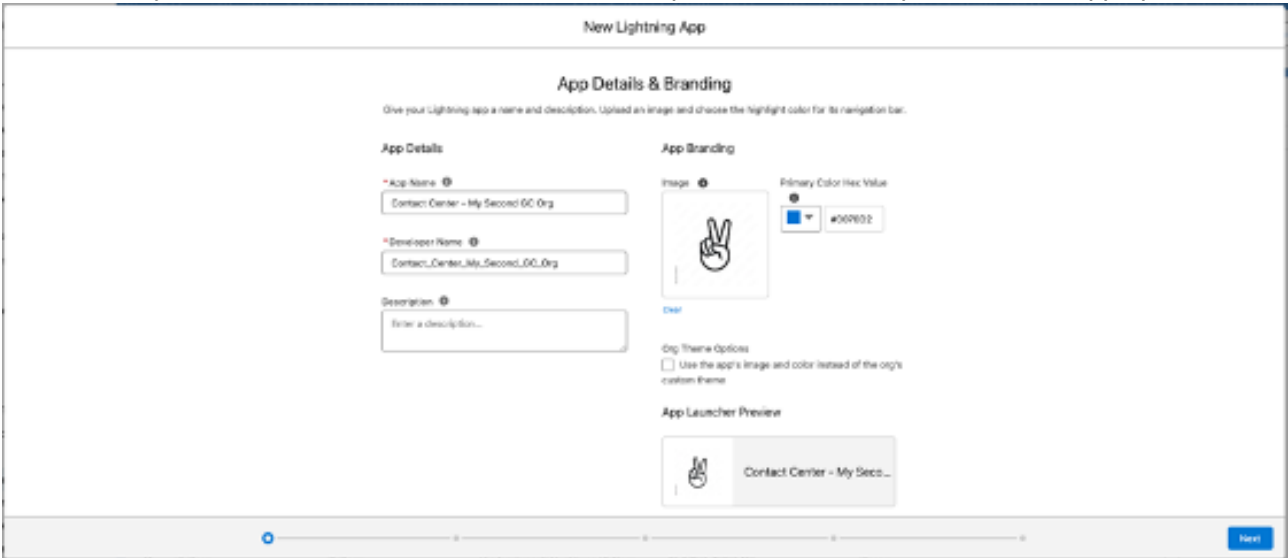
In cases where a single Salesforce Org needs to be integrated with multiple Genesys Cloud Orgs, it can be accomplished by creating separate Lightning Applications for each respective Org in Genesys Cloud. This can be done regardless of whether Salesforce Cloud Voice is used. This section details the steps /



configurations to integrate a single Salesforce Org with more than one Genesys Cloud Org. This section assumes readers have already completed the setup for a single Genesys Cloud Org, and as such omits details previously covered, yet required to complete the setup. Suffice to say, completing the setup for a single Genesys Cloud org is a prerequisite, specifically the Package Install and Post Install steps covered previously in this document. ##### Create a new Lightning App for the Respective Genesys Cloud Org In the Salesforce setup, under the “Apps” section open “App Manager” and click on the New Lightning App button



##### App Details and Branding For the first step, App Details and Branding, choose an appropriate name considering the fact it may be helpful to easily identify the respective Genesys Cloud Org the Lightning App will represent. After you’ve chosen a name and filled in the other options, click Next to proceed to the App Options menu.



##### App Options Under the Navigation Style select “Console navigation” (default options may be accepted for

### App Options

#### Navigation and Form Factor ⓘ

• Navigation Style

- Standard navigation
- Console navigation ←

• Supported Form Factors

- Desktop and phone
- Desktop
- Phone

#### Setup and Personalization ⓘ

Setup Experience

- Setup (full set of Setup options)
- Service Setup

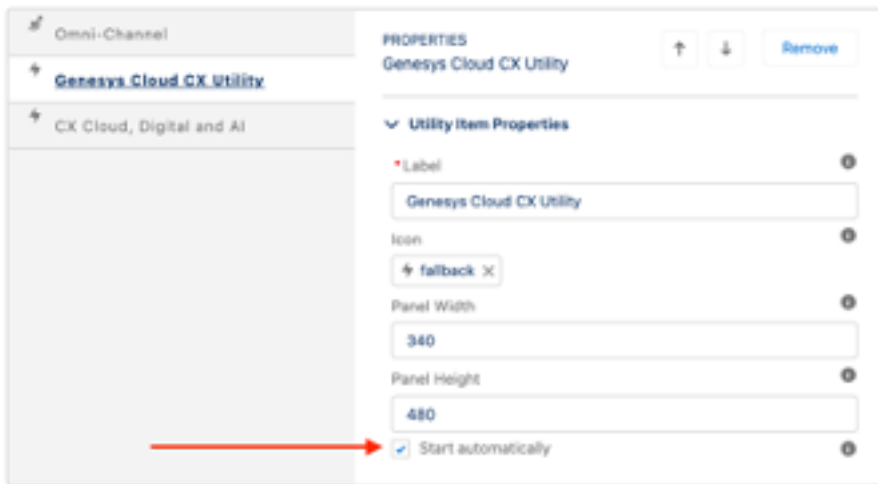
App Personalization Settings

- Disable end user personalization of navigation for this app
- Clear workspace tabs for each new console session

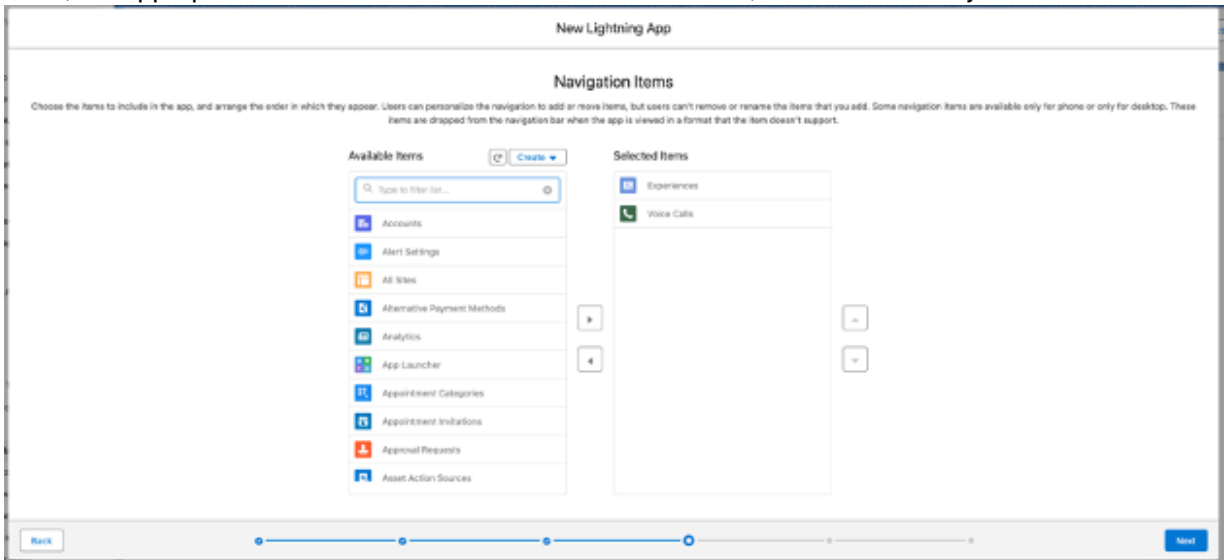
the rest) and click Next.

##### Utility Items On the Utility Items page, add the following:

1. Omni-Channel
2. Genesys Cloud CX Utility
3. CX Cloud, Digital and AI For each, be sure the Start automatically check box is selected, like so:



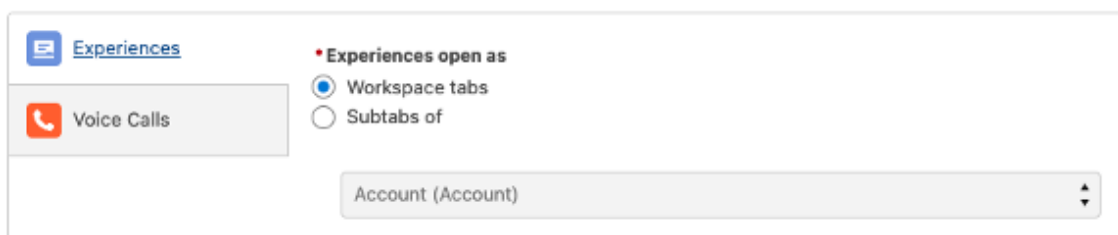
After adding CX Cloud, Digital and AI be sure to fill in the required fields. Note, when selecting whether Salesforce Cloud Voice is installed, it should be considered in the context of the Lightning App being created and the Genesys Cloud Org it reflects, meaning it could be answered negatively even if Salesforce Cloud Voice is being used in elsewhere is the Salesforce Org but not in conjunction with “this” App. Additionally, if you haven’t already created the OAuth integration in Genesys Cloud to create the Implicit Grant Client ID, this can be done following the steps previously outlined [here](#) With the Utility Items added, click Next to move to the Navigation Items menu. ##### Navigation Items On the Navigation Items page add Experiences to the list of selected items. Note, if the application also seves voice, the appropriate item should be added. In the screen shot, this is reflected by the Voice Calls item.



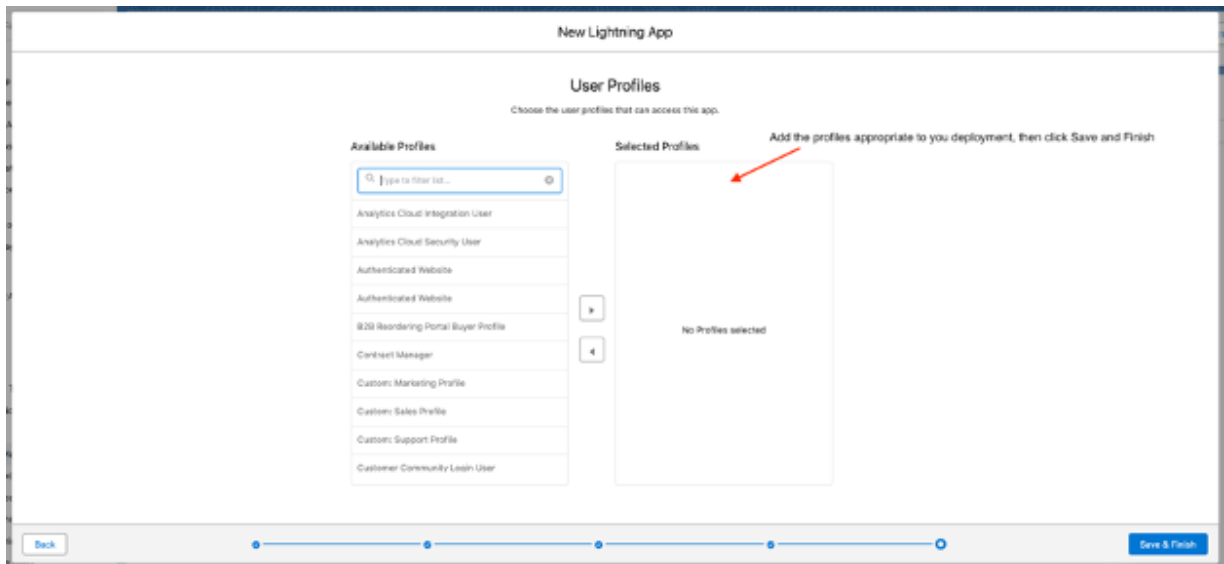
Click Next to bring up the Navigation Rules menu. ##### Navigation Rules Accept the default to have Experiences open as workspace tabs and Click Next to move to the Profiles menu

### Navigation Rules

Navigation rules determine whether to open a related record in addition to the primary record.



##### Profiles The final step to complete before the application is ready is selecting the appropriate Profiles. This will vary by deployment. Once the Profiles have been added, click Save and Finish.



This completes the setup for the new Lightning App. Provided the previous steps for installing the Package and Post Install setup were complete, you will now be able to use the newly created Lightning Application for the Agents affiliated the corresponding Genesys Cloud Org. ##### Summary This approach works by relying on the combination of the Omni Channel widget and other Utility Items configured within the Lightning Application. When Salesforce Cloud Voice is used, because an agent can only belong to a single contact center in Salesforce, when their assigned contact center represents a CX Cloud integration, when an agent opens the Lightning Application, they're automatically associated with the corresponding Genesys Cloud Org. In this manor, we can create Lightning Applications for any number of Genesys Cloud Orgs. In the cases where agents do not belong to a contact center, i.e. Salesforce Cloud Voice is not used, we determine the Genesys Cloud Org based on the CX Cloud, Digital and AI Utility Item configuration. Similarly, we can continue to add new Lightning Application as needed when SCV is not used to accommodate integration to additional Genesys Cloud Orgs.