Agent Actions ISV Guide

Creating and Packaging Agent Actions

Salesforce, Winter ’25

Last updated: November 25, 2024

CONTENTS

Get to Know Agent Actions **4**
What are Agents? 4
The Building Blocks of Agents 5
Agents 5
Topics and Actions 5
Reasoning Engine 5
Large Language Model (LLM) 6
Get Access to Scratch Orgs that have Agentforce **6**
Scratch Orgs with both Agentforce and Data Cloud **7**
Set Up Agentforce in your Scratch Org 8
Get Access in Your 1GP Packaging Org **8**
Create an Agent **8**
Create an Agent Action **9**
Create an Apex Class, Flow, or Prompt Template 9
Invocable Apex Classes 9
Autolaunched Flows 10
Prompt Templates 11
Configure Your Custom Agent Action 11
Test Your Agent Action 13
Package Your Action **13**
Create a Package 13
Add Components to Your Package 13
Upload Your Package 14
Resources **14**

Get to Know Agent Actions

Agent actions are how an agent gets things done. You can create actions from invocable Apex classes, autolaunched flows, or prompt templates.

EDITIONS

Einstein Copilot for Salesforce is available in Lightning Experience.

Available in:
- Enterprise Edition
- Performance Edition
- Unlimited Edition

USER PERMISSIONS

To build and manage agent actions:

- Manage AI Agents AND the user permission for your agent type (for example, Manage Einstein Copilot for Salesforce) OR Customize Application

  For permissions needed to build and manage individual agent types, see the respective Salesforce Help documentation. Get started with *Salesforce Help*: Explore Agent Types.

To create and manage prompt templates in Prompt Builder:

- Prompt Template Manager permission set

For permissions needed to build and manage Apex classes, flows, and managed packages, see the respective Salesforce Help and Developer documentation.

What are Agents?

An agent is a virtual assistant that boosts productivity and reduces workload in a safe way by automating routine tasks and assisting with complex ones. They're capable of helping across a wide range of workflows and tasks.

Some agents handle questions posed by users in natural language and provide answers drawn from secure, proprietary company data. Other agents are more autonomous, so they can independently identify opportunities for action, anticipate next steps, and initiate tasks within the use cases and guardrails an admin specifies.

Agents can collaborate with a user and, in some cases, act on behalf of a user.

The Building Blocks of Agents

Agents have five components: the agent, topics, actions, the reasoning engine, and the large language model (LLM).

Agents

Agents are trusted conversational AI assistants. Einstein Copilot for Salesforce, one type of agent, is seamlessly built into the Salesforce interface and can perform business tasks on behalf of the users in a Salesforce org. Other types of agents can be deployed to customer channels, including enhanced messaging channels.

To learn more about the differences between agents, Einstein Copilot for Salesforce, and bots, see *Salesforce Help:* Choosing a Conversational AI Assistant.

Topics and Actions

An agent includes a library of topics and actions.

Actions are how an agent gets things done. An action is an individual task the agent can perform. For example, if a user asks an agent for help with writing an email, the agent can launch an action that drafts and revises the email and grounds it in relevant Salesforce data. Salesforce provides some standard actions for common Salesforce tasks by default, and you can create custom actions specific to your business use cases.

A topic is a category of actions related to a particular job to be done. For example, a topic called Deal Management can contain agent actions that help a sales rep get up to speed on their day, find relevant opportunities and contacts, create to-do items, and log calls. Topics contain actions, which are the tools available for the job, and instructions, which tell the action how to make decisions.

When an agent is triggered, or when a user enters a question or request in an agent conversation, an agent compares the task or user utterance to the names and classification descriptions of all of the topics assigned to it. The agent then classifies the task or utterance into the most relevant topic. Based on the selected topic's actions and instructions, the agent can launch one or more actions. Or if the agent is chatting with a user, the agent can prompt the user for additional information (for example, a clarifying question or the required input for an action).

Reasoning Engine

The reasoning engine orchestrates how an agent launches topics and actions to accomplish a task for the user.

When an agent is triggered or when a user asks a question or makes a request, the reasoning engine works with the LLM behind the scenes. Here's what it does.

- Interprets the trigger or user request and classifies the utterance into a topic.

- Iteratively builds plans for accomplishing a goal.
- Finds and launches the right topics and actions to achieve the goal.

Large Language Model (LLM)

Agents are AI assistants, and they harness the power of an LLM to communicate with users and take action.

The reasoning engine calls the LLM at different times during a task or interaction with a user. The number and size of the LLM calls depends on the task and which topics and actions are launched.

Get Access to Scratch Orgs that have Agentforce

If you don't already have a Partner Business Org (PBO), join the Salesforce Partner Community and request a PBO.

If you are new to creating scratch orgs, follow these steps to complete the one-time Dev Hub setup in your PBO. The Dev Hub is a feature within an org that lets you create and manage scratch orgs, second-generation managed packages (2GP), and namespaces.

1. Enable Dev Hub and 2GP
2. Create a Developer Edition org using Environment Hub
3. Create a namespace in this Developer Edition org
4. Link that namespace from your PBO. Linking the namespace lets you create 2GP packages that use that namespace.

As of September 16, 2024, all active Partner PBOs have the ability to create scratch orgs with Agentforce and Prompt Builder enabled. Agentforce and Prompt Builder can ***only*** be enabled in scratch orgs that are associated with a Dev Hub in a PBO.

Partners with active PBO orgs	Enabled as of September 16, 2024
New PBO orgs (Trial or Active)	Automatically enabled when created

To create a scratch org with Agentforce and Prompt Builder enabled, use the following sample `project-scratch-def.json` file (or simply append the feature and setting below to your existing scratch org definition file).

```
Unset
{
  "orgName": "GenAI Scratch Org",
```

```
  "edition": "Partner Developer",
  "features": ["Einstein1AIPlatform"],
  "settings": {
    "einsteinGptSettings" : {
      "enableEinsteinGptPlatform" : true
    }
  }
}
```

To create a scratch org with the Einstein1AIPlatform feature, the scratch org you create must be a Partner Developer edition.

To create a scratch org, run the following command in the Salesforce CLI. Update the `definition-file` name, `alias`, and `target-dev-hub` alias as needed.

```
Unset
sf org create scratch --definition-file
config/my-agentforce-project-scratch-def.json --alias MyScratchOrg
--set-default --target-dev-hub MyHub
```

Scratch Orgs with both Agentforce and Data Cloud

For some use cases such as prompt templates that use RAG, Retrievers, and BYO LLM, a scratch org that has both generative AI and Data Cloud functionality enabled might be required.

Only include Data Cloud if it's required. Specifying Data Cloud in a scratch org significantly increases the time it takes for a scratch org creation to complete.

Note: Including Data Cloud in a scratch org has a prerequisite. You must first open a case in the Salesforce Partner Community to request for your PBO Dev Hub org to be granted permission to create Data Cloud scratch orgs. This request is only granted to PBO orgs.

```
Unset
{
  "orgName": "GenAI & Data Cloud Scratch Org",
  "edition": "Partner Developer",
  "features": ["CustomerDataPlatform",
"CustomerDataPlatformLite","Einstein1AIPlatform"],
```

```
  "settings": {
    "einsteinGptSettings" : {
      "enableEinsteinGptPlatform" : true
    }
    "customerDataPlatformSettings": {
      "enableCustomerDataPlatform": true
    }
  }
}
```

Set Up Agentforce in your Scratch Org

Once your scratch org is created, follow these steps to start developing with Agentforce.

1. Enable Agents and Einstein Copilot for Salesforce.
2. Assign User Permissions.
3. To use prompt templates with your agent actions, assign Prompt Template permissions.

Get Access in Your 1GP Packaging Org

To enable Agentforce Extensibility for ISVs on a 1GP packaging org, you must log a case with Salesforce Partner Support. In the case details, list your packaging org ID and request Agentforce Extensibility for ISVs be provisioned on your development org. Only 1GP development orgs created from Environment Hub can have Agentforce provisioned.

To add Agent Actions to a 1GP package:

1. From the Package Manager page in Setup, click on the name of your package.
2. Select **Add**, and then in the **Component Type** dropdown list, select **Generative AI Function Definition**.

Create an Agent

When you enable the Agentforce platform, an Einstein Copilot for Salesforce agent type is created for you automatically. However, you should plan and test your actions with the agent type you plan to package actions for. In addition to Einstein Copilot for Salesforce, you can create the following agent types, depending on your licenses and permissions:

- Agentforce Service Agent

- Agentforce Sales Coach
- Agentforce Sales Development Rep (SDR)

To learn more about how to create an agent from a type and available agent types in Salesforce, see *Salesforce Help*: Create Agents and Einstein Copilot for Salesforce. To learn more about the differences between agents, Einstein Copilot for Salesforce, and bots, see *Salesforce Help:* Choosing a Conversational AI Assistant.

Create an Agent Action

Before you begin:

- Review the considerations for custom agent actions.
- Review best practices for writing agent instructions.
- If you haven't already, create a custom topic.
  To assign an agent action to an agent, add the action to a topic. Your action must be assigned to an agent to test it. Learn more about agent topics and best practices for writing topic instructions in Salesforce Help. Packaging of topics isn't currently supported.

User access to a custom agent action depends on the type of Salesforce action it references. For example, if a custom action was built using a flow, the custom action adheres to the permissions, field-level security, and sharing settings configured in the flow.

Create an Apex Class, Flow, or Prompt Template

When you create a custom agent action, you build it on top of platform functionality you want to make available to agents. This platform functionality is called a reference action. You can create a reference action from an invocable global Apex class, autolaunched flow, or prompt template. Packaging of prompt templates isn't currently supported.

Invocable Apex Classes

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on the Salesforce Platform server, in conjunction with calls to the API. If you're new to Apex, learn more in the Apex Developer Guide.

Invocable Apex classes can be called by a declarative tool or external system, such as Rest, Apex, Flow, an Einstein bot, or an agent. Use the `InvocableMethod` annotation to identify methods that can be run as invocable actions. Mark your Apex class as global.

You can call an external service API endpoint in an Apex class to add an external service to an agent action.

Here's an example of an Apex class used to create an agent action.

```
global class SimpleInvocableClass {
```

```
    @InvocableMethod(label='Simple Invocable Method')
/*
If you want to call Prompt Template from the apex class, make sure that Prompt template API name matches exactly
  @InvocableMethod(CapabilityType ='FlexTemplate://PromptTemplateName')
 */
     global static void simpleMethod(List<String> inputs) {
        // Perform some operations with the inputs
        for(String input : inputs) {
            System.debug('Input: ' + input);
            // Perform additional logic here
        }
    }
}
```

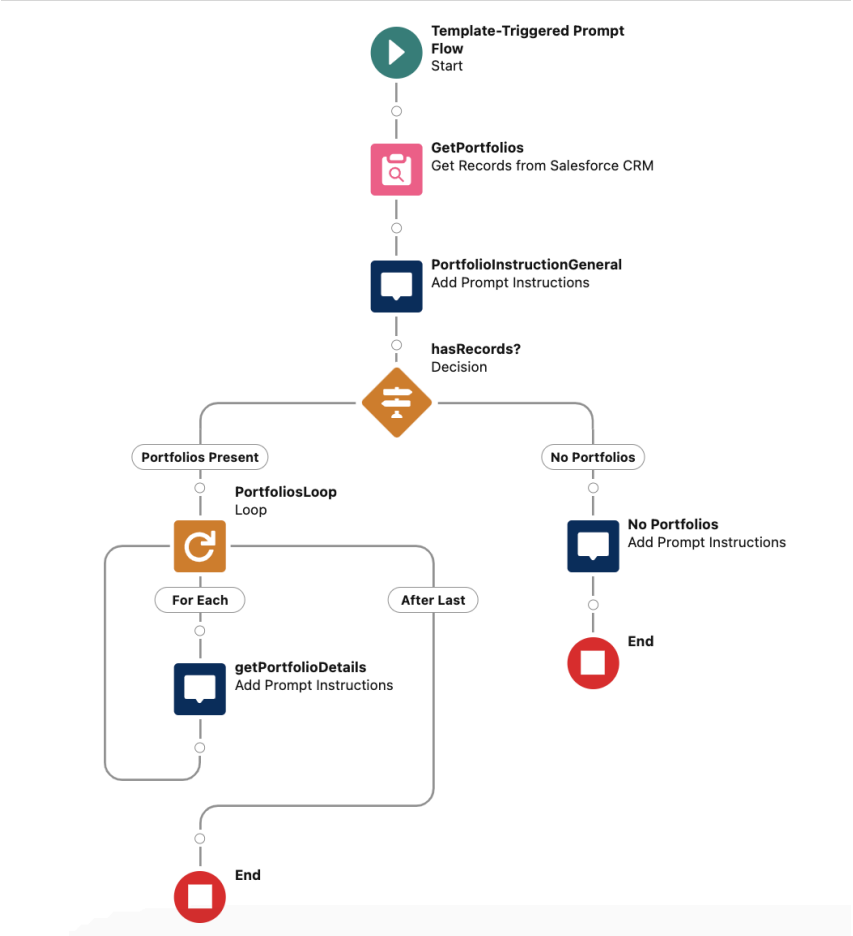
To create or manage Apex classes, from Setup, in the Quick Find box, enter `Apex Classes`, and then select **Apex Classes**.

Autolaunched Flows

Flows let you create simple or complex automations with little to no code. If you're new to flows and Flow Builder, see *Salesforce Help*: Automate Tasks with Flows to learn more.

A flow or flow version's type determines which elements and resources you can add to a flow and how you can distribute the flow. Agents and Copilot support autolaunched flows, which is a flow that is automatically launched by something, such as a change to a record or a platform event. It runs in the background and doesn't require user interaction. This type of flow doesn't support screens and local actions. To learn more about autolaunched flows, see the Salesforce Admins blog.

Here's an example of a flow used to create an agent action.



To create or manage flows, from Setup, in the Quick Find box, enter `Flows`, and then select Flows.

Prompt Templates

Use Prompt Builder to create, test, revise, customize, and manage prompt templates that incorporate CRM data from merge fields that reference record fields, flows, related lists, and Apex. You can create different types of prompt templates in Prompt Builder. If you're new to prompt templates and Prompt Builder, learn more in *Salesforce Help*: Prompt Builder.

To create or manage prompt templates, from Setup, in the Quick Find Box, enter `Prompt Builder`, and then select **Prompt Builder**.

Packaging of prompt templates isn't currently supported. However, packaging support is coming soon. Prompt Builder is available in scratch orgs, and we encourage you to experiment.

Configure Your Custom Agent Action

1. From Setup, in the Quick Find box, enter `Agent`, and then select **Agent Actions** to view all the actions in your Salesforce org.
2. Click **New Agent Action**.
3. From the Reference Action Type dropdown, select the type of action that you want to use to build your agent action. Then select the action. If you're adding a flow or a prompt template, only active flows and prompt templates appear in the list.
4. The agent action label and API name are populated with the reference action name and API name. Review these fields and, if necessary, edit the values so they accurately describe the action. Then click **Next**.
5. The instructions for the custom action and each input and output are populated with the descriptions from the reference action. Review these fields and make changes. See *Salesforce Help*: Best Practices for Einstein Agent Action Instructions.

   We copy over the descriptions from the reference action to give you a head start, but agent action instructions differ from traditional descriptions. An agent uses a large language model (LLM) and the reasoning engine to determine when to launch an action in a conversation. The instructions for the action, inputs, and outputs tell an agent what your action does and when and how to use it. Effective instructions vary by action and use case. Plan to test and iterate on your instructions to ensure that your action performs as expected.
6. For each input and output, specify any applicable settings.

Setting	Details
Collect data from user	Require this parameter to be provided by the user.
Filter from agent action	When you create an agent action, all inputs and outputs from the reference action are added to the agent action. Enable this setting when you don't want to use a parameter from the reference action with your agent action. At least one output must be used with the agent action.
Require input	Require this parameter in order to execute the action. All other inputs are treated as optional. If an input is required by the reference action, this setting is enabled by default and read-only.
Show in conversation	Allow an agent to include this parameter in a response to a user. At least one output must be available to show in an agent conversation.


Tip: When an action makes a change to a record, you can require an agent to ask the user to confirm the change before the agent can execute it. On the record page for the action, enable the **Require user confirmation** setting.

7. Click **Finish**.

Test Your Agent Action

After you create your action, add the action to a topic to assign it to an agent. Your action must be assigned to an agent to test it. Learn more about agent topics and best practices for writing topic instructions in Salesforce Help.

To test your action and preview how the output appears in an agent conversation, open an agent in the Agent Builder and start a preview conversation. Enter utterances that you expect to trigger your action, and then adjust the agent action and topic instructions based on your results.


Tip: The packaging of topics isn't currently supported. Keep in mind that the instructions for whatever topic your users assign your action or actions to will interact with your action instructions. Plan to test with many variations.

If your action isn't performing as expected, review these common troubleshooting tips.

Package Your Action

Currently, custom actions can be packaged using first-generation managed packaging. Support for second-generation managed packaging is coming soon.

Before you begin, verify that you've completed the steps in Get Access to Scratch Orgs that have Agentforce.

Create a Package

1. From Setup, in the Quick Find box, enter `Package Manager`, and then select **Package Manager**.
2. Click **New**.
3. Enter a name for your package. You can use a different name than what appears on AppExchange.
4. Agents currently support English only, so leave the Language dropdown set to English.
5. Optionally, enter a description that describes the package. You can change this description before you upload it to AppExchange.
6. Click **Save**.

Add Components to Your Package

1. From the Components tab, click **Add**.
2. From the Component Type field, choose the type of component you want to add to your package. You must add the agent action and the Apex class or flow that your action references. For the agent action, select the **Generative AI Function Definition** component type.
3. Select the component you want to add, and then click **Add to Package**.

Repeat these steps until you add all of the components you want to your package.

Upload Your Package

1. From the Package detail page, click **Upload**.
2. On the Upload Package page, enter a Version Name. As a best practice, it's useful to have a short description and the date.
3. Enter a Version Number for the upload, such as `1.0`. The format is `majorNumber.minorNumber`. This field is required for managed packages and corresponds to a Managed - Release upload.
4. For the Release Type, select **Managed - Released** to upload an upgradable version. After upload, some attributes of the metadata components are locked.
5. Update your package's description, if necessary.
6. Optionally, specify a link to release notes for the package. Click **URL** and enter the details in the text field that appears. This link appears during the installation process, and on the Package Details page after installation.
7. Click **Upload**.

When you create a managed package to publish on AppExchange, you must pass the AppExchange security review. Then create a listing with the Partner Action listing type and publish your managed package on AppExchange.

Resources

Trailhead: Quick Start: Build Your First Agent with Agentforce
Salesforce Help: Agentforce: Agents and Copilot
Salesforce Help: The Building Blocks of Agents and Copilot
Salesforce Help: Customize Your Agents and Copilot with Topics and Actions
Salesforce Help: AI Project Success
Salesforce Help: Considerations for Agents and Copilot
First-Generation Managed Packaging Developer Guide
Second-Generation Managed Packaging Developer Guide
Scratch Orgs and Package Development