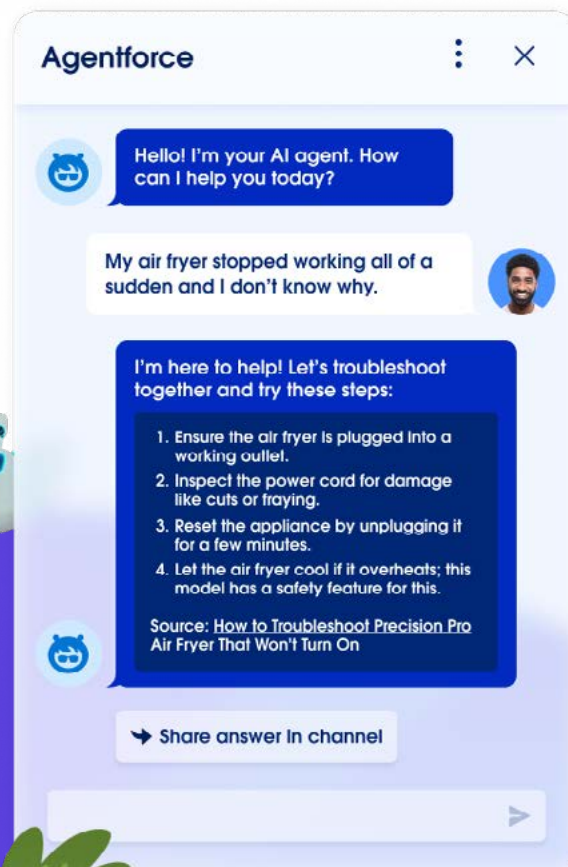




How to Get Started with Agentforce

A step by step implementation guide for creating an autonomous AI service agent.





Welcome to Agentforce! Let's get started.

With Agentforce, humans and AI agents drive customer success together. Agentforce agents can analyze data, make decisions, and take action on tasks like answering customer service inquiries, qualifying sales leads, and optimizing marketing campaigns. With Agentforce, organizations can easily build, customize, and deploy their own agents for any use case, across any industry.

This guide explains the process for configuring and troubleshooting an Agentforce Service Agent.

To do so, we will:

- Define the key building blocks of an AI service agent
- Walk you through the steps in building a service agent
- Explain how to monitor and iterate on agent responses
- Summarize a handful of best practices and sticking points



For the latest information about setting up an Agentforce Service Agent, see [Salesforce Help: Agentforce Service Agent](#). For more information about AI agents, topics, and actions, see [Salesforce Help: Agentforce: Agents and Copilot](#).

The Building Blocks of an Agentforce Service Agent

An AI service agent is made up of two fundamental building blocks: topics and actions.

Topics are the jobs you want the agent to be able to do. You can think of a topic as a job that can be achieved with a consistent set of tools (actions) and rules (instructions) for how to do it. Topics are composed of these key attributes:

- **Classification** describes what your topic does and the types of user requests that should be classified into this topic. The agent uses this description to determine when to use your topic in a conversation. You should write this description in a way that mirrors how customers ask the questions you want the agent to answer.
- **Scope** defines the boundaries of the job – what the agent can and cannot do. When you write this, think of it as a “job description” that will constrain what the agent can do within this topic.
- **Instructions** provide context and guidelines on how to do the job. These can include clarifying common terms, which actions to use, when and how to use them, or how to formulate responses.



The Building Blocks of an Agentforce Service Agent

Here are some examples of topic compositions:

Topic	Order Management	Login and Access Issues
Classification Description	Answer questions related to orders, returns, past purchases, shipping status, or repairs.	Help customers with questions about issues accessing the site, challenges logging in, questions about login failing, or finding their username or password.
Scope	Your job is only to answer questions about a customer's order status, return status, or return and repair policy. Never initiate or generate an order or return.	Your job is only to help customers who cannot log in, by resetting their password or looking up their username.
Instructions	If a customer asks for the status of an order or return, always offer all options to look up the status using either their email address, order date, or the order ID.	Always clarify what type of device (iOS or Android) the customer is using before using the "Answer Questions with Knowledge" action to retrieve troubleshooting information, and always include the device type in the search query for knowledge. Promo codes are used for discounts during checkout, while registration codes are used to register and gain access to loyalty discounts and early access to new products.



The Building Blocks of an Agentforce Service Agent

Actions are the tools that the Service Agent uses to do a job.

Examples of standard actions available with Agentforce include:

Action	Description
Get Case	Retrieve the information from a specified case
Create a case	Create a net new case
Password Reset	Replace an existing password with a new password
Update Contact	Replace contact information, like name, phone number, email, etc.
Get Reservation	Retrieve reservation details

You can extend actions using:

- **Flows:** Workflows to automate business processes and collect data in Salesforce
- **Apex:** Coding language for developers to build custom features for the Salesforce platform
- **Prompt Templates:** Reusable prompts to generate AI-driven responses based on specific inputs, use cases, and requirements



Building a Service Agent: Step by Step

In Agentforce, the steps to build a service agent follow a standard path. The key is to start simple. Then you can iteratively build, test, and repeat until the agent behaves the way you expect.

- 01** Review the “Reasoning” step to determine if an issue with the instructions in the topic led to the error.
- 02** Click the Knowledge tab. Select a library for your agent. (Or, to create a library, click New Library.) If you select or create a library, you can configure the knowledge fields or file uploads that you want your data library to index. To base agent responses on all Knowledge articles and fields, select All Knowledge Records and Fields. Save your changes.
- 03** Start with standard topics from the Service Agent templates. These are for common use cases like case management, account management, order inquiry, and more. Standard topics include basic instructions and actions you can use as a baseline, which you can then extend and customize to suit your particular needs.
- 04** From a standard topic, you can:
 - Click on New Version
 - Customize all fields on the topic
 - Add new actions
- 05** From the standard actions in that topic:
 - Click through to the underlying flow, edit it, and save a new version
 - The agent action will run the newest saved and active version of the flow
 - Click New Version for any action, then edit the action name and description
- 06** Can’t find a standard topic template for your use case? You can create a new topic for the job you want the agent to do.
 - Start simple: Begin with high-level topics like “Order Management” or “Product Troubleshooting”
 - If you need two sets of instructions to handle different jobs, you can easily split any topic into two or more topics



Building a Service Agent: Step by Step

- 07 Provide a detailed Classification Description. For example, what types of requests should this topic be used to address?
- 08 Add Scope text. For example, what are the boundaries or the job description of this topic?
- 09 **DO NOT immediately draft a large amount of instructions.** The best practice is to start with a minimal set of basic instructions, ideally just one, and iteratively build them up as you simulate conversations. **Remember: Less is more.**
- 10 Add actions to the topic so the agent has the tools to complete the use case, such as sending a password reset link, doing a customer information lookup, doing an order lookup, and so on.
- 11 Once you've added some basic actions, you should have a topic that clearly defines the job to be done and contains the required actions to do that job. Now it's time to see how it works.
- 12 The Planner is the canvas within Agent Builder where you can test and monitor your agent's reasoning and its responses. You can simulate conversations with your service agent in the Planner's preview panel by:
 - Asking a question or sending a message you expect to trigger that topic
 - Stepping through the conversation in the same way you expect customers will interact with the agent
- 13 If the service agent does not respond how you want it to, iterate on the configuration of your topic by adjusting its classification description, scope, or instructions. (Details on this process are in the next chapter.)
- 14 Clear the conversation preview and re-test until your service agent works as desired.



Fine-Tuning Your Agent

When a service agent is not behaving the way you want, how do you know which lever to pull to get it on the right track?

The key is to analyze the agent's responses to your prompts in the Planner canvas, understand why the service agent is responding the way it is, and adjust accordingly.



1

Problem 1: Service Agent Picks the Wrong Topic

This issue occurs when the service agent does not choose the right topic for a customer question. You can diagnose it by checking the “Select Topic” box in the plan canvas.

To fix this topic, adjust the Classification Description on the agent’s topic(s) to better capture the types of customer statements that should invoke each topic. Adding key terms, examples, or common questions that should trigger this topic often helps.

Select Topic

General_FAQ

Questions about Cirrus policies, cost, and other information on solar panels, turbine blades, procedures, troubleshooting solar panels, troubleshooting turbines, searching help articles, and answering any general questions about Cirrus.

> Instructions 2

> Actions 1

Check the “Classify Topic” box in the Planner canvas to confirm that the agent chooses the right topic for a customer question.



2

Problem 2: Service Agent Picks the Wrong Action

This issue occurs when the service agent does not invoke the right action when you want it to. You can diagnose this in the Planner canvas by confirming that, even though it picked the correct topic, it didn't run the expected action.

There are two potential levers to address this issue.

- First, make sure the description of your actions accurately reflects what the action is and when it should be used.
- Second, add instructions to the relevant topic to guide the agent on when to use each action, or provide it with additional context that can help it interpret common customer statements (context) that will help in choosing the right action.

When adding an instruction that expresses some sort of condition (“if this, do that” or “after doing this, then do that”), be as specific as possible.

Keep in mind that instructions rely on the large language model (LLM) for interpretation, and the LLM lacks knowledge about your specific use case. Any logic unique to your business, like whether or not a refund should be issued, should be baked into an action in a declarative way. For example, a flow-based action can assess the criteria for issuing the refund rather than relying on an LLM to make this decision..



Identify Record by Name

Searches for Salesforce CRM records by name and returns a list of matching Salesforce CRM record IDs.

Input

```
{
  "recordName": "acme"
}
```

Output

```
{
  "id": "006xx0000001a2obAAA",
  "data": {
    "Type": "Opportunity",
    "Id": "006xx0000001a2obAAA",
    "Name": "Acme - 200 Widgets"
  }
}
```

Add an instruction to invoke the right action when you want it to.

3

Problem 3: Action Output Is Incorrect or Problematic

In this scenario, the service agent has invoked the correct action, but that action is not returning the expected output for a given input. Solving this problem requires troubleshooting the action itself. If the action is based on Flow or Apex, you need to evaluate the Flow or Apex for any issues.

If the issue is with the Knowledge Action, consider the following options:

- Check the query that the service agent passed to the knowledge action. Does the query provide enough context that it is reasonable to expect the action to find the right answer? If not, consider adding instructions on how to form effective search queries for that topic.
- Is the action returning irrelevant knowledge? If so, revisit the Knowledge setup for this agent to be sure only the relevant fields and records are available to the agent.
- Is the action not finding the right article, or is it finding the right article but not extracting the right information from it? In either case, you may need to explore adjusting your articles to ensure the answers to common questions are clear in your knowledge base. Q&A articles with non-overlapping areas of focus and clear titles work the best.



Select Action

18.95 sec



Answer Questions with Knowledge

Answers questions about company policies and procedures, troubleshooting steps, or product information. For example: "What is your return policy?" "How do I fix an issue?" or "What features does a product have?"

Input

```
{
  "query": "how to troubleshoot a
solar panel"
}
```

Output

```
{
  "knowledgeSummary": "To
troubleshoot a solar panel, follow these
steps:\n\n1. **Inspect All Wiring**
Check"
```

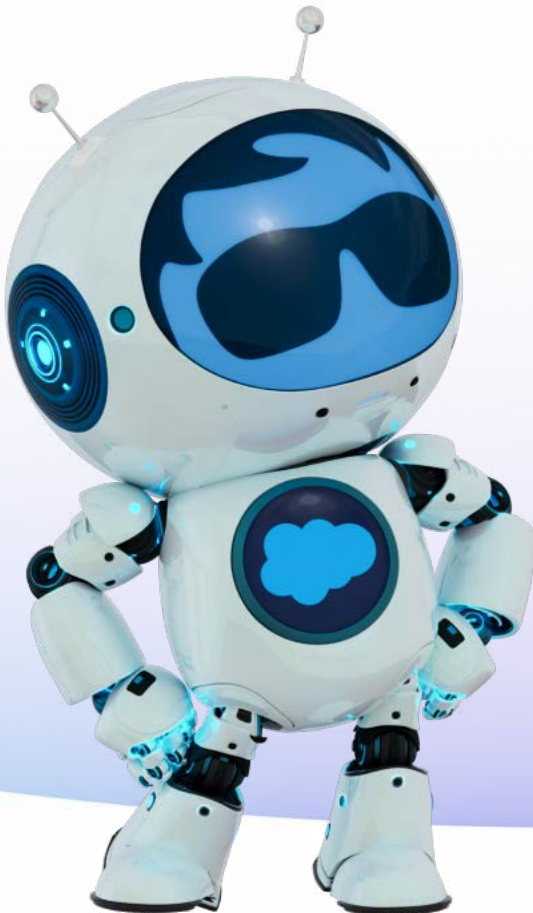
Check the Knowledge setup to troubleshoot the action.

4

Problem 4: Right Topic, Right Action, Bad Response

In this scenario, the service agent chose the right topic and ran the right action – but the end response to the customer did not meet your expectations. For example, the answer to a customer may correctly share information from an order status lookup action, but not in the tone or format you want to use with customers.

Adding instructions to your service agent on how to format customer replies, and what style and tone to use, can solve these issues quickly and easily.



5 Problem 5: Something Went Wrong

In this scenario, the service agent returned an error message that said:

"I'm sorry, I can't help with that."

An AI service agent is made up of two fundamental building blocks: topics and actions:

1. Conflicting or confusing instructions lead the agent to be unsure how to respond.
2. A legitimate bug/error.

A good place to start troubleshooting this issue is to review the Reasoning Step in the Planner canvas. The LLM validates each agent response before sending to the customer to ensure it does not conflict with any instructions and that it provides only information from the available instructions and actions. The Reasoning Step can help reveal if an issue with the instructions in the topic led to the error.

This is often the toughest problem to troubleshoot. In a worst-case scenario, you may need to remove instructions from the topic and slowly add them back in until you can figure out the root cause of the confusion..

Reasoning

GROUNDING: The response accurately reflects the data retrieved from the function call, listing the records related to "Acme" and their details.

Review the "Reasoning" step to determine if an issue with the instructions in the topic led to the error.





Additional Tips, Tricks, and Best Practices

Here are some final best practices to keep in mind:

- **Be careful with absolutes in instructions.** In Agentforce, service agents take “always” or “never” literally.
- **A winning strategy for Q&A** with the knowledge action is to add instructions to ask clarifying questions to narrow the search space, then include the answers to those questions in the search query.
- **Do not rely on the visual order of instructions in the user interface.** Be explicit in your instructions about the conditions that will exist when something should happen. For example, an instruction that reads, “After that, do XYZ” will not work well. Instead, write: “After searching knowledge, do XYZ.”
- **Treat your agent as a complete novice.** Do not expect that it knows anything beyond what it can retrieve via an action or has available to it in your topic configuration. Define terms in instructions and use those terms consistently. Providing context via instructions can improve performance dramatically.
- **If all else fails and your agent isn’t working the way you’d like, strip out instructions.** Troubleshooting often begins with reducing the number and complexity of instructions to reach consistent baseline functionality.

Here are some final best practices to keep in mind:

- [Trailhead: Build Your First Agent with Agentforce](#)
- [See What Agentforce Can Do for You](#)





The information provided in this report is strictly for the convenience of our customers and is for general informational purposes only. Publication by Salesforce, Inc. does not constitute an endorsement. Salesforce.com does not warrant the accuracy or completeness of any information, text, graphics, links, or other items contained within this guide. Salesforce.com does not guarantee you will achieve any specific results if you follow any advice in the report. It may be advisable for you to consult with a professional such as a lawyer, accountant, architect, business advisor, or professional engineer to get specific advice that applies to your specific situation.

© Copyright 2024, Salesforce, Inc. All rights reserved.