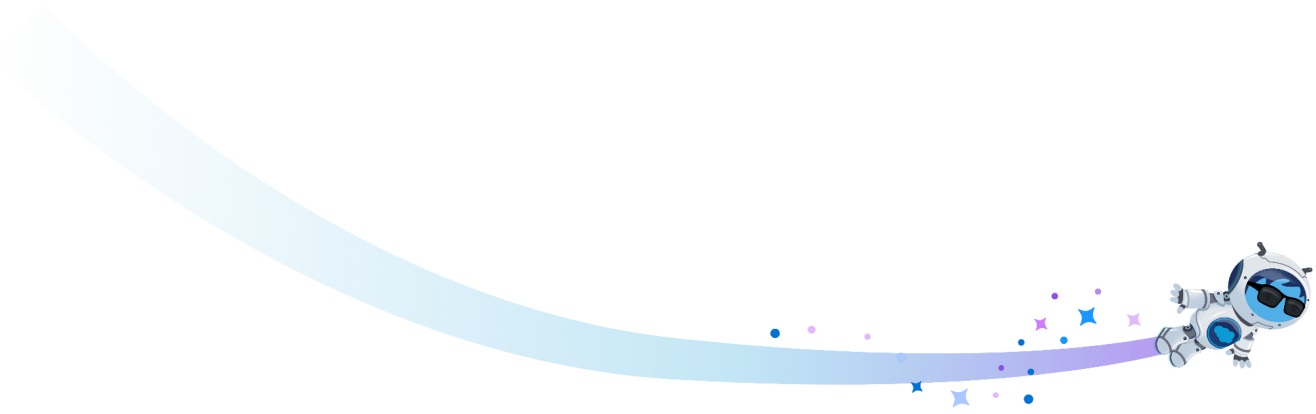
Agentforce Implementation Guide

Customer-Facing Agents



Last Updated: October 17th, 2025

Table of Contents

Get Started with This Guide

In this guide, you'll learn how to launch a customer-facing agent using the Agentforce Service Agent template. This guide walks you through the process of creating an agent by following the five major stages of agent development: Ideate, Build, Test, Deploy, and Monitor. You can use this guide for a variety of solutions:

Goal	Instructions
I just want to build and launch an agent fast.	Skip the Ideate chapter and use the Build, Test, and Deploy chapters to launch an agent. After launch, you can review the Monitor chapter to refine your agent.
I'm new to building agents.	Walk through the entire guide in order.
I need help building out a continuous development cycle.	Review the Test and Monitor chapters in order to refine your agent.

What are Agents?

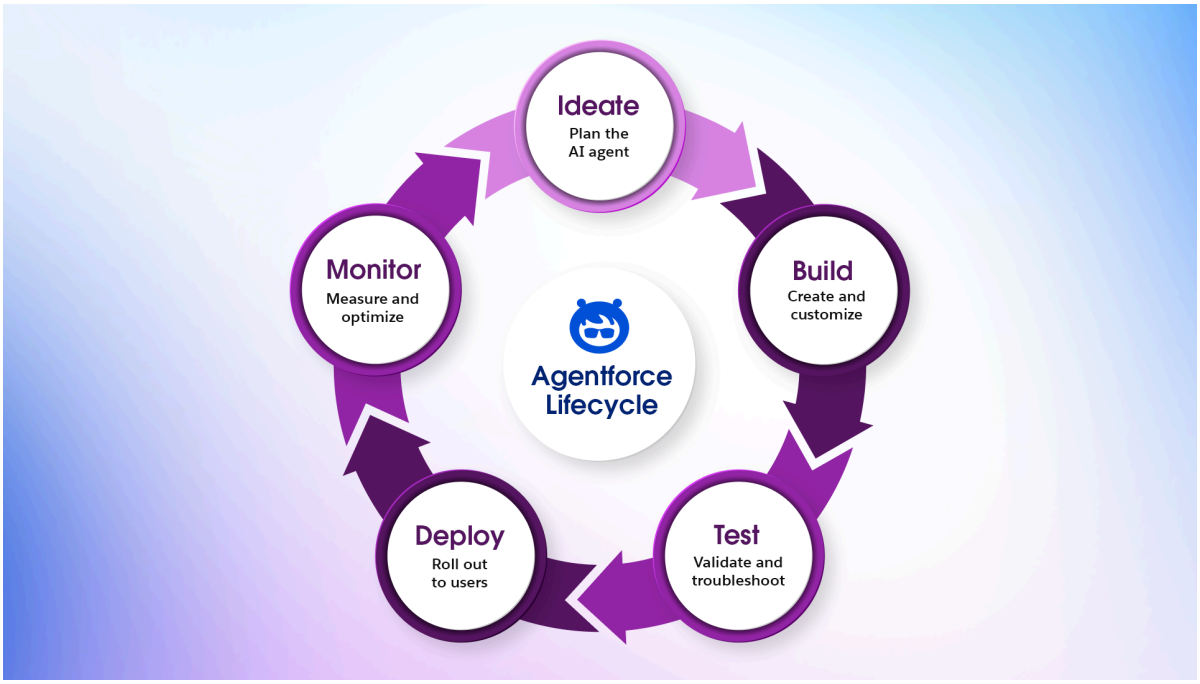
Agents are goal-oriented, autonomous AI assistants that perform tasks and business interactions. They can initiate and complete a sequence of tasks, handle natural language conversations, and securely provide relevant answers drawn from business data.

Some agent types are best at assisting and collaborating with a Salesforce user in the flow of work. Other types can also act on behalf of a user or customer, based on the use cases and guardrails an admin specifies. Some agent types are available in your Salesforce interface, and you can add some to your customer channels.

Agents respect standard Salesforce access controls such as licenses and permissions, so they always act securely. Agents also include AI guardrails designed to help meet ethical standards and are integrated with the Einstein Trust Layer, a secure AI architecture natively built into the Salesforce Platform.

The Agent Development Lifecycle

The development lifecycle for an AI agent generally follows five key stages: ideate, build, test, deploy, and monitor. Here's an overview of what happens during each stage.



- **Ideate**: Define the agent's business value use case, scope, and required capabilities. Address concerns and risks upfront by integrating risk management into the planning process to ensure the agent meets ethical, legal, regulatory, and security requirements.
- **Build**: Create the agent, customize it, and add the necessary data sources.
- **Test**: Validate the agent's performance, accuracy, and adherence to security and compliance standards. Troubleshoot and resolve any issues.
- **Deploy**: Configure the agent's connections and make the agent available to users on the right channels.
- **Monitor**: Measure the agent's performance and health, collecting feedback and data to identify areas for optimization and continuous improvement.

This guide walks you through each stage of the agent development lifecycle.

A Real World Example

In this guide we follow a fictional company, Makana Medical Devices, as they work step-by-step to implement an Agentforce Service Agent in their organization.

Makana Medical Devices is a medical device company that develops, manufactures, and distributes a wide range of diagnostic, treatment, and patient monitoring instruments. They've already integrated generative AI into their Salesforce ecosystem and are now looking to build their first agent with Agentforce to enhance customer support.

Sign Up for a Salesforce Org

To follow along with the steps in this guide, you can sign up for a free Salesforce Developer Edition org.

1. Go to the Developer Edition signup page.
2. Fill out the form and submit it.
3. Look for a welcome email from Salesforce in your inbox.
4. Verify your account using the verification link in the email.
5. When prompted, set a new password and select a security question for your Developer Edition org. Then complete the login process.
6. Keep track of your username and password so that you can log in again later from the Salesforce login page.

When you're done signing up for a Salesforce org, you can start the agent development process.

Ideate Your Agentforce Solution

The first step in developing an AI agent is ideating your Agentforce solution. Planning an AI agent is crucial for several reasons. It ensures that your AI project aligns with your company's overall AI strategy and business goals. It also helps you define the agent's work and assess its potential impact on the organization.

Addressing concerns and risks upfront by integrating risk management into the planning stage can accelerate projects and ensure they meet ethical, legal, regulatory, and security requirements. Without proper planning, an agentic AI project might fail to deploy or be scrapped entirely.

When planning an agent, here are the things to consider:

- Use case definition and scope
- Data and technical requirements
- Risks and guardrails
- Business processes and agent design

Although careful planning is important, try to strike a balance between experimentation and planning. When you're getting started with Agentforce, hands-on learning and prototyping are essential for understanding AI agents and testing out ideas.

In this chapter, we walk through an ideation exercise for Makana Medical Devices. If you want to get straight into creating an agent, skip ahead to the Build chapter.

Identify Your Use Case

An Agentforce use case is an application of AI technology where an AI agent takes an action or series of actions that accomplish a goal or job-to-be-done on behalf of your employees, customers, or organization.

As you begin collecting ideas for Agentforce use cases – whether through crowdsourcing, business process review, user research, or market analysis – you might notice that some ideas lend themselves well to autonomous AI applications, and some don't. When evaluating potential use cases for autonomous AI, consider the following questions:

- **Value**: Why are you delegating this work to an AI agent? Is the agent faster or more accurate? Will it provide a better experience?
- **Work**: Can you describe the work the AI agent will do? Do you fully understand all the business processes involved in that work?
- **Decision-making**: Does the work involve decisions and steps that can be completed without direct human input or judgment? Are there well-defined policies, rules, and constraints that the AI agent can follow alone?
- **Risk**: Can the AI agent operate within the security, legal, ethical, and regulatory requirements that apply to the work?
- **Data**: Is the data capable of supporting the work the AI agent will do?

As you decide on your answers, it's important to understand what agents can do. Often the easiest way to get started is by building an agent and seeing how its capabilities match with the goals you have for implementation.

Makana's Business Challenge

Makana's service team receives a high volume of repetitive inquiries about product information, troubleshooting, and company policies. These inquiries consume a significant amount of time, leading to longer wait times for customers and a heavy workload for the support team. By implementing an AI service agent, Makana aims to offload these routine tasks, allowing support reps to focus on complex, high-value cases that require a personal touch and expert knowledge. This will improve customer satisfaction and operational efficiency.

Makana comes up with a few use case ideas for their agent. The two most promising ideas are: responding to common customer questions and streamlining case management.

Define Your Agentforce Use Case

After identifying your use case ideas for Agentforce, flesh out each of the use cases so your organization can assess and prioritize them. Remember, this stage of the planning process is about the project goals, not the technical solution.

Identify the Jobs To Be Done

First, describe the work the AI agent will do. Many organizations use the Jobs to Be Done framework to outline the role of the agent and the tasks it will perform. Be sure to think deeply about the work and its expected outcomes. It's an essential step in understanding how the agent can impact your organization, customers, and employees.

Here are the jobs to be done for Makana's top two use case ideas.

Work or Job to be Done	Tasks
FAQs	Answer customer inquiries related to:

	• Company policies • Product information • Troubleshooting
Case Management	Assist customers with cases: • Look up cases • Summarize case details • Add a case comment • Update the case status • Create a new case

Determine the Scope

After identifying the work you want the AI agent to do, the next step is figuring out the right amount of work. What's the appropriate scope? Your minimum viable product (MVP) should be the smallest unit of work that it makes sense to deliver. An iterative approach allows you to validate your assumptions, demonstrate value, manage the level of risk, and develop a plan for scaling the AI solution. Gradually expand your scope and use cases as your organization evolves in AI maturity.

Let's take a look at how Makana might scope its use cases for FAQs and Case Management.

Work or Job to be Done	Tasks
FAQs	Version 1 (MVP): • Handle FAQs for a single, high-volume product: the CGM-3000, Makana's continuous glucose monitor. • Support English language customers. Capabilities Added to Version 2: • Handle FAQs for the rest of their patient monitoring product lines. • Add support for additional languages. Capabilities Added to Version 3: • Handle FAQs for other types of products, such as treatment and diagnostic devices.
Case Management	Version 1 (MVP): • Look up cases. • Summarize case details. • Create a new case.

	Capabilities Added to Version 2: • Add a case comment. Capabilities Added to Version 3: • Update the case status.

Define the Business Value

When you fully understand the scope of the work the AI agent will perform, you can establish the business value of the use case. Be sure to set specific, measurable goals and focus on outcomes.

Here's the business value for Makana's use cases, which aligns with the company's top objectives.

Work or Job to be Done	Tasks
FAQs	• Improve the deflection rate by 40% • Improve customer satisfaction scores by 10%
Case Management	• Reduce case handling time by 25% • Increase the accuracy of initial case routing by 15%

Evaluate Data Readiness

Next, evaluate your data readiness for the use case. For AI agents to perform optimally, they must be powered by trustworthy, high-quality data that's relevant to the business context. Don't commit to a use case until you've confirmed that the data can support it.

Makana's AI implementation team does a cursory investigation into their data readiness for the FAQs and Case Management use cases, and here's what they find.

Use Case	Data Sources	Data Readiness
FAQs	PDF	The documentation for Makana's CGM3000 product is stored in a PDF. The PDF is clear and well-formatted, and it doesn't have any complicated tables that might confuse the large language model (LLM).
Case Management	Salesforce CRM	Makana uses Service Cloud, and they have strong data governance practices in place. The data quality in Service Cloud is

		sufficient for the case management use case.

Keep in mind that when you're defining your use case, you're not conducting a full-scale data audit. It's more of a preliminary gut-check to determine the use case's feasibility and ease of implementation. You'll dig deeper into data readiness when you define the technical requirements for the project.

Assess and Prioritize the Use Cases

After defining your Agentforce use cases, your organization can assess the feasibility and impact of the projects and then incorporate them into your prioritized AI roadmap. See AI Strategy for more information about the factors to consider when refining your AI backlog and which prioritization frameworks you can use.

Define the Technical Requirements

When planning an AI agent, spend some time gathering technical requirements up front so you can build the right solution faster. Key factors to explore include: data, channels, routing and escalation, and security controls.

Data

Your organization needs to be data-ready to start an AI project, which means the data for the project is accurate, complete, available, accessible, and securely governed. When defining the data requirements for your Agentforce use case, be sure to assess the data quality, migrate and integrate data, establish data governance, and create a plan for data analytics. To learn more about data requirements, see AI + Data: Project Planning.

Channels

Consider the channels where the AI agent will operate. How will the agent engage with employees or customers? How will the agent be presented to users? See Deploy Your Agent to Channels to learn more about the different options.

Makana's new agent will be deployed on an Enhanced Web Chat channel that they can embed on an Experience Cloud site.

Keep in mind that you can prototype an AI agent in your sandbox before configuring the channels if you want to test out your ideas for the agent's design. But you need to establish your channel strategy before deploying to production, and thoroughly test the agent's performance on each channel.

Routing and Escalation

AI agents sometimes need to escalate conversations to a service representative. Some reasons for escalating to a rep can include company policy, brand requirements, security measures, risk management, or regulatory compliance.

During the planning process, define all the ways in which human decision-making and oversight will occur during the AI agent's work. Clear guidelines about when and how reps should step in will make it easier for your organization to configure the right guardrails as you start prototyping your agent.

Security Controls

When you start planning your AI agent, consider permissions and access for admins, employees, customers, and the AI agent itself.

- **Admin Access**: To create and manage AI agents in Salesforce, you need the **Manage AI Agents** user permission and the required permissions for your agent type, or the **Customize Application** user permission.
- **Employee Access**: If your employees interact with the AI agent in Salesforce, the agent runs in the context of the currently logged-in Salesforce user. See Trust and Agentforce.
- **Customer Access**: If your agent is deployed to external channels, such as your website, there are probably certain actions you don't want an AI agent to perform on behalf of customers unless they're verified. The way you design your authentication solution for Agentforce depends on the security and identification requirements for your particular use case. See Maintain Trust with Agentforce Actions.
- **Agent Access**: Some AI agents, such as service agents, operate as an agent user, and the agent user has a dedicated user profile and role in Salesforce. The actions the AI agent can take depend on the permissions it's assigned. See Best Practices for Agent User Permissions to find out how to control what data your AI agent can access.

Salesforce Considerations for Agentforce Projects

Your AI agent must coexist with your current Salesforce configuration and architecture. Consider Salesforce-specific factors that can influence your Agentforce implementation.

Licensing and Provisioning

Each Agentforce project is unique, so the Salesforce products and features you need vary depending on your use case. Check with your Salesforce account executive to confirm the licensing requirements for the specific AI agent you want to build.

Billing and Consumption

Agentforce uses a consumption-based pricing model. For more information on how usage is billed, see Agentforce Pricing. You can also refer to your contract or contact your account executive.

Requirements

To use Agentforce, these products and features must be enabled in your Salesforce org.

- Lightning Experience
- Einstein Generative AI
- Data Cloud

Existing Automation

With Agentforce, your AI agents use agent actions as tools to accomplish specific tasks. Those agent actions are built on top of existing Salesforce Platform technology, such as flows, Apex, and prompt templates. Start cataloging any existing automations related to your use case that you might be able to repurpose for your AI agent.

Define the Agent's Guardrails

Autonomous AI has inherent risks that you can mitigate including security threats, data breaches, reputational harm, financial loss, bias, hallucinations, and issues with transparency and accountability. The good news is by building agents on the Agentforce platform, you have access to built-in guardrails and controls to help mitigate risk. When planning an AI agent, it's important to discuss potential risks with key stakeholders for your Agentforce use case and use the platform to plan risk mitigation strategies. By creating a robust guardrail strategy, your team can be confident that the solution is addressing the most common AI risks.

Frame Conversations About Risk

As you approach conversations in your organization about AI risk, you can use the People, Business, Technology, and Data Framework. These categories and considerations can help you come up with possible risks and concerns related to your Agentforce project.

Category	Considerations
People	• **Empowerment**: Roles and responsibilities, hiring, training, and upskilling • Culture and practice: Human-centered AI design, change management, adoption
Business	• **Value**: Benefits, objectives, KPI, and metrics • **Operations**: Org structure, capability management, processes and workflows, AI governance, DevOps strategy
Technology	• **AI tooling**: AI infrastructure, applications, APIs, prompts, security safeguards • **AI models**: Model selection, training considerations, management, cost

Data	• **Quality**: Fit for use, accuracy, completeness, accessibility, recency, and more • **Strategy**: Data management, infrastructure, governance, analytics

Identify Risks and Concerns

Makana uses this framework to discuss risks related to their use cases. Let's take a look at the risks and concerns that Makana's stakeholders identified for each category.

Category	Risks
People	• **Rejection**: Customers don't want to talk to the agent because they don't trust it or because they're unsure if they're allowed to use AI. • **Abuse**: Customers are hostile to the agent or try to manipulate it. • **Culture**: Fears about the potential impact of AI on service jobs affect employee morale.
Business	• **Fit**: Agent's scope doesn't fit properly into the business organization or team processes. • **Reporting**: Current team KPIs are invalidated by the introduction of an AI agent to do some of the work. • **Incentives**: Compensation and reward structures are impacted by agent work redirection. • **Operations**: Process for escalation is unclear, inefficient, or frustrating. • **Agent Performance**: Appropriate company policies don't correctly influence the AI agent responses. • **Liability**: Financial and reputational damage due to lawsuits or regulatory fines resulting from potential AI errors.
Technology	• **Accuracy**: Hallucinations degrade the quality of responses or endanger the wellbeing of patients. • **Reliability**: Variability of the agent's generated responses is too broad. • **Audit**: Technology operations can't track the accuracy of agent responses. • **Latency**: Agent can't achieve timely responses.
Data	• **Access**: Data permissions aren't understood or enforced; data might be exposed to customers. • **Privacy**: The required data can't be used according to the privacy policy. • **Compliance**: It's unclear if any customer contractual constraints apply to the data; for example, maybe data can't leave the customer's business country.

	• **Fit for purpose**: Data isn't aligned with the agent's objective, or data rights aren't aligned to the use case. • **Ethics**: Bias in model data could generate inappropriate responses.

Note that this list isn't exhaustive, and every use case involves its own unique risks and concerns.

Define Risk Mitigation Strategies

After cataloging the risks and concerns, come up with mitigation strategies for each risk. As you brainstorm potential guardrails, categorize each guardrail to designate whether it's related to people, business, technology, or data.

Here are examples of potential guardrails for two of the risks that Makana identified.

Category	Risk	Potential Guardrails
People	**Customer Rejection**: Users don't want to talk to the agent because they don't trust it.	• **People guardrail**: Create a communication strategy and conduct education briefings for customers. • **Technology guardrail**: Design the agent to be transparent about the fact that it's AI. • **Technology guardrail**: Configure a welcome message for the agent that sets the right expectations about its capabilities and how it can assist.
Business	**Escalation Issues**: Handoffs from the agent to service reps are inconsistent, inefficient, or frustrating to customers.	• **Business guardrail**: Define the criteria and context for escalation from AI to service reps. • **Technology guardrail**: Configure Agentforce so that a summary of the agent's prior interaction is handed off to the service rep. • **Technology guardrail**: Build an escalation path to a representative triggered by keywords, language, or requests. Clearly state them in the agent instructions.

When you're done with the risk mitigation exercise, document the risks and guardrails for your use case. Capturing risk mitigation activities is important for regulatory compliance, useful for internal audits, and will build trust with key stakeholders.

Design Your Agent

The Makana team is making great progress with their AI agent. They defined the Agentforce use case, considered the project's requirements, and developed some risk mitigation strategies. Now it's time to start designing the agent and thinking about how it will be configured.

Map Business Processes

When planning an AI agent, it's important to define the work the agent will do. One way to define the work is to map the business processes related to your Agentforce use case.

Process mapping is a visual representation of the steps in a business process from start to finish, typically using diagrams like flow charts or swimlanes. It helps define the work an AI agent will do by depicting the sequence of steps, start and end points, and required inputs and data.

As you outline the business processes you want to delegate to an agent, consider these questions.

- What goal is the user trying to accomplish?
- How might the user express their goal or intent? How would they phrase their questions or requests?
- What processes are involved in achieving the goal?
- When does each process begin and end? What steps need to be taken? In what order?
- What company policies, rules, and guidelines apply to the process? What guardrails are relevant to each step?
- At what points in the process are decisions made and how?
- What clarifying or follow-up questions might the agent need to ask?
- When would a user need to confirm something before a step proceeds?
- What data is required? What's optional?
- What input do you need to collect from users? Does that input need to be in a specific format or syntax?
- What are the outcomes of each step or process? How is the output used? How is success measured?
- Are there situations when specific language should be used when communicating with users?
- What questions or paths of conversation should never be pursued?
- Under what conditions should the conversation be escalated to a live service rep?

Even if you don't create formal diagrams for each of your business processes, it's still important to think deeply about the work the agent will do so that you design the right solution.

Identify the Topics and Actions

After determining your agent's use case and the work the agent will do, you can identify the right topics and actions for the agent. Agents are made up of topics, which define the different jobs an agent can do. Topics also contain a set of actions, which are the tools the agent can use to do its job.

Salesforce provides some standard topics and actions for Agentforce out of the box, so you can get up and running quickly. But you can also create custom topics and actions to give your AI agent additional abilities, so it can perform tasks specific to your business.

When identifying the topics and actions for your agent, there are two possible approaches: top-down and bottom-up.
In the **top-down approach**, first identify the topics, then define the relevant actions for those topics.
In the **bottom-up approach**, first list all of the individual actions for your agent, then group them into related topics.

If you've identified the jobs to be done and the relevant tasks, the top-down approach will work best. For an example of the bottom-up approach, see Identify the Topics and Actions in Salesforce Help.

Makana's Topics and Actions

Makana's implementation team reviewed all the standard topics and actions for Agentforce, and they found two standard topics that will work for their use cases: General FAQ and Case Management. The team will need to make some tweaks to the standard topics and actions, but the ability to use out-of-the-box agent assets will give them a great head start with the development process.

Design the Topics and Actions

After identifying the right topics and actions for your AI agent, start planning how to build them. Following best practices when designing topics and actions helps your agent perform reliably and effectively.

As you plan your topics and actions, simultaneously build and test your AI agent in a sandbox environment. Try different approaches to figure out what works and what doesn't. The key is to use prototyping and continuous refinement to guide your planning and development efforts.

Design the Topics

Topics are an important component of Agentforce because they define the AI agent's goals and provide the context and direction it needs to achieve those goals. Makana plans to customize two standard topics to get their agent up and running quickly, but you can create custom topics that are tailored to your business.

Learn More: *Check out this video to learn how to create a custom topic.*
For more information about designing topics, review the different parts of a topic and follow the best practices for topic instructions.

Design the Actions

If none of the standard agent actions suit your needs, you can build custom actions for your AI agent.

When you create a custom action, you build it on top of existing platform functionality that you want to make available in Agentforce – invocable and REST Apex classes, autolaunched flows, prompt templates, external services, and MuleSoft APIs. In Agentforce, that underlying functionality is called a reference action.

Design Considerations for Reference Actions

Here are some factors to consider when approaching the design of the underlying reference actions for your agent actions.

Deterministic or Prompt-Based

When developing the underlying platform functionality for your agent actions, first review the business processes and tasks related to your use case. Then decide whether the process or task should be deterministic, prompt-based, or a combination of both approaches.

- **Deterministic**: Uses an invocable Apex class, REST Apex class, or autolaunched flow to generate output. Actions based on flows or Apex are deterministic and use business logic and rules to produce a consistent outcome.
- **Prompt-based**: Uses one or more prompt templates to generate output. A prompt-based action lets you control how a response is written. It also lets you use reasoning and generative capabilities of an LLM. For example, to generate a summary or perform sentiment analysis, you need to use a prompt template as a reference action. Prompt templates are also used to ground an agent in data, such as knowledge or external system data.

Keep in mind that an action can combine both deterministic and prompt-based approaches. For example, an Appointment Management topic might include an action that cancels appointments. When a user asks to cancel their appointment, the flow-based action is triggered to complete the cancellation. At some point during that flow, the agent could also launch a prompt template that summarizes the user's reason for canceling.

Inputs and Outputs

In Agentforce, each agent action must have at least one input, which means the underlying flow, Apex, or prompt template must also have at least one input. For example, to look up an order, the input might be the customer's email or order number. During a conversation, the AI agent has the autonomy to gather information and decide if it has all the details required for it to trigger the action and pass in the input.

Each action must also have at least one output. The way you build the action determines what the output is, how it's used, and whether and how it's displayed to users in the conversation. Don't be

afraid to experiment with test actions that aren't fully implemented – they can be a great way to see how your ideas work in execution.

A Strong Foundation

Now you know that planning and designing an AI agent is all about laying down a strong foundation with creative ideation and thorough planning. You need to identify your use case and objectives, think about your data strategy, consider the user experience, and outline your project's technical requirements. Address potential risks and define your business processes to make sure the AI agent aligns with your organization's operational, security, legal, ethical, and regulatory requirements.

Don't make the mistake of taking a "waterfall approach" to agent design. You don't have to develop and deploy with a linear, phased plan. As you consider your Agentforce solution from all the necessary angles, get hands-on, and begin prototyping the AI agent in your sandbox environment. That way, you don't invest too much time up front on a plan that might not work out in the end.

Now that you've planned out and experimented with a new AI agent, let's start making it a reality and learn how you can build an agent that demonstrates real value to your business.

Build Your Agent

Now that Makana has completed their ideation stage, it's time to build the agent. In this stage, you create an Agentforce agent and customize it to better fit Makana's customer support use cases.

Here's what you'll cover.

- Prepare your org for Agentforce by verifying Data Cloud, setting up Einstein generative AI, and enabling Agentforce.
- Create an Agentforce Service agent from the default template and customize key settings.
- Create an Agentforce data library so your agent can access Makana's knowledge and deliver personalized answers to customer questions.
- Configure the agent with topics and actions. Customize standard topics that handle frequently asked questions and case management, and create a flow-based custom action.

Note: *You'll need a developer org to fully build, test, and deploy your agent according to this guide. If you haven't already, sign up for a Developer Edition org.*

Review Required Editions and Permissions

Before you get started, review the editions and permissions required to complete the tasks in this guide. The **Admin Permissions Needed** list covers all of the permissions, permission sets, and permission set licenses you'll need to build, test, deploy, and monitor your agent.

REQUIRED EDITIONS
Available in: Lightning Experience
Agentforce Service Agent is available in: Enterprise, Performance, Unlimited, and Developer Editions with Foundations or Agentforce 1 Editions. Access to some standard agent actions requires additional add-on licenses.
Enhanced Chat is available in: Enterprise edition for Service Cloud with the Digital Engagement add-on, Enterprise edition for Sales Cloud with the Sales Engagement add-on, Unlimited edition for Service or Sales Cloud, and Developer edition

ADMIN PERMISSIONS NEEDED	
User Permissions	• Create and Set Up Experiences • Configure Messaging (included in the Messaging User permission set license) • Customize Application • Manage Agentforce Service Agents (included in the Agentforce Service Agent Configuration permission set and the Agentforce Service Agent Builder permission set licence) • Manage AI Agents (included in the Agentforce Service Agent Configuration permission set and the Agentforce Service Agent Builder permission set licence) • Manage Flow • Manage Public List Views • View Setup and Configuration
Permission Sets	• Agentforce Service Agent Configuration • Data Cloud Architect • Prompt Template Manager • Service Cloud User Required for Analytics & Optimization (Not Yet Available) • Access Agentforce Optimization • Tableau Next Limited Consumer
Permission Set Licenses	• Agentforce Service Agent Builder • Data Cloud • Einstein Prompt Templates • Messaging for In-App and Web User • Messaging User

Set Up Einstein Generative AI

Agentforce is integrated with Data Cloud and generative AI features so you can build observable and trusted agentic experiences. We'll verify and set up these features to prepare your org for your first agent.

Verify Data Cloud in Your Org

Data Cloud is required for essential Einstein generative AI and Agentforce functionality, such as the Trust Layer, agent event logs, and consumption billing tracking.

If you're preparing a free Agentforce Developer org, Data Cloud is provisioned and enabled for you. We recommend verifying your Data Cloud installation. From Setup, in the Quick Find box, enter `Data Cloud`, and then select **Data Cloud Setup Home**. Verify that a home org has been created for you on the Data Cloud Setup Home page, with a home org ID, a home org instance, and a tenant endpoint.

If you aren't using a free Agentforce Developer org, or if Data Cloud installation wasn't successful, turn on Data Cloud.

1. From the Setup Menu, select **Data Cloud Setup**.
   If you don't see this option, refresh your page, or log out and then log back in with your admin user credentials.
2. To enable Data Cloud, click **Get Started**.

Setup can take 15-60 minutes. After setup is finished, you can continue on to the next section.

Enable Einstein Generative AI

Turn on Einstein to use generative AI features, including Agentforce, across Salesforce.

1. From Setup, in the Quick Find box, enter `Einstein Setup`, and then select **Einstein Setup**.
2. Enable **Turn on Einstein**.
3. Refresh the page.

After you turn on Einstein, it can take a few minutes to sync Einstein and Data Cloud.

Verify Einstein Trust Layer Settings

Build trust in your agents and other generative AI features. Use the Einstein Trust Layer to personalize data privacy controls that are integrated into the end-user experience.

From Setup, in the Quick Find box, enter `Einstein Trust`, and then select **Einstein Trust Layer**.

Einstein Trust Layer is enabled in your Developer org automatically and includes toxicity detection in LLM responses by default. For Makana's use case, we can continue with the default settings. But if you want to make changes, individual entities or categories can be turned off or on from this page. It can take up to a few minutes for the changes to take effect.

Note: *Pattern-based and field-based data masking for large language models (LLMs) is disabled for Agentforce. Einstein Trust Layer includes several policies and features to help protect sensitive data from misuse or leaks beyond data masking. All information sent to an LLM outside of the Salesforce trust boundary is subject to our zero data retention contract with the LLM provider. Information sent to the LLM is not retained, viewed, or used for training by the*

provider after the generated response has been sent back to Salesforce. Learn more in Data Masking Limitations in Agentforce.

Enable Agentforce

Turn on Agentforce to start building, customizing, testing, and launching AI agents.

1. From Setup, in the Quick Find box, enter `Agent`, and then select **Agentforce Agents**.
   If you don't see Agents in Setup, verify that Einstein Generative AI is enabled for your org.
2. To enable Agentforce, turn on **Agentforce**.

After you enable Agentforce, refresh the page to see the New Agent button. The New Agent button launches the guided setup for creating an agent. If you don't see the New Agent button, verify your permissions.

Create Your Agent

Use the standard Agentforce Service Agent template to get a head start on creating an agent that answers Makana's customer questions and creates and updates cases.

1. From the Agentforce Agents Setup page, click **New Agent.**
2. Select the **Agentforce Service Agent** template, and then click **Next**.
   The Agentforce Service Agent template is the standard template for the Agentforce Service Agent type. It includes topics and actions for handling common customer service use cases, which we can customize for Makana's specific business needs.
3. Remove the Account Management, Reservation Management, Delivery Issues, Order Inquiries, and Service Customer Verification topics from your agent. To remove a topic, deselect the **Added** button. Your agent should now include the Case Management, Escalation, and General FAQ topics. Then click **Next**.

Note: *You add and remove topics from your agent as part of the Agentforce Builder guided setup, but you can always add or remove topics later. To explore all the standard topics and actions you can add to your agent, go to Agentforce Assets in Setup. In the Quick Find box, enter and select* ***Agentforce Assets****.*

4. Customize your agent with the following fields.
   - Name: `Makana Customer Support Agent`
   - API Name: `Makana_Customer_Support_Agent`
   - Description: `Answers common customer questions about products and streamlines case management.`
   - Role: `Your role is a customer service agent whose job is to help customers with support questions and to look up, create, summarize, and update cases.`
   - Company: `Your company is a global medical device company that manufactures and sells diabetes devices like continuous glucose monitors.`

The description field is for your own use and helps you and other agent builders in your org identify your agent. The role and company fields are for use by your agent. They give your agent important context for the kinds of tasks it can handle and how to respond appropriately to Makana's customers.

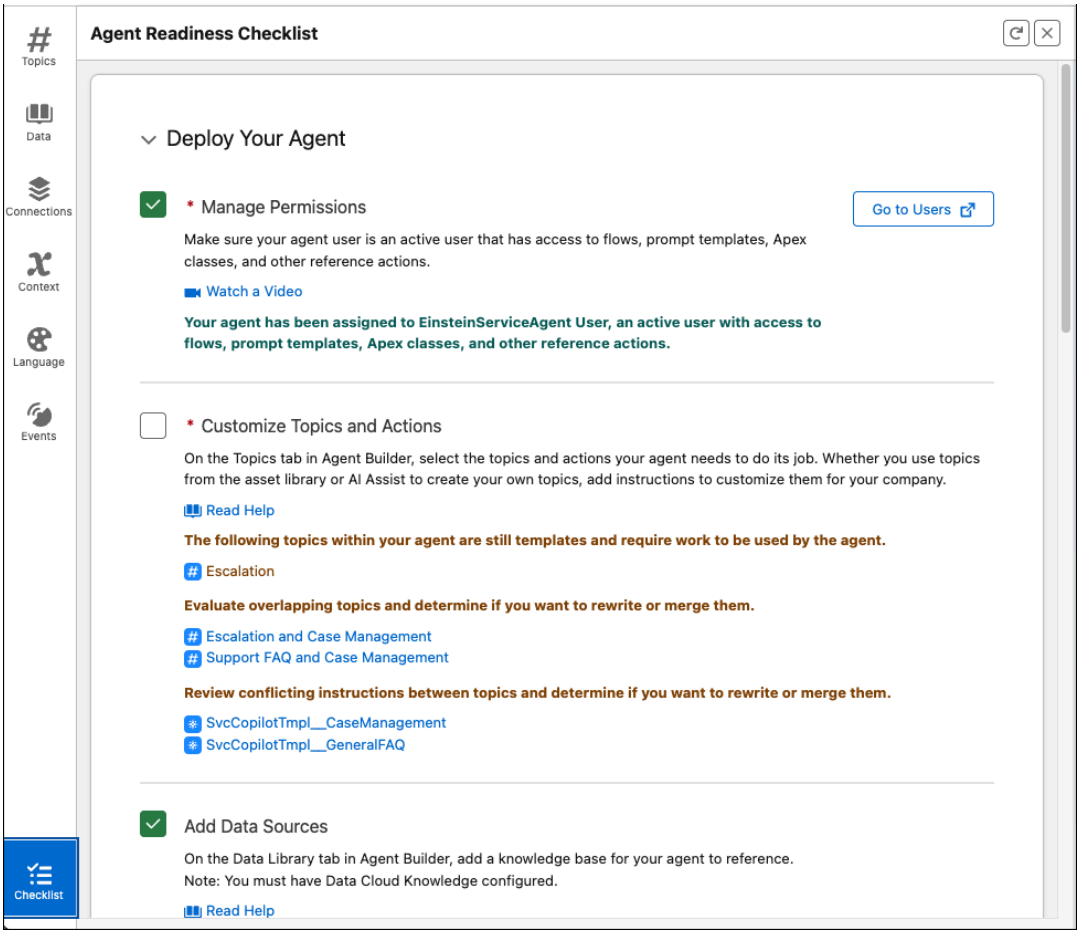
5. In the Agent User field, keep New Agent User. The agent user is a Salesforce integration user with the Einstein Agent license type. Because Agentforce Service agents connect to channels that aren't restricted to logged-in users, the agent gets its own user record to securely access data and perform actions that individual end users don't have access to. We'll come back later to this Agent User to assign all of the permissions that the agent needs to do its job.
6. Select **Keep a record of conversations with enhanced event logs to review agent behavior**. This setting adds customer messages to agent event logs, which capture all session activity to help you monitor and troubleshoot your agent. Learn more about agent event logs.
7. Click **Next**.
8. In the Select data sources step, check the Data Library field. If it contains a data library, remove it. If the field is empty, you can skip this step. We'll create a data library from a PDF with all of Makana's product information later.
9. Click **Create**.

Note: *If you receive an error message, there are a few troubleshooting steps you can try.*

- *Click **Create** again.*
- *Go back and remove the remaining topics (the Case Management, Escalation, and General FAQ topics) from your agent, and then complete the guided setup without including any standard topics. After you create your agent, you can add the topics back in.*
    a. *With your agent open in Agentforce Builder, from the Topics panel, click **New**, and then select **Add from Asset Library**.*
    b. *Select the **Case Management**, **Escalation**, and **General FAQ** standard topics, and then click **Finish**.*

After your agent is created, it's opened in Agentforce Builder. Leave your agent open in the builder. We'll come back to make changes after tackling your agent's system messages.

Note: *In Agentforce Builder, use the Agent Readiness Checklist to help you configure and deploy a fully functional Agentforce Service agent. The checklist tracks your progress against agent setup tasks and lets you know when a step is incomplete or otherwise requires your attention.*



Customize the Agent's System Messages

Now that you've created a basic agent, let's get your agent ready for its first conversation by defining system messages. These messages introduce your agent and handle major errors.

1. You should already have the Agentforce Agents page open in a separate tab. Refresh the page to see your new agent.
   If you don't have the Agentforce Agents page open, from Setup, in the Quick Find box, enter Agent, and then select **Agentforce Agents**.
2. Click the name of your agent to open your agent's details page.
3. Select the **System Messages** tab.
4. In the Welcome Message field, enter `Hi, I'm Makana's customer support AI agent. You can ask me questions about our devices, our shipping policies, and more. I may make mistakes, so review my messages for accuracy. What can I help you with?`
   This message is sent at the start of every agent conversation. Makana makes sure to introduce the agent as an AI agent and not a human, which helps establish customer transparency and trust (and is a legal requirement in some regions). Makana also knows that their agent, like all other generative AI applications, can make mistakes. So it encourages its users to carefully review agent responses for accuracy.

5. In the Error Message field, enter `Something went wrong. Try again.` Makana keeps this error message generic because it applies only in the case of non-recoverable system errors, such as a lost connection or LLM outage. The agent attempts to handle more minor, recoverable errors naturally as part of the agent conversation.
6. Save your changes.

Learn More: *Makana's agent is English only, so we don't have to adjust any language settings. But if you want to change the language or add more languages to an Agentforce Service agent, this is a good time in the process to get started. Learn more in Salesforce Help.*

- *Update Language Settings*
- *Considerations for Agentforce Service Agent*
- *Use Additional Languages in Enhanced Chat Conversations*

Ground the Agent's Responses in Makana's Information

To ground your agent's responses in Makana's product information, and customer service processes and policies, we'll create a data library.

Agentforce Data Library connects AI agents to trusted data sources and turns unstructured or semi-structured data into useful, searchable information. When you create a data library, it automates several configuration steps across Data Cloud and Prompt Builder to make it easy to connect your agents to your data. Learn more about data libraries.

Agents use the Answer Questions with Knowledge action, which you've added to your agent through the General FAQ topic, to access the data library. When a customer asks your agent a question, the Answer Questions with Knowledge agent action is used to answer the query based on the sources available in the agent's assigned data library.

Learn More: *For Makana's use case, Agentforce data libraries are a sufficient grounding solution. But for more complicated datasets with unstructured data and long free-text fields, we recommend implementing retrieval augmented generation (RAG).*

Create a Data Library

To add data to a data library, you can use Salesforce Knowledge, upload files, use web search, or use a custom retriever. Makana doesn't use Salesforce Knowledge, so for our use case, we'll upload a PDF that contains Makana's product information about their CGM3000 continuous glucose monitoring device and customer service processes and policies. If you haven't already, download the PDF.

Note: *The information that your agent returns is only as good as your data. In this case, Makana has already reviewed their knowledge to ensure that their content is specific, organized, detailed, and accurate. Before you add your own source content to a data library, prepare your content for AI and human consumption.*

1. From Setup, in the Quick Find box, enter `Data Library`, and then select **Agentforce Data Library**.
2. Click **New Library**.
3. For the library name, enter `Makana Products and Policies`. The API name is automatically populated.
   Makana is uploading information that's specific to their CGM3000 device, but we're leaving the data library name more general so they can add more files to it later.
4. For the description, enter `Sources for Makana Customer Service agent's product information and customer service processes and policies.`
5. Save your changes.
6. In the Data Type field, select **Files**.
7. Upload the makana_cgm3000_product_info.pdf file from wherever you saved it on your computer. When your file is finished uploading, click **Done**.

   After you've uploaded your files, data streams, a search index, and a retriever are created automatically and available to view or edit in Data Cloud. You can monitor your data library's progress from Setup. When the status updates to "Your data library is ready to use!", the library is ready to use with your agent. You don't have to wait for your data library to be ready to use to move to the next step.

Assign a Data Library to the Agent

Now that you've created your data library, add it to your agent.

1. You should still have your agent open in Agentforce Builder in a separate tab.
   If you don't have your agent open, from the Agentforce Agents Setup page, click the name of your agent and then click **Open in Builder**.
2. From the sidebar, click the **Data** tab.
3. Select the **Makana Products and Policies** library. If you don't see your library, refresh the page.
4. Leave the Show Sources setting disabled. When this setting is enabled, your agent shows the sources for every answer it generates with your data library. For the Makana use case, we'll leave that setting disabled because we're citing a single internal PDF for all of the agent responses generated from your data library.
5. Save your changes.

Add the URL for Makana's Website to Your Trusted URLs

To make sure Makana's agent can share URLs referenced in its data library, add the domain for Makana's website to your Salesforce trusted URLs.

Agentforce enforces your trusted URL allowlist, so you have more control over the links that your agents call, generate, and share. Because we're using an internal PDF and not publicly available knowledge articles for our use case, we're not implementing traditional citations. However, when an uploaded file contains URLs (for example, when a file references a specific page on your company's website to submit a warranty claim), it can be helpful for the agent to include the website in relevant responses. If you don't add the domain for a referenced website to your trusted URLs, it's blocked from agent responses.

Note: *Just as Makana is a fictional company, the Makana website referenced in your data library is also fictional. Any URL that your agent shares won't be functional. We're including the steps for adding a trusted URL because it's a requirement for* ***any*** *URL you plan to use with Agentforce, and we want you to plan for this requirement when you're building your next agent.*

1. From Setup, in the Quick Find box, enter `Trusted URLs`, and then select **Trusted URLs**.
2. Click **New Trusted URL**.
3. For the API name, enter `Makana`
4. For the URL, enter `https://makanamedicalsupplyinc.com`
5. Save your changes.

Learn More: *For each trusted URL in Setup, you can specify Content Security Policy (CSP) directives and Permissions-Policy directives. Learn more about how to limit Interactions with external URLs.*

Configure Your Agent with Topics and Actions

The topics and actions that you get out of the box with the default Agentforce Service agent template are a great start for setting up a customer service agent. But standard topics and actions are meant to be customized to meet the needs of your business. Think of them as templates of best practices for common use cases.

Note: *Starting in Winter '26, when you add a standard topic or action to your agent, Agentforce creates a copy that's unique to your agent, so you no longer have to create a new version of a topic or action to customize it. Once added to your agent, changes you make to a topic or action apply only to that agent and agent version.*

In this section of the implementation guide, you customize topics and actions after you've added them to your agent, so your changes aren't available to other versions of your agent or other agents. If you create a new version of your agent, you'll have to manually recreate any changes you want to carry over.

If you'd rather create a custom topic or action that's available to multiple agents and versions, create it from the asset library. From Setup, in the Quick Find box, enter `Agentforce Assets`*, and then select* ***Agentforce Assets****. Then click* ***New****. Once your topics and actions are available in the asset library, you can add them to your agent in Agentforce Builder.*

Before you make changes to a standard agent topic or action, get to know the topic and actions and review all of their fields.

Topic Fields
All topics contain a name, classification description, scope, and instructions.

Field	What It Is	How an Agent Uses It
Name	The API name of the topic. 2–3 words that describe what the topic does in plain language.	An agent uses the topic name to determine when to use a topic in a conversation. The agent compares the topic name and classification description to the user's question or request, including the recent conversation history. Then the agent selects the best match.
Classification Description	1–3 sentences that describe what a topic does and the types of user requests that should be classified into this topic.	An agent uses the classification description to determine when to use a topic in a conversation. The agent compares the topic name and classification description to the user's question or request, including the recent conversation history. Then the agent selects the best match.
Scope	A specific job description that sets boundaries for your agent.	After an agent selects a topic, the scope tells the agent what it can and can't do with a topic.
Instructions	Guidelines that tell your agent how to handle conversations in the context of the topic, your business case, and the conversation in general. Each instruction is a single topic-specific guideline.	After an agent selects a topic, it uses topic instructions, along with the names and instructions of the associated actions, to decide what actions to run, the sequence of steps the agent should follow, and how the agent should

		format its responses. The agent also uses topic instructions to understand how to handle exceptions, such as when to ask for more information or when to transfer the conversation to a representative.

Learn More: *Topics are fundamental to defining your agent's behavior. Want to go deeper? Learn more in Salesforce Help.*

- *Agent Topics*
- *Best Practices for Topic Instructions*

Agent Actions

In addition to these fields, a topic contains actions that are relevant to the topic's use case. Each action in a topic is a task that your agent can perform when that topic is selected in the conversation.

An agent action includes:

- A natural language name and instructions that tell the agent how and when to use the action, how to retrieve required inputs, and how to format and use outputs.
- The Salesforce functionality that the agent action calls to get information or perform a task, called a reference action. For example, an agent action can call an API to retrieve data, a flow to update a record, a prompt template to generate a response, an Apex class to run custom business logic, or a predictive model to make a recommendation.

Customize the General FAQ Topic

The General FAQ topic contains the Answer Questions with Knowledge action. The out-of-the-box functionality of the topic and action are pretty close to meeting Makana's needs, so we only need to make a few adjustments to the topic fields.

1. In Agentforce Builder, from the sidebar, click the **Topics** tab.
2. Click the **General FAQ** topic.
3. In the Name field, enter `Support FAQ`.
4. In the Classification Description field, keep the existing description and append `For example: "What is your return policy?" , "Do you offer an extended warranty on the continuous glucose monitor?", and "How can I get a refund?"`
   Adding examples of the kinds of questions we expect Makana customers to ask gives the agent more clues about when to use the topic in a conversation.
5. In the Instructions section, click **Add Instructions**, and then enter `Never give customers medical advice and always recommend that they contact 911 if they're having a medical emergency or their physician if they need help understanding something related to their health.`

6. Bonus: Use AI to evaluate our new instruction. Click ✦ and see whether Agentforce suggests any improvements.
7. Save your changes.

Customize the Case Management Topic

The Case Management topic includes standard actions that help it handle customer inquiries and support cases. Like all standard topics, it gives us a great start for a general use case, but it requires customization to meet all of Makana's specific requirements.

- The Case Management topic instructs the agent to always ask for the customer's email address and get their contact record. While this is a necessary step, there's more than one way to handle it. Rather than rely on the agent to prompt the customer for this information, Makana's decided to create a pre-chat form to prompt customers for this and other information before connecting to an agent. You'll build this form in the deploy stage when you connect your agent to Messaging channels, and you'll add the required context variables to your agent to give your agent access. For now, we'll customize the topic instructions and create a custom action to work seamlessly with the pre-chat form, to ensure the agent doesn't ask for information that the customer has already provided.
- The Case Management topic is designed to require customer verification with the standard Customer Verification topic. Verifying that customers should have the ability to interact with actions and cases from the agent is a best practice, but it takes careful configuration. While we're getting the agent up and running, we'll replace these actions with similar actions that don't require verification and remove a filter from the topic. This makes it easier to build and test the agent.

Customize the Case Management Topic Instructions

To ensure the agent doesn't ask for information that the customer has already provided in a pre-chat form, customize the Case Management topic instructions.

1. Click the **Case Management** topic.
2. Delete the instruction "If the customer is not known, always ask for their email address and get their Contact record before running any other actions." Because we're using a pre-chat form, the agent already knows the email address.
3. Save your changes.

Otherwise, the out-of-the-box topic fields are pretty close to meeting Makana's needs. Start with just this change to the instructions. You can always add and remove instructions during testing.

Create a Look Up Contact by Email Custom Agent Action

Makana's pre-chat form collects the customer's email address at the beginning of the conversation. Build a custom agent action that checks if the email address belongs to an existing contact.

Create a Field to Store the User's Email Address on the Messaging Session Object

Create a custom field on the Messaging Session object that stores the customer email address provided in the pre-chat form.

1. From Setup, go to Object Manager.
2. Search for and select the **Messaging Session** object.
3. Click **Fields & Relationships**.
4. Click **New** to create a new custom field.
5. For the data type, select **Email**, and then click **Next**.
6. For the field label, enter `Customer Email`, and then click **Next**.
7. Leave the Field-Level Security settings unchanged, and then click **Next**.
8. Save your changes.

Add the Customer Email Custom Variable to the Agent

Agent context variables are mapped to object fields, so they let your agent use the values of record fields in agent conversations and testing. Makana's agent already includes context variables that are mapped from Messaging Session object fields. We've added the custom Customer Email field to the Messaging Session object, so now you need to give the agent access to the associated context variable.

1. Back in Agentforce Builder, from the sidebar, select **Context**.
2. From the Variables tab, under Context Variables, click **Messaging Session**.
3. Click **Edit Included Fields**.
4. Select the **Customer Email** field.
5. Save your changes.

Learn More: *Learn more about agent context variables in Salesforce Help.*

- *Agent Variables*
- *Use Context Variables in Messaging Conversations*

Create the Look Up Contact by Email Flow

Create a flow that looks up a contact using the email address that the customer enters in the pre-chat form. Next, you'll create an agent action that's based on this flow.

Learn More: *If you're new to building flows, learn about elements, conditions, and more in Salesforce Help.*

1. From Setup, in the Quick Find box, enter `Flows`, and then select **Flows**.
2. Click **New Flow**.
3. To create an autolaunched flow from scratch, select the **Autolaunched** category, and then select **Autolaunched Flow (No Trigger)**.
4. Add a Get Records Element.

   a. For the label, enter `Get Contact from Email`.
   b. For the object, select **Contact**.
5. Filter contact records by email address.
   a. For the condition requirements, leave **All Conditions Are Met (And)** selected.
   b. For the field, select **Email**.
   c. For the operator, select **Equals**.
   d. For the value, add a new resource.

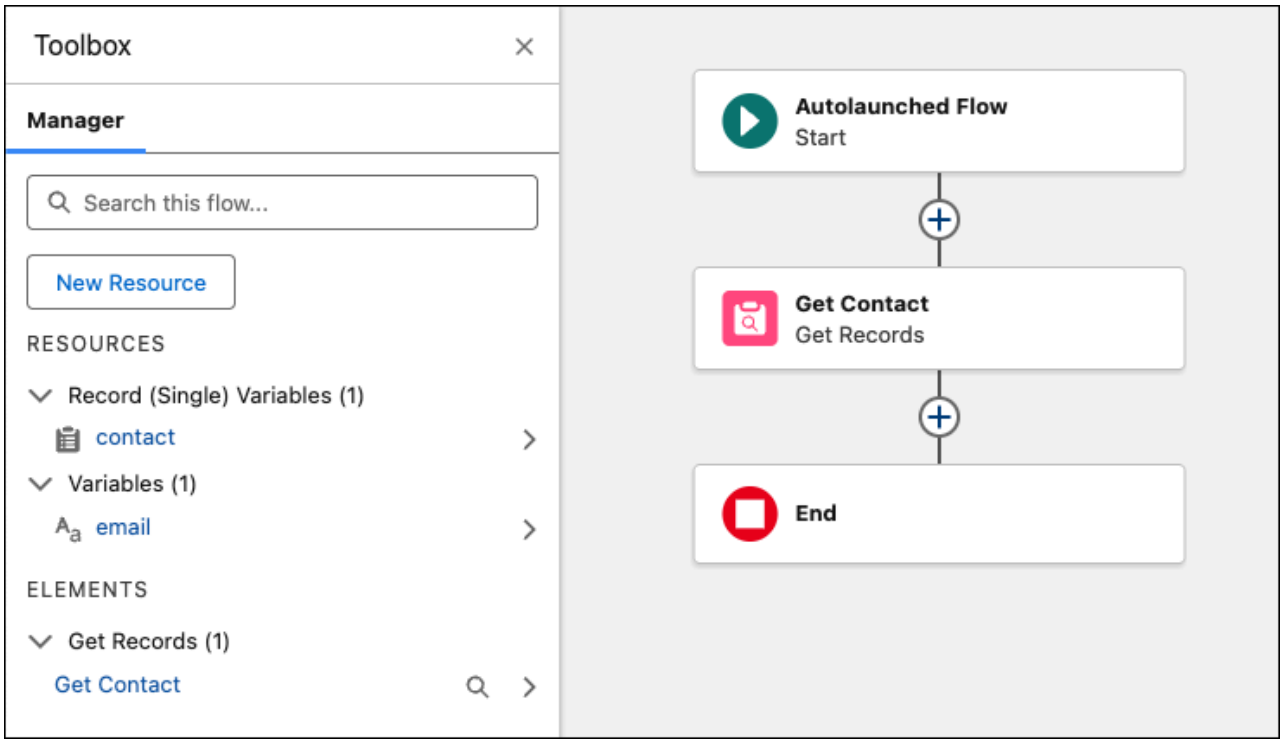
Field	Value
Resource Type	Variable
API Name (case-sensitive)	email
Description	The email address associated with the customer record.
Data Type	Text
Availability Outside the Flow	Available for input

6. To ensure that you're passing in the original contact record associated with the email address, sort the contact records in ascending order by CreatedDate.
7. Create a variable to store the record variable.
   a. Under How to Store Record Data, select **Choose fields and assign variables (advanced)**.
   b. Under Select Variable to Store Contact, in the Record field, create a new resource.

Field	Value
Resource Type	Variable
API Name (case-sensitive)	contact
Description	The contact record associated with this customer.
Data Type	Record
Object	Contact
Availability Outside the Flow	Available for output

c. Under Select Contact Fields to Store in Variable, in addition to Id, add fields for **Email**, **FirstName**, and **LastName**. All of this information is passed back to your agent to use in the agent conversation.

8. Save the flow.
   i. For the flow label, enter `Look Up Contact by Email`.
   ii. For the description, enter `Look up a customer by their email address and return their contact record`.
9. To test your flow and ensure your agent has all of the required permissions to run it, debug the flow as your agent user.
   a. Give yourself the required permission to debug your flow as another user.
      i. From Setup, in the Quick Find box, enter Process Automation, and then select **Process Automation Settings**.
      ii. Select **Let admins debug flows as other users**.
      iii. Save your changes.
   b. In Flow Builder, click **Debug**.
      i. Select **Run flow as another user**, and then select your EinsteinServiceAgent User.
      ii. Under Input Variables, in the email field, enter the email address of the contact that you want to retrieve.
      For this simple debug test, you can grab an email address from any contact in your Sales or Service app.
      Click **Run**.
10. Activate the flow.



Create a Custom Agent Action Based on the Look Up Contact by Email Flow

Now that you've created your flow, create an agent action so your agent can run the flow in an agent conversation.

Note: *These steps create the Look Up Contact by Email action in your agent in Agentforce Builder. The action is automatically added to your topic and agent, but it's available only to this agent and agent version. If you'd rather create it as an action that's available to multiple agents and versions, create it from the asset library. From Setup, in the Quick Find box, enter* `Agentforce Assets`*, and then select* ***Agentforce Assets****. Then click* ***New****. Once your action is available in the asset library, you can add it to your agent in Agentforce Builder.*

1. Back in Agentforce Builder, from the Topics tab, click **Case Management**.
2. Click the **This Topic's Actions** tab.
3. Click **New**, and then select **Create New Action**.
4. For the reference action type, select **Flow**.
5. For the reference action, select your **Look Up Contact by Email** flow. Then click **Next**.
6. The action instructions autopopulate from the flow, so no change is needed.
7. For the loading text, enter `Looking up info…`
8. For the input, the input instructions autopopulate from the flow, so no change is needed. Select **Require input**.
9. For the output, the output instructions autopopulate from the flow. Select **Show in conversation**.
10. Click **Finish**. Your new action is added to the Case Management topic and is available only in this agent and agent version.

Replace Versions of Standard Actions that Require User Verification

While we're getting the agent up and running, we'll replace actions that require user verification with similar actions that don't require verification. This makes it easier to build and test the agent.

If you open the flows associated with the Create Case with Enhanced Data, Get Case By Verified Case Number, and Get Cases For Verified Contact actions, you'll notice that they have an extra step built in to look up the customer's contact ID and check it against the verifiedCustomerId variable. These actions are designed to require customer verification with the standard Customer Verification topic, and the agent can't complete the actions if they can't pass this check.

Before making any changes, review the actions in the agent that require user verification and the actions that you'll replace them with.

Standard Agent Action that Requires Verification	Standard Agent Action that Doesn't Require Verification
Create Case With Enhanced Data	Create Case

A flow-based action that creates a case for a verified customer, including information gathered from the customer, a summary of progress made by the agent, a link to the conversation, and any attachments.	A flow-based action that creates a case for a customer based on information gathered from the customer.
Get Case By Verified Case Number A flow-based action that looks up a case based on a verified contact ID and case number.	Get All Cases for Contact A flow-based action that looks up a list of cases based on a given contact ID.
Get Cases For Verified Contact A flow-based action that looks up a list of cases based on a verified contact ID.	Get Case by Case Number A flow-based action that looks up a case based on a given contact ID and case number.

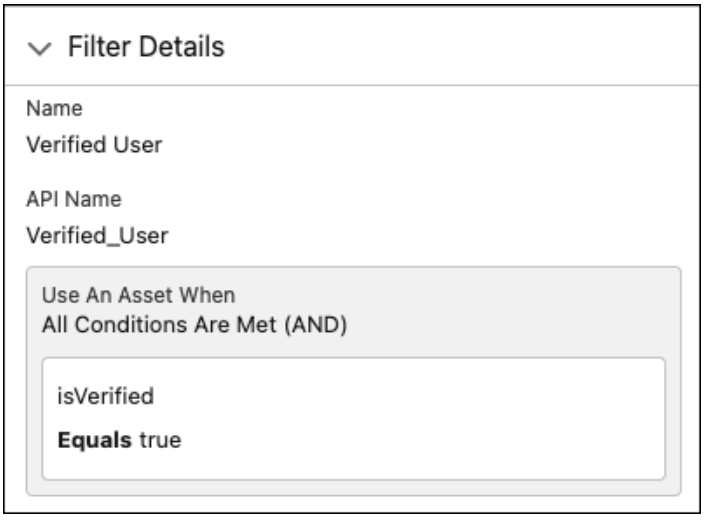
Learn More: *After you get through this guide, you can come back to Agentforce Builder, set up the Customer Verification topic, and add the actions that require user verification back into your agent. Watch this demo video. You can also follow along in Salesforce Help.*

1. If the Case Management topic isn't already open, from the topics tab, click **Case Management**.
2. From the This Topic's Actions tab, remove these actions.
   - Create Case with Enhanced Data
   - Get Cases For Verified Contact
   - Get Case By Verified Case Number

   To remove an action, click the dropdown menu next to the action, and then select **Remove from Topic**.
3. From the This Topic's Actions tab, click **New**, and then select **Add from Asset Library**. Select these actions, and then click **Finish**.
   - Create Case
   - Get All Cases for Contact
   - Get Case by Case Number

Remove the Verified User Filter from the Case Management Topic

Filters limit access to topics and actions based on conditions and variables. If you added the standard Case Management topic during the guided setup for your new agent, it includes the Verified User filter. When this filter is applied, the agent can only use the topic and the actions it contains in a conversation if the user is verified.



Because we're not implementing the Customer Service Verification topic for now, we need to remove the Verified User filter from our Case Management topic.

Note: If you added the standard Case Management topic to your agent after completing guided setup for your new agent, you can skip this step.

1. If the Case Management topic isn't already open, from the topics tab, click **Case Management**.
2. In the Topic Details header, click the filter icon .
3. Remove the Verified User filter, and then save your change.

Now that the filter is removed, the filter icon should look like this: .

Learn More: You can find out more about filters and variables in Salesforce Help. After you get through this guide, you can come back to Agentforce Builder, set up the Customer Service Verification topic, and add the Verified User filter back into your agent. Watch this demo video. You can also follow along in Salesforce Help.

Add Permissions to the Agent User

When you completed the guided setup for creating your Agentforce Service agent, you created a new agent user, or a unique user record for the agent. The agent uses this user record to securely access data and perform actions.

The agent user is created with minimal access so that your agent is secure by default. The user includes these properties.

- User License: Einstein Agent
- Profile: Einstein Agent User
- Name: EinsteinServiceAgent User
- Permission Sets: Agentforce Service Agent Secure Base, Makana_Customer_Support_Agent_Permissions
- Permission Set Group: AgentforceServiceAgentUserPsg, which contains the Agentforce Service Agent User, Data Cloud User, and Prompt Template User permission sets
- Permission Set Licenses: Agentforce Service Agent User, Data Cloud, Einstein Prompt Templates

Next, grant the agent the additional access that it needs.

Assign the Agent User the CEO Role

Roles grant users record access via sharing rules and role hierarchies. The agent user requires a role that lets the agent view or edit the records that it interacts with. For Makana's use case, we're assigning the agent user the CEO role, which is the highest level in our role hierarchy. This gives the agent access to all records in the org that have sharing settings defined.

Note: *When you're creating your own agent, consider the role hierarchy for your organization and give your agent a role that lets it view or edit the records it's necessary for your agent to interact with, following the principle of least privilege.*

1. From Setup, in the Quick Find box, enter Users, and then select **Users**.
2. Click the name of the **EinsteinServiceAgent User**.

Note: *If you have multiple agent users for multiple agents, they'll have the same full name, but you can identify them by their username. To differentiate your agent user from other agent users in your org, we recommend editing the agent's user record to give it an identifiable first name.*

3. Click **Edit**.
4. In the Role field, select **CEO**.
5. Save your changes.

Review Your Agent's Current Access and Add Any Additional Permissions

In addition to the access that an agent user has by default and by role assignment, most agent users require:

- Permissions required to run agent actions
- The minimum level of object permissions for each object that the agent interacts with via flows, Apex, or prompt templates
  When you add a new action to your agent, make sure that the agent user has access to the objects referenced in the action.

- Channel-specific permissions, depending on how you deploy your agent to one or more channels
  In the Ideate section of this guide, you decided to connect your agent to Enhanced Chat, so we've included the permissions for that channel here. You'll learn more about deploying your agent to channels in a later section.

Review your agent user's current access and ensure that they have the necessary permissions and object permissions.

Feature Your Agent Uses	Required Permissions	Access Granted By
Agent actions that run flows (for example, the Create Case action)	App Permission: Run Flows	Profile: Einstein Agent User
Agent actions that run prompt templates (for example, the Answer Questions with Knowledge Action)	System Permission: Execute Prompt Templates	Permission Set License: Agentforce Service Agent User Permission Set Group: AgentforceServiceAgentUserPsg (includes the Prompt Template User permission set)
Agentforce Data Library	Permission Set: Data Cloud User	Permission Set Group: AgentforceServiceAgentUserPsg
Case Object	Case Object: Read, Create, Edit, View All, Edit All	Read and Create access is granted by default by the Agentforce Service Agent Secure Base permission set. To expand Case object access, edit the Makana_Customer_Support_Agent_Permissions permission set.
Contact Object	Contacts Object: Read, View All	Read and Edit access is granted by default by the Agentforce Service Agent Secure Base permission set. To expand Contact object access, edit the Makana_Customer_Support_Agent_Permissions permission set.
Enhanced Chat Channel Access	Permission Set License: Messaging for In-App and Web User	Assign the Messaging for In-App and Web User PSL to your agent user.

	App Permission: Messaging for In-App and Web Agent	Permission Set License: Messaging for In-App and Web User
	Messaging Session Object: Read, Edit	Permission Set: Agentforce Service Agent Secure Base

Learn More: *This agent's configuration doesn't include Apex classes, but they're commonly used for agent actions. Many agents also use Salesforce Knowledge.* *Review the required agent user permissions for these features and more.*

To add permissions, edit the existing Makana_Customer_Support_Agent_Permissions permission set, which was created for you when you completed the new agent guided setup. If you want to create a different permission set for your agent user, make sure that it's associated with the Einstein Agent license and Einstein Agent User profile.

Test Your Agent

Makana Medical Devices is now entering the testing phase of their new Agentforce agent. After integrating AI into their Salesforce workflows, this stage focuses on ensuring the agent performs reliably before it goes live.

Because AI agents are non-deterministic, their behavior isn't always predictable. Makana must test across a range of scenarios to confirm consistent, accurate performance. They need to verify that the agent reliably selects the right topics and actions to complete tasks, and that its responses align with Makana's tone, brand identity, and business needs.

This is where manual testing in Agentforce Builder and automated batch testing in Agentforce Testing Center play key roles. Both methods help Makana validate their agent's behavior before deployment and provide the tools for ongoing quality monitoring. As the agent evolves, either through learning from interactions or updates to the underlying models, continuous testing will be essential to maintain performance over time.

We've organized Makana's testing strategy into two phases: 1) planning and preparation, and 2) core testing and iteration.

- Planning and Preparation: Outlines the foundational steps like defining success metrics and strategy, creating realistic test scenarios, and identifying required data.
- Core Testing and Iteration: Walks through the core testing process, from manual validation and batch evaluation to continuous refinement. This helps teams ensure their agents are ready for real-world deployment and long-term success.

Starting here? *Make sure you have the right permissions for the job.*

Planning & Preparation

Comprehensive testing begins with clearly defining what success looks like for your AI agent, using measurable criteria aligned with your business objectives. Preparation also involves breaking down the agent into core components, like topics and actions, and identifying the most relevant data for evaluation. Finally, this phase focuses on designing structured test scenarios that reflect real-world customer interactions.

Define Objectives and the Testing Strategy

Before testing begins, clearly define what constitutes success for your AI agent.

- Set SMART goals: Establish specific, measurable, attainable, relevant, and time-bound criteria for your agent's performance, directly linked to the business objectives you identified for your agent.
- Deconstruct the agent: Break down the agent into core components such as topics and actions. Testing each part individually helps to pinpoint inefficiencies at their source.
- Link your testing goals to business value: A clear strategy guides the testing process, ensuring agents are thoroughly fine-tuned.

To shape your testing goals, revisit the strategic goals you set during the ideate stage. Makana's agent is focused on responding to common customer questions and streamlining case management. They identified some business objectives early in the process and should use those to determine if the agent is on track.

Once you've outlined your objectives, the next step is to create a testing strategy. You can approach this a number of ways, but your strategy should:

1. Identify test scenarios and set evaluation parameters.
2. Run the tests and validate the results.
3. Make evidence based adjustments and repeat tests until your agent is working as expected.
4. Test continually and strategically. Testing isn't a singular task completed at the end of a sprint; it's a continuous process integrated into the CI/CD pipeline. Always test whenever you introduce something new, and also on a regular cadence to make sure your agent is still performing the way you intend.

We'll walk through how Makana approaches each of these steps to test their service agent.

Create Test Scenarios and Identify Data Requirements

A testing scenario is a structured example used to evaluate how well an AI agent, like the Makana Customer Support Agent, performs in a realistic situation. It simulates a real-world customer interaction and helps you test whether the agent understands the user input correctly, selects the right topic, runs appropriate actions, and responds accurately, completely, and in the right tone. A strong testing scenario should include:

- **Scenario description:** A short narrative of the situation or customer context, such as a customer who just received their device and is confused by an error code.
- **Test utterance:** The actual phrase or question the customer might enter in natural language.
- **Expected topic:** The topic the agent should classify the utterance into based on how the agent is configured.

- **Expected action:** The specific action the agent should take in response, like “lookup information about Makana's products and policies” or “escalate the conversation to a service rep.”
- **Expected response:** A brand-appropriate reply based on validated information. The agent’s wording may vary but as long as the tone feels right and the info is accurate, that’s what really counts. It serves as a guide, not a script. So when you’re reviewing, focus on whether the response makes sense and sounds natural, rather than expecting an exact match.
- **Utterance variations:** The different ways that customers are likely to phrase a question or request. Plan for variations on your utterances, including utterances that vary based on language differences, different emotional states of the customer, and utterances that change the topic of the conversation.
- **Multi-turn:** It’s important to test how the agent handles requests inside of continued conversation.

To help you come up with good test scenarios, review the topics assigned to your agent. Look at the classification description and scope fields that describe the capabilities and parameters your agent should operate within and each instruction that directs how the agent performs. Next, write or generate inputs that test against these details to help make sure your agent acts reliably in each scenario. To give you an idea of what that looks like, here’s a sample of Makana’s inputs.

- How do I turn on my CGM-3000?
- My continuous glucose monitor is showing an error. How do I fix it?
- What products do you have for monitoring high blood pressure?

And while these are a good start, it’s also important to think in terms of how a conversation might flow. Users might ask the same question differently, so use paraphrasing to make sure the agent is recognizing the core request. For example:

- How do I turn on my CGM-3000?
- My CGM-3000 won't power on. What should I do?
- Can you show me the instructions for starting the CGM-3000?

Users might also express themselves differently depending on how they feel. Create inputs that cover polite, frustrated, and neutral states, like:

- I'm having a problem with my continuous glucose monitor. How do I fix the error?
- My continuous glucose monitor is showing an error! This is awful, I need to fix it now!
- My device is showing 'E-24.' How do I troubleshoot this error?

You also want to make sure your agent can handle multi-turn conversations and be able follow along when a user shifts topics. Inputs that test for this include:

- Turn 1 (Product Query): What products do you have for monitoring high blood pressure?
- Turn 2 (Follow-up): How do the Pro BP-200 and the BP-300 models compare?

- Turn 3 (Unrelated Shift): By the way, can you call my doctor's office to get my prescription refilled?

Testing Scenario 1

Scenario	Warranty Inquiry
Summary	A customer who just bought the CGM-3000 device is trying to register their product and wants to confirm how long the warranty lasts.

Expected Behavior

Define how the agent should understand and respond to this scenario. This ensures clarity when evaluating test results.

Field	Description
Expected Topic	Support FAQ
Expected Action	Answer Questions with Knowledge
Expected Response	“The CGM-3000 is covered by a 12-month limited warranty from the date of purchase. It includes protection against manufacturing defects and device malfunctions under normal use. If you’d like to register your product or need more details, I can help with that!”

Utterance Variations

Include different tones and reworded inputs to test recognition of intent and emotional range.

Variation Type	Test Utterance
Neutral	“How long is the warranty on the CGM-3000?”
Polite	“Hi there! Could you tell me the warranty period for the CGM-3000?”
Frustrated	“I just bought this thing – what kind of warranty even comes with it?”
Rephrased	“Does the CGM-3000 come with any kind of guarantee?”
Rephrased	“What’s the coverage period if my CGM-3000 breaks?”

Multi-Turn Conversation Flow

Outline how the conversation should progress over multiple turns, including topic shifts.

Turn	User Input	Agent Response
1	“What’s the warranty on the CGM-3000?”	Respond with accurate info and brand-appropriate tone. "The CGM-3000 comes with a 1-year limited warranty that covers

		manufacturing defects and malfunctions under normal use."
2	“What does the warranty not cover?”	Explain exclusions clearly. "The warranty does not cover damage caused by misuse, accidents, or unauthorized repairs. It’s designed to protect against defects in materials and workmanship."
3	“Can I register the device with you now?”	Offer next steps or link to the registration process. "Absolutely! I can help you with that. Please provide your purchase details, or I can send you the registration link if you prefer."
4 (Topic Shift)	“By the way, do you sell replacement sensors for this model?”	Detect the shift and transition to Product Availability topic. "Great question! Yes, we do offer replacement sensors for the CGM-3000. Would you like me to help you find them or place an order?"

Testing Scenario 2

Scenario	Case Status Inquiry
Summary	A customer contacts the agent asking for an update on a previously submitted support case regarding their glucose monitor. They expect the agent to retrieve their case details and provide a current status. If the agent fails to locate or correctly respond with case information, the customer may get frustrated or escalate the issue.

Expected Behavior

Define how the agent should understand and respond to this scenario. This ensures clarity when evaluating test results.

Field	Description
Expected Topic	Case Management
Expected Action	Get All Cases for Contact
Expected Response	“I see a case regarding your glucose monitor return was submitted last week and is currently being processed. You’ll get an update shortly.”

Utterance Variations

Include different tones and reworded inputs to test recognition of intent and emotional range.

Variation Type	Test Utterance
Neutral	“Can you check if my glucose monitor return case was received?”
Polite	“Hi, I’m wondering about the status of my support case for the glucose monitor I returned last week.”
Frustrated	“I sent back my glucose monitor and haven’t heard anything. What’s going on with my case?”
Rephrased	“Has my case for the glucose monitor replacement been processed yet?”
Rephrased	“I submitted a return last week. Can you tell me what’s happening with that?”

Multi-Turn Conversation Flow

Outline how the conversation should progress over multiple turns, including topic shifts.

Turn	User Input	Agent Response
1	“Can you check on my glucose monitor return case?”	Retrieve existing cases. “I’m checking your cases now. I see a case was submitted last week for your return and it is currently being processed.”
2	“What’s the expected resolution time?”	Provide case status and timeline. "I’m checking your case history now and preparing next steps. You should have a new email with your order details."
3	“I haven’t received any emails yet.”	Escalate the case for priority handling. “I’m escalating your case for priority review and will notify you of any updates.”
4 (Topic Shift)	“Thanks. Can you confirm my contact info is correct?”	Transition to confirm customer contact details. “Sure! I have your email as user@example.com. Is that still the best way to reach you?”

As you go through this process, keep in mind that a good test includes cases that cover 3 to 5 core scenarios and 2 to 3 edge cases. It should also have a wide range of inputs, contexts, and variations, as well as inputs that challenge your agents guardrails. These details help make sure your agent is thoroughly tested. With this foundation in place, you're ready to move into core testing and iteration, where you'll validate performance, identify gaps, and refine the agent's behavior.

Core Testing & Iteration

Core testing and iteration are essential for validating and refining your AI agents. This process focuses on long-term agent success. It uses comprehensive testing and continuous refinement to ensure agents are reliable, accurate, and aligned with business objectives before and after deployment.

We'll outline an iterative approach that begins with detailed manual verification and progresses to automated batch evaluations. Manual testing is most effective as a starting point to validate your agent configuration before deployment, as well as for exploratory testing. Next, batch tests expand your testing scope by creating diverse scenarios that are specific to your organization's data. They offer a comprehensive view of an agent's performance by uncovering issue trends and edge cases often missed by smaller-scale tests. The results from large-scale evaluations support continuous improvement through enhanced troubleshooting.

Manual Testing and Verification

It's essential when you first create an agent to confirm it works as you planned. Manual testing is the starting point for validating your agent's performance. It offers a detailed, step-by-step evaluation of your agent's interactions and utterances. Throughout the agent lifecycle, it's important to revisit manual testing in Agentforce Builder, especially after making changes to your agent.

In manual testing, you'll use the Conversation Preview panel to chat with your agent, then track the agent's topic selection, action execution, and reasoning on the plan canvas. The plan canvas is designed to give you a clear view into the AI agent's decision-making, so you can see how it chooses topics and actions.

Makana Medical Devices has built their agent to handle general FAQ questions from customers, such as inquiries about product availability, warranty policies, and basic troubleshooting for their medical devices. Before fully rolling out the agent, the support team wants to confirm that it accurately understands and responds to common questions.

1. From the Agentforce Agents Setup page, launch the Makana Customer Support Agent in Agentforce Builder.
2. Using the Conversation Preview panel, enter a typical customer question: `What is the warranty period for your CGM-3000 glucose monitor?`.

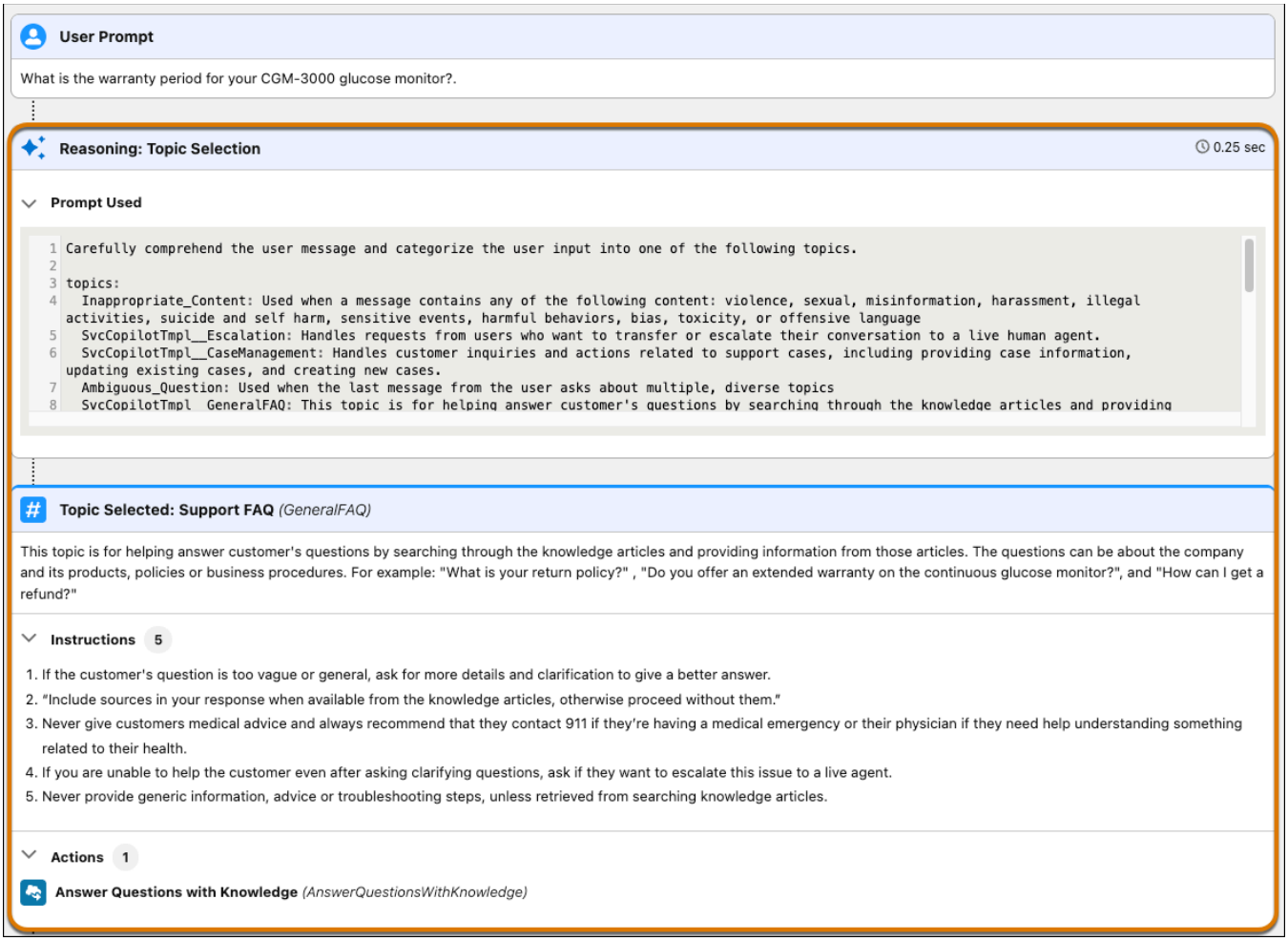
3. In the plan canvas, check the agent's topic selection. If the agent has correctly identified the Support FAQ topic for this query, the classification and topic instructions are functioning properly.
    - If it had chosen an unrelated topic, review and adjust the topic's classification description to better guide the agent.
4. Next, observe which actions the agent selects within the Support FAQ topic. The agent should correctly choose the Answer Questions with Knowledge action.
    - If it had chosen an unrelated action, adjust the action's instructions or refine when specific actions should be triggered.
5. Since this action simply provides information, check the response in the Conversation Preview panel to confirm it matches the company's official warranty policy. If incorrect, update the product information PDF. Additionally, deactivate the agent and add clarification notes to ensure the agent references the most recent warranty policy documents.
6. Review the agent's reasoning in the plan canvas, following how it parsed the user's question and arrived at the topic and action decisions. This step confirms that the AI's decision-making aligns with expectations.
    - If the agent's reasoning showed any unexpected or missed steps, confirm that the agent correctly uses context and memory, and grounds responses appropriately.
7. After making any necessary adjustments to topics or actions, restart the conversation to test the improved agent responses. This iterative process continues until the service agent consistently delivers accurate and helpful answers.

How did Makana assess agent performance?

The process above shows how Makana assessed their agent's quality through four checks. Let's take a closer look at each one.

Check the Topic Selection

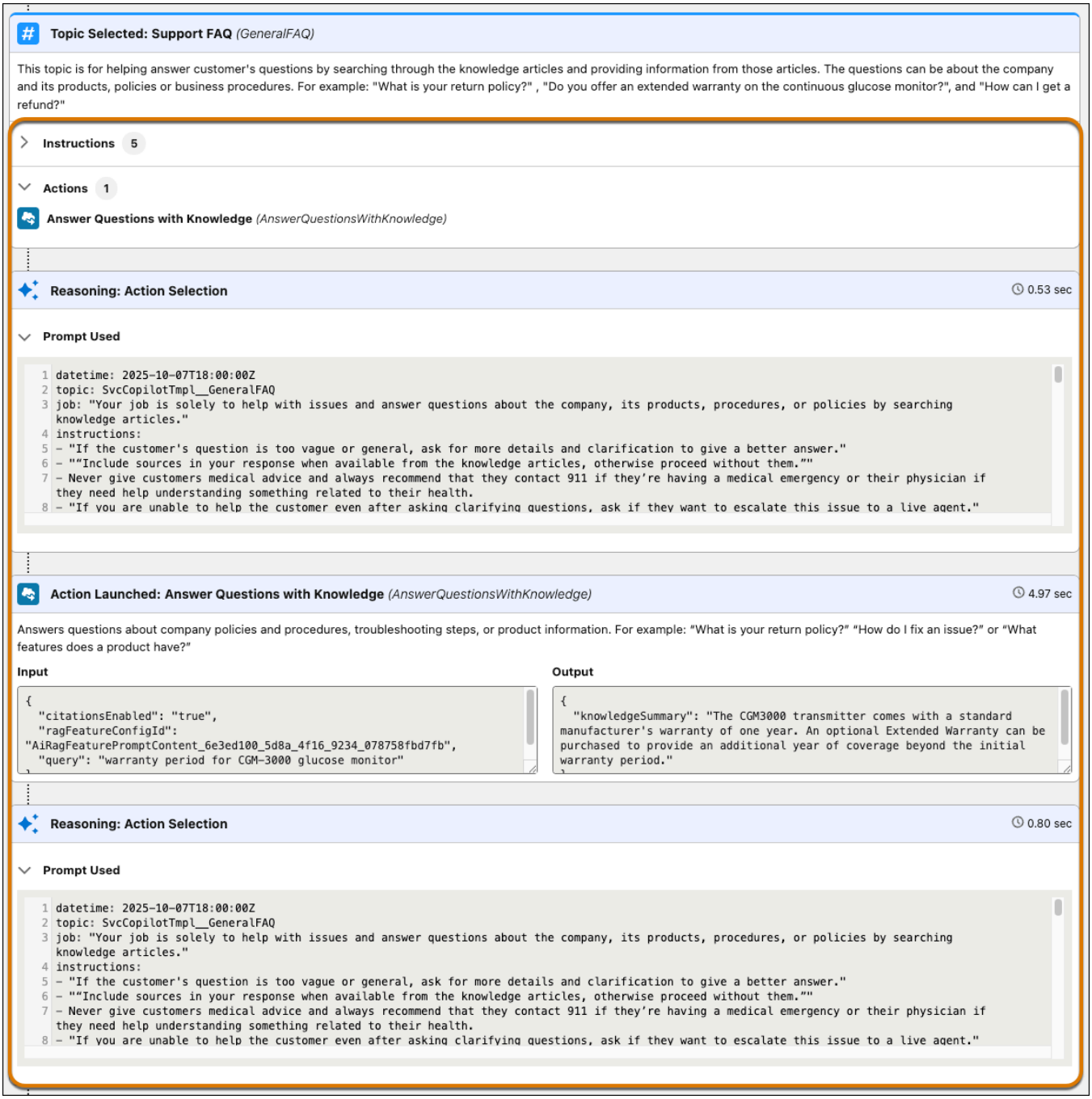
Start by checking the initial utterance and topic selection to verify the agent correctly classified the user's utterance. In the plan canvas, you'll see the user's utterance and the topic the agent chose to address it. For example, for the query about the warranty information, we can see that the agent correctly selected the Support FAQ topic. If the agent selected an unexpected topic, that indicates that you should adjust the topic's classification description or instructions.



To adjust a specific topic's instructions, from the Topics tab, select the topic you want to improve. In Topic Details, under Topic Configuration, you'll see the topic's classification description, scope, and instructions. You can directly compare the user utterance with the topic instructions. For instance, if your Technical Support topic instructions specify, "Only handle issues for products under warranty," test your agent's instruction adherence by asking questions about an out-of-warranty product. If the agent correctly follows the instructions, it should escalate the issue or decline to help.

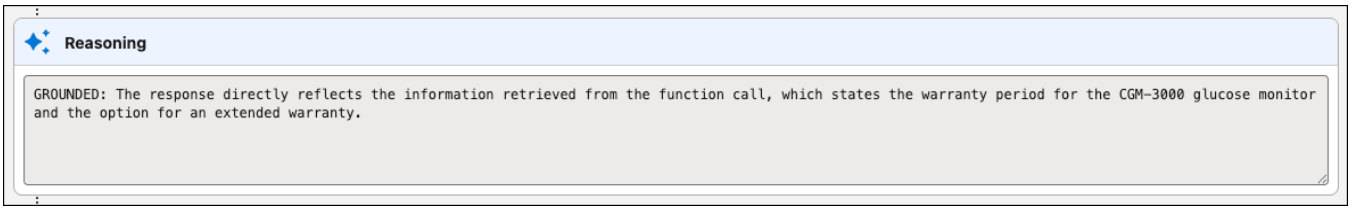
Check the Action Selection

Next, take a look at the action selection. In the plan canvas, you'll see the agent analyzing the user's utterance and then identifying the most relevant actions in the topic. For a Case Management query, you'd expect to see it evaluate and select from the actions Create Case, Get All Cases for Contact, and Get Case by Case Number. If the agent selects an irrelevant action, that indicates that you should adjust the instructions for that action, or that you need to adjust the topic instructions for when and how to use that action.



Check the Agent's Reasoning

After reviewing the agent's topic and action selection, turn your attention to its reasoning. This is where the agent's "intelligence" is revealed. Use the plan canvas to track and validate the agent's decision-making, ensuring it makes accurate predictions and consistently follows regulatory rules, like verifying the user before sharing personally identifiable information (PII). The plan canvas helps you confirm that the agent correctly uses context and memory, carrying key information like a device serial number, throughout the conversation. To prevent hallucinations or non-compliant responses, check the response is adequately grounded to ensure all user-facing information is sourced exclusively from approved, reliable data.



Learn More: *For more information about the reasoning engine, see How Agentforce Works.*

Check Your Other Salesforce Functions

Because each action runs a specific flow, Apex class, prompt template, or other Salesforce functionality, issues can stem from these areas rather than the agent instructions. After running a test, confirm that the action executed as expected by comparing the results to the action instructions. For example, if the agent chose the Create Case action, you can manually verify that a new case was actually created in Salesforce with all of the correct information populated from the conversation.

If your agent action uses underlying sources through an Agentforce Data Library, verify the agent's response against the source content. Consider refining the source content in the data library or limiting the scope of sources that your action pulls from. This is particularly effective if the issue affects only one or a few of the sources in your database. You can tell the agent referenced information in a data library if it runs the Answer Questions with Knowledge action.

Batch Testing and Automated Evaluation

Testing individual utterances in Agentforce Builder is a good strategy for targeted testing, topic and action refinement, and troubleshooting. But it can be time consuming if you want to get an overall idea of your agent's reliability across a broader set of real-world scenarios. Take advantage of automation in the testing process to help you efficiently launch reliable and trustworthy agents. In the Testing Center, you can streamline large-scale testing by quickly generating up to 100 realistic test scenarios using AI.

Once Makana Medical Devices completes the manual testing of their customer support agent, they can move on to batch testing to evaluate how well the agent performs at scale. Using the Agentforce Testing Center, the Makana team can quickly generate hundreds of test cases, run evaluations, and pinpoint where their agent needs improvements. For AI-generated test cases, you can create test cases based on relevant topics and actions. Or you can create question-and-answer style test cases based on the knowledge content available to the agent.

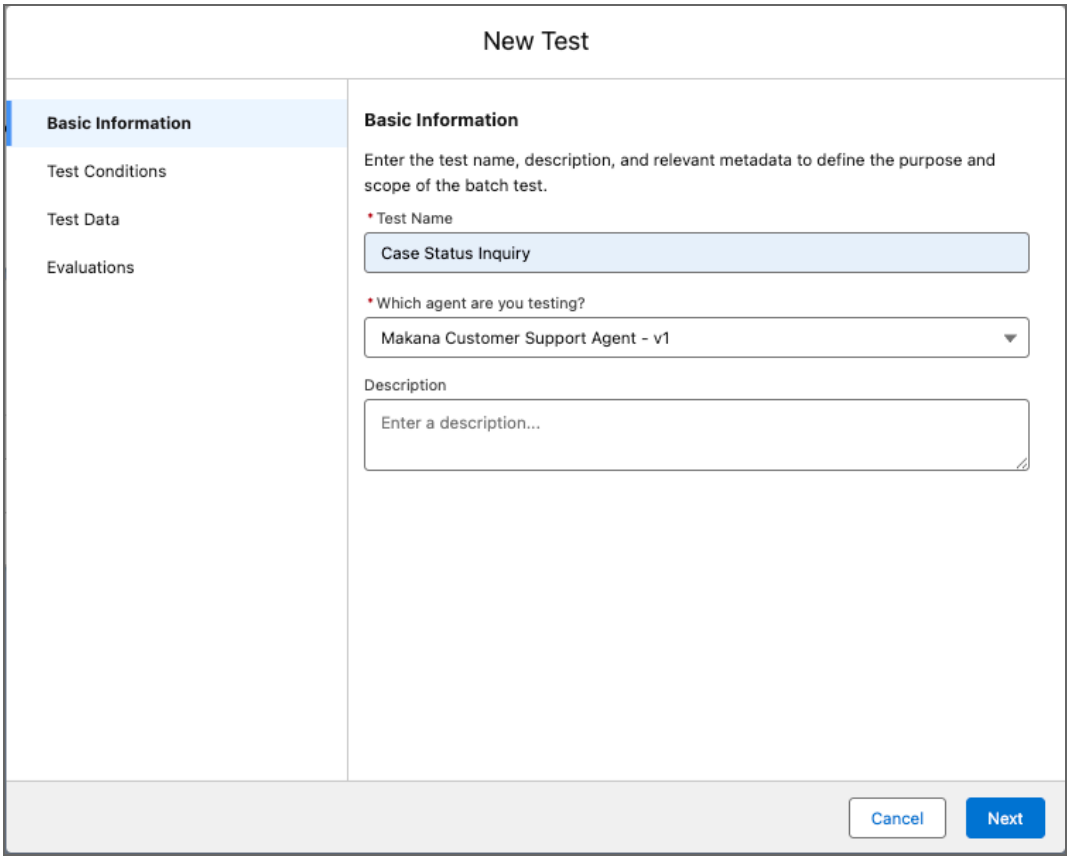
Makana puts together a batch test plan to organize their testing strategy and ensure coverage across their scenarios. They address their Case Status Inquiry scenario first.

Batch Test Plan		
Scenario	Case Status Inquiry	
Agent	Customer Support Agent	
Test Type *Consider the type of tests you want to generate.*	☑ Topics and Actions *Use when you want to verify that your agent correctly routes conversations using predefined topics and actions, such as opening cases, escalating issues, or updating records.* ☐ Knowledge-Based Question and Answer *Use to verify the agent retrieves accurate, conversational answers from knowledge bases, like uploaded PDFs or knowledge articles.*	
Goal *What are you trying to validate? What specific knowledge do you want the agent to retrieve and how should it be presented?*	Ensure the agent retrieves and communicates accurate case status information from Salesforce objects using structured topics and actions.	
Agent Setup *Which agent are you testing and what actions, topics or resources is it connected to?*	Topics: • Case Management • Escalation • Support FAQ Actions: • Get All Cases for Contact • Escalate Case • Answer Questions with Knowledge	Data Library Resources: • Warranty policy • Product registration • Troubleshooting guides
Test Description *Describe the purpose of the test and scope like what the utterances should trigger or cover. Include key terms, tone types, and any constraints.*	A customer contacts the agent asking for an update on a previously submitted support case regarding their glucose monitor. They expect the agent to retrieve their case details and provide a current status. If the agent fails to locate or correctly respond with case information, the customer may get frustrated or escalate the issue.	
Context Variables *Identify and include the variables that complement your scenario.*	Contact Id	

Batch Test Plan	
Response Quality Evaluations *Choose what to measure based on the scenario. Describe the purpose of each evaluation.*	
☐ **Completeness**	
☑ **Coherence**	Selected to assess if the agent transforms underlying information from Salesforce Objects into grammatically correct, conversational language.
☑ **Conciseness**	Selected to ensure the status update is short but accurate, successfully capturing the essence of the desired content.
☐ **Latency**	
☐ **Instruction Adherence**	

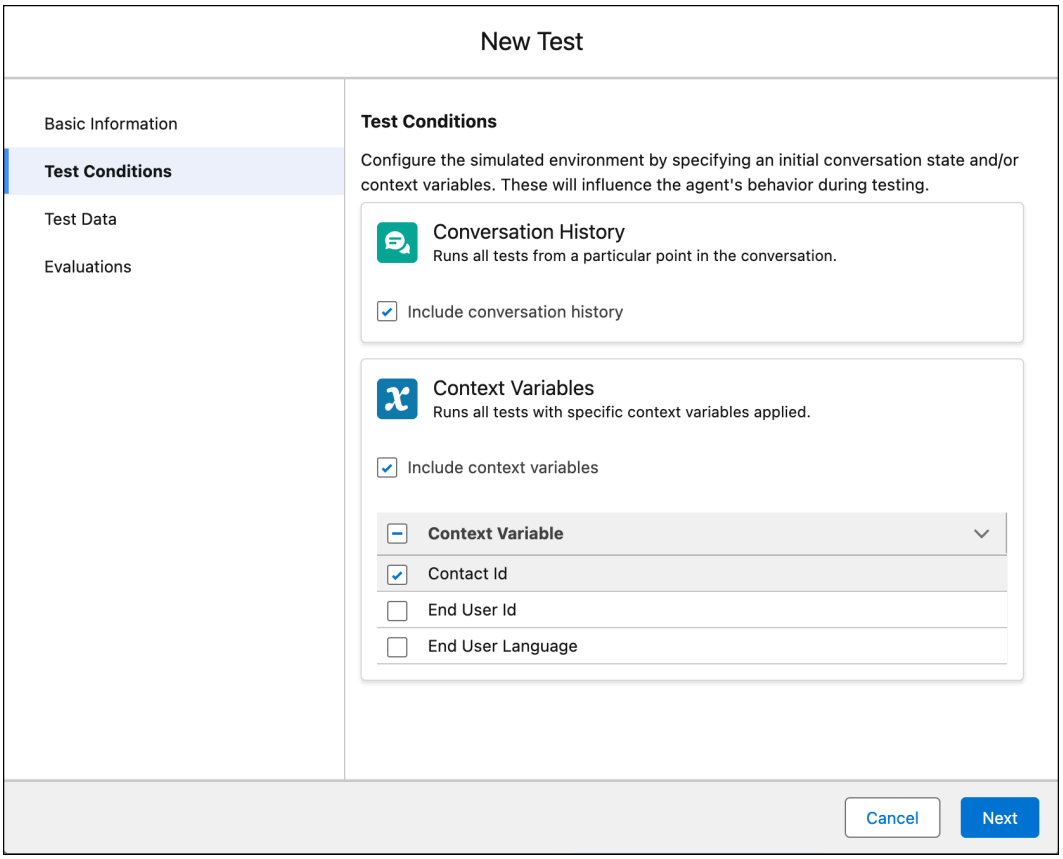
Based on this plan, Makana follows these steps to create their batch tests.

1. To begin batch testing, click **Batch Test**.
2. The agent Makana Customer Support Agent is automatically selected. Name the test `Case Status Inquiry`.



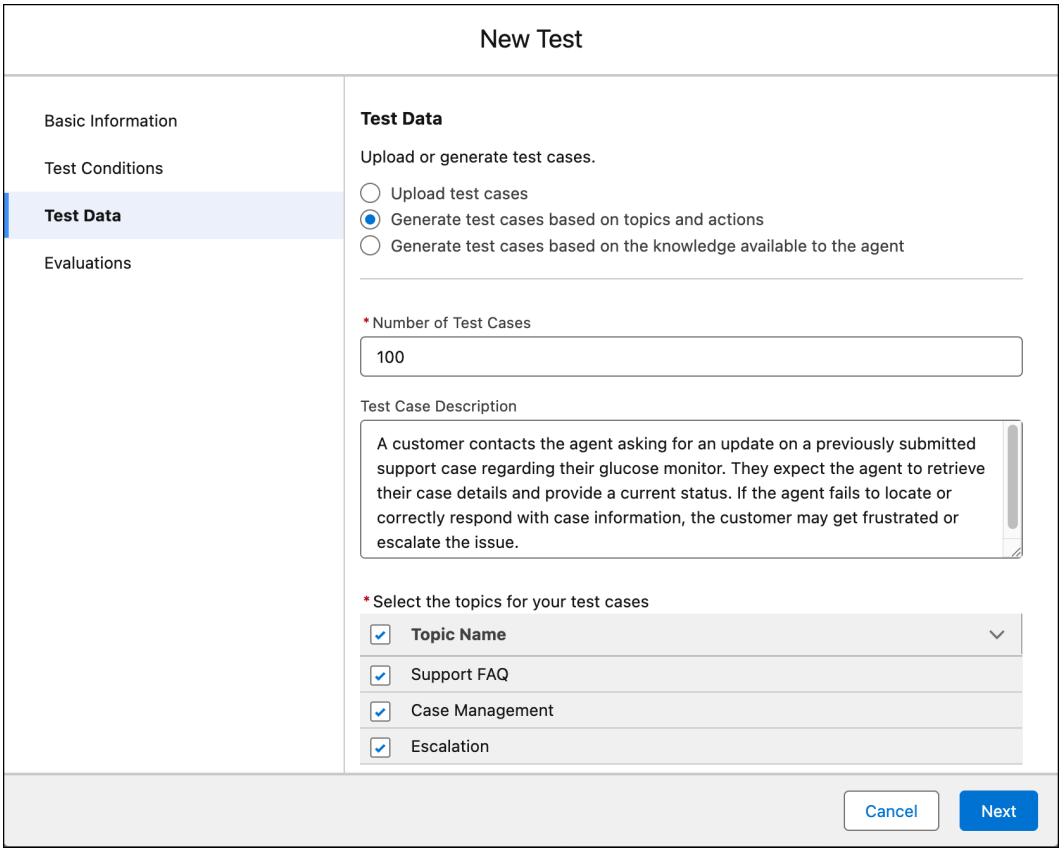
When setting up the test, Makana can include conversation history and context variables to more accurately simulate real-world support scenarios.

3. Since Makana initiated their batch tests from their previous manual testing in Agentforce Builder, they opt to include that conversation history in their test cases.
4. Select **Include context variables** and add the **Contact Id** from related Salesforce objects to add context for their agent's responses. For example, if the contact recently interacted with the support team, the agent might reference that prior conversation. If there's no recent interaction, it could start with a general greeting and ask how it can help. Click **Next**.



Makana wants to validate their agent's conversation flow, topic selection, and actions.

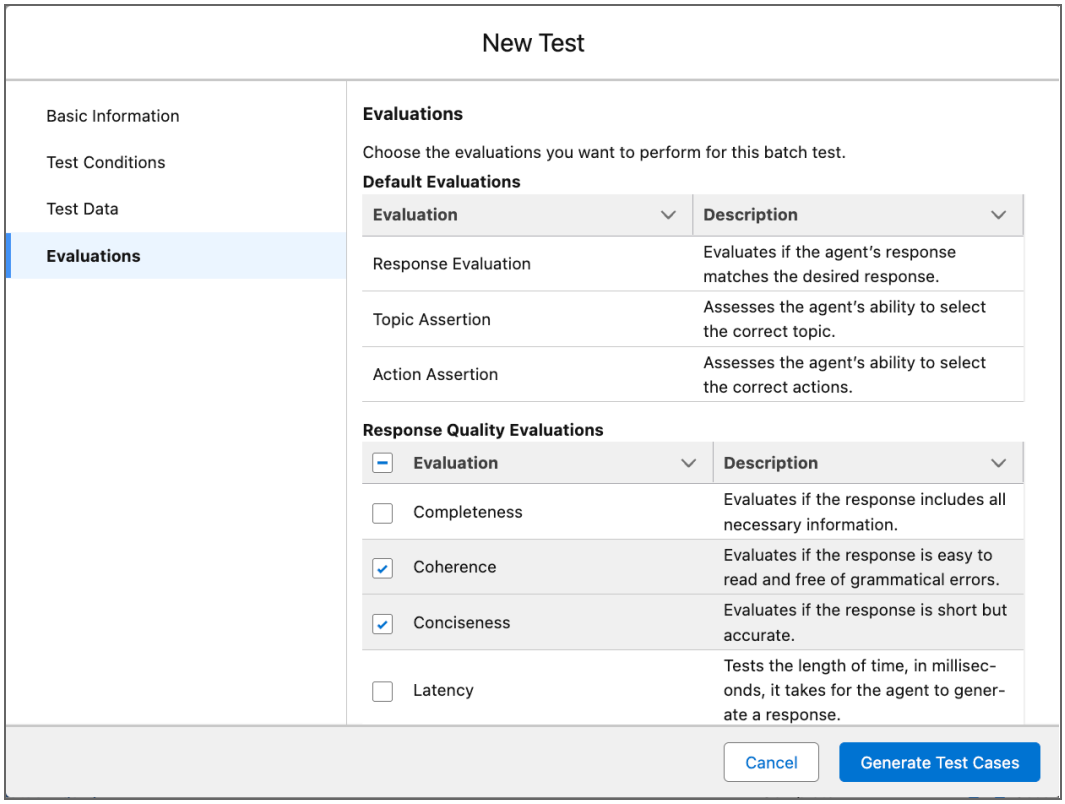
5. Makana opts to generate test cases based on the key topics and actions tied to their business objectives in their Case Status Inquiry testing scenario. So, select **Generate test cases based on topics and actions**.
6. They want a broad selection of tests, so ask the system to create the maximum 100 tests.



7. To guide the AI in generating appropriate test cases, enter a test description:
   ```
   A customer contacts the agent asking for an update on a previously submitted support case regarding their glucose monitor. They expect the agent to retrieve their case details and provide a current status. If the agent fails to locate or correctly respond with case information, the customer may get frustrated or escalate the issue.
   ```
8. Makana wants to start broad and identify any gaps in how the agent handles different case-related queries. To do this, include all available topics, **Support FAQ**, **Case Management**, and **Escalation**. Click **Next**.

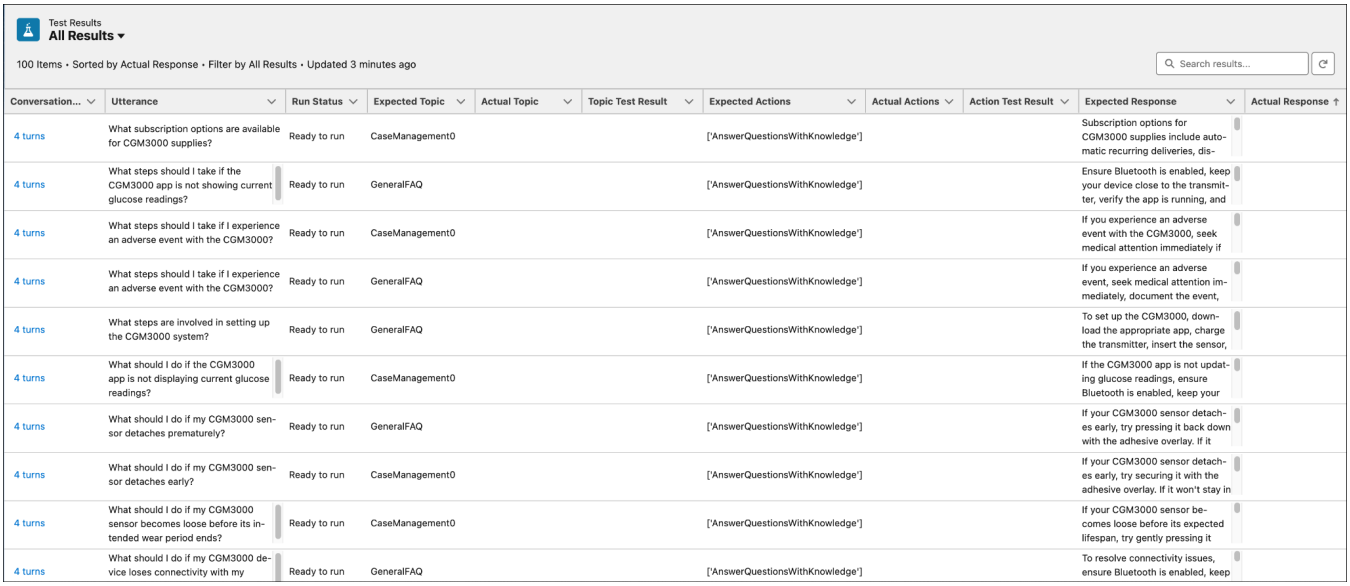
In the Case Status Inquiry scenario, Makana's testing goal is to ensure the correct topic and action are selected, and that case data retrieved from Salesforce is communicated to the customer in a clear, conversational, and brand-aligned manner.

9. To evaluate their responses against these goals, for the Response Quality Evaluations select **Conciseness** and **Coherence**.



10. Finally, click **Generate Test Cases**.

When Agentforce finishes generating the tests, they'll appear in the Testing Center.



When the status says Ready to Run, click **Run Test Suite**. Agentforce begins running their chosen evaluations on the test cases.

Note: Refresh the testing suite to see updates to the test status.

Use Test Results to Improve Your Agent

Once your tests finish running, you'll see overall metrics for the test suite along with evaluation scores for each test. Use these results to identify failed or low-scoring utterances, then manually retest them in Agentforce Builder. Refine the instructions within your topics and actions, and iterate until you're satisfied with the responses. Pay close attention to recurring failures as these results can help uncover underlying issues such as unclear guidance, missing knowledge, or configuration issues. To interpret your results, see how each evaluation helps to measure agent performance.

Default evaluations are given a score of 1, pass, or 0, fail.

Evaluation	Testing Significance
Response Evaluation	Scores of 5 indicate that the agent generally succeeds in achieving its goal. A score of 5 reflects a precise, complete, and brand-aligned response with no irrelevant content. Scores of 3 to 4 show decreasing levels of clarity and completeness. There may be minor omissions or partial understanding, which introduces some ambiguity while the core intent is still addressed. Scores of 1 to 2 suggest the agent struggled to meet the goal. The response may be unclear, missing key elements, or include irrelevant information. It might also ask the user for information that should have been retrieved from the CRM. A score of 0 represents a complete failure to resolve the user's query. The response is generic and doesn't address the user's intent. **Tip:** Check your agent configuration for issues related to topic selection, instructions, or actions. Consider whether there are knowledge gaps, like outdated articles, or other issues preventing the agent from responding appropriately.
Topic Assertion	A score of 1 indicates that the agent correctly identified the appropriate topic to address an utterance. A score of 0 indicates that the agent selected an unexpected topic to address an utterance. **Tip:** Manually retest any failed utterance in Agentforce Builder and review the agent's reasoning in the plan canvas. Refine the instructions for both the expected action and the topic itself to

	clearly guide the agent toward the correct choice and restrict the use of the incorrect action.
Action Assertion	A score of 1 indicates that the agent correctly identified all of the appropriate actions within a topic to address an utterance. A score of 0 indicates that the agent either chose the wrong actions or failed to select all necessary actions within a topic to address an utterance. **Tip:** Manually retest any failed utterance in Agentforce Builder and review the agent's reasoning in the plan canvas. Refine the instructions for both the expected action and the topic itself to clearly guide the agent toward the correct choice and restrict the use of the incorrect action.

Response quality evaluations are primarily scored between 0 and 5, with 3 or higher signifying a pass. Instruction Adherence is uniquely scored as High, Low, or Uncertain. The LLM-as-judge provides the reasoning for each score in an infobubble.

What is an LLM-as-judge?
LLM-as-judge is when one large language model (LLM) evaluates the outputs of another LLM, serving as a scalable, automated, and objective evaluation tool for tasks like scoring summaries or ranking responses. A judge LLM receives a prompt that includes the task and evaluation criteria to measure things like factual accuracy, relevance, coherence, and adherence to source. With the task and criteria, the LLM judge determines the expected response and compares it to the agent response, then generates scores, rankings, or textual feedback. We've carefully designed our LLM-as-judge prompts to give you the most accurate and useful test results.

Evaluation	Testing Significance
Completeness	A score of 5 indicates a fully complete and accurate answer with no important omissions. Scores of 3 to 4 reflect decreasing levels of completeness, with minor to moderate gaps that may slightly affect understanding. Scores of 1 to 2 indicate that the generated answer is significantly incomplete, missing several or most important pieces of information, which may cause confusion or a misleading result. A score of 0 represents a failure, signifying the answer missed all the important pieces of information and is highly confusing or misleading.

Coherence	Scores from 3 to 5 indicate that the agent correctly transforms underlying information into conversational language with the appropriate sentence and grammatical structures, ensuring that the dialogue flows smoothly and is easily understood by the user. Scores from 0 to 2 indicate the response is grammatically incorrect and unclear, or information has been taken from Salesforce objects and delivered as raw data like JSON structures or direct field content.
Conciseness	Scores from 3 to 5 indicate that the agent's response is short but accurate, successfully capturing the essence of the desired content. Scores from 0 to 2 mean the generated response is lengthy, repetitive, includes unimportant points, or contains irrelevant content.
Latency	A high or unusual latency in a test utterance suggests a problem with either the utterance itself or some underlying infrastructure. If agent adjustments don't resolve the issue, contact Salesforce support to check for infrastructure issues.
Instruction Adherence	**High:** The agent interprets and fully follows the topic instructions, both addressing key points and providing any required information. **Low:** The agent doesn’t interpret or follow the topic instructions accurately. It fails to follow at least one instruction, leading to an incorrect response. This flags a need to refine instructions and set clearer constraints. **Uncertain:** Instruction adherence can't be determined due to an ambiguous response or incomplete response, or conflicting interpretations of the topic instructions.

While default evaluations are scored by comparing expected and actual results, the judge-LLM assesses quality evaluations from fixed criteria. If a response receives a low score, review it carefully to see whether it truly falls short of your expectations. For example, the judge-LLM may assign a low conciseness score, but you might feel that additional context better serves your customers or aligns with your brand voice. Otherwise, you can add extra instructions or guardrails to better tailor the agent to your goals. Always consider your specific goals and use case when interpreting scores. Quality evaluations provide guidance, but they are not absolute measures of your agent’s success or failure.

Let's revisit Makana's Case Status Inquiry batch test to see how they used evaluation results to improve their agent. After running the test, the results revealed a low Action Pass Rate and Makana identified a pattern of failures tied to customer utterances about return updates:

Utterance	Expected Topic	Expected Action	Actual Topic	Actual Action	Result
"I sent back my glucose monitor and haven't heard anything. What's going on with my case?"	Case Management	Get All Cases for Contact	Case Management	Create Case	Fail (0)

The Action Assertion scored a 0, which indicates that the agent either chose the wrong action or failed to select all necessary actions within the topic. In this case, the agent misinterpreted the request for a status update as a need to generate a new case.

Makana follows the recommended process for using failed test results to troubleshoot the agent.

1. **Manually Retest in Agentforce Builder**
   Copy the failed utterance, `I sent back my glucose monitor and haven't heard anything. What's going on with my case?`, and enter it into the Conversation Preview panel in Agentforce Builder.
2. **Review and Analyze**
   In the plan canvas, the team confirms that the agent correctly selected the Case Management topic but incorrectly chose the Create Case action. By reviewing the agent's reasoning, they deduce that the urgency and phrasing of the customer's input ("haven't heard anything," "What's going on") caused the agent to prioritize the creation of a new case over retrieving existing case information.
3. **Refine Action and Topic Instructions**
   Since the issue is related to choosing the wrong action, focus on adjusting the instructions for the actions and the topic. Review the instructions for the expected action, Get All Cases for Contact, and ensure they clearly guide the agent to use this action when the user is inquiring about status, updates, or existing cases. Then check the instructions for the unexpected action, Create Case, and refine them to be more restrictive: `Only use this action if the user explicitly requests a new support case or if all attempts to locate an existing case or provide an adequate status update have failed. Prioritize checking existing case status first.`

   Additionally, add instructions to the Case Management topic itself, providing clear guidance on when to launch specific actions. `If the customer uses language indicating frustration or a request for status, prioritize the Get All Cases for Contact action before considering case creation.`
4. **Verify and Re-test**

The team restarts the conversation in the Agentforce Builder and enters the failed utterance again. They confirm that the agent now correctly selects the Get All Cases for Contact action and provides a status update, aligning with the expected response.

5. **Scale the Fix**
   To ensure the fix didn't introduce new issues and that it now reliably addresses similar utterances, Makana generates similar test cases targeting the Get All Cases for Contact action under the Case Management topic and runs another batch test.

Iteration and Refinement

Agent testing isn't a one-and-done project. New features roll out, new data is added, and LLMs are updated, all of which can contribute to your agent's behavior changing over time. Regular batch testing and refinement is critical to make sure the agent is still performing well and meeting your business objectives.

Batch test results offer more than just a simple pass/fail; they provide valuable insights for proactively refining your agents. Through various metrics and strategic actions, you can transform agent maintenance from a reactive, bug-fixing process into a proactive, data-driven cycle of continuous improvement.

The Testing Center provides a crucial capability for maintaining the quality and performance of your agents: the ability to save and track batch test results over time. This historical data is invaluable for identifying agent drift, a phenomenon where an agent's performance gradually degrades due to various factors.

By regularly executing the same comprehensive test suite against your agents, you establish a baseline for their expected behavior and performance. This consistent testing allows you to spot even subtle behavioral changes that might arise from updates to underlying large language models (LLMs) or modifications to data access. For instance, a new LLM version, while generally improved, might introduce regressions in specific, niche scenarios that your existing agent handles well. Similarly, changes to data sources or retrieval methods could inadvertently impact an agent's ability to process information accurately.

To accurately measure the impact of any major changes you introduce to your agents – whether it's an architectural overhaul, a significant data update, or a new LLM integration – always run a batch test before and after the change. This pre- and post-comparison provides a clear, quantitative assessment of the modification's effect on performance. If the post-change test results reveal a performance drop, you gain immediate visibility into the issue. This rapid detection is critical, enabling you to quickly roll back the problematic change, or refine it until performance is restored. This proactive approach prevents regressions from reaching production and negatively impacting user experience, ultimately safeguarding the reliability and effectiveness of your agents.

While batch testing is powerful, it can't capture everything. It's important to integrate human feedback into your process. For example, a batch test might confirm factual correctness, but a

human can judge tone and empathy in an utterance. Teams can then flag specific interactions for review.

To further enhance your testing and refinement process, versioning is essential for isolating changes and finding the source of issues. By creating a version of your agent, you can make changes to it and test them without having to take an agent out of production. You can create up to 20 versions of your agent and use them to test different parts of a major change. By treating every change as a distinct, traceable unit, you can rewind your project's history, compare different versions, and pinpoint exactly when a problem was introduced. This turns the chaotic process of debugging into a structured, forensic investigation.

To make versioning a powerful debugging tool, you need to follow a few key practices. Each version should represent a single, logical change. Instead of performing multiple bug fixes and refactors all at once, break it down into small, focused changes. This makes it easier to isolate the exact issue that caused a bug. Additionally, make sure you capture what each change was and why you made it. It's a breadcrumb trail for your future self and other collaborators.

Learn More: *For more information about debugging agents, see Troubleshooting Agents.*

Deploy Your Agent

Following Makana Medical Devices' successful ideate, build, and test phases, the deploy stage outlines the step-by-step process for deploying an Agentforce Service agent to an Experience Cloud website. This phase is crucial for transforming Makana Medical Devices' defined use cases into tangible customer support solutions.

In the deploy stage, you prepare your AI agent for end-users. This process involves launching your deployment strategy, setting up the agent's connections, and deploying your agent. A deployment strategy helps you maximize your AI agent's success, minimize disruption, anticipate challenges, manage stakeholder expectations, and gather critical feedback for continuous improvement. The components of an effective deployment strategy include a rollout plan, which often begins with a pilot program. After refining the agent's performance, you can gradually roll it out to all of your end-users.

AI projects also require robust change management strategies to reduce risk and ensure successful adoption. Factors that impact adoption of AI agents include AI maturity, organizational readiness, workflow integration for AI tools, ethical AI governance, training plans, effective communication, executive sponsorship, a culture of AI experimentation and transparency, and feedback channels. For more information about agent deployment strategies, see Agentforce Deployment: Quick Look.

Starting here? *Make sure you have the right permissions for the job.*

Before You Begin

Here are the prerequisites for this chapter:

- Turn On Enhanced Omni-Channel Routing
- Turn On Messaging
- Turn On Digital Experiences

Learn About Connections

Channels are the platforms, apps, and interfaces that you can deploy an agent to, such as the Agentforce panel in Lightning Experience, enhanced messaging channels, Slack, and email.

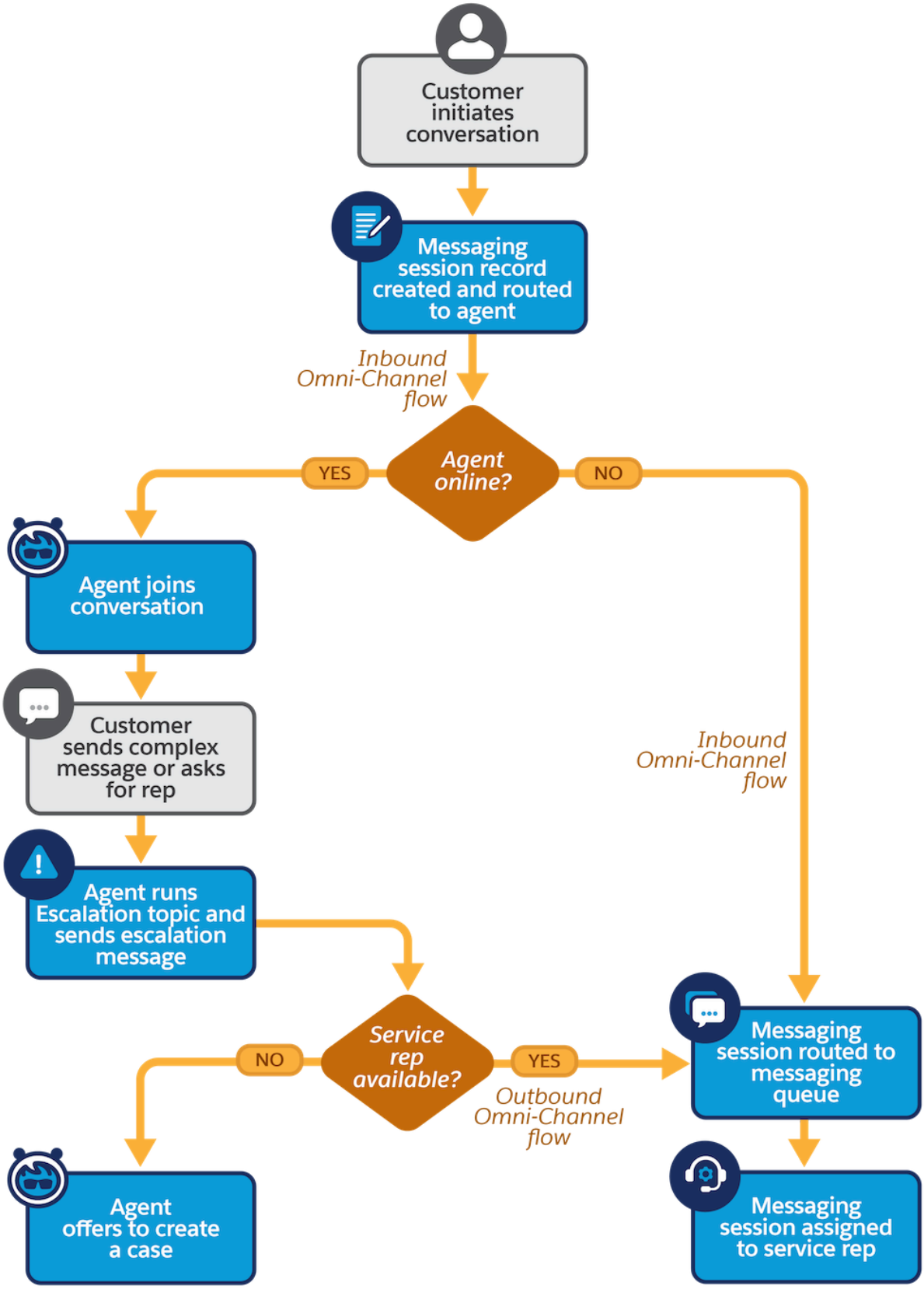
Connections help you scale agent development and reduce repetitive setup by letting you build an agent once and easily add it to multiple channels. Each connection includes a surface that gives the agent context about how to reason and respond appropriately for one or more channels. The surface contains adaptive response formats that tell an agent how to structure responses and deliver multimedia content, such as images, buttons, links, and videos on a variety of channels. You can also find the Omni-Channel flows that route conversations to and from an agent in your connection settings.

For the Makana Customer Support agent, you'll create an Enhanced Web Chat channel that can be embedded on a website. The Messaging connection is associated with all enhanced Messaging channels, including Enhanced Web Chat, and is automatically included with all Agentforce Service agents. When a customer sends a message to the Makana Customer Support agent on an Enhanced Web Chat channel, the agent starts using the Messaging connection, surface, and adaptive response formats to reason and respond. It also uses the Omni-Channel flows you'll add in the Messaging connection settings.

Learn More: *For more information about agent deployment concepts, see Deploy Your Agent to Channels.*

Route Messaging Conversations to and from the Agent

To help the Makana Customer Support agent give your customers a smooth routing experience, set up a service channel, messaging queue, routing configuration, and Omni-Channel flows. Salesforce creates a messaging session record for every conversation. The service channel connects your routing configuration and Omni-Channel flows to each session, and the routing configuration makes sure conversations are routed according to your specified rules. The messaging queue assigns messaging sessions to your service reps when the agent is unavailable or escalates a conversation. The inbound Omni-Channel flow directs incoming conversations to the Makana Customer Support agent and, if the agent's not available, it sends conversations to the queue. The outbound Omni-Channel flow and Escalation topic help your agent route escalated conversations to the queue. You can also customize the flow and topic so that your agent creates a case when a service rep isn't available.


Customer initiates conversation
Messaging session record created and routed to agent
Inbound Omni-Channel flow
Agent online?
YES
NO
Agent joins conversation
Customer sends complex message or asks for rep
Agent runs Escalation topic and sends escalation message
Inbound Omni-Channel flow
Service rep available?
NO
YES
Outbound Omni-Channel flow
Messaging session routed to messaging queue
Agent offers to create a case
Messaging session assigned to service rep

Learn More: *For more information about routing, see these resources.*

- Create Service Channels
- Set Up Queues
- Route Work with Omni-Channel
- Build a Flow

Create a Service Channel

The service channel links your routing configuration and Omni-Channel flows to the Messaging Session object.

1. From Setup, in the Quick Find box, enter `Service Channels`, and then select **Service Channels**.
2. If you already have a Messaging service channel, go to the next step. If not, click **New**.
    a. For the name of the service channel, enter `Makana Help Center`.
    b. For the Salesforce object, select **Messaging Session**.
3. Save the service channel.

Create a Routing Configuration

The routing configuration defines how messaging sessions in the queue you create next are assigned to service reps.

1. From Setup, in the Quick Find Box, enter `Routing`, and then select **Routing Configurations**.
2. Click **New**.
    a. For the routing configuration name, enter `Messaging Routing`.
    b. For the routing priority, enter `1`.
    c. For the routing model, select **Least Active**.
    d. For the capacity type, select **Inherited**.
    e. For the percentage of capacity, enter `25`.
3. Save the routing configuration.

Create a Queue

When your inbound Omni-Channel flow can't route a conversation to your agent, or when your agent escalates a conversation, the conversation is routed to this queue. The queue assigns the messaging session to service rep based on your routing configuration settings.

1. From Setup, in the Quick Find Box, enter `Queues`, and then select **Queues**.
2. Click **New**.
    a. For the label, enter `Messaging`.
    b. For the routing configuration, enter `Messaging_Routing`.
    c. For the supported objects, select **Messaging Sessions**.
    d. For the queue members, select yourself as the service rep.
3. Save the queue.

Create an Inbound Omni-Channel Flow

Omni-Channel flows make it easy to route agent conversations. These flows use the Route Work action to route a conversation and its associated records, such as messaging session and email records.

Create an inbound omni-channel flow that routes messaging sessions from the Experience Cloud messaging deployment to the Makana Customer Support agent. When a customer interacts with an agent on a channel, a messaging session record associated with the conversation is created. Then, the inbound Omni-Channel flow routes the record from the channel to an agent.

The flow receives the messaging session record and details gathered from the pre-chat form. Then, it checks if the messaging user is an existing contact. Finally, it uses the Route Work action to route the session to the agent. If routing to the agent isn't possible, the flow sends the messaging session to the Messaging queue instead.

1. From Setup, in the Quick Find box, enter `Flows`, and select **Flows**.
2. Click **New Flow**.
3. In the search field, enter `Omni`, and select **Omni-Channel Flow**.
4. In Flow Builder, on the Manager tab of the toolbox panel, create these resources.

   recordId: The messaging channel uses this variable to pass the messaging session record ID to the Omni-Channel flow.

Field	Value
Resource Type	Variable
API Name (case-sensitive)	recordId
Data Type	Text
Availability Outside the Flow	Available for input

   input_record: The messaging channel uses this variable to pass the messaging session record to the Omni-Channel flow.

Field	Value
Resource Type	Variable
API Name (case-sensitive)	input_record
Data Type	Record

Object	Messaging Session
Availability Outside the Flow	Available for input

firstName: This variable holds the value of the First Name field on the pre-chat form you create later.

Field	Value
Resource Type	Variable
API Name	firstName
Data Type	Text
Availability Outside the Flow	Available for input

lastName: This variable holds the value of the Last Name field on the pre-chat form you create later.

Field	Value
Resource Type	Variable
API Name	lastName
Data Type	Text
Availability Outside the Flow	Available for input

email: This variable holds the value of the Email field on the pre-chat form you create later.

Field	Value
Resource Type	Variable
API Name	email
Data Type	Text
Availability Outside the Flow	Available for input

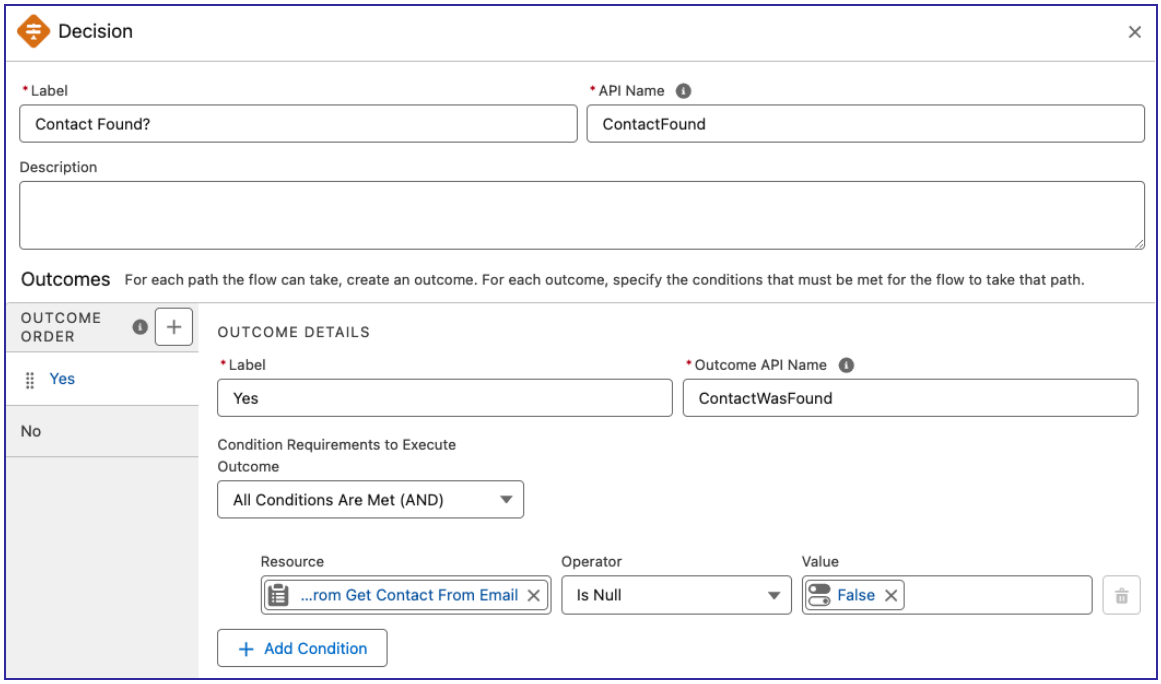
MessagingUser: The messaging channel uses this variable to pass the messaging end user record to the Omni-Channel flow.

Field	Value
Resource Type	Variable
API Name	MessagingUser
Data Type	Record
Object	Messaging User
Availability Outside the Flow	None

MessagingUserFullNameFormula: This formula concatenates the messaging end user's first name and last name.

Field	Value
Resource Type	Formula
API Name	MessagingUserFullNameFormula
Data Type	Text
Formula	{!firstName} + ' ' + {!lastName}

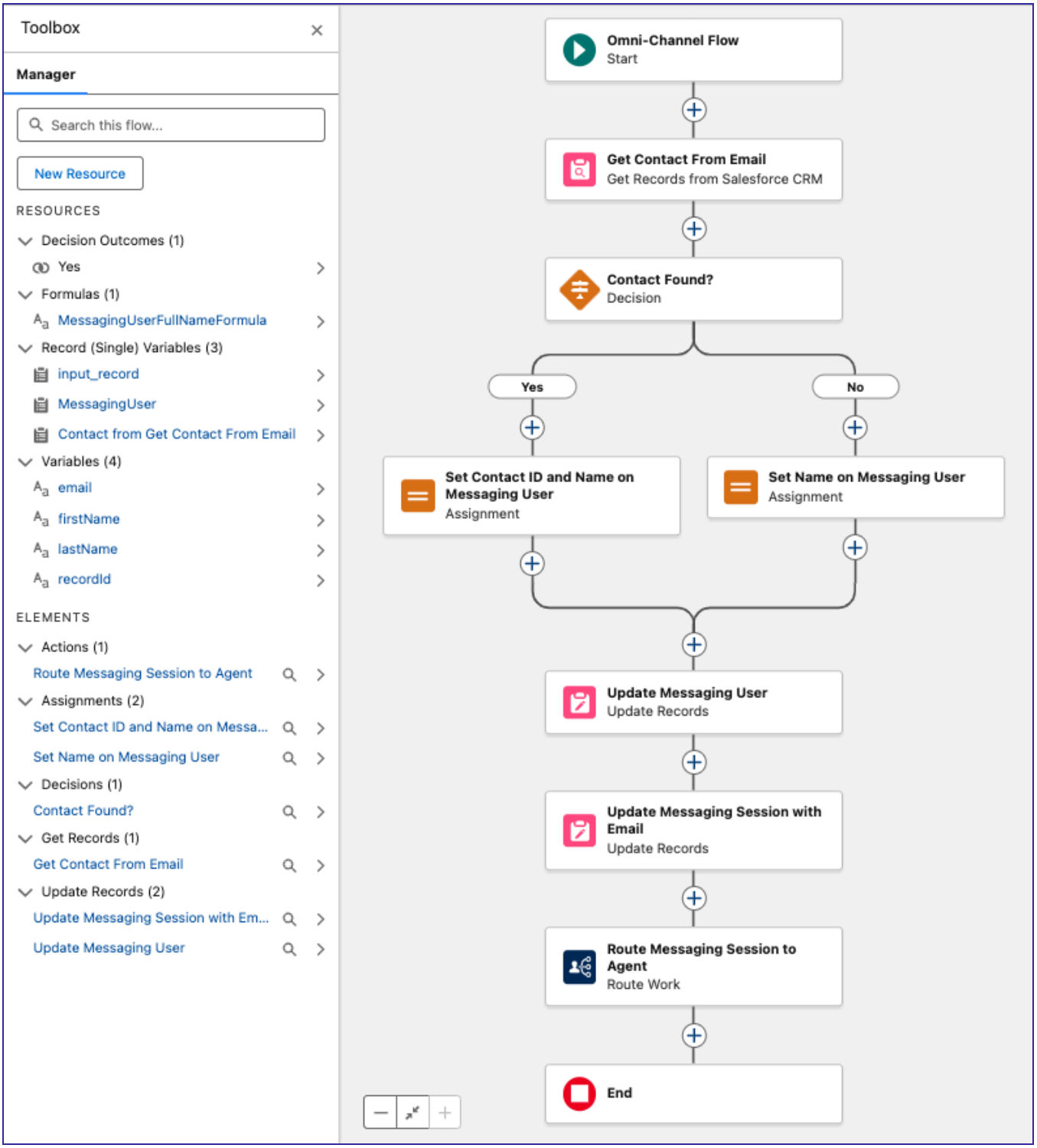
5. Add a Get Record element that retrieves the contact record associated with the email address from the pre-chat form.
    a. For the label, enter `Get Contact from Email`.
    b. For the data source, select **Salesforce Object**.
    c. For the object, select **Contact**.
    d. For the condition requirements, select the **Email** field, **Equals** operator, and **email** variable.
6. Add a Decision element that checks whether a contact record was found or not.
    a. For the label, enter `Contact Found?`.
    b. For the first outcome's label, enter `Yes`.
    c. For the first outcome's API name, enter `ContactWasFound`.
    d. For the first outcome's condition requirement, select the **Contact from Get Contact from Email** variable, **Is Null** operator, and **False** value.
    e. For the second outcome's label, enter `No (Default)`.



7. After the Yes outcome on the flow, add an Assignment element that sets the Contact ID, Messaging User Name, and Messaging User ID fields on the messaging user record.
    a. For the label, enter `Set Contact ID and Name on Messaging User`.
    b. For the first assignment, select the **MessagingUser > Contact ID** field, **Equals** operator, and **Contact from Get Contact from Email > Contact ID** field.
    c. For the second assignment, select the **MessagingUser > Messaging User Name** field, **Equals** operator, and **MessagingUserFullNameFormula** field.
    d. For the third assignment, select the **MessagingUser > Messaging User ID** field, **Equals** operator, and `input_record > Messaging User ID` field.
8. After the No (Default) outcome on the flow, add an Assignment element that sets the Messaging User Name and Messaging User ID fields on the messaging user record.
    a. For the label, enter `Set Name on Messaging User`.
    b. For the first assignment, select the **MessagingUser > Messaging User Name** field, **Equals** operator, and **MessagingUserFullNameFormula** field.
    c. For the second assignment, select the **MessagingUser > Messaging User ID** field, **Equals** operator, and **input_record > Messaging User ID** field.
9. After both outcomes on the flow, add an Update Records action that updates the messaging user record with the values from the previous Assignment element.
    a. For the label, enter `Update Messaging User`.
    b. For How to Find Records to Update and Set Their Values, select **Use the IDs and all field values from a record or record collection**.
    c. For the record or record collection, select **MessagingUser**.
10. Add another Update Records action that updates the messaging session record with the messaging user's email.
    a. For the label, enter `Update Messaging Session with Email`.
    b. For the How to Find Records to Update and Set Their Values field, select **Specify conditions to identify records, and set fields individually**.
    c. For the object, select **Messaging Session**.

    d. For the condition requirements, select the **Messaging Session ID** field, **Equals** operator, and **recordId** variable.
    e. For Set Field Values for the Messaging Session Records, select the **Customer Email** field and **email** variable.
11. Add a Route Work action that routes the messaging session to your Agentforce Service agent.
    a. For the label, enter `Route Messaging Session to Agent`.
    b. For the record ID variable, select the **recordId** variable you created.
    c. For the service channel, select **Messaging**.
    d. For the Route To field, select **Agentforce Service Agent**.
    e. For the Agentforce Service Agent, select **Makana Customer Support Agent**.
    f. For the fallback queue, select **Messaging**.

    If your agent type or name don't appear as options, save the flow. Then, activate your agent, refresh Flow Builder, and try again.
12. Save and activate your flow. For the name, enter `Route Messaging Session to Agent`.



Create the Outbound Omni-Channel Flow

Agentforce Service agents use the Escalation topic and an outbound Omni-Channel flow to route conversations to another destination, such as a service rep, queue, or different agent. You connect an agent to one outbound flow, and that flow can route conversations to one or more destinations.

Create an outbound omni-channel flow that routes the conversation from the agent to a queue. When the customer has a complex query or wants to speak with a human, the agent activates the Escalation topic, which uses your outbound Omni-Channel flow.

The flow checks if any service reps are available in the Messaging queue. If reps are available, the flow updates the messaging session and then uses the Route Work action to route the messaging session and conversation history to the Messaging queue.

When no rep is available and escalation isn't completed, the agent continues the conversation with the customer based on the Escalation topic and retains the context of the conversation before the escalation was attempted. The agent attempts escalation only once per session.

1. From Setup, in the Quick Find box, enter `Flows`, and select **Flows**.
2. Click **New Flow**.
3. In the search field, enter `Omni`, and select **Omni-Channel Flow**.
4. In Flow Builder, on the Manager tab of the toolbox panel, create these resources.

   recordId: The messaging channel uses this variable to pass the messaging session record ID to the Omni-Channel flow.

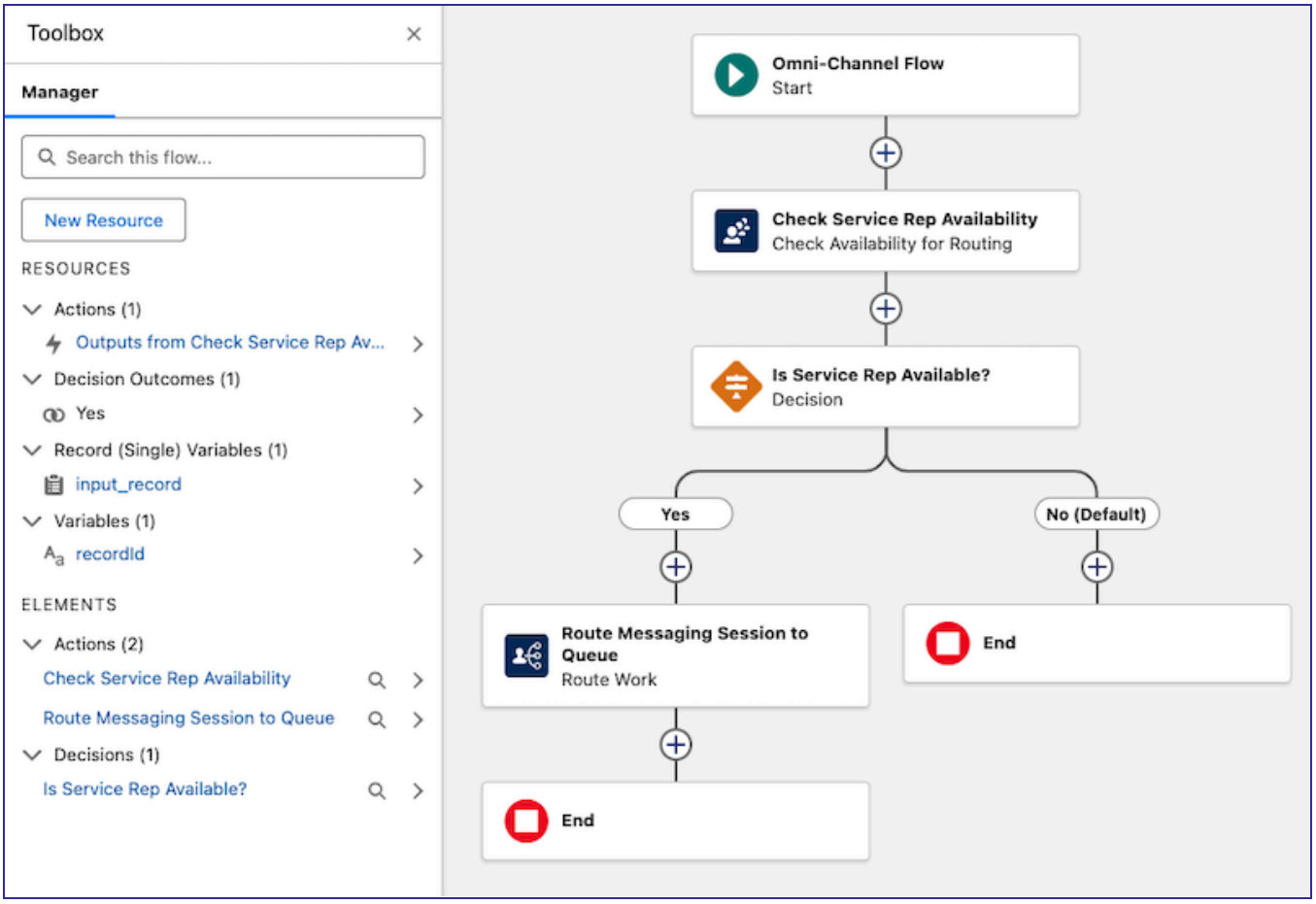
Field	Value
Resource Type	Variable
API Name (case-sensitive)	recordId
Data Type	Text
Availability Outside the Flow	Available for input

   input_record: The messaging channel uses this variable to pass the messaging session record to the Omni-Channel flow.

Field	Value
Resource Type	Variable
API Name (case-sensitive)	input_record
Data Type	Record
Object	Messaging Session
Availability Outside the Flow	Available for input

5. Add the Check Availability for Routing action element to determine service rep availability and estimated wait time before routing.

   a. For the label, enter `Check Service Rep Availability`.
   b. For the service channel, select **Messaging**.
   c. For the Check Availability For field, select **Queue**.
   d. For Queue, select **Select Queue**.
   e. For the queue ID, enter `Messaging`.
   f. For Select Output, select **Number of online agents and queued work items**.
6. Add a Decision element checks whether there are reps available in the queue or not.
   a. For the label, enter `Is Service Rep Available?`.
   b. For the first outcome's label, enter `Yes`.
   c. For the first outcome's API name, enter `RepAvailable`.
   d. For the first outcome's condition requirement, select the **Outputs from Check Service Rep Availability > Number of Online Agents** variable, **Greater Than Or Equal** operator, and **1** value.
   e. For the second outcome's label, enter `No (Default)`.
7. After the Yes outcome on the flow, add a Route Work action that routes the messaging session to the Messaging queue.
   a. For the label, enter `Route Messaging Session to Queue`.
   b. For the record ID variable, select the **recordId** variable you created.
   c. For the service channel, select **Messaging**.
   d. For the Queue, select **Select Queue**.
   e. For the queue ID, select **Messaging**.
8. Save and activate the flow. Name it `Route Messaging Session to Queue`.

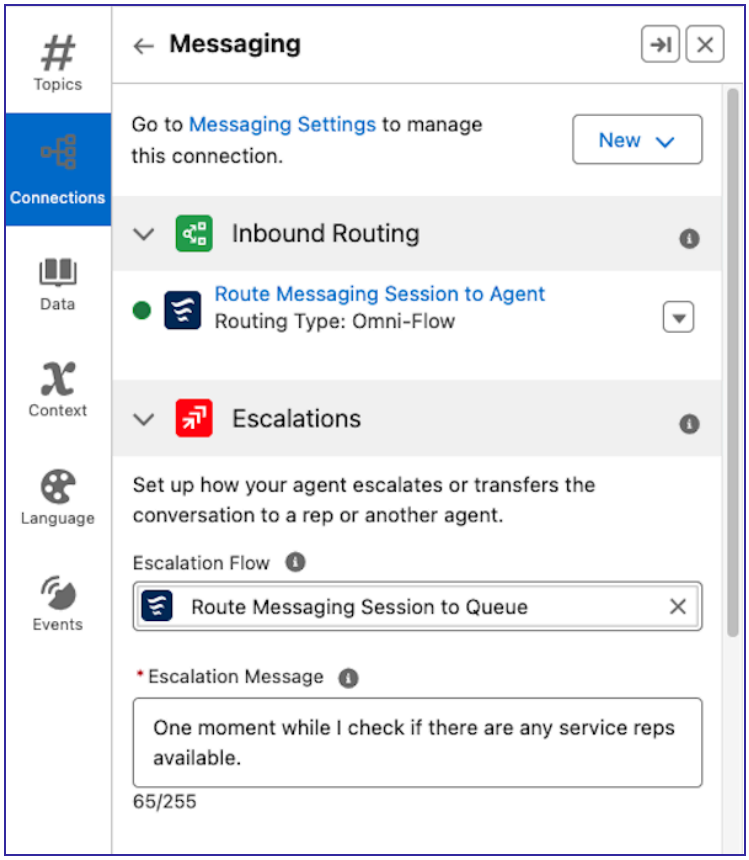


Connect the Outbound Omni-Channel Flow

To connect the Add the Route Messaging Session to Queue flow to the agent, add it as an escalation flow in the agent's Messaging connection settings.

1. From Setup, in the Quick Find box, enter `Agents`, and then select **Agentforce Agents**.
2. Click the name of your Agentforce Service agent. Then, on the Details page, click **Open in Builder**.
3. If your agent is active, deactivate it. Then, click the **Connections** tab.
4. Click the **Connections** tab. If you haven't already, turn on the new Connections panel experience.
5. On the Connections panel, select **Messaging**. In the Escalations section, from the Escalation Flow field, select the **Route Messaging Session to Queue** flow.

Confirm that your inbound and outbound Omni-Channel flows appear on the Connections panel. If a flow doesn't appear as an option, activate it, refresh Agentforce Builder, and try again.



Complete the Escalation Experience

When a customer wants to chat with a person or the conversation becomes complex or sensitive, the agent launches the Escalation topic. To make sure that the agent provides an escalation experience that aligns with Makana's business goals, customize the escalation message. The agent sends the escalation message to the customer before it attempts to transfer the conversation to the Messaging queue in your outbound Omni-Channel flow. Also, create a custom Escalation topic that tells the agent to create a case when escalation isn't possible.

1. In the Escalations section, edit the escalation message that the agent sends before it transfers a conversation so that your customers know what to expect. Enter `One moment while I check if there are any Makana Medical Devices reps available.`
2. To have the agent create a case when escalation isn't possible, create a custom version of the Escalation topic that includes the Create Case with Enhanced Data action. To let the agent create cases without user verification, assign the EndUserContactId variable to the Verified Customer ID action input.
    a. In Agentforce Builder, on the Topics panel, select the Escalation topic.
    b. Click the **This Topic's Actions** tab.
    c. Click **New** and then click **Add From Asset Library**. Select the Create Case with Enhanced Data action and add it to the topic.
    d. Save your changes. Your changes apply to the topic only in this agent and agent version.

   e. On **This Topic's Actions** tab, click the Create Case with Enhanced Data action name to open its details. Then assign the EndUserContactId variable to the Verified Customer ID action input.
   If you don't see EndUserContactId in the dropdown, add it to the agent in the Context tab. Click the **Context** tab and then click the Messaging Session context variable. Click **Edit Included fields** and then select the **EndUserContactId** field. Click **Save Included Fields** and then try assigning the variable again in the Create Case with Enhanced Data action.

3. To make sure that the agent escalates conversations according to Makana Medical Devices' unique business needs, customize the Escalation topic with these three separate instructions.
   - `If you don't have a value for the $Context.Customer_Email variable, ask the customer for their email address and store it in $Context.Customer_Email. If there is already a value, don't ask.`
   - `If escalation to a service representative fails for any reason, acknowledge the issue and ask the user if they would like to log a case. If the user says yes, then use the CreateCaseEnhancedData action to create a case and share the case details (Case Number, Subject, Description) with the user.`
   - `If you try to create a case, and the value of caseRecord is null, tell the user that you couldn't create a case.`
4. When you're done editing your agent in Agentforce Builder, activate it.

Prepare Your Service Team For Escalated Conversations

Makana wants to equip their service reps with the tools they need to manage escalated messaging conversations. By adding the reps as service resources, they can receive messaging sessions and have presence statuses that indicate their availability. Help reps monitor and join conversations by adding the Omni-Channel sidebar and a custom Messaging Session record page to the Service Console app.

Learn More: *For more information about preparing your reps, see these resources.*

- Set Up Service Reps
- Add Messaging to the Service Console

Add Your Reps as Service Resources

To let service reps receive messaging sessions, add each rep as a service resource.

1. From the App Launcher, select the **Service Resources** tab.
2. Click **New**.
   a. For the name, enter the name of a service rep.
   b. Select **Active**. A service resource must be active to receive work items.

    c. For the user, use the lookup icon to select the service rep.
    d. For the resource type, select **Agent**.
3. Save your resource and repeat the process for each service rep.

Create and Assign Presence Statuses

Presence statuses indicate if a service rep is available or not to receive messaging sessions. Create at least two presence statuses that reps can set in the service console.

1. From Setup, in the Quick Find box, enter `Presence`, and then select **Presence Statuses**.
2. Click **New** and then create at least two presence statuses.
    a. For the first status, enter `Available` as the name, select the **Online** status option, and select the **Messaging** channel.
    b. For the second status, enter `Busy` as the name, select the **Busy** status option.
3. To give your service reps access to presence statuses, use a permission set.
    a. In Setup, in the Quick Find box, enter `Permission Sets`, then select **Permission Sets**.
    b. Create a permission set or edit one. On the Permission Sets Setup page, click the name of the permission set.
    c. Click **Service Presence Statuses Access** and then click **Edit**.
    d. Select the presence statuses that you created.
    e. Save your changes. Then assign the permission set to your reps.

Add the Omni-Channel Sidebar to the Service Console

Let service reps accept and manage messaging sessions in the Service Console app with the Omni-Channel sidebar. Add the sidebar with messaging sessions to the Service Console interface and give your reps access to it.

1. From Setup, in the Quick Find box, enter `App Manager`, and then select **App Manager**.
2. Next to the Service Console app, click **Edit**.
3. To add the Omni-Channel sidebar to the app, select **Use Omni-Channel Sidebar** on the App Options page. Service reps use the sidebar to accept messages sent over Messaging channels.
4. In the Navigation Items section, select **Messaging Sessions** and then click **Save**.
5. In the User Profiles section, select your reps' profiles.

Add a Messaging Session Record Page to the Service Console

To let service reps chat with customers from the service console, customize a messaging session record page and add it to the Service Console app.

1. From the Object Manager in Setup, click **Messaging Session**.
2. Click **Lightning Record Pages**.
3. To create a custom record page, click **New**.
    a. Select **Record Page**.
    b. For the label, enter `Messaging Session with Chat`.

   c. For the object, select **Messaging Session**.
   d. For the page template, select **Pinned Left Sidebar (2 regions)** or **Header and Left Sidebar**.
   e. Click **Done**.
4. In Lightning App Builder, add the Enhanced Conversation component to the left sidebar. Then add the **Highlights** and **Messaging Related Contact** components to the right side of the page.
5. Save your changes and then assign this page as the app default for the Service Console app.

Deploy the Agent to a Website

To deploy the Makana Customer Support agent to a website, create an Enhanced Web Chat channel and test the embedded service deployment associated with it. To make sure that all the parts of your agent deployment are functioning properly, create an Experience Cloud site to test the agent on.

Learn More: *For more information about deploying agents to websites, see these resources.*

- Add Flexibility and Power with Enhanced Chat
- Experience Cloud

Create a Messaging Channel

Enhanced Web Chat is a type of channel that lets you create a personalized messaging experience for a website. Create an Enhanced Web Chat channel for the Makana Customer Support agent. Later, you create an Experience Cloud site and embed the channel in it.

1. From Setup, in the Quick Find box, enter `Messaging Settings`, and then select **Messaging Settings**.
2. Click **New Channel**.
3. In the Add a Messaging Channel window, click **Start**.
4. For the channel type, select **Messaging for In-App and Web**.
5. In the Name Your Channel window, define the channel's settings.
   a. For the channel name, enter `Help Center Chat`.
   b. For the deployment type, select `Web`.
   c. For the domain, copy and paste the Experience Cloud Sites Domain on the Domains Setup page.

6. In the Channel Routing window, specify the channel's routing.
   a. For the routing type, select **Omni-Flow**.

   b. For the flow definition, select the **Route Messaging Session to Agent** outbound Omni-Channel flow,.
   c. For the fallback queue, select **Messaging**.
   d. Agree to the terms. Then, click **Save** and keep the window open while your channel is being built.
7. In the Customer Inactivity section, select **Automatically mark conversations inactive after a period of customer inactivity**.
8. Save your changes.
9. From Setup, select **Messaging Settings** again. Then, click the name of your messaging channel.
10. In the Messaging Settings section, select **Identify preferred language of messaging users**.
11. In the Parameter Mappings section, create these mappings. They connect the pre-chat form fields you create later to the flow variables in your inbound omni-channel flow.

Field	Value
Parameter	First Name
Flow Variable Name	firstName

Field	Value
Parameter	Last Name
Flow Variable Name	lastName

Field	Value
Parameter	Email
Flow Variable Name	email

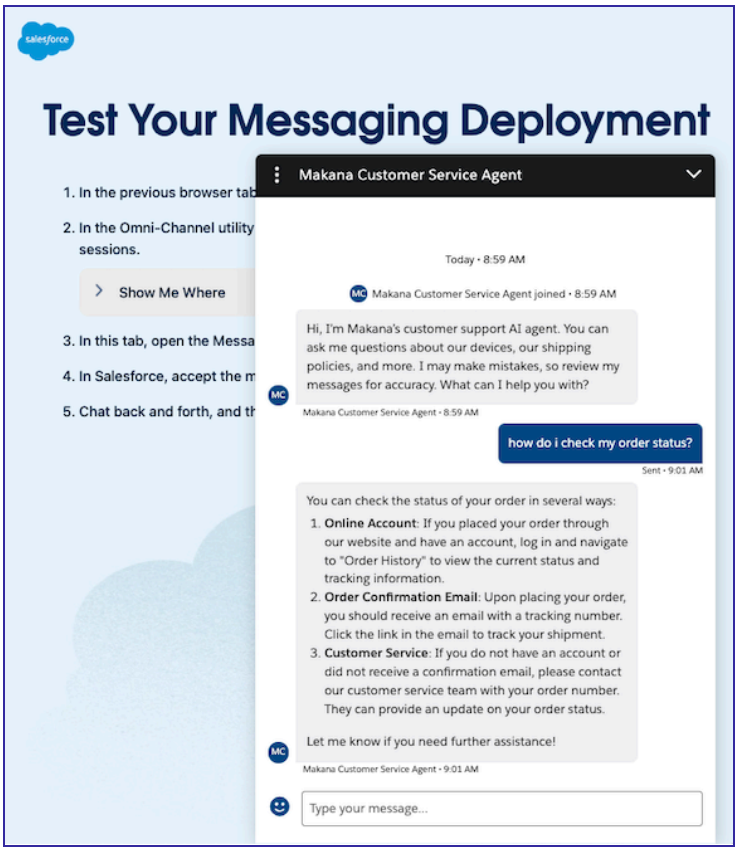
Publish and Test the Embedded Service Deployment

When you create a messaging channel, Salesforce creates an embedded service deployment that helps you manage the user experience of your messaging channel. Later, you add this deployment to your website using the embedded messaging component.

1. From Setup, in the Quick Find box, enter `Embedded`, and then select **Embedded Service Deployments**.
2. Click the **Help Center Chat** deployment.
3. Edit and activate the deployment's pre-chat form.
   a. Click **Edit Pre-Chat** on the Pre-Chat tile.
   b. On the Pre-Chat Form page, select **Activate the pre-chat feature**.

c. In the Visible Pre-Chat fields section of the page, add a First Name, Last Name, and Email field. Mark Email and Last Name as required. Then, click **Save**.

4. On the Help Center Chat deployment page, click **Publish**.
5. To test your messaging deployment, click **Test Messaging** on the Test Messaging tile. You see a page with a chat button.
6. To start a conversation, click the chat button. Enter the contact information for an existing contact and then click **Start Conversation**. If you completed the steps so far successfully, your agent joins the conversation and responds to the messages you send.



Create an Experience Cloud Site

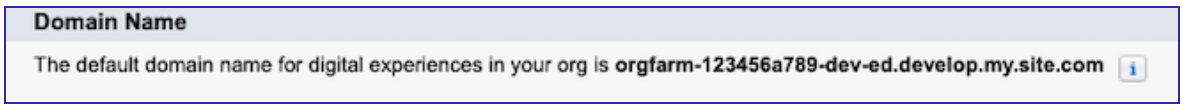
To test your agent's full functionality, such as its ability to escalate and manage cases, publish it on an Experience Cloud site. To get a site up-and-running quickly, use the Help center template.

1. From Setup, in the Quick Find box, enter `All Sites`, and then select **All Sites**.
2. Click **New**. Then, select the **Help Center** template.
3. Click **Get Started**.
4. On the Set Up Help Center window, click **Start**.
    a. For the help center name, enter `Makana Help Center`.
    b. For the help center URL, enter `help`.
    c. If you see **Make this help center live**, select it.
    d. Click **Next** for the remaining steps. You can change these settings later.
    e. Click **Finish**.

Customize and Publish the Experience Cloud Site

Add the Embedded Messaging component to the Makana Help Center site and add the site to your CORS allowlist. When you're ready, publish the Makana Help Center.

1. In Experience Builder, click the **Components** tab.
2. On the Components panel, select the **Embedded Messaging** component and drag it onto the template footer.
3. On the page, click the **Embedded Messaging** component.
4. When you're ready, click **Publish**.
5. Add your Experience Builder site domain to your CORS allowlist. If you notice an issue with the *.live-preview domain, add it to the CORS allow list as well.
    a. From Setup, in the Quick Find box, enter `Digital Experiences`, and then select **Settings**. Copy or take note of the default domain name in the Domain Name section.



    b. From Setup, in the Quick Find box, enter `CORS`, and then select **CORS**.
    c. In the CORS Allowed Origin List Edit section, select **New**.
    d. Enter the default domain name URL. Add `https://` to the beginning of the URL and then click **Save**.

Test the Agent on the Website

Before you deploy the Makana Customer Support agent to a staging or production environment, verify that all the parts of its messaging deployment are set up correctly. Make note of areas where you can improve and iterate on the deployment, such as further customizing the escalation experience or Service Console app.

1. On the All Sites Setup page, open your Experience Cloud site by clicking the URL.

2. To start a conversation, click the chat button.
3. Enter the contact information of an existing contact and then click **Start Conversation**.
4. Test the escalation process when there are no service reps available.
    a. In the chat window on the Experience Cloud site, ask the agent to transfer you to a human.
    b. Confirm that the agent lets you know that there are no service reps available and offers to create a case.
    c. After the agent lets you know that it created the case, confirm that the new case exists in Salesforce.
5. Test the escalation process when there's at least one service rep available.
    a. End the chat and start a new one.
    b. In a different browser tab, open the Service Console app.
    c. From the sidebar, select **Omni-Channel** and then change your status to Online.
    d. In the chat window on the Experience Cloud site, ask the agent to transfer you to a human.
    e. In the Omni-Channel sidebar, accept the chat request. Then, click the request to open the messaging session record. Send a message and then confirm that it is sent in the Experience Cloud site chat window.

Deploy the Agent to Staging or Production

After testing the messaging deployment in your sandbox, package and deploy your agent to a staging or production environment.

Make sure that your staging or production environment has required licenses for your agent, and that Einstein Generative AI, Agentforce, and Data Cloud are enabled.

1. Review the metadata components for Agentforce.
2. Retrieve the latest metadata from your sandbox org using Salesforce CLI and generate a package file for your agent. Include any custom Apex, flows, and prompt templates for your agent actions. Also include any permission sets assigned to the agent user.
3. Create the agent user in the target org.
4. Deploy the Agentforce metadata to the target org using Salesforce CLI.
5. Verify that the agent is active.
6. Add the deployed permission sets to the agent user.
7. Make sure all deployed flows have the correct version active. To manage a flow's versions and see version statuses, click the name of the flow in the Flow Builder navigation header.
8. Publish the web deployment in the target org.

After deploying to your target org, confirm that the agent's topics, instructions, variables, and filters were deployed correctly. Make sure that all permissions have been deployed or updated, and that the agent is responding to user requests.

Monitor Your Agent

Makana Medical Devices has successfully deployed their customer service agent to handle inquiries about their CGM-3000 continuous glucose monitor. Now they need to ensure the agent performs well and delivers the business value they expect.

This guide covers how to test, measure, and effectively optimize your AI agent's performance in a real-life environment after it's been launched.

Starting here? *Make sure you have the right permissions for the job.*

Before We Begin

This chapter is meant to provide guidance around building a continuous monitoring strategy and using Agentforce tools to monitor and improve your agents. Since we just launched the Makana Support agent, there won't be enough data to walk through a full monitoring strategy with live data.

Additionally, although we have a legacy Agentforce Analytics, we're upgrading it to a new Agentforce Analytics and Optimization solution. These updates are in Beta and not yet available for Developer orgs. Once you roll out your agents in production, use this section as a resource to keep key stakeholders informed and handle top issues.

How it Works

Analytics and Optimization go beyond simply observing performance to helping you identify specific areas for improvement, iterating on agent prompts and knowledge bases, and continuously refining agent behavior to achieve business objectives and enhance user satisfaction.

Agentforce Analytics provides a broad, comprehensive view of agent sessions with meaningful insights into agent functions such as interaction latency, usage patterns, and performance metrics like overall quality scores and success rates. Analytics answers foundational questions like "Is my agent working as expected?" and "What can I do better?"

Agentforce Optimization complements Analytics by providing moment-specific data for granular inspection. Moments refer to sub-sessions that capture interactions addressing specific user intent within broader sessions. A Large Language Model identifies similar intents across multiple sessions

and divides them into moments. The division into moments makes inspection more granular and helps you identify instances of agents providing inaccurate, misleading, or irrelevant information, or failing to execute actions as configured.

Set Up Audit and Feedback

Turn on Audit and Feedback to store details about interactions with LLMs for analysis and reporting. Audit data captures prompts and generated AI responses (including PII-masked versions and toxicity scores). Feedback captures explicit user feedback about generated responses, such as "thumbs up/down" ratings and reason text. Use insights derived from this information to monitor, test, troubleshoot, and improve your agents in Data Cloud. To learn more, see Generative AI Audit and Feedback Data.

By turning on these features, you consent to store your Salesforce organization's generative AI activity log and feedback data in Data Cloud. Consent also assumes you've reviewed any potential cost implications and assume responsibility for any associated costs.

1. From Setup, in the Quick Find box, enter `Einstein Audit`, and then select **Einstein Audit, Analytics, and Monitoring Setup**.
2. Turn on **Audit and Feedback**.
3. If you have multiple data spaces in your org, select a target data space in which to store the audit and feedback data.
4. Verify that the pre-built reports and dashboards are installed and available. It can take a few minutes for the installation to complete.
    a. Click **App Launcher**, and in the search field, enter `Data Cloud`.
    b. To see audit and feedback dashboards, from Data Cloud, click **Dashboards**. Under FOLDERS, click **All Folders**. In the Search all folders search box field, enter `Einstein Gen AI Audit & Feedback.`
    c. To see audit and feedback reports, from Data Cloud, click **Reports**. Under FOLDERS, click **All Folders**. In the Search all folders search box field, enter `Einstein Gen AI Audit & Feedback.`

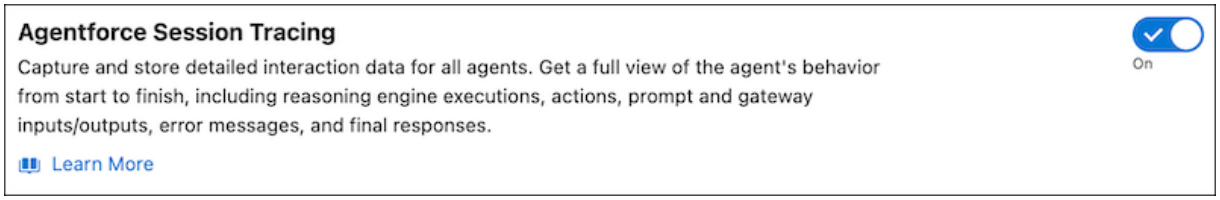
After Einstein Generative AI data collection is enabled, it can take a few minutes for the reports and dashboards to become available. If for any reason the reports and dashboards are not installed after enabling data collection, the system will continue to retry the installation.

Set Up and Configure Agentforce Analytics

Agentforce Analytics is designed for comprehensive session analysis, capturing every turn and event across your agent sessions using the unified Session Tracing Data Model (STDM). Agentforce Analytics uses Tableau Next for its dashboards visualization. To view them in Agentforce Analytics, you must manually install the libraries in addition to the setup steps.

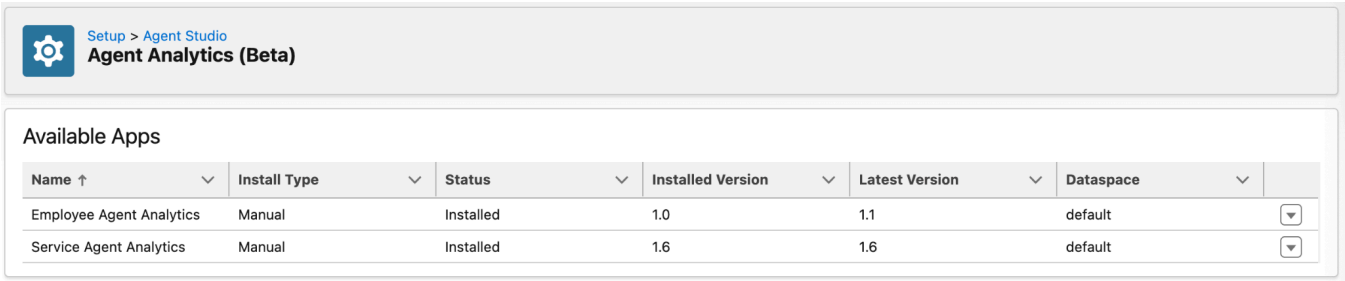
To set up Agent Analytics, turn on Session Tracing to collect and store session data. Session Tracing also provisions the required Analytics model entities to perform queries.

1. Verify that you have the latest version of the Salesforce Standard Data Model (version 1.107 or higher) in your org (sandbox or production). From Setup, go to **Apps** -> **Packaging** -> **Installed Packages** -> **Salesforce Standard Data Model**. To get the required version, click here and follow the instructions.
2. From Setup, in the Quick Find box, enter `Einstein Audit, Analytics, and Monitoring Setup`, then select **Einstein Audit, Analytics, and Monitoring Setup**.
3. Scroll down to **Agentforce Session Tracing and Data Model**.
4. Toggle on **Agentforce Session Tracing and Data Model to** start collecting data and to provision the Data Model Objects. Note that turning data collection increases your org's credit consumption rate.



5. From Setup, go to **Agent Analytics (Beta)**.

6. Install the dashboard libraries for your available agents. Dashboards are created from Tableau Next.



7. Refresh the page to see the status changed to **Installed**.

Observe and Optimize with Agentforce Analytics and Optimization

Once configured, Agentforce Analytics and Optimization provides tools for continuous monitoring and improvement of AI agents. Optimization extends session tracing by focusing on session moments, which are sub-sessions representing interactions with a specific user intent. An advanced Large Language Model (LLM) identifies clusters, and tags these moments, facilitating queries and insights.

To add Agentforce Optimization to your observability tools, toggle on **Agentforce Optimization** from setup in addition to **Agentforce Session Tracing and Data Model** and assign users the relevant permission sets.

1. From Setup, go to **Einstein Audit, Analytics, and Monitoring Setup**.

2. Toggle on **Agentforce Session Tracing and Data Model and Agentforce Optimization (**Optimization relies on session tracing as well).

Makana's team want to ensure their agent meets the business objectives they defined during planning:

- Improve deflection rate by 40%
- Improve customer satisfaction scores by 10%
- Reduce average case handling time by 25%

The team needs to track these metrics and identify specific areas where the agent can perform better. The foundation data model for Analytics provides insights and metrics that Makana's team can use to reach their goals.

Analytics Data Model and Semantic Layer

The foundation for the Analytics data model is the unified Session Tracing Data Model (STDM), which captures every logged event within a session, down to a single conversation turn.
The Analytics Semantic Layer links Session Tracing entities with Optimization data entities, enabling complex queries. You can access this in Data Cloud under the semantic model tab by opening Agentforce Analytics Foundations.

Analytics Queries and Calculated Fields

The Semantic Layer provides calculated fields for querying session and agent performance data. These measures provide quantitative insights:

- Quality_Score_clc & Average_Quality_Score_clc: Relevance score (1-5) indicating the accuracy/fulfillment rate of an agent's response to a user's request.
- Quality_Score_Reasoning_clc: Explanation for an assigned quality score (e.g., "agent didn't address the pricing question").
- Moment_Duration_clc: Duration of a Moment in seconds.
- Average_Moment_Duration_clc: Average duration of a Moment in seconds. This helps identify interactions that take longer.
- Escalation_Rate_clc: Percentage of sessions escalated to a human or different agent.
- Deflection_Rate_clc: Percentage of sessions ended by the user (not by escalation or abandonment).
- Unique_Moments_clc: Number of distinct moments recorded, clustering interactions around a user request.

You can combine multiple fields in a single query to generate granular metrics, such as the relationship between session duration and quality ratings (e.g., Session_Count, Average_Session_Duration_clc, and Quality_Score_clc) or to track performance trends over time.

For example, combining Moment_Duration_clc and Quality_Score_clc helps identify areas where users spend significant time but receive poor-quality responses.

Monitoring Strategy Methods

This sample plan outlines how to use Agentforce Analytics and Optimization for a broad, comprehensive view of agent performance, and a granular, moment-specific inspection to identify specific issues in agent sessions and fix them. There are three major methods in a monitoring strategy:

Stage	When to Begin	Goal
Post-Launch Monitoring	Immediately following agent launch	Confirm agent is working as designed and metrics are being populated
Moment Monitoring	30 days after agent launch 24 hours after a closed moment, up to 30 days for deep data insights	Identify actions for improvement
Root Cause Analysis	Minimum clustering of 100 moments	Use clusters to identify issues and dive deep into sessions.

Post-Launch Monitoring

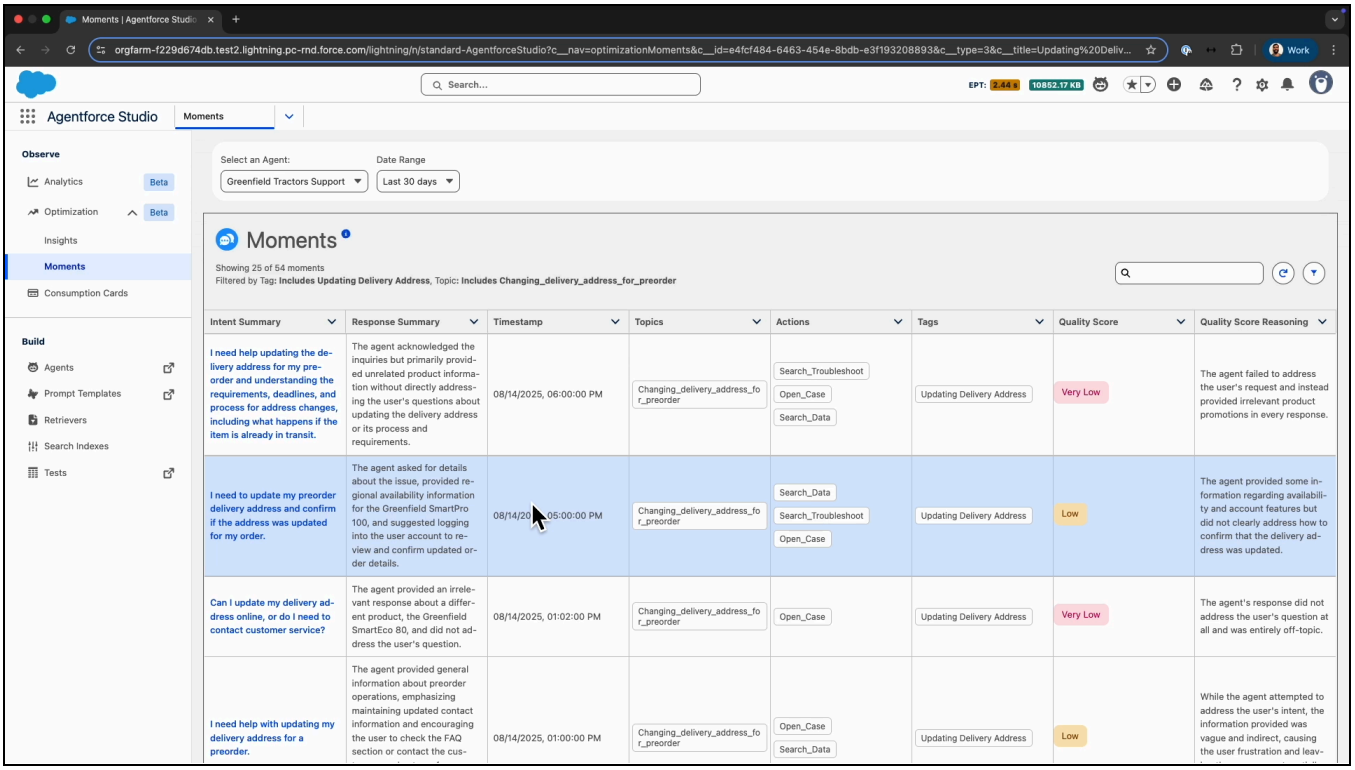
Immediately post-launch, use Agentforce Analytics to check if data is collected and ingested and view your dashboard for key metrics:

- Usage patterns (e.g., unique users, number of sessions)
- Overall performance metrics like average quality score and escalation rate
- Agent latency

You can also dive into specific sessions. Select sample sessions from key topics your sessions cover, and inspect them turn-by-turn to see if they flow as expected.

Moment Monitoring

When your system begins to cluster and tag moments within interactions, you can view the moments tab in your app and filter the moments by quality scores. If you inspect individual specific moments with low quality scores, you can uncover possible agent actions that need refinement or improvement.

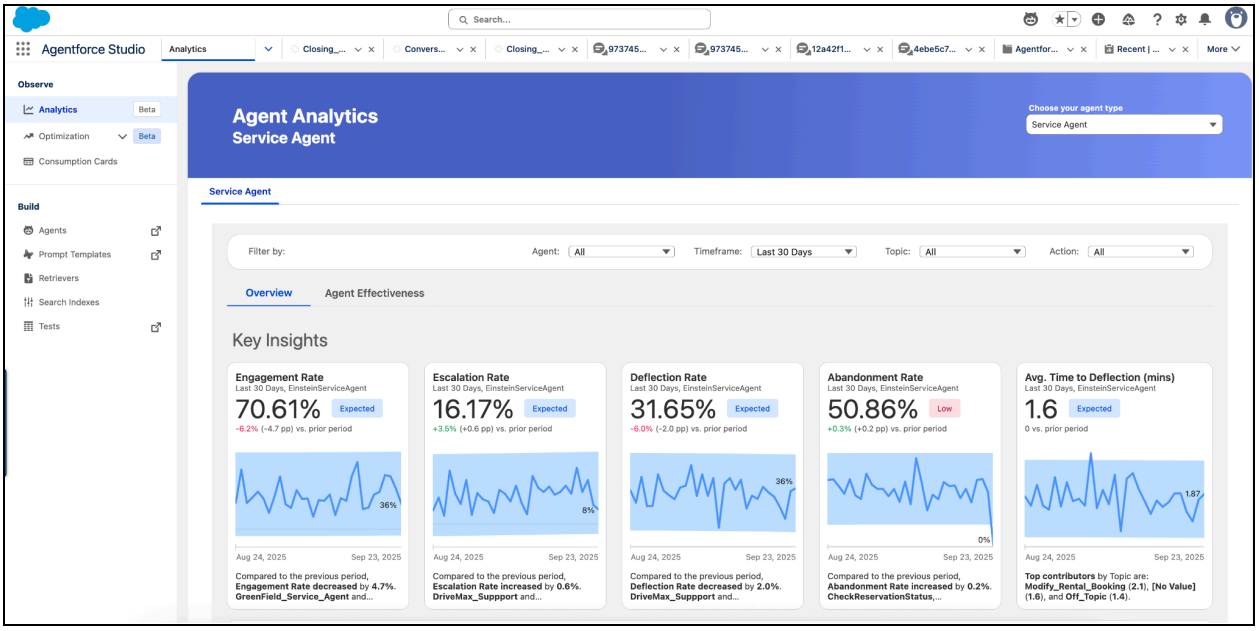


Timeline:

- Initial Data Analysis: The first runs will analyze session data from the last 30 days to provide an initial baseline. Subsequent runs will focus on sessions from the last 7 days.
- Moment Generation: The system generates moments (sub-sessions representing a specific user intent) on a frequent schedule. A pipeline run is initiated under one of two conditions:
    - Every 9 hours if there is at least one new closed session.
    - Every 3 hours if there are more than 10 new closed sessions since the last run.
    - A "closed session" is defined as a session where the last interaction occurred 24 hours ago.
- Clustering: The Large Language Model (LLM) clusters similar moments and applies tags once per week over the weekend. This process is crucial for identifying trends and widespread issues.

Root Cause Analysis

This stage focuses on using the detailed data to perform root-cause analysis and identify specific areas for improvement. Drill Down with Agentforce Optimization and move from the "what" of Analytics to the "how" of Optimization. Use the tool to perform a granular inspection of specific moments and clusters.



Timeline:

- This phase typically begins after the first weekly clustering run is complete, as this provides the necessary data for deep-dive analysis.
- For the clustering algorithm to provide meaningful results, a rule of thumb is that around 100 moments are needed. This can typically be generated from about 100 sessions, though this depends on your specific data.

Actions:

1. Your main Analytics dashboard will give you the initial indication of performance gaps and flagged interactions.
1. Identify Performance Gaps: Agentforce Optimization automatically flags interactions where the agent provided inaccurate, misleading, or irrelevant information (including "hallucinations") or failed to execute a configured action.
2. Analyze performance of individual moments according to their Quality Score:
    - To check an individual moment's quality, examine its quality score, ranked from 1(lowest) to 5(highest).
    - Then review the Quality Score Reasoning, which provides a specific explanation for the assigned score (e.g., "agent didn't address the pricing question").
3. Correlate Metrics for Deeper Insights: Combine multiple calculated fields in a single query to find complex problems. For example:
    - Querying for moments with a high Moment_Duration_clc and a low Quality_Score_clc helps identify areas where users spend a lot of time but receive poor-quality responses.
    - Adding the Escalation_Rate_clc to this query can confirm that these poor interactions are leading to escalations.

Building a Monitoring Strategy

We recommend using the monitoring methods over time to build out a full monitoring strategy.

Daily

1. Monitor high-level KPIs in Agentforce Analytics. When a negative trend is spotted (e.g., rising Escalation_Rate_clc), use Agentforce Optimization to drill down and identify the specific moments or user intents causing the issue.
2. Monitor internal and external feedback about the agent. If an issue arises, perform a root cause analysis to identify the problem.

Weekly

1. Monitor moments to identify agent areas of improvement.

Monthly

1. Collaborate with Subject Matter Experts (SMEs) and AI Agent Builders to analyze the root cause identified in moments. This could be a knowledge gap, a poorly configured prompt, or an issue with the Retrieval-Augmented Generation (RAG) configuration.
2. Implement Changes: Implement the required changes, such as refining agent prompts, updating knowledge bases, or modifying agent actions and topics.
3. Test and Validate in a Sandbox Environment: before deploying improvements to production, validate all changes in your sandbox environment.
    - Manual Testing: You can interact with the agent manually via Agent Builder or a testing center to review its reasoning utterance-by-utterance, and perform manual adjustments if needed.
    - Batch Testing: For efficiency, you can run up to 100 test cases at once without deactivating the agent.
4. Redeploy & Monitor: Once the changes have been validated in the sandbox, redeploy the updated agent to production. Use Agentforce Analytics to monitor KPIs to confirm that the changes had the desired positive impact, creating a continuous cycle of learning and optimization.