Oracle AI for Fusion Applications

How do I use AI Agent Studio?

26C

Oracle AI for Fusion Applications
How do I use AI Agent Studio?

26C

G57390-01

Author: Soniya Mathew

Contents

Get Help

There are a number of ways to learn more about your product and interact with Oracle and other users.

Get Help in the Applications

Some application pages have help icons to give you access to contextual help. If you don't see any help icons on your page, click your user image or name in the global header and select Show Help Icons. If the page has contextual help, help icons will appear.

Get Training

Increase your knowledge of Oracle Cloud by taking courses at *Oracle University*.

Join Our Community

Use *Cloud Customer Connect* to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, suggest *ideas* for product enhancements, and watch events.

Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we'd like to hear from you.

You can email your feedback to *oracle_fusion_applications_help_ww_grp@oracle.com*.

Thanks for helping us improve our user assistance!

1 Overview

Overview of AI Agent Studio

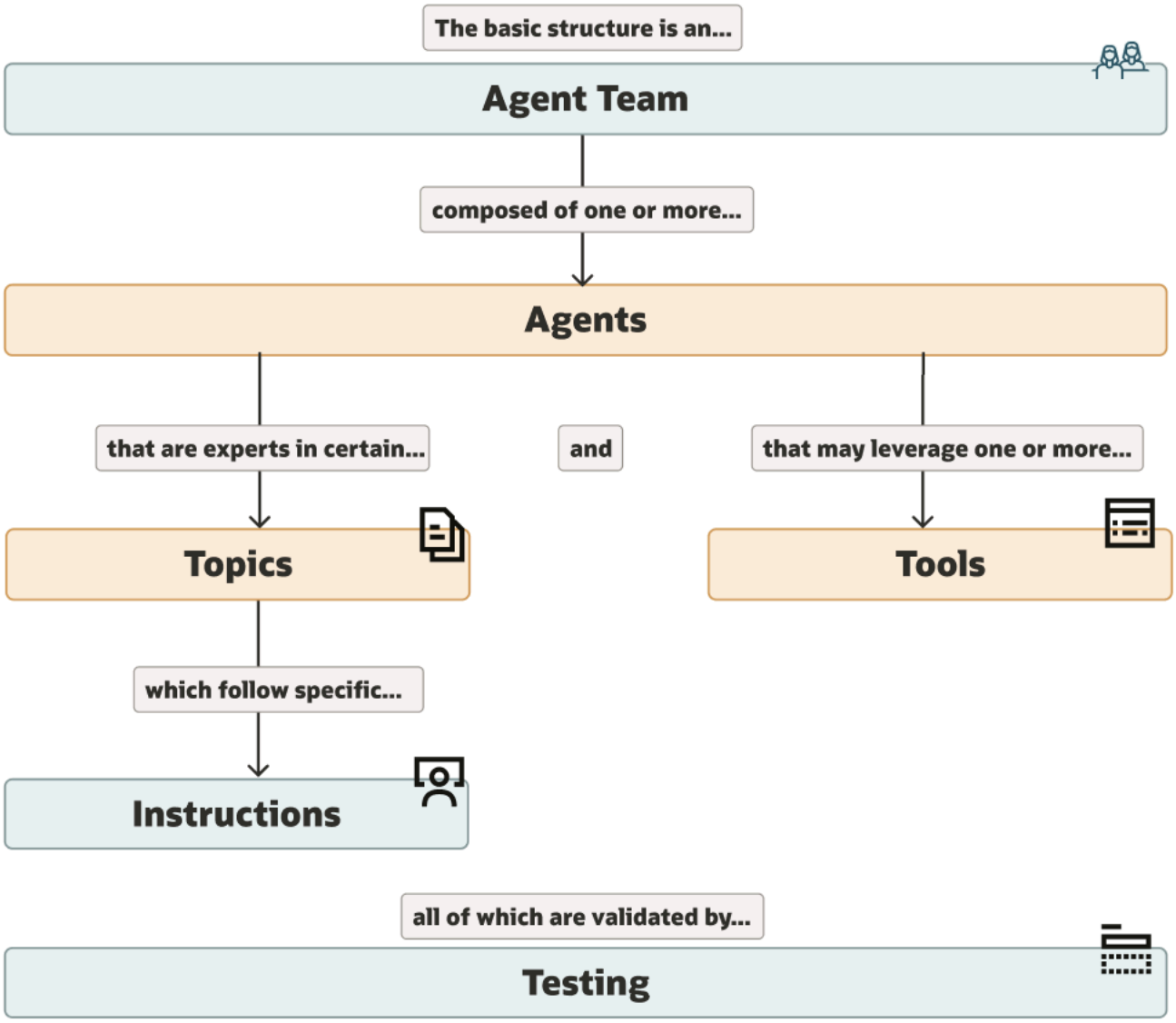
AI Agent Studio is a design-time environment that empowers you to create, configure, validate, and deploy AI agents to meet your organization's needs.

With AI Agent Studio, you can easily extend preconfigured agent templates, and even build new agents and multiagent flows. AI Agent Studio is fully integrated into Oracle Fusion Cloud Applications, providing secure and seamless access to the knowledge stores, tools, and APIs of Fusion Applications. This integration enables agents to be deployed directly into the flow, ensuring an efficient process.

Key Capabilities of AI Agent Studio

Feature	Description
Agent template libraries	Use templates and natural language prompts to create or fine tune agents for common business scenarios, such as opportunity-to-quote processing and shift scheduling.
Agent team orchestration	Configure multiple agents to collaborate on multistep processes, integrating user approvals where necessary.
Agent extensibility	Change and extend existing agents in Oracle Fusion Cloud Applications by incorporating new data sources, prompts, and APIs to fit specific business needs or industry requirements.
Native integration with Fusion Applications	Directly access APIs and tools in Fusion Applications, ensuring seamless deployment of agents without complex modifications.
Third-party system integration	Connect with external systems and collaborate with third-party agents for end-to-end automation with secure API support.
Trust and security framework	Automatically applies the security configurations, policies, and access controls of Fusion Applications, ensuring compliance with enterprise security standards.
Validation and testing tools	Use built-in tools to make your agents reliable, repeatable, easy to explain, and secure by verifying AI-driven flows before deployment.

Components of AI Agent Studio



Component	What It Does	Example
Agent Team	Consists of a single agent or a group of agents that follow a structured sequence of steps or actions to accomplish a specific business task or answer a user query. The agent team is the component that can be deployed for use. • Consists of conversation logic, system integration, and user support flow. • Defines how the agent acts for a particular use case — what to do, when to do it, and how to respond based on user inputs or back-end system data.	A recruitment agent (that might consist of multiple agents) schedules interviews, screens resumes, calculates salary, and generates offers, based on policies and approvals.

Component	What It Does	Example
Agents	Leverages a large language model to reason, create action plans, and interact with users to gather information and take direction. On behalf of users, the AI agent can do tasks that enhance productivity, efficiency, and the overall user experience. An agent must be added to an agent team, so that it can be deployed for use. AI agents can be categorized into various types. • User-proxy agent: Acts on behalf of a business user to provide input to another agent or group of agents. It's sometimes referred to as a conversational agent. • Supervisor agent: Orchestrates the use of agents within an agentic flow. • Specialist or utility agent: Focuses on a specific role or expertise and can be skilled in using a particular tool. AI agents can also have one or more of these characteristics: • Persona-based agent: Represents a specific role, such as benefits administrator, customer service representative, and finance administrator. • Tool user: Uses technology-related tools, such as calculators, web search queries, document embedding, and calendar schedulers. • Task-oriented agent: Understands their assignment or task, as a single agent or as part of a multiagent flow.	A scheduling or calendar agent that manages your workday by following your instructions. It can accept new calendar invites and propose alternative times when needed.
Topics	Defines the areas of expertise through instructions that set the boundaries and constraints for agent conversations and abilities.	An employee benefits agent can contain topics such as Health Savings Account (HSA), retirement benefits, and stock plans.
Tools	Defines the additional utilities an agent can use to accomplish a task. One or more tools are assigned to agents, and they're reusable among agents.	• Calculator tool • Connector tool • Email tool • Business object tool • User query tool • Document retrieval tool for retrieval-augmented generation (RAG)
Instructions	Specifies the natural language rules that define the rules or conditions applied to a given topic. Instructions are part of the prompts that are sent to the underlying large language model. They can also contain guidelines and guardrails that set the parameters of an agent response.	Instructions for the payroll deduction topic: Make sure you've information regarding the number of dependents either by asking the user or querying the system. If you don't know the answer, don't make up a response.
Testing	Enables administrators to test the agent team design, ensure correct tool, topic, and instruction configurations, and validate reasoning and sources cited by the agent.	Provide example responses to a series of questions a user would likely ask an agent and details about how the agent arrived at its response.

2 Access Requirements for AI Agent Studio

Access Requirements for AI Agent Studio

You can give access to AI Agent Studio by assigning predefined duty roles to job roles. Also, make sure to complete these prerequisites:

- Enable security console to work with permission groups
- Run scheduled processes to import security data
- Run scheduled processes to get help from the integrated AI agent
- (Optional) Assign privilege to use external REST API tools
- (Optional) Assign permission groups for channels
- (Optional) Assign role to schedule workflow agent teams
- (Optional) Assign permission group to view monitored agent records
- Give users access to AI agents

Watch video

Enable Security Console to Work with Permission Groups

For the Security Console to work with permission groups and related objects, set the **Enable Security Console External Application Integration** (ORA_ASE_SAS_INTEGRATION_ENABLED) profile option at the site level.

1. In the Setup and Maintenance work area, search for the **Manage Administrator Profile Values** task using the search link in the panel.
2. Search for the profile option and set the value for the **Site** profile level to **Yes**.

Run Scheduled Processes to Import Security Data

To import resources from LDAP, and transfer the necessary information into the security tables of Fusion Applications, run these two scheduled processes sequentially.

1. Import Resource Application Security Data
2. Import User and Role Application Security Data

You must run the processes one after the other.

1. Go to **Navigator** > **Tools** > **Scheduled Processes**.
2. Click **Schedule New Process**.
3. Leave the type as **Job**.
4. Search for and select the process.
5. Submit the process.

Run Scheduled Processes to Get Help from the Integrated AI Agent

You can get answers to questions about developing AI agents from the AI help agent integrated into AI Agent Studio. Using this conversational agent, you can get answers to questions about existing agents, search for agents, tools, and topics using natural language, and receive AI powered suggestions for relevant resources to use in your agents.

To use the integrated AI help agent, run these two scheduled processes daily.

- Index AI Agent Studio Assistant Documents
- Index AI Agent Studio Assistant Objects and Attributes

Run each process using these steps:

1. Go to Scheduled Processes, and click **Schedule New Process**.
2. Leave the type as **Job**.
3. Search for and select the process.
4. Submit the process.

Assign Privilege to Use External REST API Tools

To create External REST API tools in AI Agent Studio, the Create and Edit Backends for Visual Builder Studio (ORA_FND_TRAP_PRIV) privilege must be added to the custom role assigned to the user. You can add this privilege while creating or editing a custom role.

1. Go to the Security Console.
2. To use a new custom role, create it. To use an existing custom role, search for the custom role and edit it.
3. Go to the Function Security Policies page and select **Add Function Security Policy**.
4. Add the Create and Edit Backends for Visual Builder Studio (ORA_FND_TRAP_PRIV) privilege to the role and save it.
5. Save the custom role and assign to the user.

Assign Permission Groups for Channels

To create channels from Credentials tab in AI Agent Studio, additional permission groups must be added to the custom role assigned to the user. You can add these permission groups to a duty role and assign the duty role while creating or editing a custom role.

1. Go to the Security Console and create a new duty role for permission groups.
2. Open the **Permission Groups** page and select **Add Permission Groups**.
3. Search for and add these permission groups:
    - create:ChannelManifest
    - create:ExternalChatCorrelation
    - delete:ChannelManifest
    - delete:ExternalChatCorrelation
    - read:ChannelManifest
    - read:ExternalChatCorrelation
    - update:ChannelManifest
    - update:ExternalChatCorrelation

4. Add security view for each permission group.
    a. Select the permission group added.
    b. In Details section, open Security Views tab.
    c. Select **Add Security Views** and add the **AllRowsAllFields** security view.
    d. Add the security view for all the permission groups.
5. Save the duty role and assign this duty role to the custom job role.
6. To use a new custom job role, create it. To use an existing custom job role, search for the custom job role and edit it.

    > **Note:** Make sure to enable permission groups.

7. Go to the **Role Hierarchy** page. From the Roles and Permission Groups tab select **Add Role**.
8. Search for and add the duty role you've created.
9. Save the custom role and assign to the user.

Assign Role to Schedule Workflow Agent Teams

You can call a workflow agent team using a scheduled trigger. To schedule the workflow, the Fai Batch Job Manager Duty (ORA_DR_FAI_BATCH_JOB_MANAGER_DUTY) role must be added to the custom role assigned to the user.

1. Go to the Security Console.
2. To use a new custom role, create it. To use an existing custom role, search for the custom role and edit it.

    > **Note:** Make sure that permission groups are enabled for the custom role.

3. Go to the Role Hierarchy page and open the Roles and Permission Groups tab.
4. Add the Fai Batch Job Manager Duty (ORA_DR_FAI_BATCH_JOB_MANAGER_DUTY) role.
5. Save the custom role and assign to the user.

(Optional) Assign Permission Group to View Monitored Agent Records

To view records in the **Monitoring and Evaluation** tab of AI Agent Studio, an additional permission group must be added to the custom role assigned to the user.

1. Go to the Security Console.
2. To use a new custom role, create it. To use an existing custom role, search for the custom role and edit it.

    > **Note:** Make sure that permission groups are enabled for the custom role.

3. Open the **Permission Groups** page and select **Add Permission Groups**.
4. Search for and add the `read:Generative AI Workflow Execution` permission group.
5. Add security view for the permission group.
    a. Select the permission group added.
    b. In the **Details** section, open the Security Views tab.
    c. Select **Add Security Views** and add the `AllRowsRestrictedFields` security view.
6. Save the custom role and assign to the user.

Give Users Access to AI Agents

After your agents are created and ready for use, provide access for users to interact with the AI agents. For information, see *How can I give users access to AI agents?*

Assign Predefined Duty Roles to Job Roles

Assign predefined product-specific duty roles to the appropriate job roles, and make sure permission groups are enabled. You can give people access to configure AI agents in all or specific products.

> **CAUTION:** Using predefined roles might account for subscription consumption irrespective of whether you purchased the cloud service or not. See *Guidance for Assigning Predefined Roles*.

Provide Access to Configure AI Agents in all Products

1. Go to the Security Console and create a new custom job role.

   > **Note:** Make sure to enable permission groups.

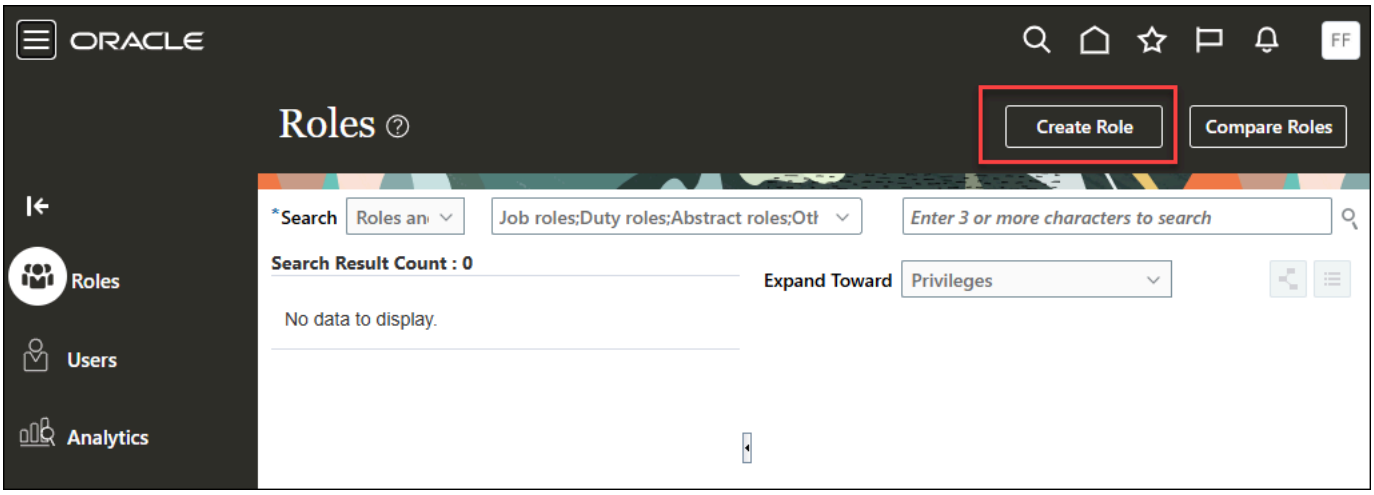
2. On the Role Hierarchy page, open the Roles and Permission Groups tab and add these duty roles:
   - Fai Genai Agent CX Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_CX_ADMINISTRATOR_DUTY)
   - Fai Genai Agent FIN Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_FIN_ADMINISTRATOR_DUTY)
   - Fai Genai Agent GRC Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_GRC_ADMINISTRATOR_DUTY)
   - Fai Genai Agent HCM Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_HCM_ADMINISTRATOR_DUTY)
   - Fai Genai Agent PRC Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_PRC_ADMINISTRATOR_DUTY)
   - Fai Genai Agent PRJ Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_PRJ_ADMINISTRATOR_DUTY)
   - Fai Genai Agent PSC Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_PSC_ADMINISTRATOR_DUTY)
   - Fai Genai Agent SCM Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_SCM_ADMINISTRATOR_DUTY)
3. Open the Roles and Privileges tab and add the Manage All Intelligent Agents (ORA_FAI_MANAGE_ALL_AI_AGENTS) role.
4. Save the custom role and assign it to users who want access.

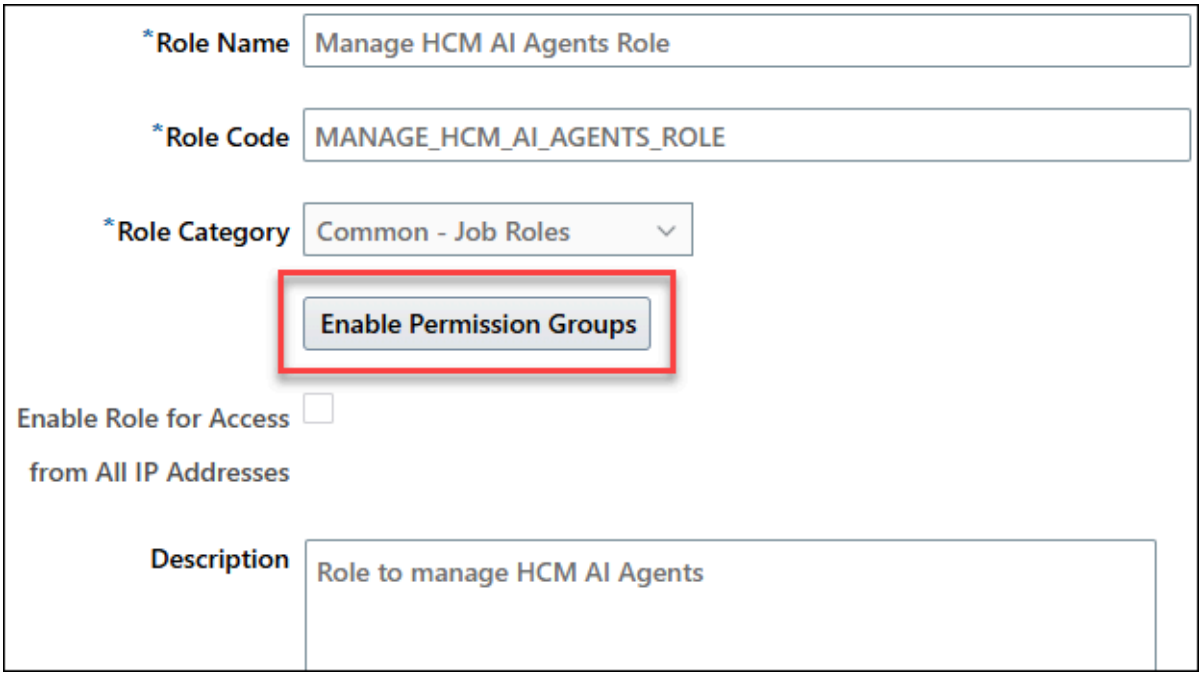
Provide Access to Configure AI Agents in Oracle Fusion Cloud Human Capital Management

To give access to users without the Human Capital Management Application Administrator Job Role:

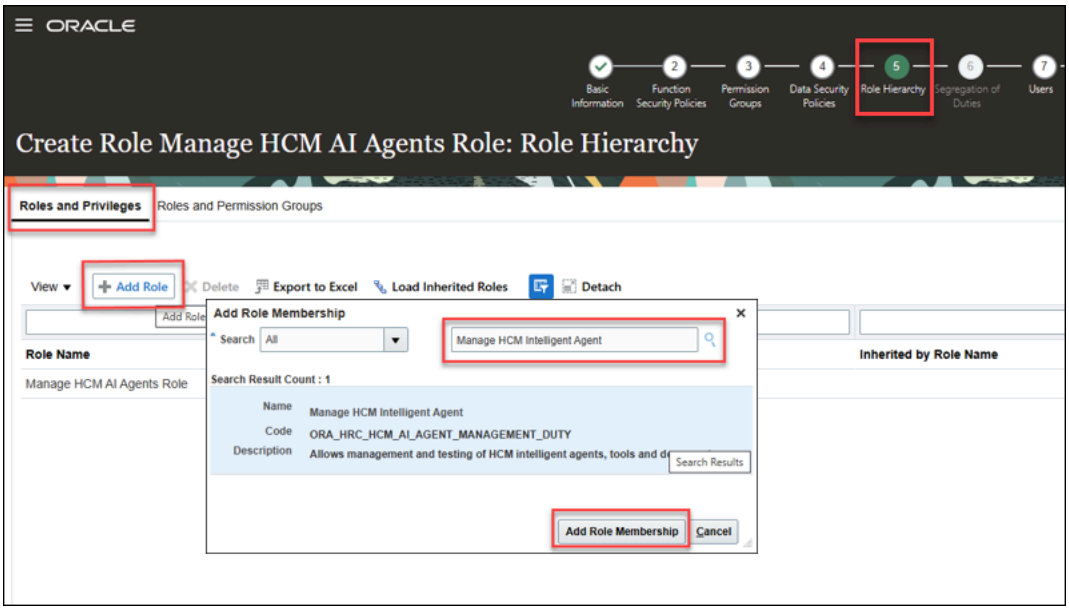
1. Go to the Security Console and create a new custom job role.

   Note: Make sure to enable permission groups.

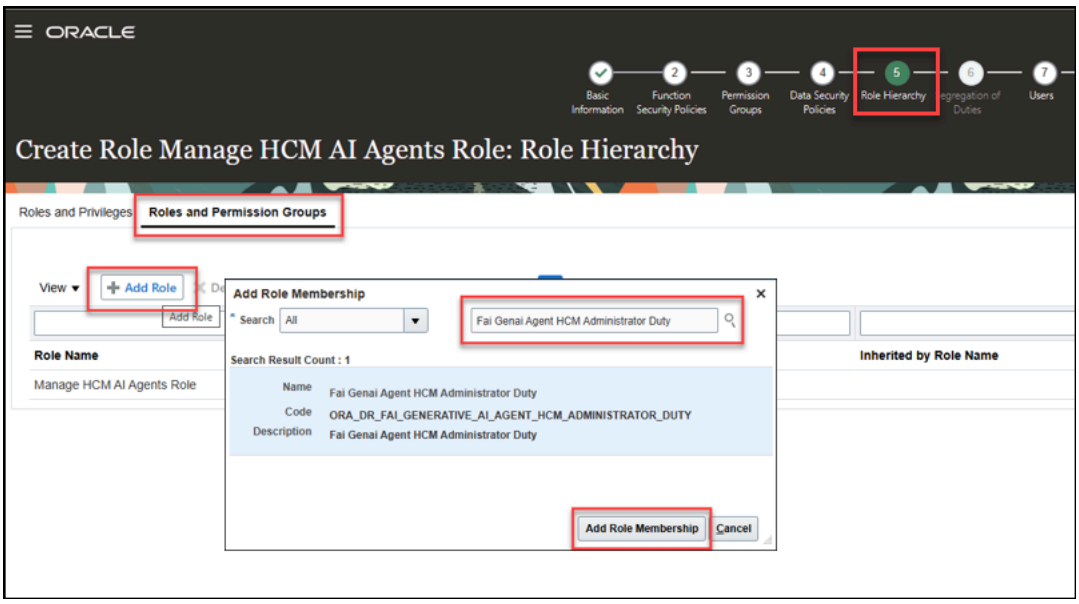




2. Go to the Role Hierarchy page.
    - Open the Roles and Privileges tab, and add the Manage HCM Intelligent Agent (ORA_HRC_HCM_AI_AGENT_MANAGEMENT_DUTY) duty role.



    - Open the Roles and Permission Groups tab, and add the Fai Genai Agent HCM Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_HCM_ADMINISTRATOR_DUTY) role.

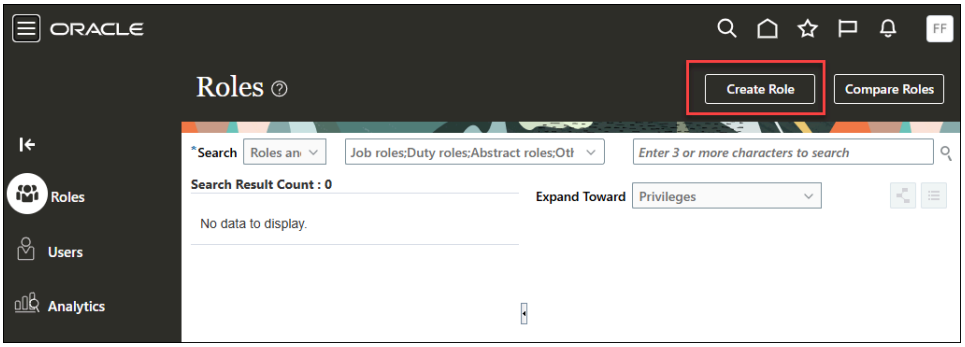


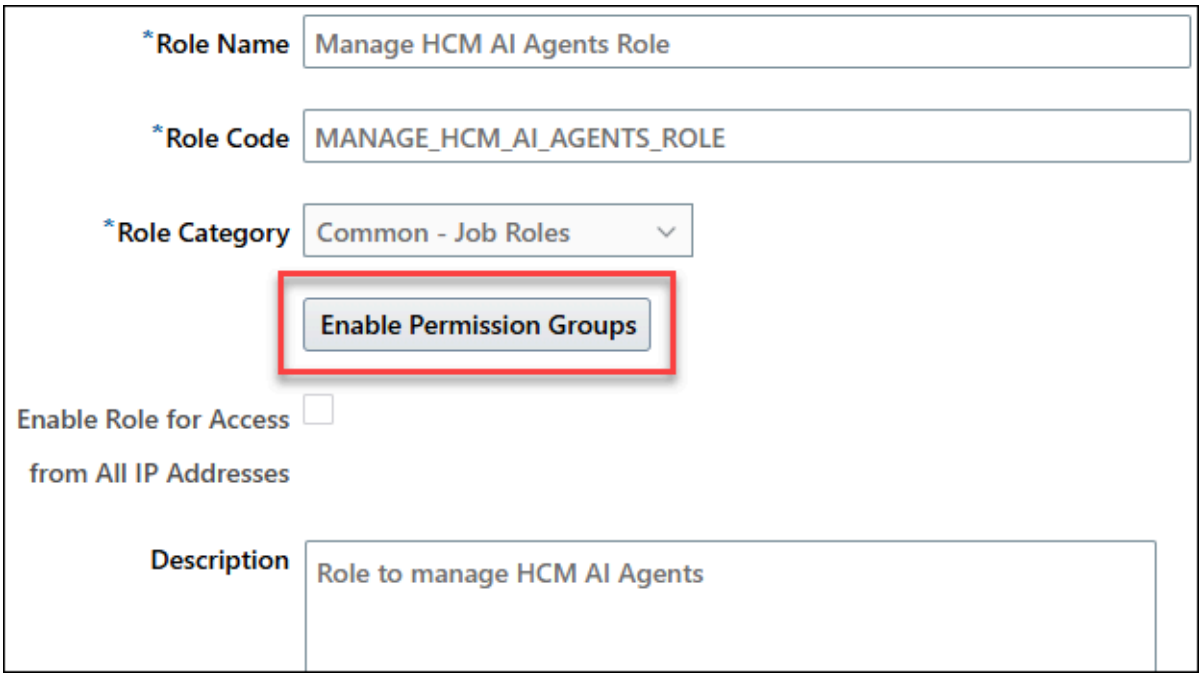
3. Save the custom role and assign to the appropriate job roles.

To give access to users with the Human Capital Management Application Administrator Job Role:

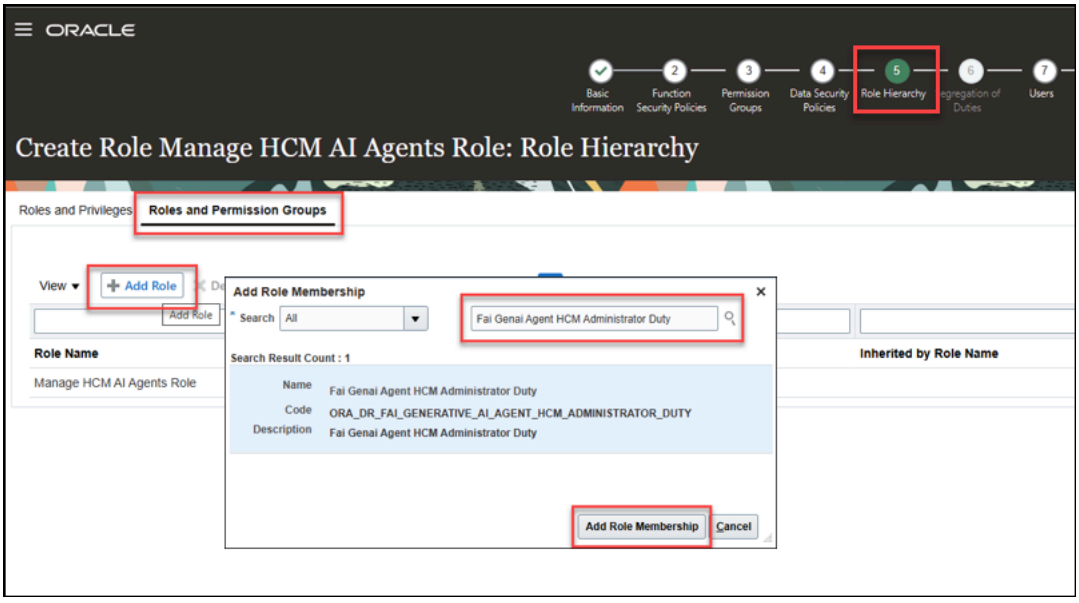
1. Go to the Security Console and create a new custom job role.

   Note: Make sure to enable permission groups.





2. On the Role Hierarchy page, open the Roles and Permission Groups tab, and add the Fai Genai Agent HCM Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_HCM_ADMINISTRATOR_DUTY) role.



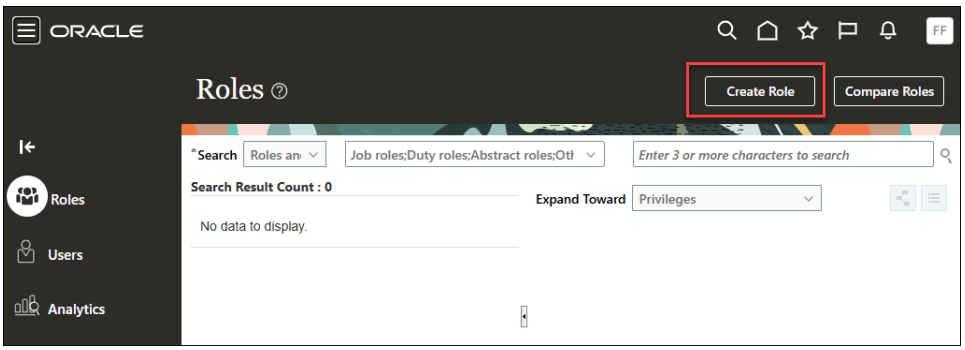
3. Save the custom role and assign to the appropriate job roles.

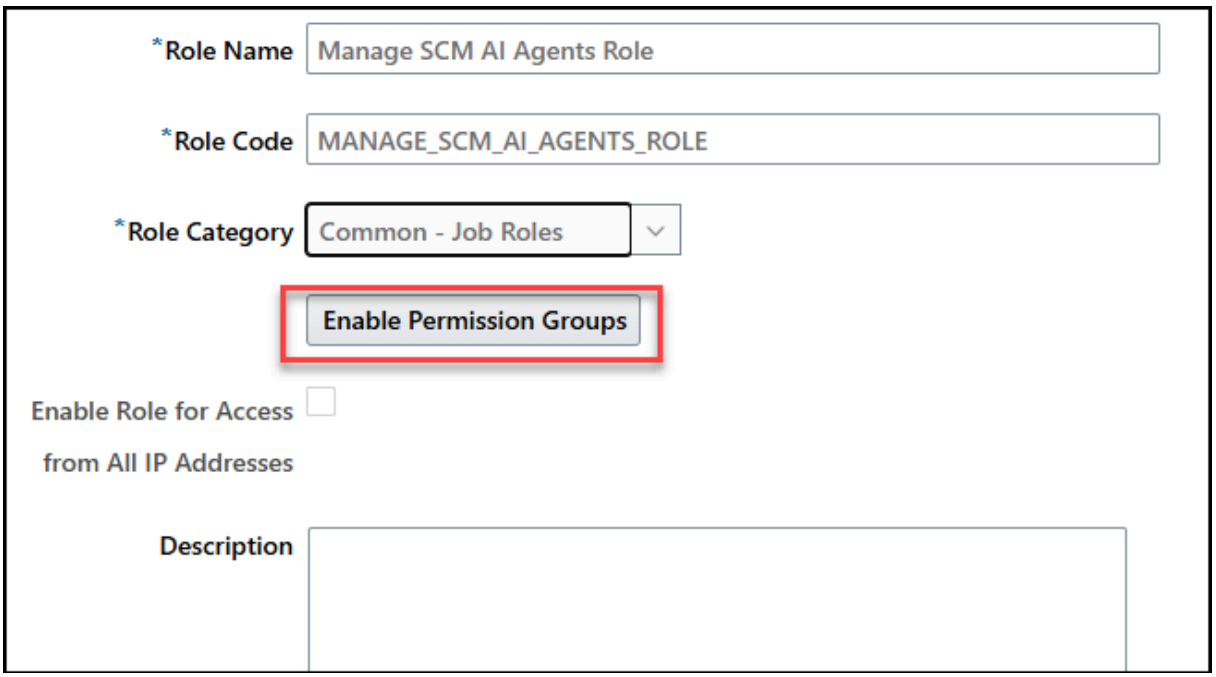
Provide Access to Configure AI Agents in Oracle Fusion Cloud Supply Chain & Manufacturing

To give access to users without the Supply Chain Application Administrator Job Role:

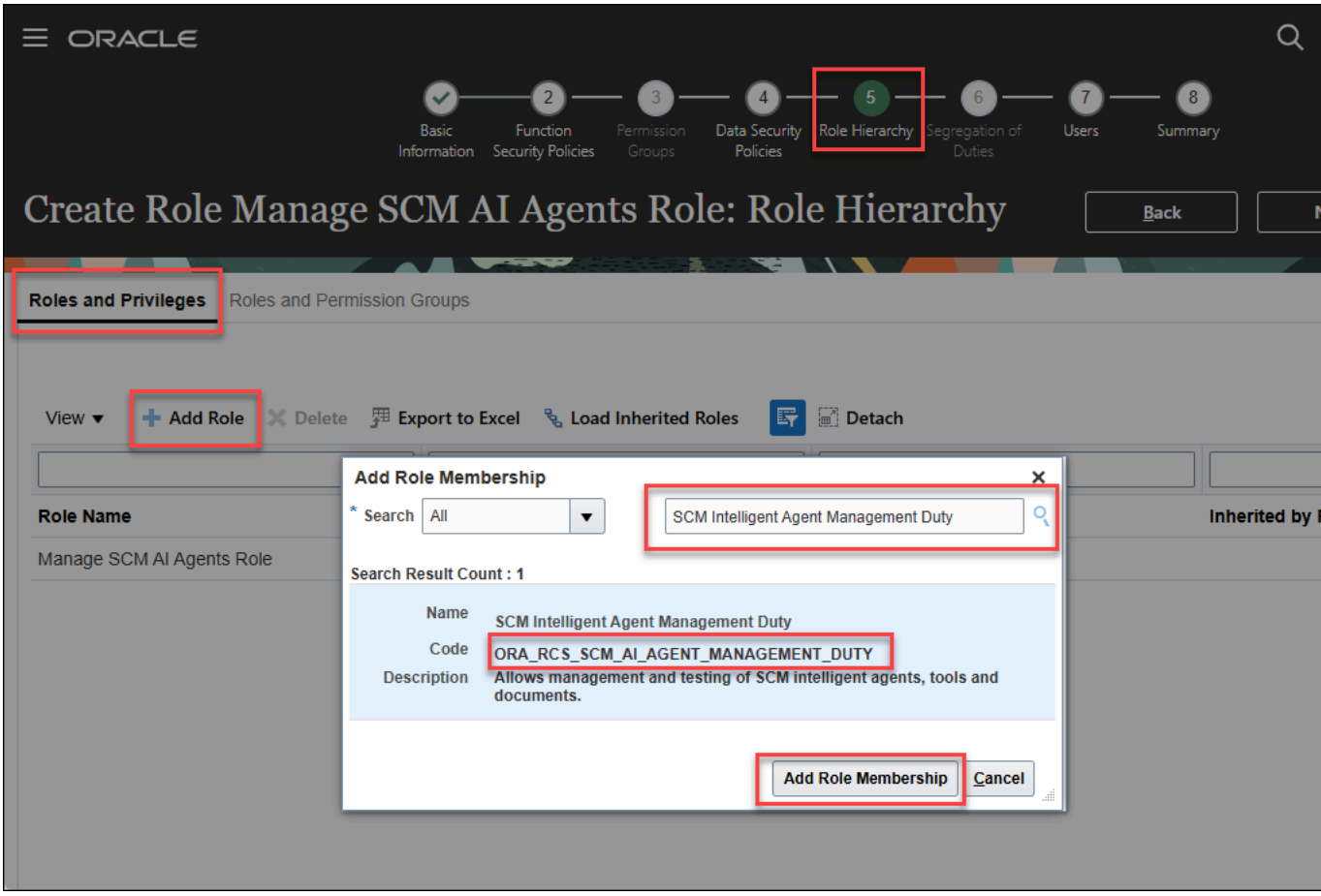
1. Go to the Security Console and create a new custom job role.

> **Note:** Make sure to enable permission groups.

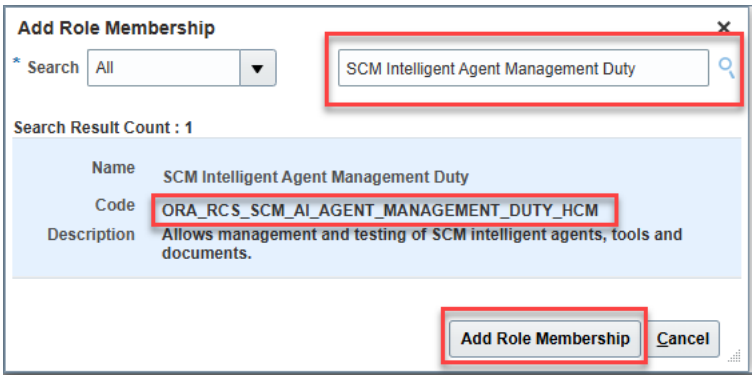



*Role Name
Manage SCM AI Agents Role
*Role Code
MANAGE_SCM_AI_AGENTS_ROLE
*Role Category
Common - Job Roles
Enable Permission Groups
Enable Role for Access from All IP Addresses
Description

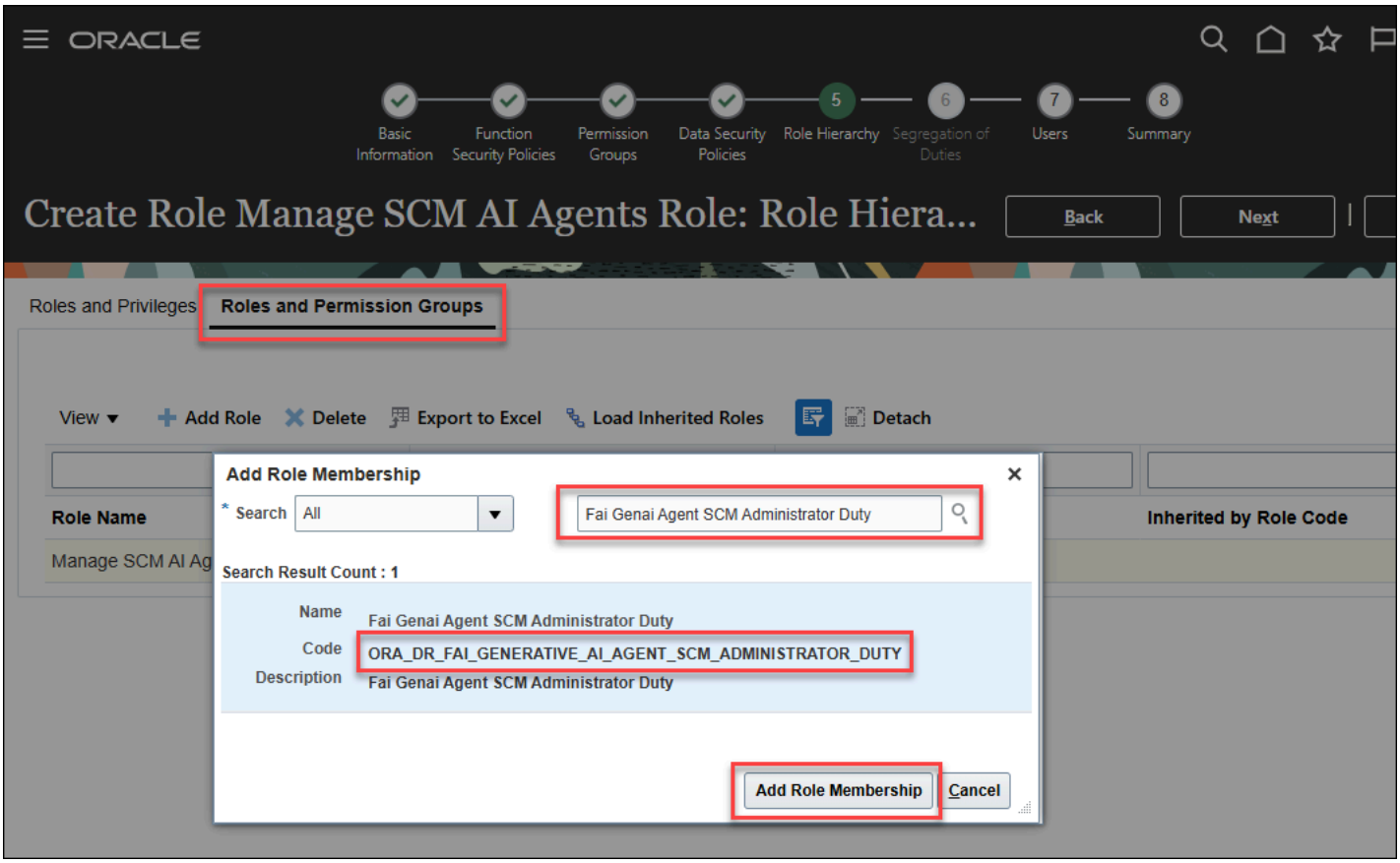
2. On the Role Hierarchy page, open the Roles and Privileges tab and add these roles:
   - SCM Intelligent Agent Management Duty (ORA_RCS_SCM_AI_AGENT_MANAGEMENT_DUTY)



   - SCM Intelligent Agent Management Duty (ORA_RCS_SCM_AI_AGENT_MANAGEMENT_DUTY_HCM)



3. Open the Roles and Permission Groups tab and add the Fai Genai Agent SCM Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_SCM_ADMINISTRATOR_DUTY) duty role.

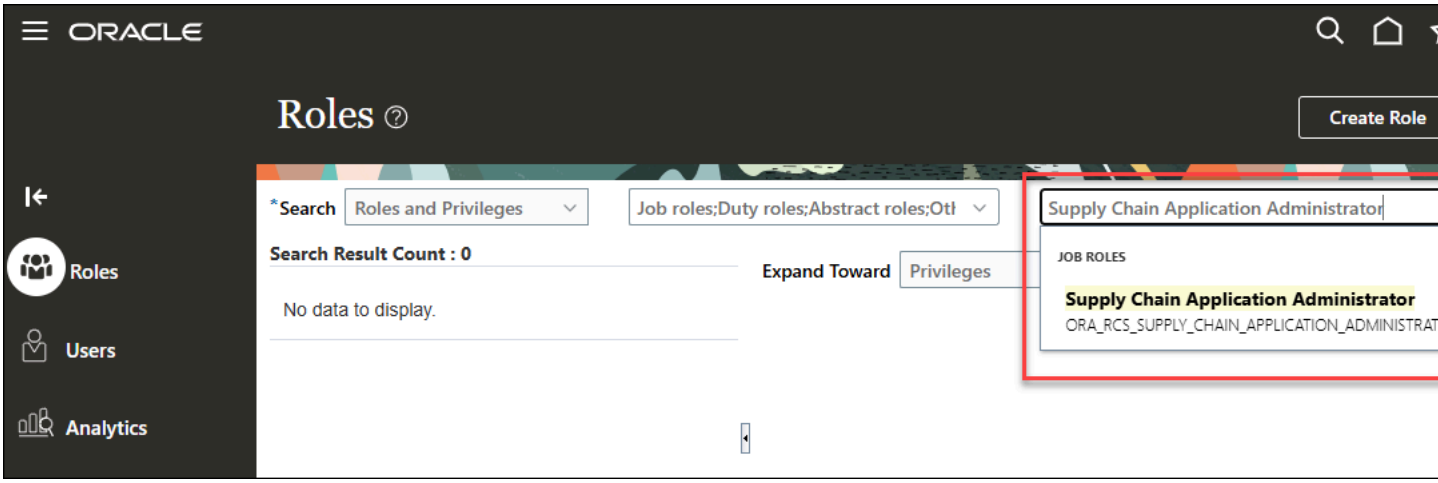


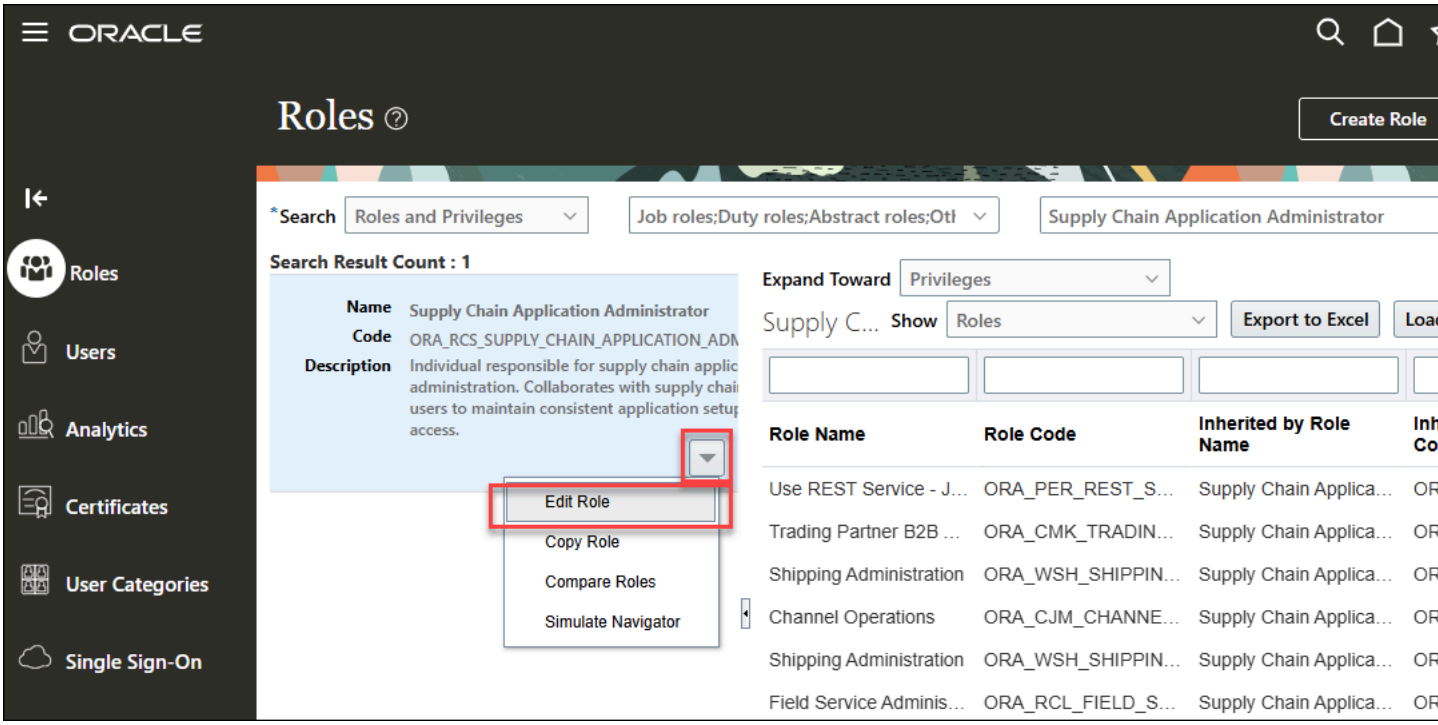
4. Save the custom role and assign this custom role to the appropriate job roles.

To give access to users with the Supply Chain Application Administrator Job Role:

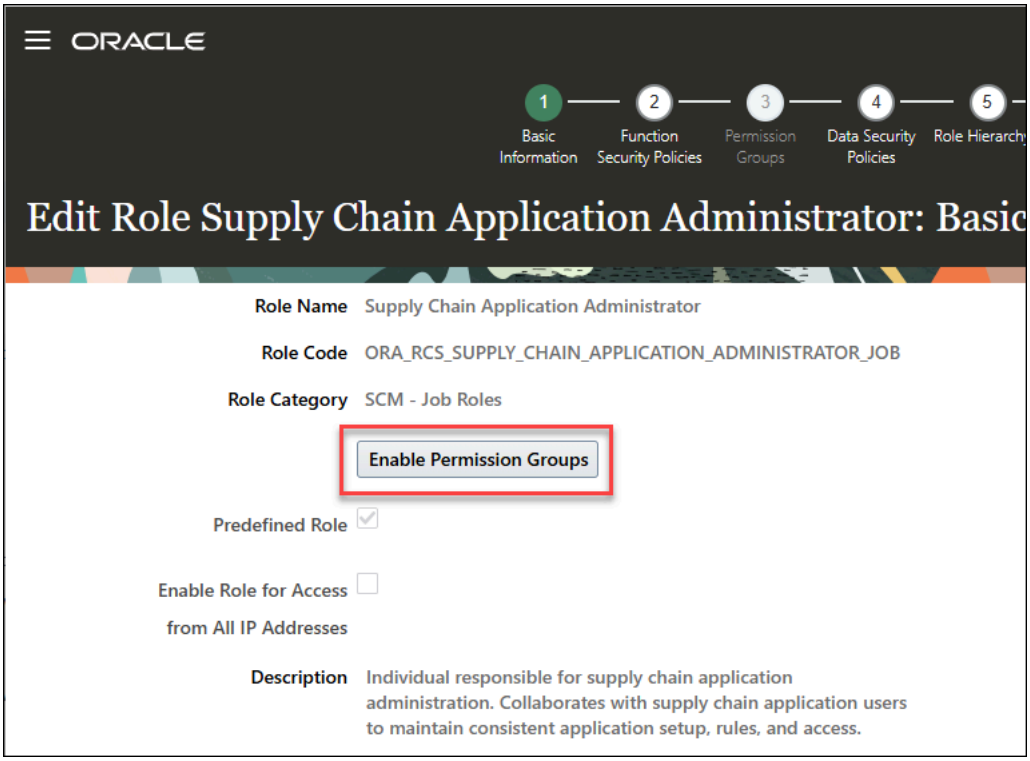
1. Go to the Security Console.

2. Search for the Supply Chain Application Administrator (ORA_RCS_SUPPLY_CHAIN_APPLICATION_ADMINISTRATOR_JOB) job role, and edit it.





3. Enable permission groups and save the job role.

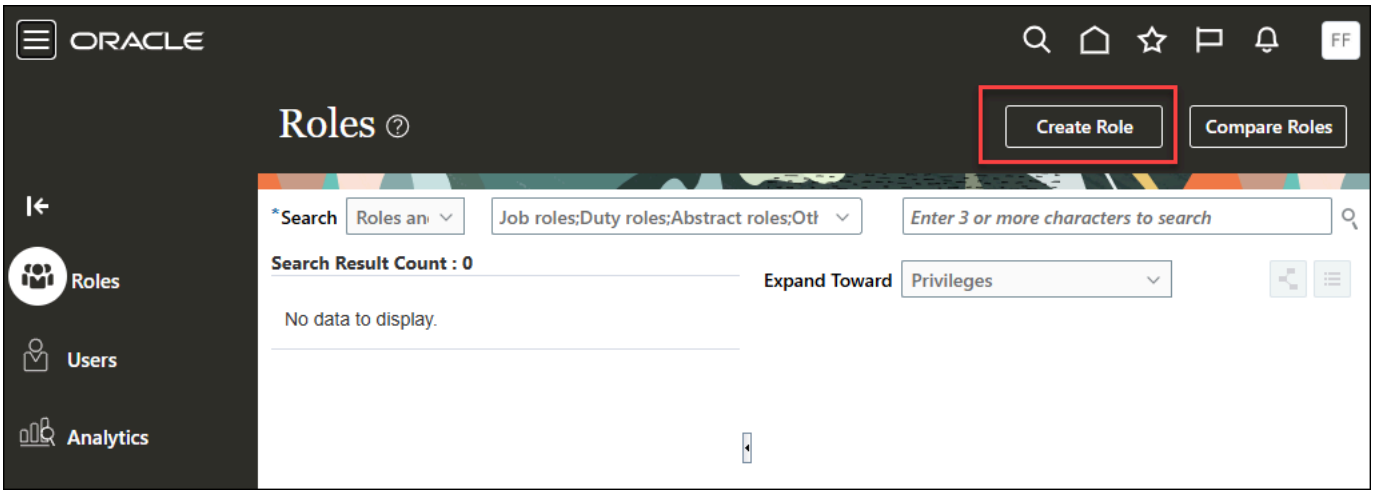


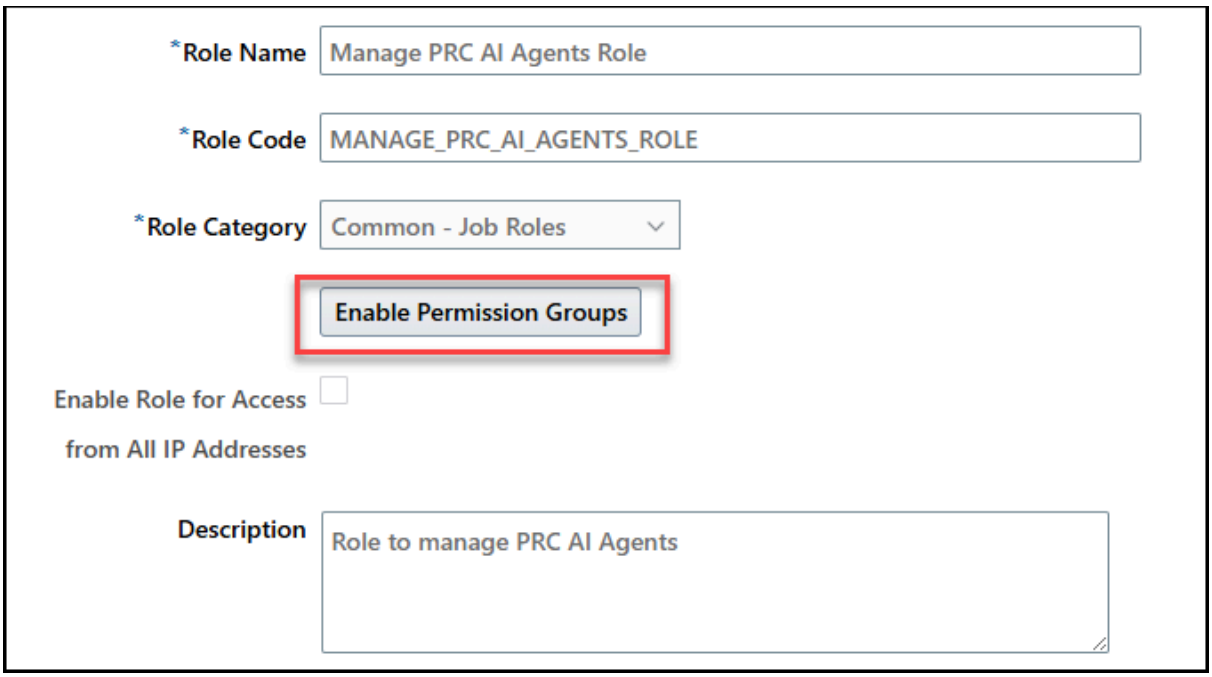
Provide Access to Configure AI Agents in Oracle Fusion Cloud Procurement

To give access to users without the Procurement Application Administrator Job Role:

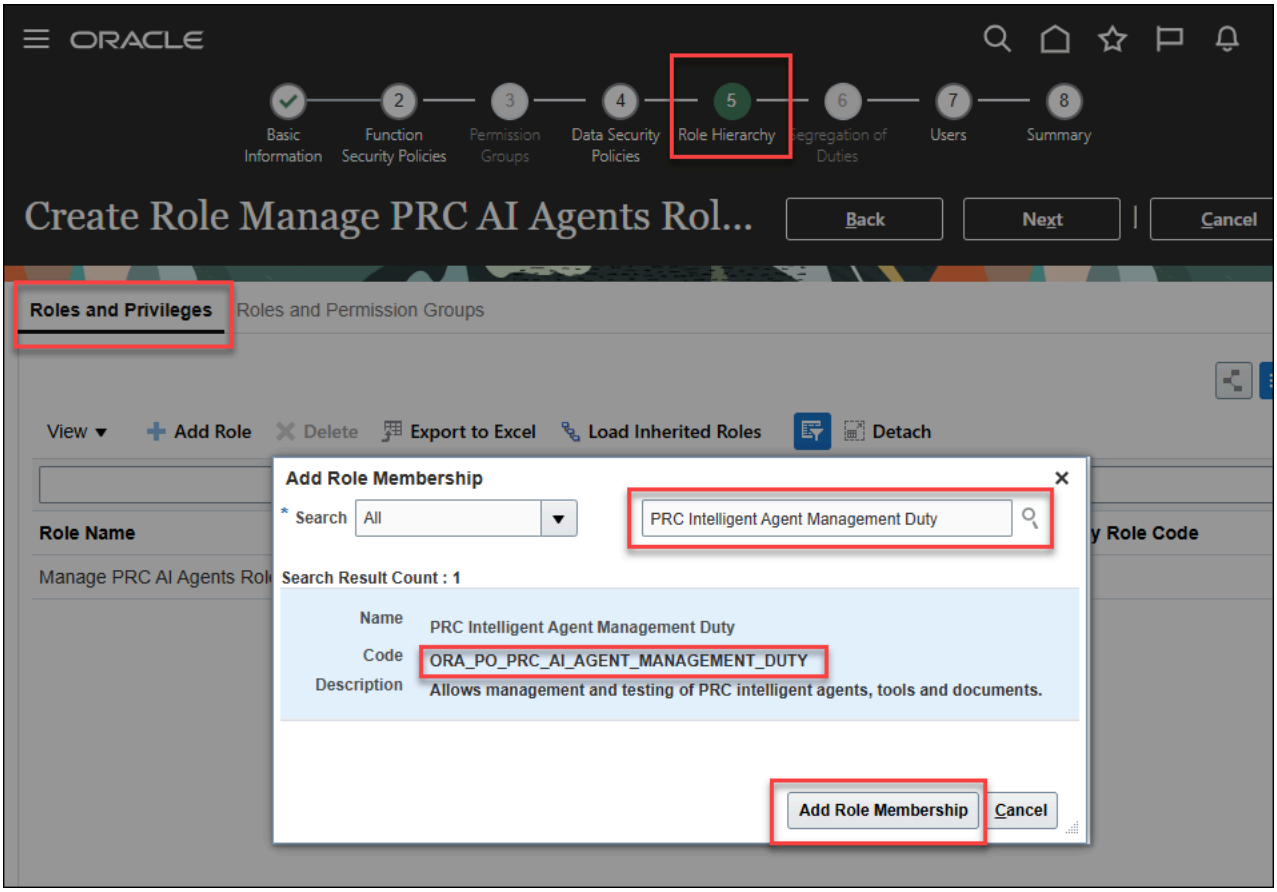
1. Go to the Security Console and create a new custom job role.

   Note: Make sure to enable permission groups.

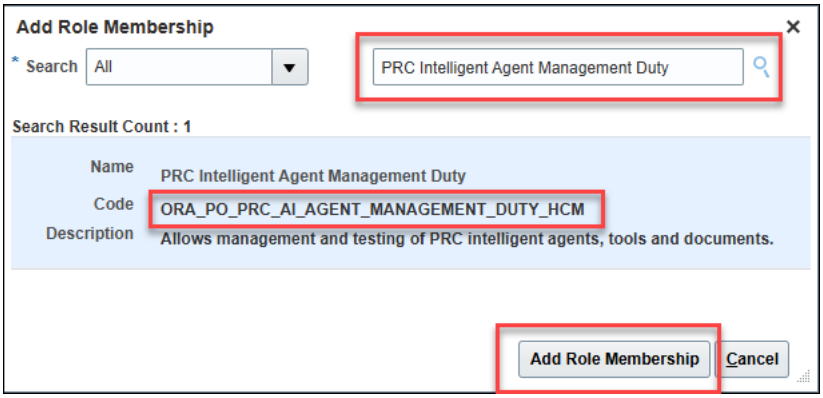




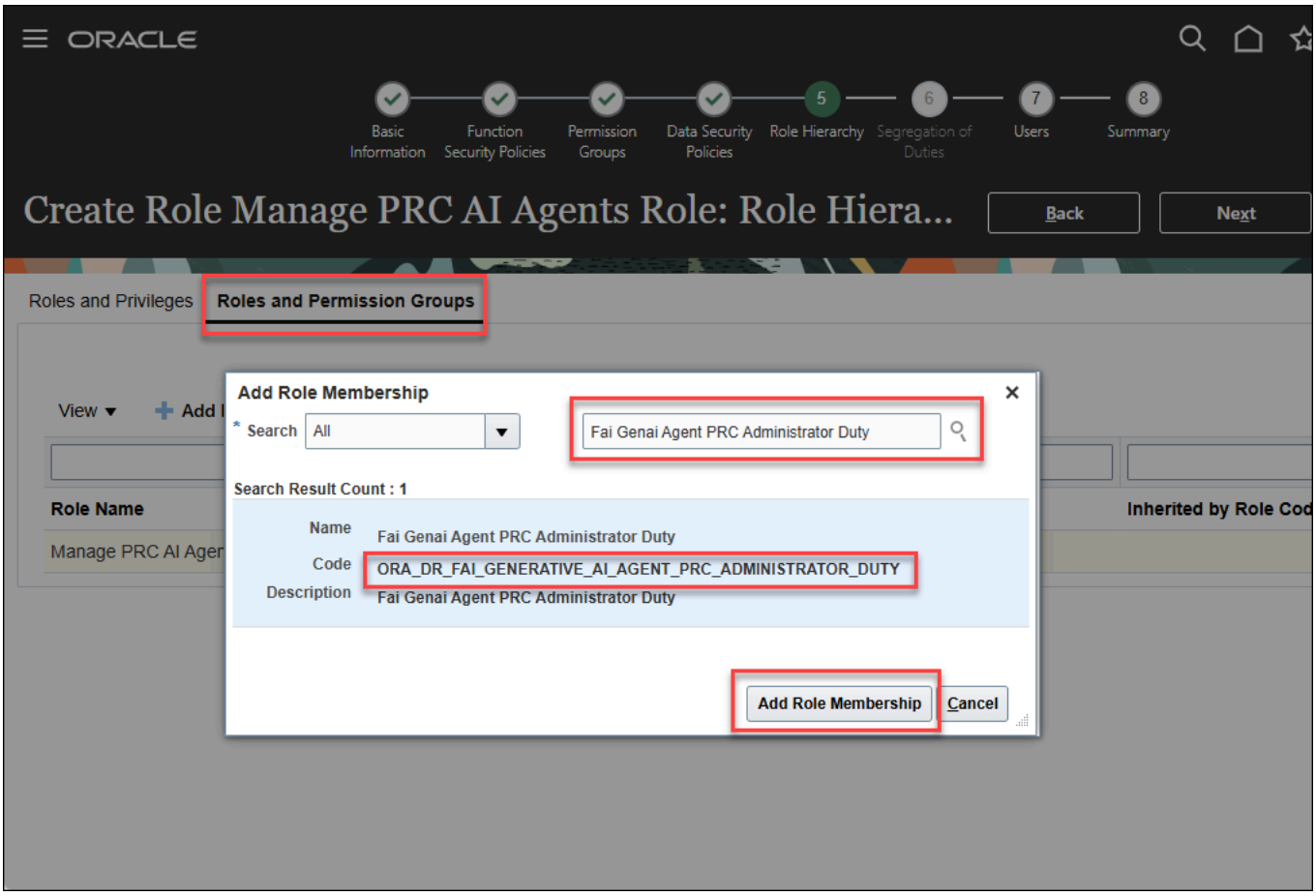
2. On the Role Hierarchy page, open the Roles and Privileges tab and add these roles:
   - PRC Intelligent Agent Management Duty (ORA_PO_PRC_AI_AGENT_MANAGEMENT_DUTY)



   - PRC Intelligent Agent Management Duty (ORA_PO_PRC_AI_AGENT_MANAGEMENT_DUTY_HCM)



3. Open the Roles and Permission Groups tab and add the Fai Genai Agent PRC Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_PRC_ADMINISTRATOR_DUTY) duty role.

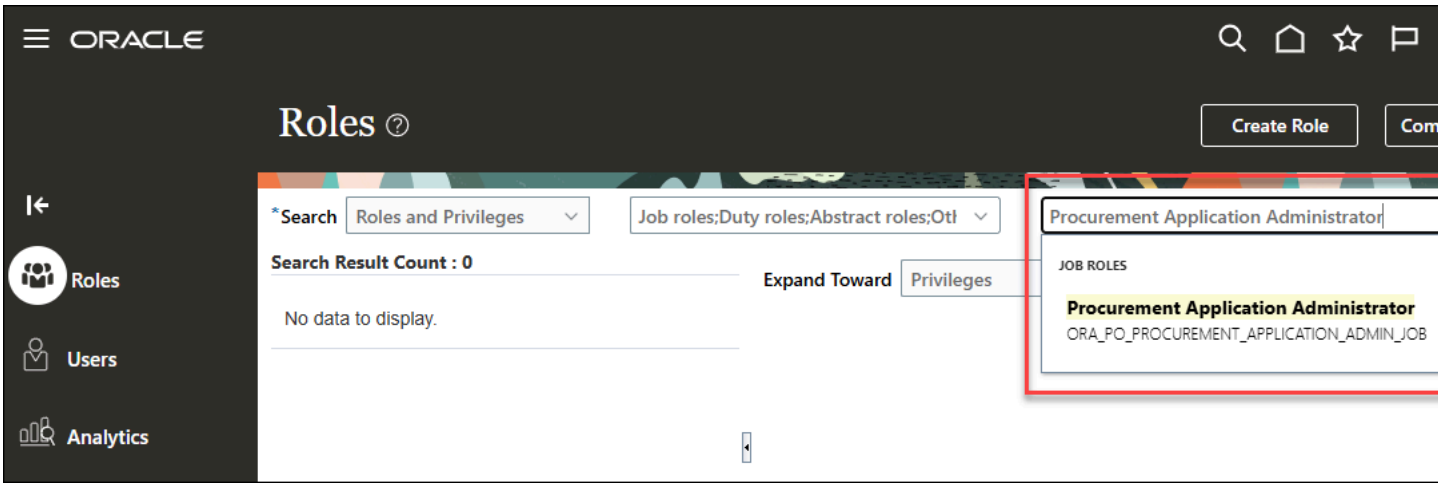


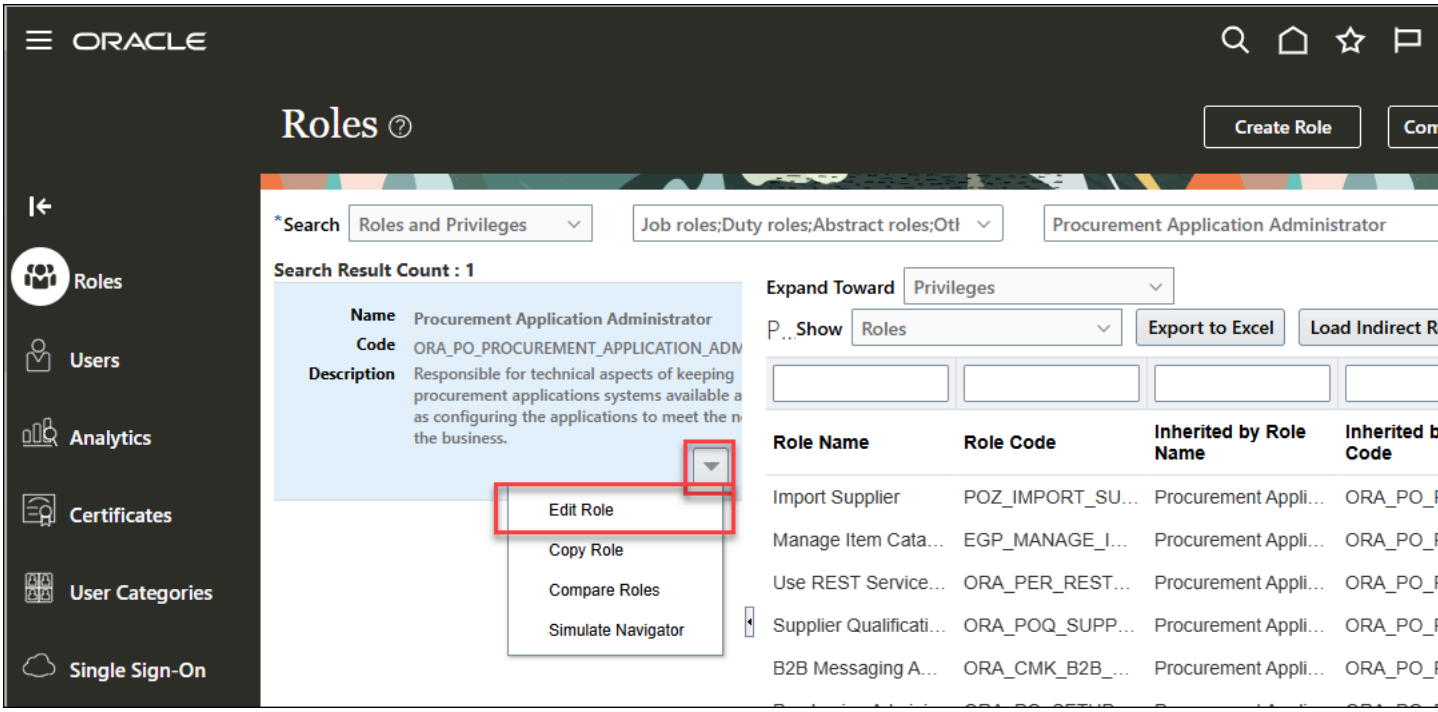
4. Save the custom role and assign this custom role to the appropriate job roles.

To give access to users with the Procurement Application Administrator Job Role:

1. Go to the Security Console.

2. Search for the Procurement Application Administrator (ORA_PO_PROCUREMENT_APPLICATION_ADMIN_JOB) job role, and edit it.





3. Enable permission groups and save the job role.

Provide Access to Configure AI Agents in Oracle Fusion Cloud Financials

To give access to users without the Financial Application Administrator Job Role:

1. Go to the Security Console and create a new custom job role.

   Note: Make sure to enable permission groups.

2. Go to the Role Hierarchy page.
   - Open the Roles and Privileges tab, and add these duty roles:
     - Manage Financials Intelligent Agent (ORA_FUN_MANAGE_FIN_AI_AGENT_HCM)
     - Manage Financials Intelligent Agent (ORA_FUN_MANAGE_FIN_AI_AGENT)
     - AI Agent Connector Management(ORA_CSO_CONNECTORS_AI_AGENT_MANAGEMENT_DUTY)
   - Open the Roles and Permission Groups tab, and add the Fai Genai Agent FIN Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_FIN_ADMINISTRATOR_DUTY) duty role.
3. Save the custom role and assign it to the appropriate job roles.

To give access to users with the Financial Application Administrator Job Role:

1. Go to the Security Console.
2. Search for the Financial Application Administrator (ORA_FUN_FINANCIAL_APPLICATION_ADMINISTRATOR_JOB) job role, and edit it.
3. Enable permission groups and save the job role.

Provide Access to Configure AI Agents in Oracle Fusion Cloud CX

To give access to users without the Sales Administrator Job Role:

1. Go to the Security Console and create a new custom job role.

   > **Note:** Make sure to enable permission groups.

2. On the Role Hierarchy page, open the Roles and Privileges tab and add these roles:
   - Manage CX AI Agents (ORA_ZCA_MANAGE_CX_AI_AGENTS)
   - Manage CX AI Agents (ORA_ZCA_MANAGE_CX_AI_AGENTS_HCM)
3. Open the Roles and Permission Groups tab and add the Fai Genai Agent CX Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_CX_ADMINISTRATOR_DUTY) duty role.
4. Save the custom role and assign this custom role to the appropriate job roles.

To give access to users with the Sales Administrator Job Role:

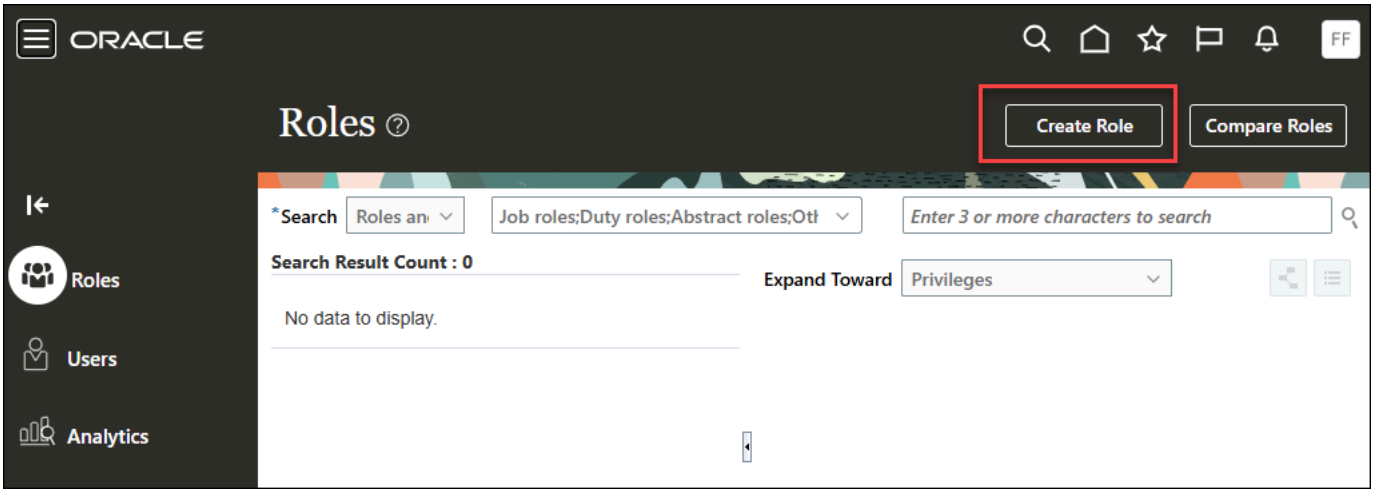
1. Go to the Security Console and search for the Sales Administrator (ORA_ZBS_SALES_ADMINISTRATOR_JOB) job role, and edit it.
2. Enable permission groups and save the job role.

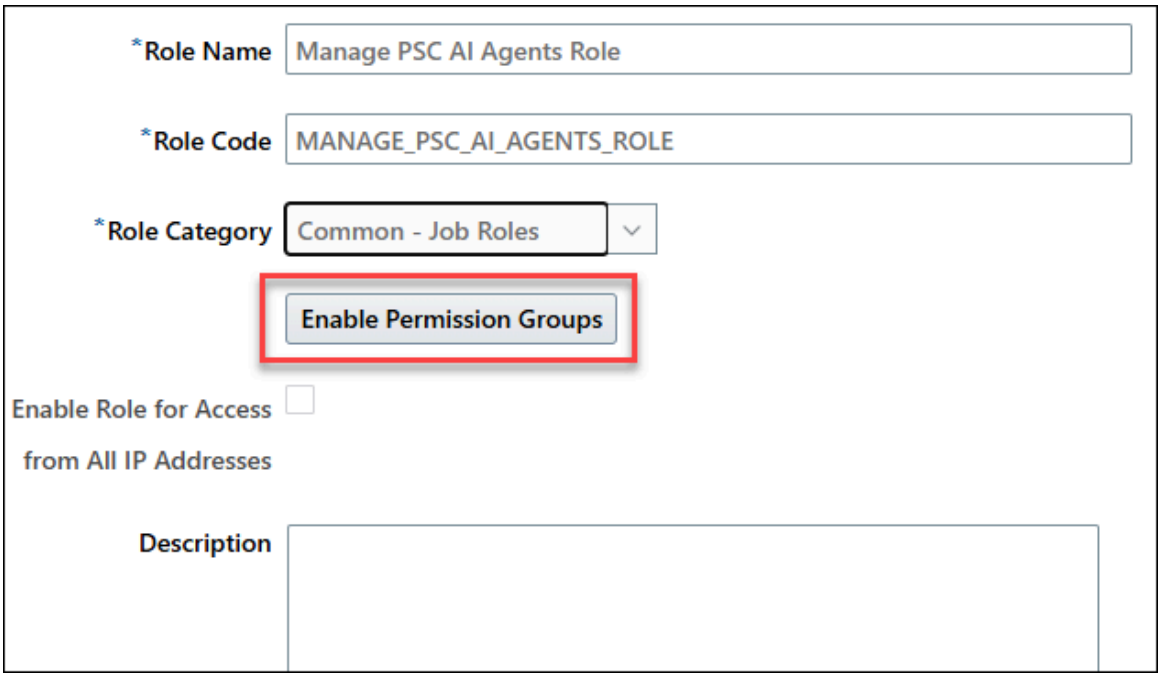
Provide Access to Configure AI Agents in Oracle Permitting and Licensing

To give access to users without the PSC Application Administrator Job Role:

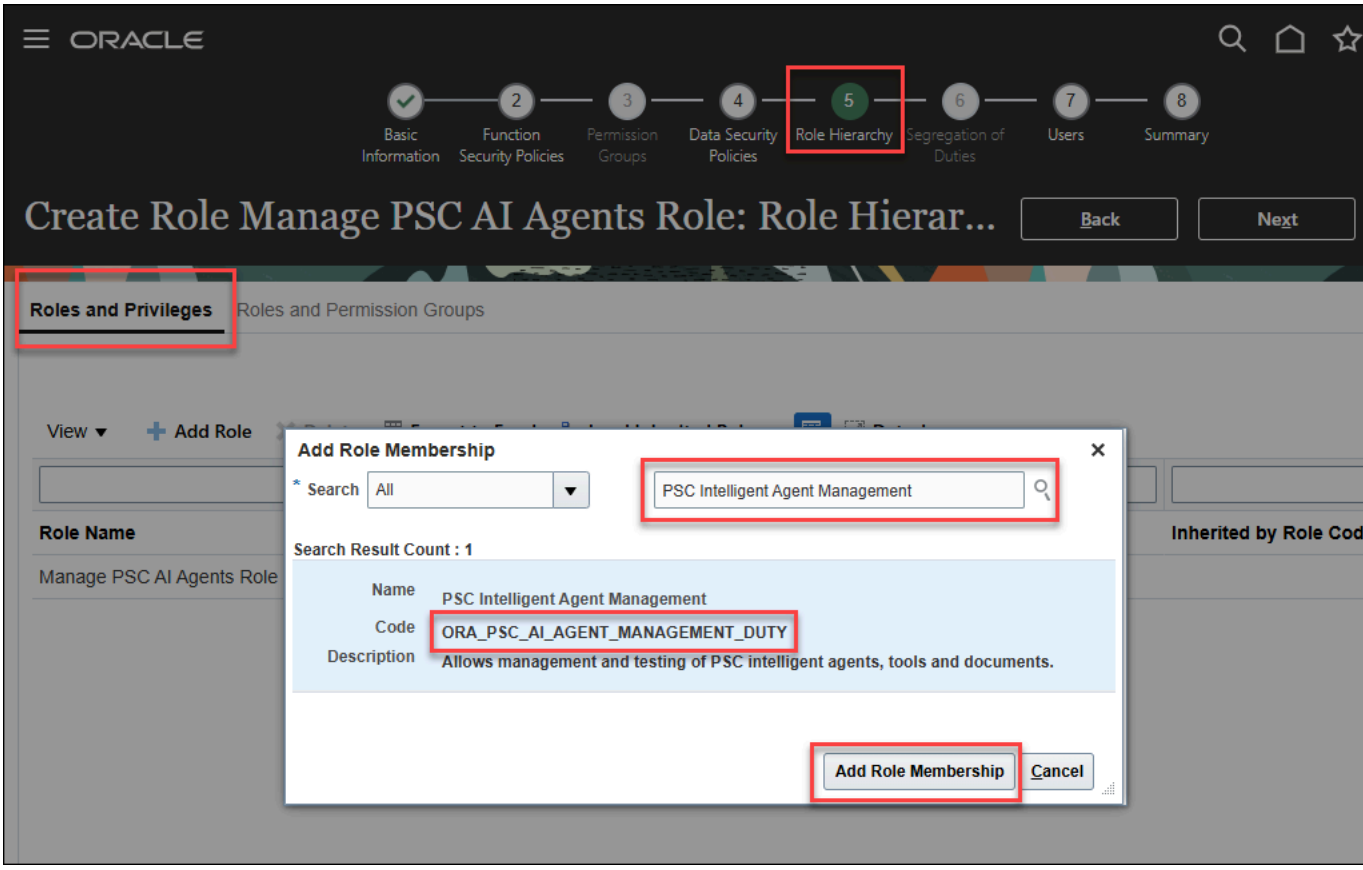
1. Go to the Security Console and create a new custom job role.

Note: Make sure to enable permission groups.

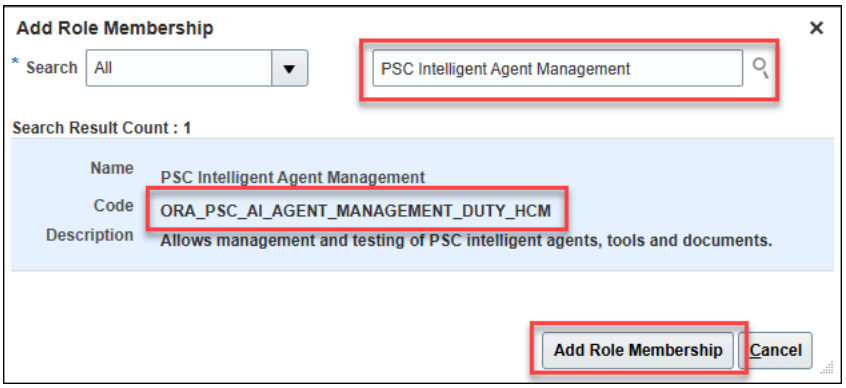




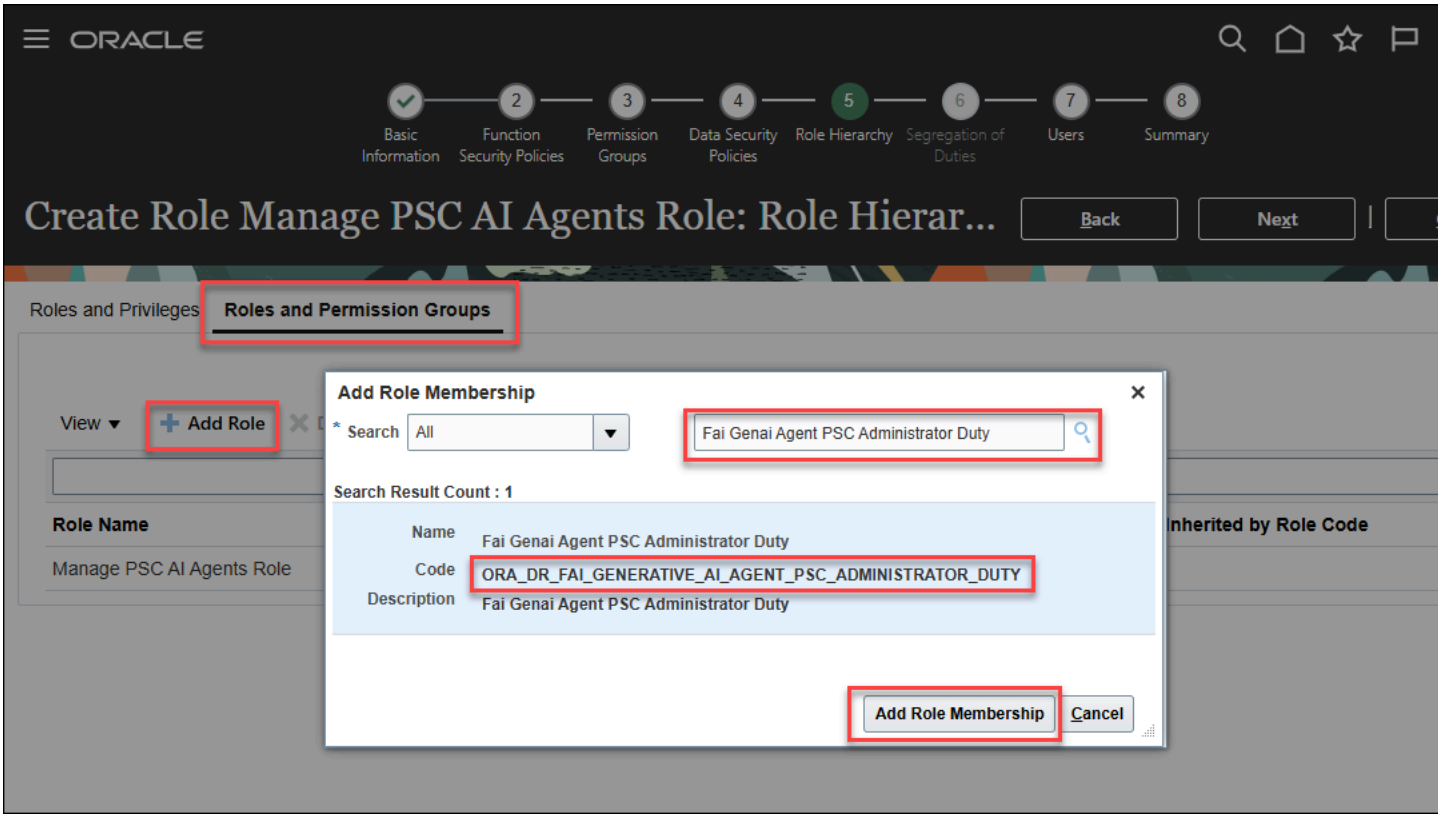
2. On the Role Hierarchy page, open the Roles and Privileges tab and add these roles:
   - PSC Intelligent Agent Management (ORA_PSC_AI_AGENT_MANAGEMENT_DUTY)



   - PSC Intelligent Agent Management (ORA_PSC_AI_AGENT_MANAGEMENT_DUTY_HCM)



3. Open the Roles and Permission Groups tab and add the Fai Genai Agent PSC Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_PSC_ADMINISTRATOR_DUTY) duty role.

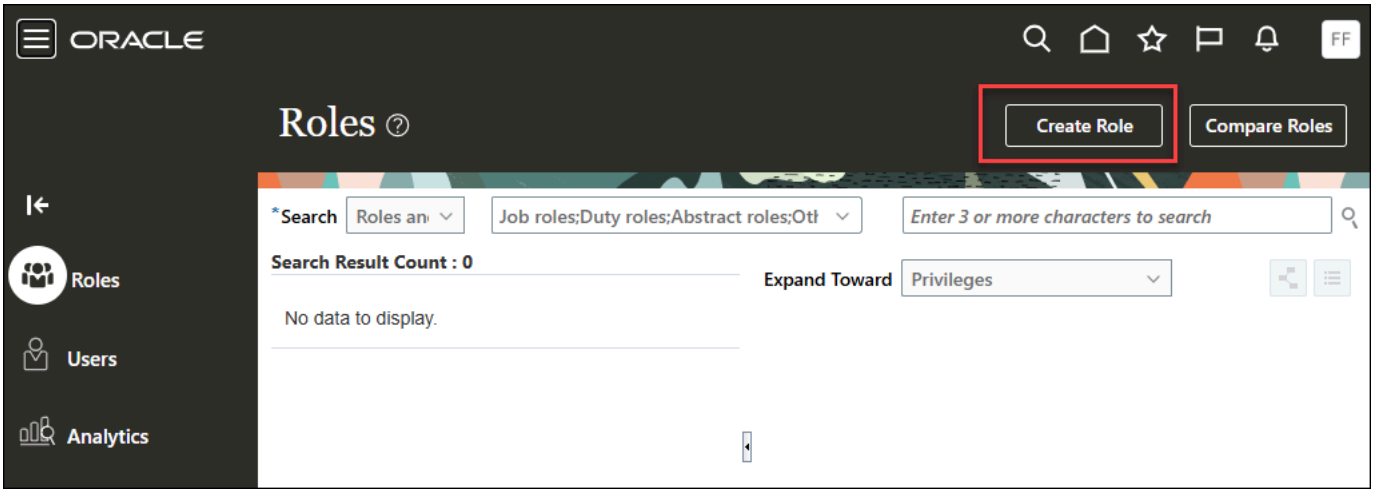


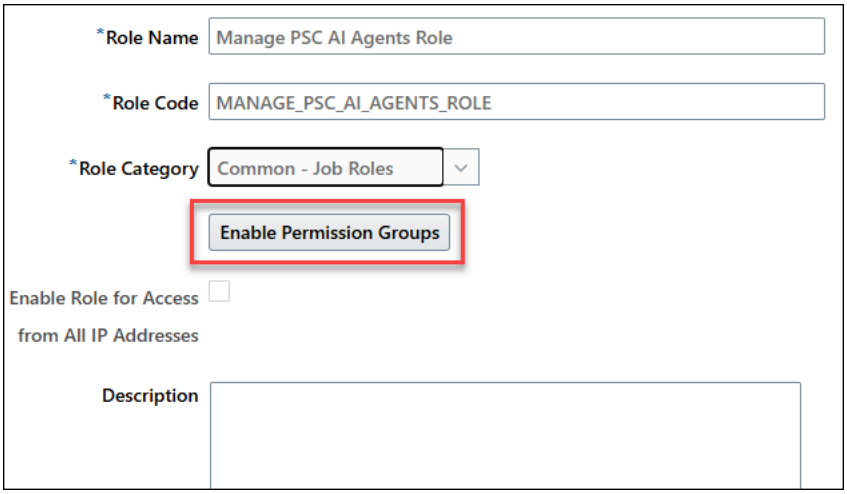
4. Save the custom role and assign this custom role to the appropriate job roles.

To give access to users with the PSC Application Administrator Job Role:

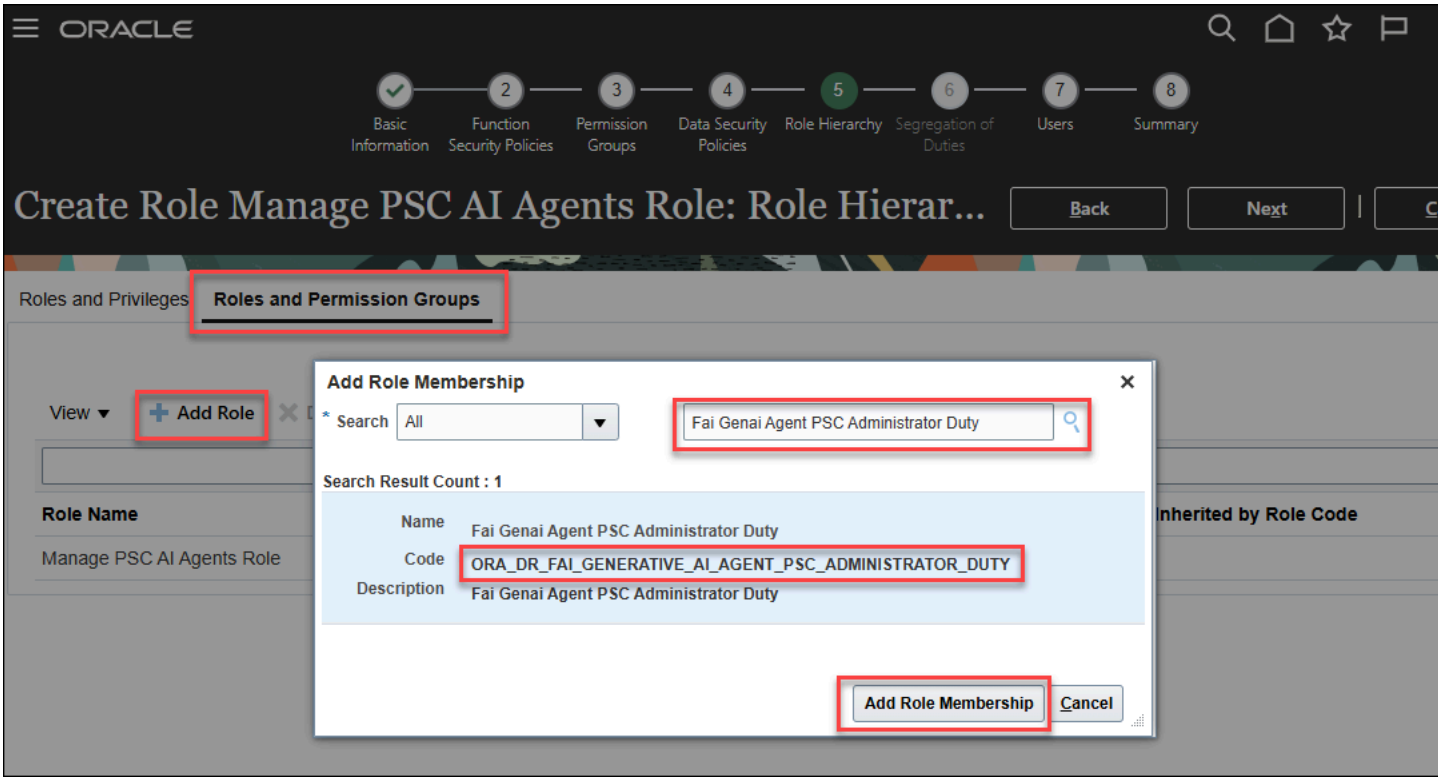
1. Go to the Security Console and create a new custom job role.

   Note: Make sure to enable permission groups.





2. On the Role Hierarchy page, open the Roles and Permission Groups tab and add the Fai Genai Agent PSC Administrator Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_PSC_ADMINISTRATOR_DUTY) duty role.



3. Save the custom role and assign this custom role to the appropriate job roles.

3 Before You Begin

Choose How to Create AI Agents

You can either use one of the preconfigured templates, modify them, or create your own agents.

Here are some examples of how you choose between creating custom agents or using preconfigured templates, based on your use case.

Use Case	How to Create AI Agents
You want to turn on Benefits Agent, which allows employees to ask questions about their benefits, within AI Agent Studio. To ensure that this agent can understand the benefits specific to the organization, the documents specific to that organization must be uploaded.	Create an AI agent using a preconfigured agent team template. You can edit the existing agent, without adding any additional tools. **Note:** You can either use or copy a template and edit it. When you copy an agent template, you can automatically add a suffix to all the components to easily differentiate them.
You want to create a custom agent to help with onboarding by answering new hire questions and providing deep links to internal new hire resources if the agent is unable to answer. You can tailor the agent's topics and prompts to ensure they effectively address the specific needs of new hires.	Create a custom agent, and add the necessary tools and topics.
You want to create a Contract Assistant Agent to generate a new contract with auto-filled fields. This assistant will be provided with multiple contract templates to generate the new contract. The agent also reviews existing contracts to perform risk analysis, recommend changes and route the contract for final approvals. For these activities that need distinct expertise (template management, legal validation, recommendation, and routing), the administrator will need to build a supervisor agent that collaborates with other worker agents.	You can use either of these methods: • Create a team of custom agents, with one agent added as a supervisor to manage the others. • Use a preconfigured agent team template, and add more agents and a supervisor agent to it.
You want to create a Supplier Quote Assistant agent to parse through quotes from multiple suppliers. The agent	Create an agent team of type Workflow, to run the predetermined set of tasks. Each node can perform a defined function, for example, extracting data, calling a business object, running an LLM, or sending an email in the order you define.

Use Case	How to Create AI Agents
processes the quote document and maps the attributes. It then creates a draft requisition and submits for approval. For these activities, administrator will need to create a workflow agent that runs the tasks in the specified order.	

Get Started with AI Agent Studio

You can start with a preconfigured agent template or create your own agent team.

When using a preconfigured agent team template, the artifacts such as agents, tools, and topics, aren't directly editable. To change these artifacts, you create a copy of the artifact and add the copy to the agent team.

Note: Agents created directly within an agent team are directly editable within that team.

Here's a broad outline of the tasks involved in creating an agent team.

Task	Details
Define tools	To effectively define the tools required by an agent, you need to first identify the types of questions users might ask and then decide which tools the agent needs to answer those questions accurately. These are the available tools: • Oracle Fusion Cloud Applications business object: Business Object tools allow AI agents to retrieve, update, create, or delete business object records within Fusion Applications. Using this tool, agents can securely access information or call specific functions in the application. You can control what data the agents can access, by selecting the business objects and fields to use or ignore. AI agents adhere to the native security and role-based access controls of Fusion Applications, ensuring protection and privacy for your enterprise data. • Connector: You can use this tool to connect your AI agents to various data sources such as SharePoint, web pages, and other custom content source. For more information, see *How do I use connectors in AI Agent Studio?* • Document tool: You can upload specific documents to be used by the AI agent, and provide natural language instructions on how the agent should use these documents. The agent can then search for information in the documents to provide a more exact answer to a user's question. • Email: This tool can access the email client to send emails that include summaries of interactions or details pulled from a knowledge store. • Deep link: A deep link will send a user directly to the part of Fusion Applications where they can update underlying information. For example, if the user moved and wants to update their home address in the HR system, a deep link can quickly route the user to the page for making that update. • External REST: You can connect to internal and external SaaS applications or public APIs using External REST tool. To connect to internal and third-party services, add the authorization

Task	Details
	parameters. For example, you can retrieve the real-time weather information for a specific location by creating an External REST tool that connects to the appropriate external API. • MCP: You can use the Model Context Protocol (MCP) tool and securely connect to external MCP servers, without building additional REST wrappers or plugin logic.
Define topics	Topics define the focus of the agent to a specific area of expertise. They are prompts that you can reuse across multiple systems or summarization prompts. For example, within a Benefits Administrator agent, we might define topics such as health policy coverage, vision policy coverage, and benefits enrollment. Use Topics to efficiently streamline your interactions with the agent. • Specify the instructions that help the agent decide which tools to use. • Enable the agent to better understand user intent by letting it identify and select the relevant topic based on the user's question. • Give each topic a clear, specific name, and include natural language instructions to ensure it's used correctly. You can reuse topics across agents.
Define credentials	To enable access to more services, you need to provide the necessary connection credentials. You can add credentials for this artifact: Custom LLM: In addition to the LLMs provided by Oracle, you can use other LLMs you've access to. You can add credentials for your LLM, and select it while creating your agent, node, or agent team.
Build new agents	Define the capabilities and scope of the agent. • Agent name • Product area the agent will work in • Natural language instructions to allow the agentic flow, or other agents, to understand the capabilities of this particular agent • Tools and topics the agent will need to use. In addition to the tools and topics you created, you can also add the predefined ones. In the Tools and Topics tabs, the predefined tools and topics are indicated by the icon.
Add a user (human) in the loop	If required, add an approval step for some actions that your AI agent will perform. The review step can be added at any point in the process for oversight and control over key actions, such as sending an email or updating a record.
Build the agent team	Create an agent team and add agents and other artifacts to it. Types of agent teams are: • Supervisor: A supervisor agent manages other agents and artifacts in the agentic flow. • Workflow: An agentic flow that does the tasks in a predetermined order. Agents and artifacts are added as nodes in the workflow, and each node performs a defined function, for example, extracting data, calling a business object, running an LLM, or sending an email. The node then passes its output to the next step.

Task	Details
	Note: You can also add predefined agents to your agent team. In the Agents tab, the predefined agents are indicated by the icon.
Test the agents	Make sure to test the agent before deploying to production. Ask a test query, and determine the accuracy and relevance of the agent's response. You can also see the instructions the agent is following, and the actions the agent has taken to arrive at the response.
Deploy the agent team	After defining and testing your agent team, you can deploy it directly from AI Agent Studio. • Embed the agent conversation chat experience into any website or application. • Trigger the agent from an external resource using Webhooks, or seamlessly embed the chat experience into HTML and React web pages.

Migrate RAG Agents to AI Agent Studio

If you've previously created any RAG agents in Fusion Applications, we recommend replacing your existing agent with a new one created in AI Agent Studio. For more information, see *Migrate Document Tools of RAG Agents*.

Add External REST Tool

You can enable your agents to securely connect to internal or external SaaS applications and public APIs by adding External REST tools in AI Agent Studio. Make sure you have a role with the Create and Edit Backends for Visual Builder Studio (ORA_FND_TRAP_PRIV) privilege assigned to it.

1. Go to **AI Agent Studio**.
2. From the Tools tab, click **Add**.
3. Add a new tool of type **External REST**.
4. Enter the name, code, and description for the tool, and select the appropriate family and product.
5. Select **Require Human Approval** to ensure that a person reviews and approves any action before the tool runs.
    a. Message: Enter instructions for the reviewer.
    b. Actions: Select the required action.
6. In the Authorization subtab, add connection details for the third-party service.
    a. Instance URL: Enter the base domain of the external API, with no paths, parameters, or trailing slashes. For example, *https://api.weather.gov* is correct.
    b. Authentication: Define how your agents will authenticate with the external system.
    c. Description: Provide details about the tool, including what the tool does, and when the agent must call the tool.
7. In the Functions subtab, provide endpoint details.
    a. Name: Assign a unique name to this endpoint. For example, **getWeatherByCoordinates**.
    b. Operation Type: Select the HTTP method required to process the REST call against the target endpoint.
    c. Resource Path: Specify the relative resource path to be appended to the Instance URL. Don't include the protocol or base domain. For example, **/points/{latitude},{longitude}** is correct.
    d. Description: Provide a clear summary of the endpoint's function and the specific task it performs within the workflow.

8. In the Parameters subtab, define each dynamic variable in the path, including data type and purpose. For example, you can define latitude as the first parameter and longitude as the second.
9. In the Sample Queries subtab, list example user queries that should trigger this tool.
10. In the Headers subtab, specify any required metadata and security credentials for the external server.

The tool now appears in the Tools tab, and you can select it when creating agents, nodes, or agent teams.

Add MCP Tool

Using Model Context Protocol (MCP) tool, you can securely connect to external MCP servers, and use their capabilities inside agents and nodes, without building additional REST wrappers or plugin logic.

1. Go to **AI Agent Studio** and open the Tools tab.
2. Add a new tool of type **MCP**.
3. Enter the name, code, and description for the tool, and select the appropriate family and product.
4. Add MCP server connection details.
    - Instance URL: Enter the endpoint URL of the MCP server your agent will connect to.
    - Transport Type: Select the communication method:
        - Server Sent Events (SSE): Enables real-time, one-way streaming of data from a server to a client.
        - StreamableHTTP: Enables servers to independently handle multiple client connections using the HTTP POST and GET requests.
    - Credential Type: Select the authentication method required by the MCP server.
5. Update the details and create the tool.

Your tool now appears in the Tools tab, and you can select it when creating agents, nodes, or agent teams.

Add Your LLM

In addition to the LLMs provided by Oracle that are available for selection when creating an agent, node, or agent team, you can also use other LLMs. You can use the additional LLMs available in AI Agent Studio by adding the credentials for it. If the model you want to use isn't available in the list of models for the supported provider, you can submit a request through *My Oracle Cloud Support* to get the model added.

> **Note:** Oracle updates the list of Bring Your Own LLM options on a monthly basis and may remove models that have been deprecated, based on information provided by their respective model providers. You're responsible for obtaining and managing all required third-party provider credentials, associated service costs, and custom model configurations. You must also monitor model lifecycle changes, keep configurations up-to-date, and complete any required migrations to supported model versions within the timelines specified by the model provider.

You can add an LLM from any of these supported providers:

- OpenAI (through Microsoft Azure)
- Google Gemini (through Google Vertex AI)

- Google Gemini (through direct API)
- Anthropic (through Google Vertex AI)
- Oracle Cloud Infrastructure (through Oracle)

To add the credentials for available LLMs, do these steps:

1. Go to AI Agent Studio and open the **Credentials** tab.
2. In the LLM subtab, add the API key.
3. Add the details for the provider and save it.
    - Model: Choose the model you need from the available options.
    - API Key: Add the API key for the selected model.
    - Basic URL: Enter the endpoint URL that will receive requests for the LLM.

Your LLM now appears in the LLM tab, and you can select it when creating agents, nodes, or agent teams.

To request a new LLM model, do these steps:

1. Create a service request through *My Oracle Cloud Support*.
2. Add the details for the approver.
    - Supported provider name shown in AI Agent Studio.
    - Requested model and version.
    - Business use case.
    - Required timeline.
3. Submit your request.

The time required to make a requested model available in AI Agent Studio depends on the complexity of the model integration. In most cases, requests are completed within two to four weeks. Each request is reviewed, and an estimated availability timeline is provided as part of the service request process. The model will appear in the Models list, where you can add LLMs.

Add Email Tool

Using the Email tool, you can securely enable agents to send, receive, and manage emails within automated agent teams, without building additional email integrations or custom communication logic.

1. Go to **AI Agent Studio**.
2. From the Tools tab, select **Add**.
3. From the **Tool Type** list, select **Email**.
4. Enter the tool name, description, and select the appropriate family and product. The tool code will be automatically populated based on the tool name.
5. (Optional) Select **Require Human Approval** to ensure that a person reviews and approves any action before the tool runs.
    a. Message: Enter instructions for the reviewer.
    b. Actions: Select the required action to be done by the reviewer.
6. In the **Delivery options** section, select **Add** and configure email notification details.
7. From the **Delivery Option** list, select **Email Notification**.

8. Enter the recipient information and email subject.
9. Create the email template as needed. You can also include tokens in your template.
    a. From the **Tokens** list, select a token.
    b. Select **Insert Token** to add the token to the template.
10. Select **Enable Attachments** to include attachments in the email.
11. Save and create the email tool.

Your email tool now appears in the Tools tab, and you can select it when you create Supervisor type agents.

Add Email Accounts

To use an email trigger or a human approver in your agent teams of type Workflow, you must register an email account with AI Agent Studio.

1. Go to **AI Agent Studio** and open the Credentials tab.
2. In the **Email Accounts** subtab, select **Add Account**.
3. Add either a **Microsoft** or **Google** email account.
4. Select **Inbound** or **Approver** to designate the purpose of the email account.
    - Inbound: Receives emails used to initiate a workflow
    - Approver: Receives emails to either approve or reject requests

    These email account designations are configured in the same way and they differ only in where they appear in the UI. You can't use the same email account for both the inbound and approver accounts.
5. Enter the account information.

Field	Description
Email Folder	Folder name in the email account that's used for monitoring requests
Polling Interval	Interval at which AI Agent Studio checks the folder in the email account

Microsoft and Google accounts require different authorization details to manage the account. You can obtain the required information from the vendor's cloud console.

Account	Authorization Fields
Microsoft	- Tenant ID - Client ID - Client Secret
Google	- Service Account Email - Private Key

Create Documents to Add to Document Tool

You can upload files to your document tools for agents to query from. Each document is a logical collection of attachments (files). You can also upload multimodal files.

These file types are supported for attachments:

- PDF-TAGGED / SCANNED
- HTML
- JSON
- MARKDOWN
- XML
- TXT
- DOCX
- DOCX-MULTIMODAL
- XLSX
- CSV
- PPTX
- JPG
- PNG
- ZIP

The content and structure of a document are key aspects to fetch the correct answer for your query. Here are some best practices for attachments:

- Make sure that a single document doesn't contain more than five images or scanned pages. This applies to the following types of files:
  - Fully scanned PDFs
  - Image-only documents
  - Multimodal files combining text and images.

  If your content includes more than five images, split it into multiple smaller documents before uploading.
- Avoid heavy files. To ensure reliable processing and avoid latency or timeouts, make sure that your attachments follow these standards:
  - Avoid documents larger than 25 MB
  - Use image compression where possible
  - Prefer optimized PDFs over high-resolution raw scans
  - Remove unnecessary pages, blank pages, or duplicated content before uploading

Document Structure

It’s highly recommended to structure your document using headings, subheadings, and so on.

Headings must clearly describe the meaning of the content in the subsequent section. Each section (under a heading) must contain semantically related concepts. If you find that a section contains a mix of concepts, it's advisable to create sections. Also, the various heading levels must be clearly distinguishable. For example, use font size 12 for Heading 1, font size 10 for Heading 2, font size 8 for Heading 3, and so on.

A well-constructed document structure enables the agent to split the document into semantically meaningful chunks. This improves the quality of the generated answers.

Here's a comparison of two approaches to structuring a document:

Approach 1	**Approach 2**
Scope of Plan Coverage This plan covers Employees of ACME. An "Employee" shall mean a common law Employee of ACME, Inc. (ACME), ACME International Corporation and ACME Software Technology GmbH. Independent contractors and/or "leased workers" engaged by a staff leasing company aren't Employees for purposes of the Plan and therefore aren't eligible to participate in the Plan. Full and part time employees are covered as follows – You might participate in one of the medical, dental, vision, life/accidental death and dismemberment (AD&D), long term disability (LTD), and health care and dependent care flexible spending accounts (FSA) benefits offered under the Plan when you meet one of the eligibility requirements listed below. • Regular Full-Time Employee on the ACME U.S. payroll scheduled to work 30 or more hours per week. • Regular Part-Time Employee on the ACME U.S. payroll scheduled to work 20-29 hours per week. • Interns and temporary Employees aren't eligible to participate in the Plan as they're not regular Employees of ACME. INPATRIATES are covered as follows – An Inpatriate (and eligible Dependents) residing in the United States for 90+ days and scheduled to work 30 -- 40 hours/	Scope of Plan Coverage This plan covers Employees of ACME and their dependents. Eligible Employees An "Employee" shall mean a common law Employee of ACME, Inc. (ACME), ACME International Corporation and ACME Software Technology GmbH. Independent contractors and/or "leased workers" engaged by a staff leasing company aren't Employees for purposes of the Plan and therefore aren't eligible to participate in the Plan. FULL & PART TIME EMPLOYEES You might participate in one of the medical, dental, vision, life/accidental death and dismemberment (AD&D), long term disability (LTD), and health care and dependent care flexible spending accounts (FSA) benefits offered under the Plan when you meet one of the eligibility requirements listed below. • Regular Full-Time Employee on the ACME U.S. payroll scheduled to work 30 or more hours per week. • Regular Part-Time Employee on the ACME U.S. payroll scheduled to work 20-29 hours per week. • Interns and temporary Employees aren't eligible to participate in the Plan as they're not regular Employees of ACME. INPATRIATES An Inpatriate (and eligible Dependents) residing in the United States for 90+ days and scheduled to work 30 -- 40 hours/week for an ACME Entity will be enrolled, upon notifying , in the following benefits: • ACME Premium PPO Medical • Dental Plan II • Vision Service Plan (VSP) Option 1 • An Inpatriate (and eligible Dependents) are not eligible for Life/Accidental Death and Dismemberment (AD&D) and Long-Term Disability (LTD) coverage. ELIGIBLE DEPENDENTS Your eligible Dependents may be covered by the medical, dental, vision, and life insurance options under the Plan when Your Dependent meets one of the eligibility requirements listed in this section. NOTE: Your Dependents may not enroll in a particular benefit option offered under the Plan unless You are also enrolled.

Approach 1	Approach 2
week for an ACME Entity will be enrolled, upon notifying , in the following benefits: • ACME Premium PPO Medical • Dental Plan II • Vision Service Plan (VSP) Option 1 • An Inpatriate (and eligible Dependents) are not eligible for Life/Accidental Death and Dismemberment (AD&D) and Long-Term Disability (LTD) coverage. Your eligible dependents may be covered by the medical, dental, vision, and life insurance options under the Plan when Your Dependent meets one of the eligibility requirements listed in this section. NOTE: Your Dependents may not enroll in a particular benefit option offered under the Plan unless You are also enrolled	

Approach 2 is highly recommended. Breaking the document into semantically related content, organized by the various coverage categories, enables the agent to provide more relevant answers to questions about benefits coverage. Also, note that the various heading levels are visually distinguishable.

Add Connector Tool

You can use the Connector tool to connect your AI agents to various data sources such as SharePoint, web pages, and other custom content sources. For more information, see *How do I use connectors in AI Agent Studio?*

4 Use Cases

Use Cases for Agents of Type Supervisor

Single Agent with Multiple Workers

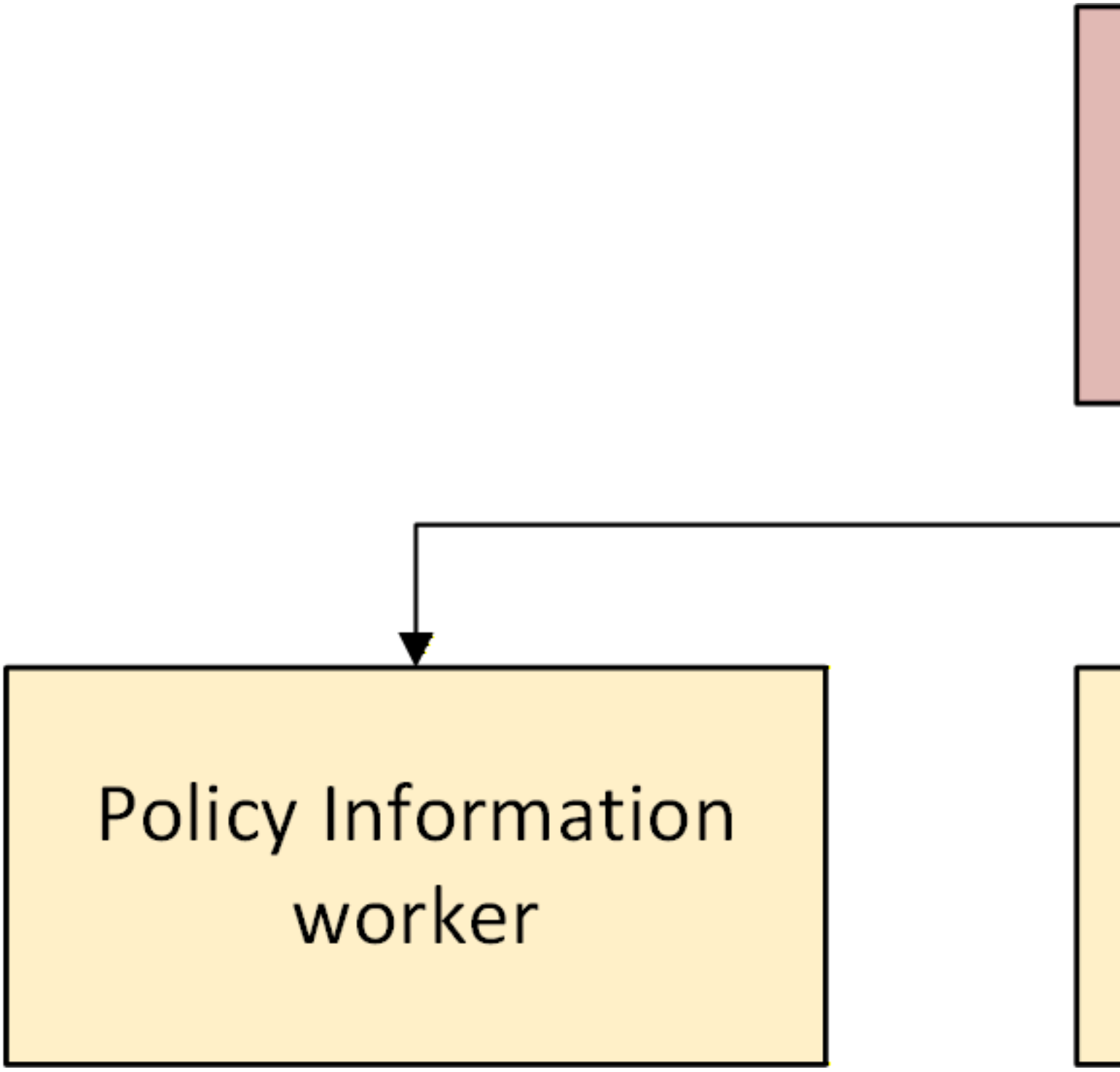
A worker is a specialized agent designed to perform a narrowly defined task or skill, and act as a modular building block for supervisor and workflow agents. Each worker is configured with specific instructions and has access to the tools required to perform that role effectively. Workers don't make high-level decisions or independently orchestrate tasks; instead, they perform well-scoped operations with precision.

Workers are designed to operate as part of a single or multi-agent system, and not as standalone entities. Workers are typically embedded within supervisor or workflow agents that coordinate their usage. This includes deciding when and how to initiate workers, pass appropriate context, and integrate the output from the worker into the overall task operation.

For example, a supervisor agent, based on the problem context, might dynamically select and initiate workers in a non-deterministic manner. A workflow agent might call workers in a predefined, deterministic sequence. In both cases, workers remain task-focused and don't control the broader flow of operations.

In this pattern, a Benefits Advisor acts as a central agent, helping employees understand and make the most of their benefits packages. The agent coordinates several specialized workers to provide a personalized support for medical, dental, and vision coverage.

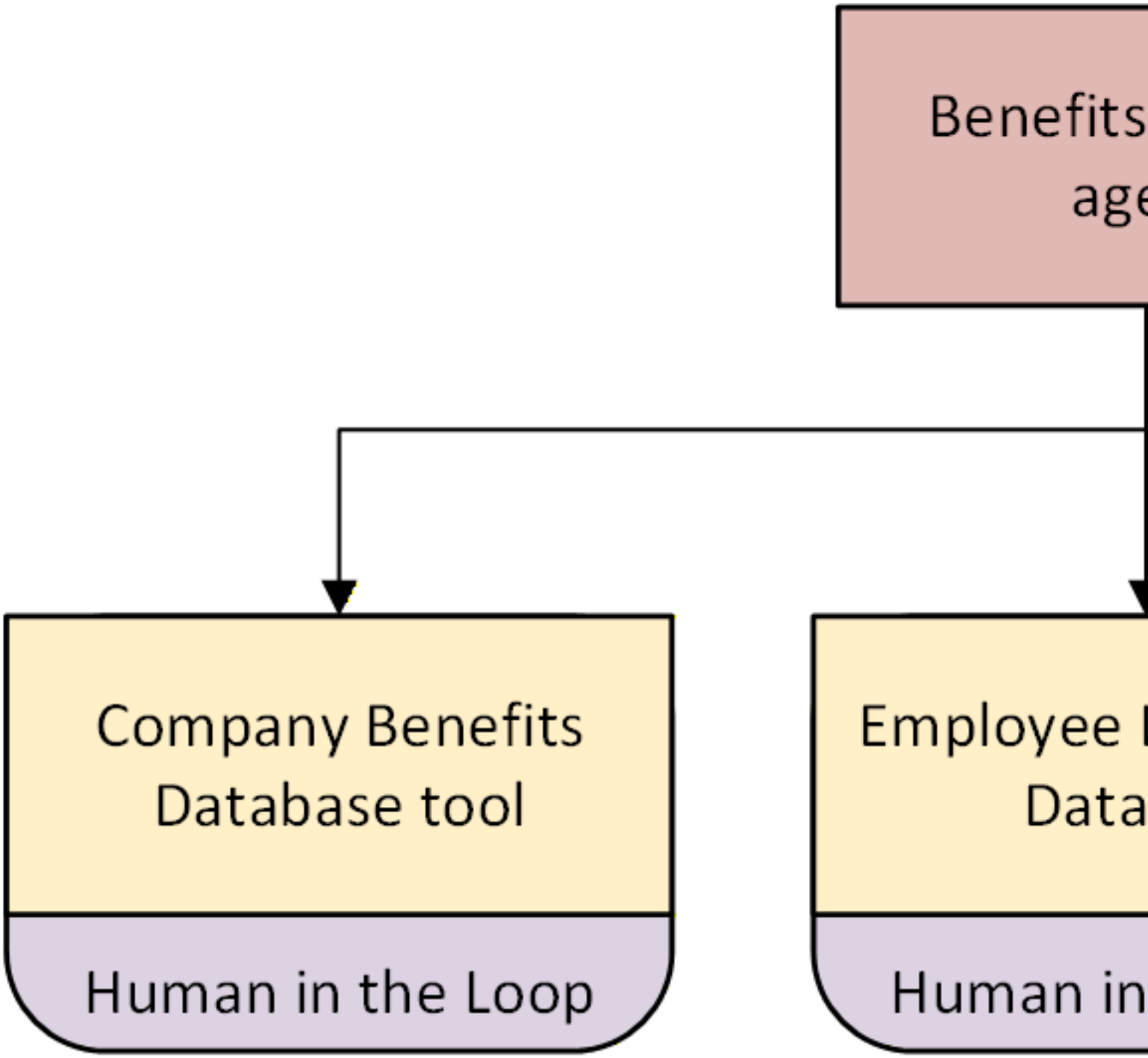
- The **Benefits Advisor** agent serves as the initial contact, helping the employee understand their benefits, and answers any preliminary questions.
- The **Policy Information** worker provides the employee with detailed information on available medical, dental, and vision benefits.
- The **Personal Enrollment** worker uses the employee's enrollment details to offer tailored advice on how they can optimize their benefits package.
- The **Eligibility and Assistance** worker answers queries regarding eligibility, coverage, helping to resolve issues related to benefits.



Single Agent with Multiple Tools and a Human in the Loop

This pattern shows the Benefits Advisor as a central agent that helps employees understand and optimize their benefits. This agent coordinates specialized workers to provide personalized insights on medical, dental, and vision coverage.

- The **Benefits Advisor** agent starts the interaction with a friendly introduction and asks for the employee's needs regarding benefits.
- The **Company Benefits Database** tool retrieves information on the company's benefits policies, eligibility, and plan details.
- The **Employee Enrollment Data** tool uses the employee's personal enrollment information to provide tailored advice, helping them understand and optimize their coverage.
- The **Benefits Optimizer** tool analyzes the employee's selections and offers personalized suggestions for improvement.
- The *Human-in-the-Loop* flag determines whether a user must provide explicit approval before the agent retrieves sensitive data, such as salary or personal information.



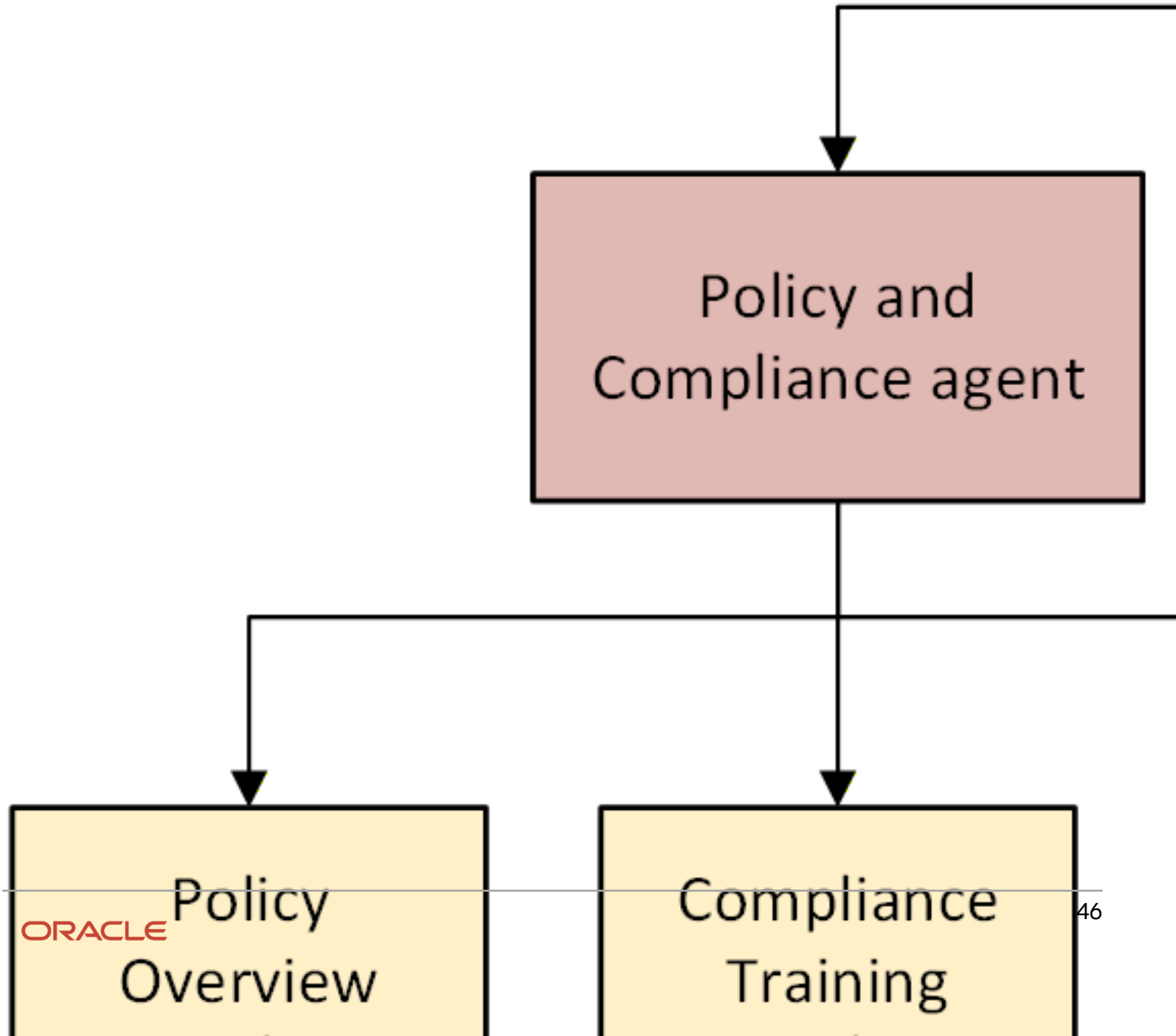
Multiple Agents With Multiple Tools and Delegations

In this pattern, the Central Onboarding Assistant agent acts as a supervisor, guiding new employees through their on-boarding process. The supervisor agent delegates tasks to specialized agents, each focused on a specific aspect of on-boarding, supported by dedicated workers.

Agent	Description
Policy and Compliance Agent	This agent helps employees understand and comply with company policies, and is supported by these workers. • **Policy Overview worker:** Shares key policies like leave, dress code, and workplace behavior. • **Compliance Training worker:** Provides access to mandatory training. For example, safety and social media policies. • **Document Management worker:** Ensures the employee completes and submits all required forms, such as tax forms and nondisclosure agreements.
Cultural Integration Agent	This agent helps employees adapt to the company's culture and connect with teams, and is supported by these workers. • **Company Values worker:** Explains the vision, mission, and goals of the organization. • **Community Introduction worker:** Connects employees with internal groups, events, and networking opportunities. • **Workplace Norms worker:** Provides guidance about unwritten rules, such as communication etiquette and meeting culture.
Tools and Resources Agent	This agent ensures that employees have access to tools, systems, and knowledge required for their job, and is supported by these workers. • **Account Setup worker:** Helps employees set up email, communication tools, and HR platforms. • **Resource Guide worker:** Provides access to handbooks, FAQs, and internal documentation. • **Technical Support worker:** Troubleshoots issues with tools or platforms.

Process

- The Central Onboarding Assistant agent acts as the entry point that gathers basic information about the employee's role and specific needs, and delegates tasks to specialized agents based on the query.
- The Policy and Compliance agent handles queries, or tasks, related to company rules and legal documentation, such as, "What's the process for requesting leave?"
- The Cultural Integration agent handles queries that focus on helping employees feel part of the team and aligned with the company values, such as, "How do I join company events?"
- The Tools and Resources agent handles queries about gaining access to systems and essential knowledge, such as, "How do I log into the time-tracking tool?"


Policy and Compliance agent
Policy Overview
Compliance Training

Product-Specific Use Case Examples

Oracle Fusion Cloud CX

Agent Type	Key Characteristics	When to Use	Example
Customer service agent	• Interactive • Rule-based • Escalation-friendly • Customer-focused	• Repetitive, high-volume queries • Customer support • Query resolution	Chatbot for customer support agent answers FAQs and resolves issues.
Learning agent	• Adaptive • Feedback-driven • Continuously improves • Explainable	• Adaptive systems • Chatbot improvement • Demand forecasting refinement	Chatbot improvement agent learns from customer feedback to improve responses.

Oracle Fusion Cloud HCM

Agent Type	Key Characteristics	When to Use	Example
Copilots	• Collaborative • Proactive • Decision-making • Domain-specific	• Complex tasks requiring expertise • Draft documents • Input validation	Hiring copilot agent helps to build job descriptions and suggests interview questions.
Document referencing agent	• Knowledge-intensive • Accurate • Up-to-date • Query-driven	• Knowledge-intensive tasks • Answers to policy or FAQ questions • Compliance checks	Benefits agent answers employee questions about benefits.
Evaluating agent	• Assessment-focused • Criteria-based • Actionable insights • Feedback-driven	• Performance evaluation • Candidate ranking • Feedback analysis	Candidate evaluation agent ranks candidates based on interview performance.
Scheduling agent	• Task-oriented • Reactive • Limited decision-making • User-centric	• Routine task scheduling • Calendar management • Appointment booking	Interview scheduling agent coordinates availability between candidates and hiring managers.
Task automation agent	• Repetitive task handler • Efficient • Error-resistant	• Repetitive, high-volume tasks • Payroll processing	Payroll processing agent automates time sheet collection and payslip generation.

Agent Type	Key Characteristics	When to Use	Example
	• Trigger-based	• Form generation	

Oracle Fusion Cloud SCM

Agent Type	Key Characteristics	When to Use	Example
Collaborative workflow agent	• Multi-stakeholder • Real-time coordination • Role-based task allocation • Communication-driven	• Complex processes requiring collaboration • Supplier collaboration • Crisis management	Supplier collaboration agent coordinates with procurement, supplier, and warehouse bots to work together.
Decision-making agent	• Data-driven • Recommendation-focused • Transparent • Adaptive	• Data-driven tasks • Supplier evaluation • Candidate shortlisting	Supplier evaluation agent evaluates suppliers based on cost and quality.
Goal-seeking agent	• Optimization-focused • Real-time adjustments • Objective-driven • Explainability	• Resource allocation • Route planning • Workforce scheduling	Route optimization agent plans optimal delivery routes.
Monitoring agent	• Proactive • Alert-driven • Data-intensive • Response-oriented	• Proactive management • Inventory monitoring • Social media tracking	Inventory monitoring agent tracks in-stock levels and alerts for replenishment.

5 Create AI Agents

Create AI Agents Using Preconfigured Templates

You can create and deploy AI agents using preconfigured templates available in these two subtabs:

- AI Agents: Access the preconfigured templates built and provided by Oracle. When using a preconfigured agent team template, the artifacts such as agents, tools, and topics aren't directly editable. To change these artifacts, you create a copy of the artifact and add the copy to the agent team.
- Marketplace: Access the preconfigured templates built and provided by Oracle partners. View template details using the ⓘ icon. To create a copy of any template, click **Create**.

To use preconfigured agent team templates from the AI Agents subtab:

1. Go to **AI Agent Studio**.
2. Select **Copy Template** from the required agent team, enter a suffix for the new agent team and then select **Continue**.

   > **Tip:** You can choose the **Use Template** option to provide all details for the new agent team before creating it.

3. Your copy of the template is created. If needed, select **Agent Team Settings** ⚙ to edit the agent team details, and select **Update**.

   > **Note:** You can add expressions to fields using **Insert Expression** ⊕ ⌄. For more information, see *Expressions in AI Agent Studio*.

   Details Tab

Field	Description
Family	Select the family to which this agent team belongs.
Product	Select the product within the family to which this agent team belongs.
Maximum Interactions	Indicate the number of times an agent within this agent team can interact with the topics and tools assigned to it.

   LLM Tab

Field	Description
Provider	Choose to use the default model or select a model. When using a custom model, specify the model properties.

   Security Tab

Field	Description
Add	Select the roles which will have access to this agent team.

Questions Tab

Field	Description
Starter Questions	Enter initial questions for the agent team.
Follow-up Questions	Enable this to indicate that the agent team can ask follow-up questions based on the user's conversation history.
Prompt	Enter the prompt to be used for the follow-up questions. For example: **using {chatHistory} generate 3 follow up questions in json format. JSON Schema format mentioned below. Remove the ``json markdown from the output. Here is the JSON Schema format the output should adhere to:** **[{"question": "<put first question generated here>"}, {"question": "<put second question generated here>"}, {"question": "<put third question generated here>"}]**
Insert Expression	Add additional variables to the prompt. For example, to add the current system date to your prompt, select the **Current Date Time** option.

Chat Experience Tab

You can enable users to upload attachments while interacting with agents. Users can upload up to five files, with a total combined size limit of 50 MB. These are the supported file types users can upload to the chat: PDF-Tagged / Scanned, HTML, JSON, MARKDOWN, XML, TXT, DOCX, DOCX-Multimodal, XLSX, CSV, PPTX, JPG, PNG, and ZIP.

Field	Description
Enable file upload	Select this option to allow users to upload files from local storage.

Field	Description
Enable third-party upload	Select this option to allow users to upload files from connected cloud storage accounts. To enable this functionality, add credentials for your provider by navigating to the Credentials tab in AI Agent Studio and completing the Chat Experience setup.

> **Note:** To make sure your agent can process attachments, add the common tool of **Runtime File Processor** type, **MultiFileProcessor**.

For more information, see *What file types, limits, and processing capabilities are supported in AI Agent Studio?*

Output Tab

Define the overall structure of the agent's output using JSON schema, to specify the exact output.

Field	Description
Specification Mode	Select this mode to directly modify the JSON schema for the output.
Simple Mode	Select this mode to define the output values and types. The corresponding JSON schema will be generated automatically and displayed in the specification mode for any further changes.

4. Continue to edit and add details for the agent team. If you want to edit the artifacts in the agent team including agents, tools, and topics, you can create a copy of the artifact and edit it. After editing, remove the original artifact from the agent team, and then add the copy you created. See *Can I edit a preconfigured agent team?*

   > **Note:** The agents and other artifacts included within an agent template are optimized to provide the best usage of the agent team. We recommend not to change the basic functionality of the artifacts because that might impact your agent team's performance.

5. If needed, use ▷ to test the agent team. For any required fine-tuning, you can edit the agent team using ⚙.
6. Publish your agent team.

Users can view the published agents from the AI Agents page. To access this page, go to **Me** > **Quick Actions** > **Show More** > **AI Agent Studio** > **AI Agents**. If you're using **Home with Ask Oracle** as your home page layout, search for the **AI Agents** page.

> **Note:** Make sure that your users have access to interact with the AI agents. For information, see *How can I give users access to AI agents?*.

Create Custom AI Agents of Type Supervisor

The supervisor agent orchestrates and plans tasks for your agent team in Supervisor flows. It directs other agents on how to interact and generate responses. You can configure supervisor agents as a single agent or within a multiple agent pattern. To see some of the common supervisor agent patterns, go to *Use Cases for Agents of Type Supervisor*.

Tip: For recommendations about improving your agent's efficiency, see *How do I make agents respond faster?*.

1. Go to **Navigator** > **Tools** > **AI Agent Studio**.
2. From the Tools tab, add the required tools.

   For example, to create an HR benefits administrator agent that can answer questions related to medical, vision, dental, retirement, and stock plans, these are some of the tools needed.
   - Document tools, with the organization's health and financial benefits documents.

     Note: Before you add any document tools to an agent, make sure you've set the status of the document tool as **Ready to Publish**.
   - Business object tools, to fetch the employee enrollments data.
   - Calculator tool, to check balance amounts and percentages.
3. From the Topics tab, add the required topics. Make sure you include instructions about these key areas.
   - What the topic is about
   - Tools to use with the topic
   - Examples of possible questions
   - Any guidelines and guardrails
4. From the Agents tab, add an agent.
   - In **Maximum Interactions** field, specify the number of times the agent can interact with the topics and tools assigned to it.
   - Describe the persona and role of the agent, including the tone to use.
   - Add the prompt for the agent, and select any variables from **Insert Expression**.
5. Assign the required tools and topics to the agent, and create it.
6. From the **Agent Teams** tab, add an agent team and provide details for the new agent team.

   Note: You can add expressions to fields using **Insert Expression**  . For more information, see *Expressions in AI Agent Studio*.

   - **Details Tab**

Field	Description
Family	Select the family to which this agent team belongs.
Product	Select the product within the family to which this agent team belongs.
Type	Select **Supervisor** to create an agent team where one agent acts as supervisor and manages other agents and artifacts.

Field	Description
Maximum Interactions	Indicate the number of times an agent within this agent team can interact with the topics and tools assigned to it.

- **LLM Tab**

Field	Description
Provider	Choose to use the default model or select a model. When using a custom model, specify the model properties.

- **Security Tab**

Field	Description
Add	Select the roles which will have access to this agent team.

- **Questions Tab**

Field	Description
Starter Questions	Enter initial questions for the agent team.
Follow-up Questions	Enable this to indicate that the agent team can ask follow-up questions based on the user's conversation history.
Prompt	Enter the prompt to be used for the follow-up questions. For example: using {chatHistory} generate 3 follow up questions in json format. JSON Schema format mentioned below. Remove the ```json markdown from the output. Here's the JSON Schema format the output should adhere to: [{"question": "<put first question generated here>"}, {"question": "<put second question generated here>"}, {"question": "<put third question generated here>"}]
Insert Expression	Add additional variables to the prompt. For example, to add the current system date to your prompt, select the Current Date Time option.

- **Chat Experience Tab**

  You can enable users to upload attachments while interacting with agents. Users can upload up to five files, with a total combined size limit of 50 MB. These are the supported file types users can upload to the

chat: PDF-Tagged / Scanned, HTML, JSON, MARKDOWN, XML, TXT, DOCX, DOCX-Multimodal, XLSX, CSV, PPTX, JPG, PNG, and ZIP.

Field	Description
Enable file upload	Select this option to allow users to upload files from local storage.
Enable third-party upload	Select this option to allow users to upload files from connected cloud storage accounts. To enable this functionality, add credentials for your provider by navigating to the Credentials tab in AI Agent Studio and completing the Chat Experience setup.

> **Note:** To make sure your agent can process attachments, add the common tool of **Runtime File Processor** type, **MultiFileProcessor**.

For more information, see *What file types, limits, and processing capabilities are supported in AI Agent Studio?*

- **Output Tab**

Define the overall structure of the agent's output using JSON schema, to specify the exact output.

Field	Description
Specification Mode	Select this mode to directly modify the JSON schema for the output.
Simple Mode	Select this mode to define the output values and types. The corresponding JSON schema will be generated automatically and displayed in the specification mode for any further changes.

7. Select **Create**.
8. Select and click **New Supervisor Agent**.
9. Enter details for the supervisor agent and create it. This supervisor agent is specific only to the agent team in which it's created, and can't be reused.
10. Select and add any existing agents as worker agents, or create worker agents.

    When you create worker agents from within the agent team, those agents are available only for this agent team.
11. Add any needed artifacts such as tools and topics.
12. If needed, use to test the agent team. For any required fine-tuning, you can edit the agent team using .
13. Publish your agent team.

Users can view the published agents from the AI Agents page. To access this page, go to **Me** > **Quick Actions** > **Show More** > **AI Agent Studio** > **AI Agents**. If you're using **Home with Ask Oracle** as your home page layout, search for the **AI Agents** page.

> **Note:** Make sure that your users have access to interact with the AI agents. For information, see *How can I give users access to AI agents?*.

Create Custom AI Agents of Type Workflow

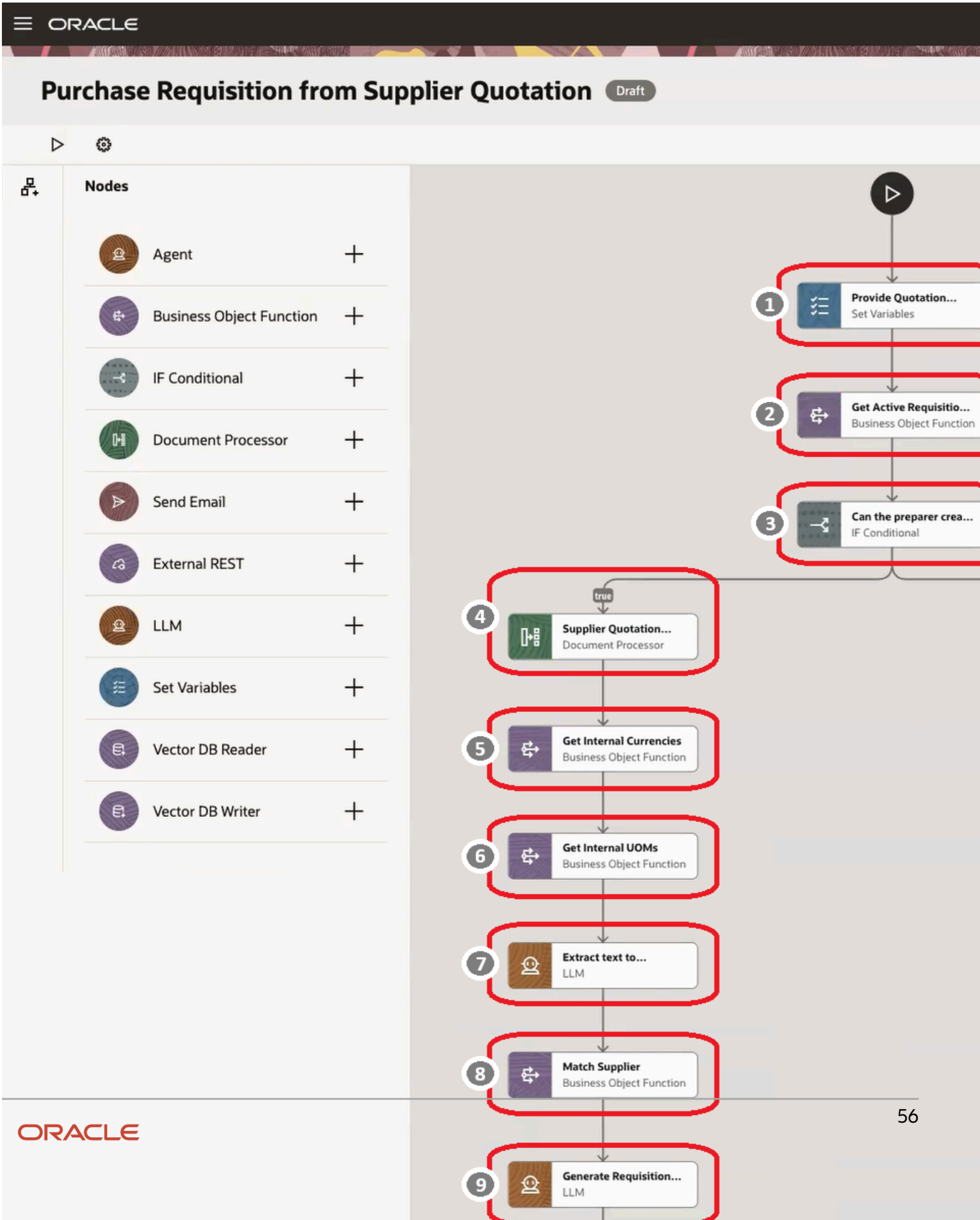
Agent teams of type Workflow consist of a sequence of nodes used for deterministic, rule-based orchestration of tasks in which every step is preconfigured. Workflows are ideal for scenarios where compliance, repeatability, and governance are essential. The workflow tasks are represented as a connected sequence of nodes. Each node within the workflow performs a defined function, such as extracting data, calling a business object function, running an LLM, or sending an email. The output from each node is seamlessly passed to the following step, ensuring a controlled and efficient process flow.

When designing your workflow, keep these process breakdown steps in mind:

- Model the manual workflow. Start from a high-level process and break it down into smaller, automatable steps.
- Identify task dependencies and group nodes by type. For example, distinguish between back-end logic tasks and those driven by LLMs. Doing so helps to streamline operation and minimize unnecessary LLM usage.
- For each task, define the node type, required input values, expected outputs, and any transformation logic needed.
- For recommendations about improving your agent's efficiency, see *How do I make agents respond faster?*

Let’s look at an example to illustrate how a workflow operates in practice. Suppose an employee needs to buy a product or service and contacts a supplier to request a quotation. The supplier responds with a quotation, which might vary in format and content, but typically includes key information such as the seller’s details, a list of items, and the total quoted price. The quotation is processed internally, and a requisition order is created.

Here's an example workflow for this scenario, created in AI Agent Studio.


ORACLE
Purchase Requisition from Supplier Quotation
Draft
Nodes
Agent
Business Object Function
IF Conditional
Document Processor
Send Email
External REST
LLM
Set Variables
Vector DB Reader
Vector DB Writer
1
Provide Quotation...
Set Variables
2
Get Active Requisitio...
Business Object Function
3
Can the preparer crea...
IF Conditional
true
4
Supplier Quotation...
Document Processor
5
Get Internal Currencies
Business Object Function
6
Get Internal UOMs
Business Object Function
7
Extract text to...
LLM
8
Match Supplier
Business Object Function
9
Generate Requisition...
LLM

Callout Number	Node Name	Node Type	Description
1	Provide Quotation	Set variables	Sets the values for the variables.
2	Get Active Requisition	Business Object Function	Retrieves the default preferences of the requester.
3	Can the preparer create a requisition?	IF conditional	Validates whether the user has permission to create a requisition. • If true, the workflow continues to the following nodes. • If false, the workflow indicates that a requisition wasn't created, sends a failur
4	Supplier Quotation	Document processor	Downloads the document from the WebCenter Content server and parses the cont
5	Get Internal Currencies	Business Object Function	Pulls the valid internal currency value.
6	Get Internal UOM	Business Object Function	Pulls the valid internal units of measure value.
7	Extract text to LLM	LLM	Uses LLM to convert the unstructured text from upstream nodes to a structured JS
8	Match supplier	Business Object Function	Checks and validates the supplier information extracted earlier.
9	Generate requisition payload	LLM	Maps all gathered data from the document and database to generate a valid post r requisition payload.
10	Create purchase requisition	Business Object Function	Calls the API to create the draft requisition. After this call, the requisition is saved to
11	Is requisition created?	IF conditional	Checks if the requisition is successfully created. • If true, the workflow proceeds to the success confirmation node. • If false, the workflow proceeds to the send failure email node.
12	Success confirmation	LLM	Displays a message indicating the successful creation of the requisition.
13	Send failure email	Send email	Sends an email to the user with failure details.

Here are the steps to create your own workflow.

1. Go to **AI Agent Studio**.

2. From the **Agent Teams** tab, add an agent team and provide details for the new agent team.

> **Note:** You can add expressions to fields using **Insert Expression** . For more information, see *Expressions in AI Agent Studio*.

Details Tab

Field	Description
Family	Select the family to which this agent team belongs.
Product	Select the product within the family to which this agent team belongs.

Field	Description
Type	Select **Workflow**.

LLM Tab

Field	Description
Provider	Choose to use the default model or select a model. When using a custom model, specify the model properties.

Security Tab

Field	Description
Add	Select the roles which will have access to this agent team.

Triggers Tab

Create triggers to begin the workflow. Triggers define the conditions or events that start the workflow and identifies when and why it begins.

Field	Description
Type	Select the data type of the trigger.

For more information about setting up triggers, see *Triggers*.

Variables Tab

Define workflow-level variables, making them accessible to all nodes within the workflow. Variables are ideal for storing IDs, constants, or any values you need to share and use across different parts of the workflow.

Field	Description
Type	Select the data type of the variable.

Chat Experience Tab

You can enable users to upload attachments while interacting with agents. Users can upload up to five files, with a total combined size limit of 50 MB.

> **Note:** To make sure your agent can process attachments, add a worker agent with the tool of **Runtime File Processor** type. For more information, see *Create Custom AI Agents of Type Supervisor*.

These are the supported file types users can upload to the chat: PDF-Tagged / Scanned, HTML, JSON, MARKDOWN, XML, TXT, DOCX, DOCX-Multimodal, XLSX, CSV, PPTX, JPG, PNG, and ZIP.

Field	Description
Enable file upload	Select this option to allow users to upload files from local storage.
Enable third-party upload	Select this option to allow users to upload files from connected cloud storage accounts. To enable this functionality, add credentials for your provider by navigating to the Credentials tab in AI Agent Studio and completing the Chat Experience setup.

For more information, see *What file types, limits, and processing capabilities are supported in AI Agent Studio?*

Output Tab

Define the overall structure of the workflow's output using JSON schema, to specify exactly what you'll receive when the workflow is completed. If no schema is specified, the output type defaults to that of the individual node. When a schema is provided, the system validates the outputs against the schema and enables auto-complete for those fields.

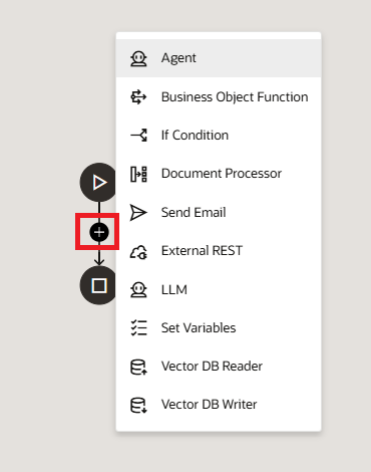
Field	Description
Specification Mode	Select this mode to directly change the JSON schema for the output.
Simple Mode	Use this mode to define output values and types. The corresponding JSON schema will be generated automatically and displayed in the specification mode for you to edit.

Error Handling Tab

Specify where to send an email when a workflow instance runs into a permanent error. You can also use context expressions in these fields for more clarity.

3. Click **Create**.

4. Add nodes to the workflow by hovering over the line between the start and end of the workflow. Click ⊕ to choose the type of node you want to add.



Tips

- Connect nodes in sequence to control how input values are received, processed, and forwarded throughout the workflow.
- You can insert new nodes before, after, or between existing nodes as needed.
- For more information about Vector nodes, see *Vector Write and Read Nodes* .
- Once created, you can't rearrange the nodes. To change a node's position, delete it and create it again in the desired spot.

5. If needed, use ▷ to test the agent team. For any required fine-tuning, you can edit the agent team using ⚙ .
6. Publish the agent team.

Users can view the published agents from the AI Agents page. To access this page, go to **Me** > **Quick Actions** > **Show More** > **AI Agent Studio** > **AI Agents**. If you're using **Home with Ask Oracle** as your home page layout, search for the **AI Agents** page.

> **Note:** Make sure that your users have access to interact with the AI agents. For information, see *How can I give users access to AI agents?*.

Vector Write and Read Nodes

To automate business processes effectively, your AI workflows require access to high-quality, well-structured, and reusable knowledge. You can use Vector Write and Read nodes in AI Agent Studio to store knowledge as embeddings and retrieve it intelligently with semantic search and metadata filters. By leveraging these nodes, you can design workflows that capture, manage, and reuse critical business knowledge across processes, thereby improving accuracy, scalability, and reliability in an enterprise environment.

Here's what you need to know to effectively use Vector Write and Read nodes in your workflow agents, along with some best practices for organizing and managing enterprise knowledge for the best possible retrievals.

Operational Limits and Defaults

To ensure efficient vector operations, keep node entries within specified sizing and retrieval limits.

Parameter	Recommendation
Document size per entry	50KB
Number of retrievals	15 retrievals by default, consisting of 10 semantic and five text
Maximum results per read	Up to five

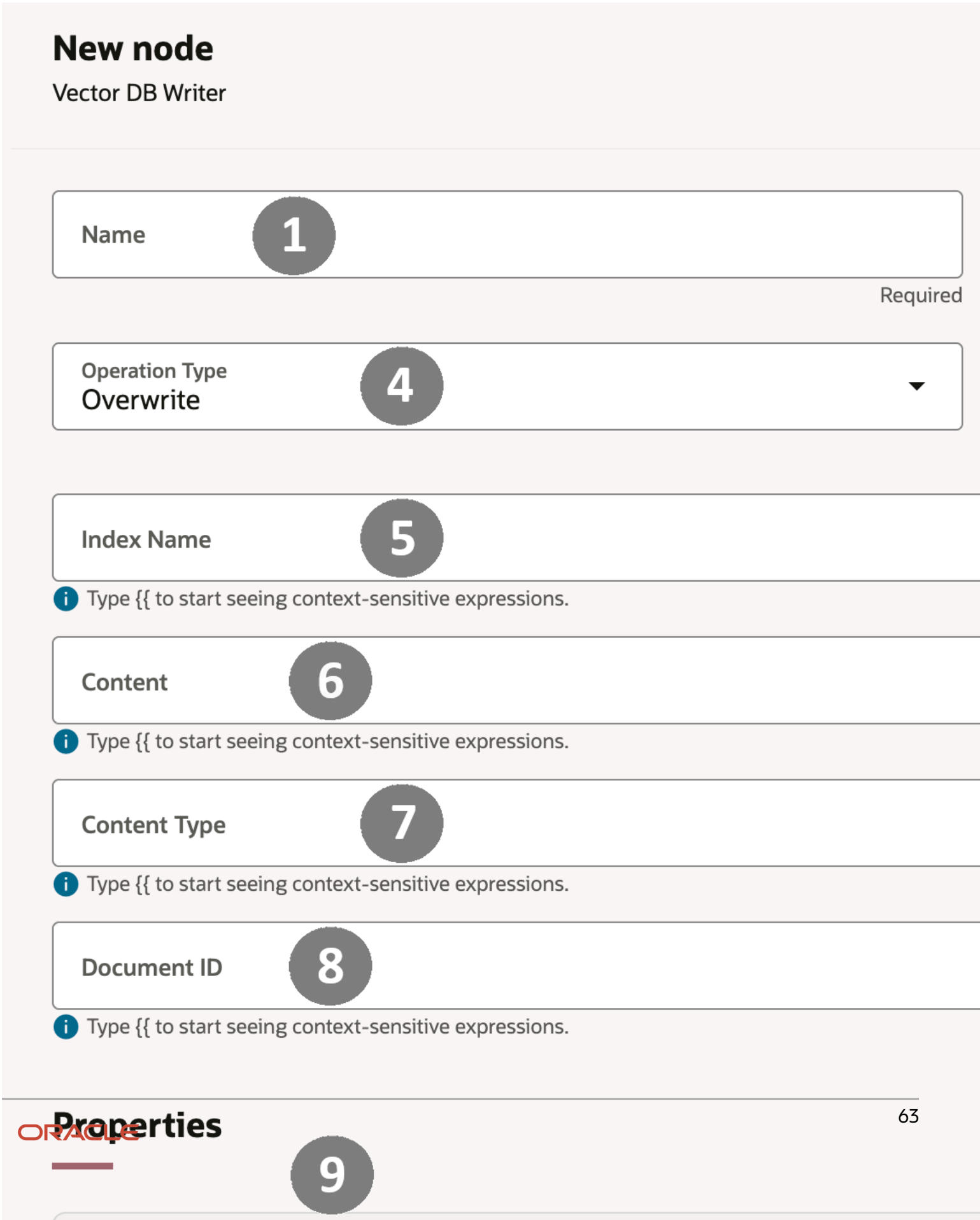
Here are some sizing tips:

- Keep the number of retrievals to a minimal amount of relevant items.
- Don't store more than 500 objects per workflow and use strong metadata filters to improve precision.
- Apply a semantic match score threshold before using results. If the scores are low or have no value, implement a fallback path to a deterministic look-up or a human-in-the-loop intervention.
- Avoid noisy, raw, or sensitive data.
- Keep entries fresh with UPSERT or OVERWRITE for updates and DELETE for stale records.

Vector Write Node

Vector write nodes store high-value knowledge as embeddings for future semantic retrieval. These nodes are also referred to as vector store nodes.

When you create a node, you can configure all values as expressions that resolve dynamically at runtime. Here's an example of the node creation window. Each numbered field is described in detail in the following table.


New node
Vector DB Writer
Name
1
Required
Operation Type
Overwrite
4
Index Name
5
Type {{ to start seeing context-sensitive expressions.
Content
6
Type {{ to start seeing context-sensitive expressions.
Content Type
7
Type {{ to start seeing context-sensitive expressions.
Document ID
8
Type {{ to start seeing context-sensitive expressions.
Properties
9

Callout Number	Field	Description	Example
1	Name	Design-time name for the node	Use descriptive names, like `WriteResolutionSummaryToVectorDB`, so the workflow is easy to understand.
2	Code	Programmatic identifier in the workflow schema	This field is auto-generated but can be changed to a user-defined one, such as `write_resolution_vector`. Use lowercase and underscores.
3	Error Handler	Fallback path on failure	Select an error branch or a dedicated handler node to manage failures cleanly.
4	Operation Type	Method of how the document is written to the index	Use `INSERT` for new entries, `OVERWRITE` to replace content, `UPSERT` to safely update or create, or `DELETE` to remove entries.
5	Index Name	Name of the vector index to write to	Choose an existing index or specify a new one, such as `support_ticket_summaries` or `product_docs_index`.
6	Content	Textual data to embed	Summarize this in a clean and structured manner. Avoid raw logs. Instead, use LLM-generated summaries, extracted facts, or curated knowledge.
7	Content Type	Type of content being indexed or embedded	Typical values include `json` or `text`.
8	Document ID	Unique identifier for this record	Use stable identifiers, such as `ticket_1123` or `customer_450_profile`.
9	Properties	Other optional metadata key-values	`{objectId:"a12345", region:"NA", severity:"High"}`

Best Practices for Building Vector Write Nodes

A vector store works best when you write clear, meaningful, metadata-rich knowledge while avoiding noise, duplication, and topic drift.

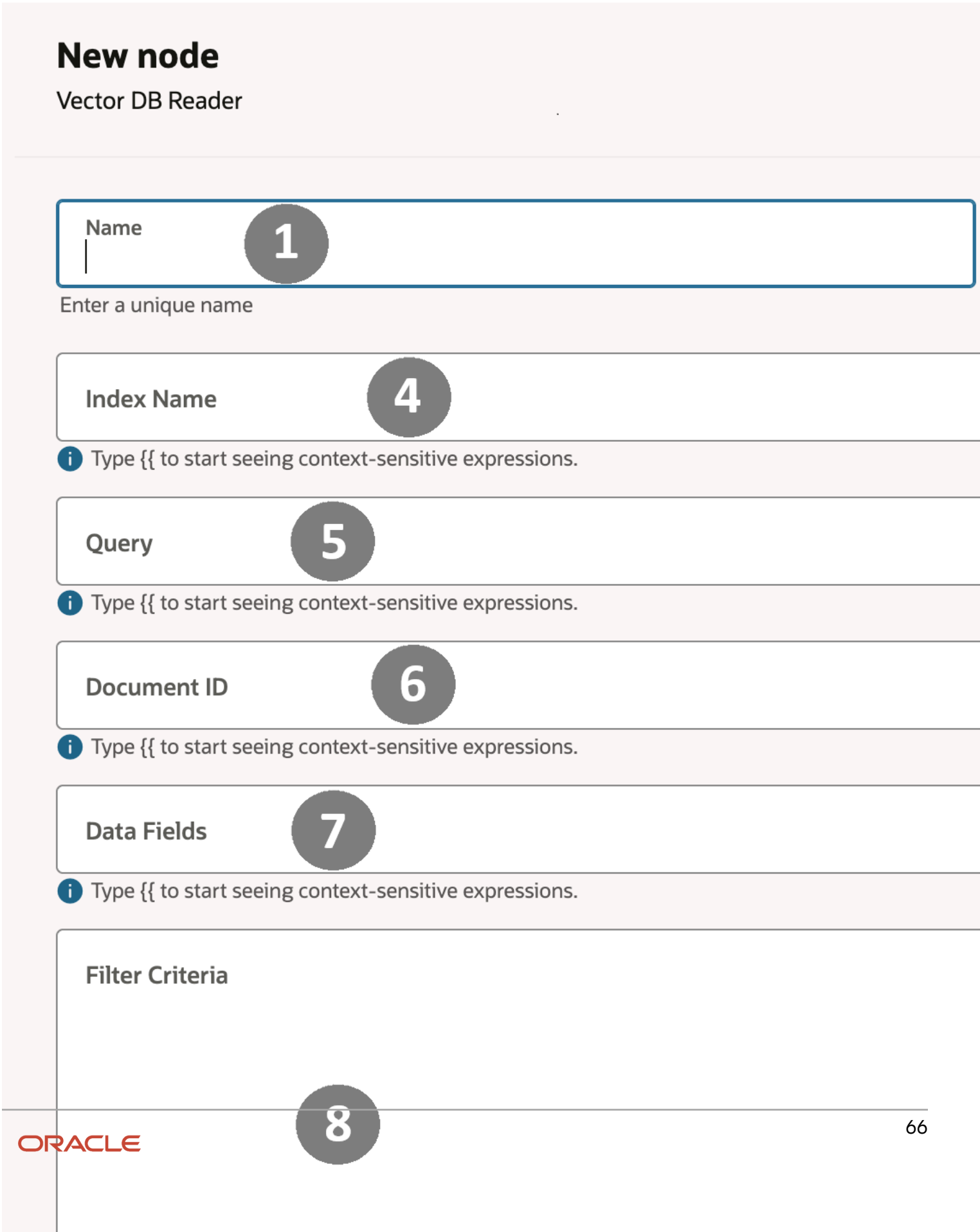
Best Practice	Description
Store only public, reusable knowledge	Exclude sensitive, permission-controlled, or private information, for example, personally identifiable information (PII), credentials, confidential documents, or any information requiring permission checks.
Focus on durable, high-value content	Store information that will have long-term value for future workflows, such as case resolutions, structured summaries, or validated insights. Avoid writing noisy, one-off, or ephemeral content, like raw chat logs or temporary instructions.
Clean and normalize before writing	Always standardize content, summarize documents, deduplicate overlaps, remove irrelevant details, and attach key metadata.

Best Practice	Description
Keep data up to date	Vector writes must reflect the latest truth. Stale entries decrease accuracy and lead to incorrect answers. • Set auto-refresh triggers to update vectors whenever business objects change (for example, case closures, policy updates, PO resolutions). • Use OVERWRITE or UPSERT to keep summaries up to date. • Delete outdated or unused entries to prevent stale data from accumulating. • Maintain freshness so workflows rely on correct, up-to-date knowledge.
Follow smart update practices	Update vectors in a controlled and intentional manner. Don't create more entries for each update, or store conflicting or outdated data. • Avoid creating new entries for every update. Refine existing data instead of duplicating. • Use UPSERT to update and improve existing summaries. • Use OVERWRITE for significant changes or new, higher-quality content. • Prune redundant or conflicting entries to keep the vector store clean. • Establish clear versioning strategies for policies, product versions, and configuration changes to prevent fragmentation.
Always include rich metadata	Tag with business object IDs, product numbers, version information, and other searchable attributes. Use a consistent metadata schema across agents and indexes.
Prevent data pollution	Check for existing knowledge to avoid duplicates and inconsistent tags. Regularly remove low-value content.
Exclude business object data	Duplicating this data in vector memory creates noise, redundancy, and version drift.

Vector Read Node

Vector read nodes retrieve the most relevant knowledge using semantic similarity and metadata filters.

Here's an overview of what the node creation window looks like. When creating a node, all values can be configured as expressions that will be resolved at runtime.


New node
Vector DB Reader
Name
1
Enter a unique name
Index Name
4
Type {{ to start seeing context-sensitive expressions.
Query
5
Type {{ to start seeing context-sensitive expressions.
Document ID
6
Type {{ to start seeing context-sensitive expressions.
Data Fields
7
Type {{ to start seeing context-sensitive expressions.
Filter Criteria
8

Callout Number	Field	Description	Example
1	Name	Name of the node	Use clear names like `RetrieveTicketContextFromVectorDB` to keep the workflow readable.
2	Code	Internal programmatic identifier	This field is auto-generated but can be changed to a user-defined one, like `read_ticket_context_vector`.
3	Error Handler	Defines what occurs on failure	Route to an error handler or fallback logic to avoid empty or invalid agent responses.
4	Index Name	Vector index to search	Choose the same index used by the writer, such as `support_ticket_summaries` or `employee_profile_index`.
5	Query	Natural language search query	Avoid vague queries and ask for intent-specific info, such as, “What troubleshooting steps were taken?”
6	Document ID	Unique identifier for this record.	Specify this when you want details tied to a known record, such as `ticket_12345_summary`.
7	Data Fields	Metadata fields to return	Specify an array of strings that will be used in the filter.
8	Filter Criteria	Logical filters applied before ranking	Specify this field to enable high precision and to constrain results from a retrieval, such as `product = Payroll`, `region = US`, or `severity >= High`.
9	Maximum Results	Maximum number of ranked results returned	Specify an integer value for the max number of results to be displayed.

Best Practices for Building Vector Read Nodes

A vector read node works best when you write clear, specific queries that use metadata filters to ensure semantically correct responses, while preventing topic drift and hallucinations.

Best Practice	Description
Write clear, intent-driven queries	Ensure queries are specific and aligned with the exact workflow or agent task. For example, use precise prompts like "What resolved similar issues?" instead of vague ones like "Help me with this."
Use metadata filters for precision	Apply filters (for example, entity type, business object ID, and product number) to target only relevant content.
Validate retrieved results before using them	Check metadata and context before trusting or using results in workflows. This validation prevents incorrect answers, silent workflow failures, and LLM hallucinations grounded in bad evidence.

Best Practice	Description
Set an appropriate `maxResults` value	Set the `maxResults`field to return three to five, or more results. Don't depend on a single query result.
Add graceful fallback logic	Implement backup steps, such as business objects or API lookups, to handle cases where queries return no results.
Avoid broad or irrelevant retrieval	Scope queries to just what's needed. Skip irrelevant or noisy indexes to boost performance and accuracy of retrieved results.

Triggers

Workflow triggers are automated, predefined events or conditions that initiate a workflow when specific criteria are met.

Note: Triggers activate after the workflow is published.

Webhook

A webhook is an automated, event-driven mechanism for sending real-time data from an application over HTTP. You can configure a webhook trigger to call a workflow.

1. Open the workflow, select **Agent Team Settings** , and go to the **Triggers** tab.
2. Select **Add** for webhook trigger, and enter the input name and type.
3. Save and publish the workflow.

Email

You can configure an email trigger to call a workflow when an email is received in a monitored account. Each workflow can monitor only one unique inbound email account. You're not allowed to use the same inbound email account for another workflow.

To allow the workflow to be triggered by incoming emails:

1. Register the inbound email account in **AI Agent Studio**. See *Add Email Accounts*.
2. Open the workflow, select **Agent Team Settings** , and go to the **Triggers** tab.
3. Select **Add** for email trigger and select your registered inbound email account.
4. Save and publish the workflow.

To verify that emails received in the selected inbound email account will now trigger this workflow:

1. Go to the **Monitoring and Evaluation** tab and open the **Activities** subtab.
2. Go to **Inbound Emails** to view the contents of the trigger email messages and verify the messages are initiating the workflow correctly.

Schedule

You can configure a scheduled trigger based on a defined time interval, to call a published workflow without user interaction or API calls. Scheduled triggers are useful when workflows need to run regularly or at fixed intervals.

> **Note:** Make sure you have the FAI Batch Job Manager (*ORA_DR_FAI_BATCH_JOB_MANAGER_DUTY*) permission assigned in the Security Console.

To allow the workflow to be triggered at a scheduled date and time:

1. Open the workflow, select **Agent Team Settings** , and go to the **Triggers** tab.
2. Select **Add** for schedule triggers and choose the type of schedule to create.

Type	Description
Interval	Run the workflow between fixed time periods, for example every 100 minutes, every 10 hours, or every 5 days.
Recurrence	Run once at a specific time or repeatedly on calendar patterns, such as daily, weekly, or monthly. Enter a start time and an optional date and time for the recurrence to end.

3. Save and publish the workflow.

To verify that your scheduled trigger is starting the workflow:

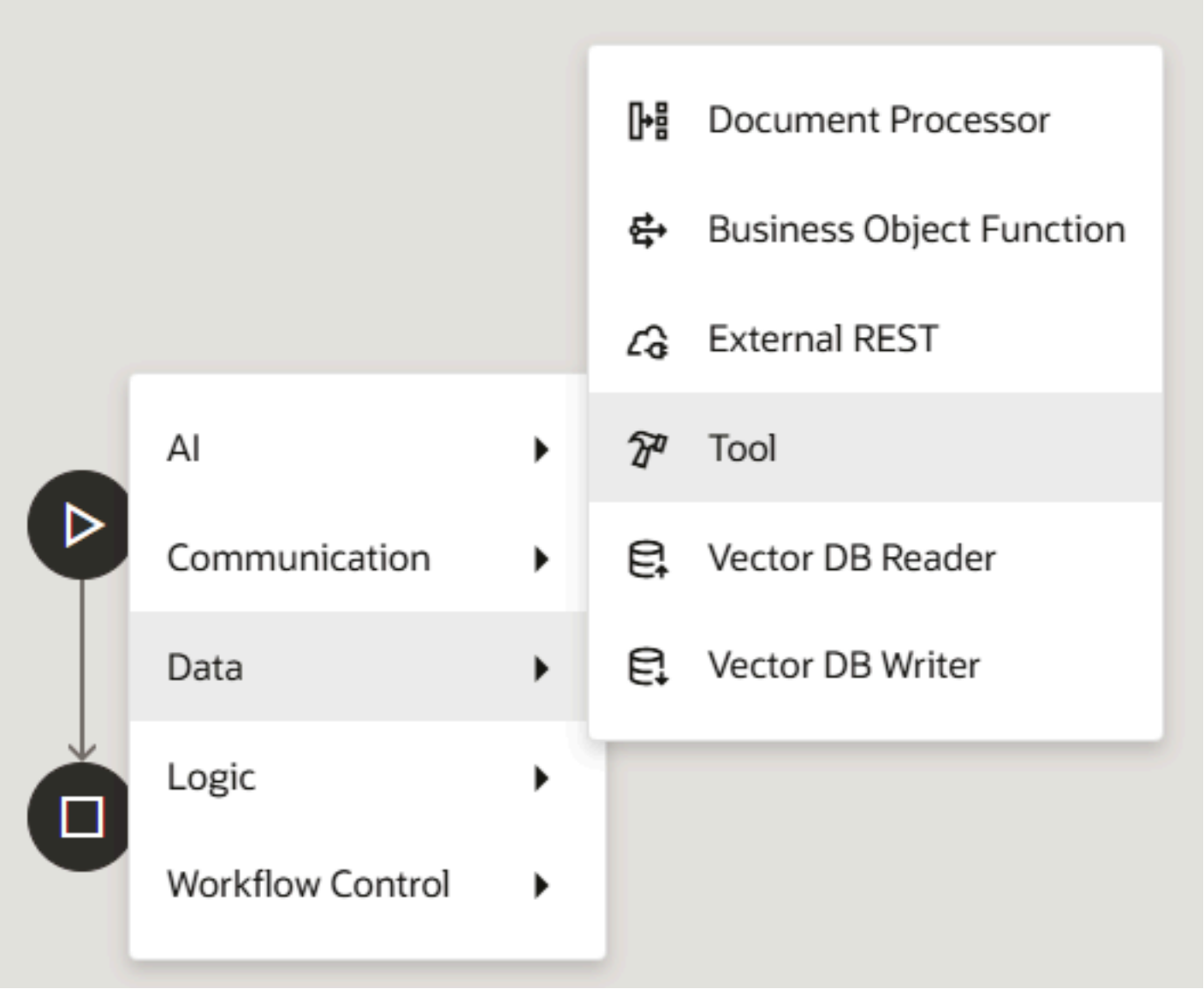
1. Go to the **Monitoring and Evaluation** tab and open the **Activities** subtab.
2. Go to **Schedules** to view the entry and verify the scheduled trigger is initiating the workflow correctly.

Tool Nodes

Tool nodes provide direct access to some common tools that are available to use in workflow agents. Here are the tools that can be used in a Tool node:

- Deep Link
- Intent change indicator
- Chat attachments reader
- User session

To add a tool node in your agent team of type Workflow, add a node of type Data → Tool.



For each Tool node, you'll need to specify these common configuration items:

- Enter a unique name for your tool.
- The code field will automatically populate based on the name field. You can change this if you wish.
- Select an error-handling option from the list.

Based on the selected tool type, the fields will vary.

Tool Type	Description
Deep Link	Enables workflows to reference and execute an existing Deep Link tool without recreating it. This tool is used to generate context-aware navigation links to Oracle Fusion Cloud Applications; for example, links to business objects or transactions.

Tool Type	Description
	This tool mirrors the functionality of Deep Link usage in supervisor agents. It references preexisting Deep Link tools, returns a URL that points to a specific business function, and enables seamless navigation from the workflow to the application UI. Select **Deep Link** as the type and enter these details: • **Family**: Select the family of products to associate this tool node with. • **Product**: Select the product you're working with. • **Tool Name**: Select the associated tool name. • **Parameters**: Displays the parameters (if any) associated with the tool you selected from **Tool Name**. Select **Edit** to change any parameters shown.
Chat attachments reader	Enables workflows to process files uploaded during a chat session. The output is the extracted content, provided as plain text in Markdown or Base64 format. This tool is also known as Multifile Processor. Select **Chat attachments reader** as the type and enter these details: • **Conversation Identifier**: Select the variables to uniquely identify conversations and apply them to the attachments. Type {{ to add a variable, or select from the **Insert Expression** menu. You can add multiple variables. • **Parse files**: Select to enable extraction of readable content or metadata. This option is on by default and is best for most text-based use cases. • **Get raw file content in Base64 format**: Select to get the original bytes of the attachment for downstream processing or storage. For example: o Pass files to downstream APIs that accept Base64, like vLLM. o Preserve the original file for storage when the integration expects Base64. o Process nontext or hard-to-parse files like images, scans, and proprietary formats. o Re-create the file later, for example, by decoding Base64 back into the original binary in another step.
Intent change indicator	Enables workflows to reset or segment the chat history based on the intent specified in the expression entered in the Intent field. Select **Intent change indicator** as the type and enter this detail: **Intent**: Select the variables to uniquely identify intent change indicator. Type {{ to add a variable, or select from the **Insert Expression** menu. You can add multiple variables.
User session	Enables adding the User Session tool as a node and returns the data from the tool relevant for the signed in user, such as `PersonNumber` or `Username`. Select **User session** as the type.

Tool Type	Description

Add an Approval Process Channel to a Human Approval Node of Workflows

Approval processes determine how approval requests are routed, who must approve them, how approvers are notified, and which rules or conditions apply.

Associate a published approval process with a human approval node by selecting the Approval Process channel. Approval process channels provide a structured, reusable framework for managing approvals in agentic flows. At runtime, the workflow is paused while the approval request is routed through the selected approval process. The workflow resumes after the required approval outcome is received.

Create an Approval Process

1. Go to AI Agent Studio.
2. Open the **Approvals** tab, and select **Add**.
3. Enter a name. The code is automatically populated based on the name.
4. To add a description, first select **Generate**, add a short description, and then select **Go**.
5. Modify the description if required, and select **Create**.
6. From **Notification Channels**, select the channels to use for approval notifications. For an email, specify the email account.
7. Add the approval payload fields that the approval process requires.
8. Add the approval levels and configure the approver details.
    a. Enter a level name and description.
    b. Select the approver type and specify the approver details.
    c. To add a rule logic, first select **Generate rule**, add a logic in plain language, and then select **Go**.
9. Add a condition by selecting a field, operator, and value.
10. Publish the approval process.

Published approval processes appear in the Approvals tab and can be selected in human approval nodes within workflows.

Associate an Approval Process to a Human Approval Node

1. From the Workflows tab of AI Agent Studio, edit a workflow and add a human approval node to it.
2. Enter a name and description for the human approval node. The code is automatically populated based on the name.
3. Select an error handler node to define failure handling.
4. From the Channel list, select **Approval Process**.
5. If required, enable feedback and configure the feedback options.
6. If you select **Yes**, choose the node to return to when feedback or retries are enabled.
7. Specify the maximum number of times the node can run.
8. From the Approval Process list, select a published approval process.

9. You can provide additional details such as subject and message. Otherwise, the node uses the properties of the selected approval process.
10. Publish the workflow.

When the workflow runs, the human approval node sends the approval request through the selected approval process.

Add a Document Schema to a Document Processing Node of Workflows

Document schemas define the structure and fields of data to be extracted from documents in agentic flows.

You can generate a document schema from a sample document, publish it for reuse, and associate it with a document processor node. When the workflow runs, the node uses the selected LLM and schema to extract structured information from documents and return the extracted data in the format defined by the schema.

Create a Document Schema

1. Go to AI Agent Studio and open the **Resources** tab.
2. Select the **Document Schema** subtab.
3. Select **Add**.
4. Enter a name and description for the document schema. The code is automatically populated based on the name.
5. In **Generate Schema From File**, select **Choose File** and upload a supported sample document. See *What file types, limits, and processing capabilities are supported in AI Agent Studio?*
6. Select **Generate Schema**.
7. Review the generated schema definition.
8. Publish the document schema, or save and close it to publish later.

Published schemas appear in the Document Schema subtab and can be reused across workflows. If you save and close a document schema, a draft copy appears in the Document Schema subtab. You can open the draft, make changes, and publish it. You can also open a published document schema, go to the draft, update it, and republish the schema.

Associate a Document Schema to a Document Processor Node

1. From the Workflows tab, edit a workflow and add a document processor node to it.
2. Enter a name and description for the document processor. The code is automatically populated based on the name.
3. Select an error handler node to define failure handling.
4. Select a product family and a product.
5. Select the business object that provides the functions and parameters for the node.
6. Select the function to run on the selected business object.
7. In the **Data Extraction** section, set **Use LLM for data extraction** to **Yes**.
8. Use the default LLM configured for the workflow or select another model.
9. From the **Schema Template** list, select a published document schema.
10. Publish the workflow.

When the workflow runs, the document processor uses the selected schema and LLM to extract information from the document. The extracted data is returned in the format defined by the schema, so workflow nodes, applications, or agents can process it more easily.

Add a Reference Block to a Reference Node of Workflows

Reference blocks define reusable logic in agentic workflows.

You can include artifacts such as agents, LLMs, tools, and other workflow nodes in a reference block, associate the block with a reference node, and reuse that logic multiple times in a workflow. When the workflow runs, the reference node uses the logic defined in the reference block.

Create a reference block

1. Go to AI Agent Studio.
2. From the Workflows tab, edit a workflow and add a Reference Block node to it.
3. Enter a name and description for the reference block. The code is automatically populated based on the name.
4. Select an error handler node to define failure handling.
5. Add any required parameter names and their data types.
6. Add the nodes and logic that you want to reuse within the workflow.

The reference block is now available for use by reference nodes in the workflow.

Associate a Reference block with a Reference Node

1. From the Workflows tab of AI Agent Studio, edit a workflow and add a Reference node to it.
2. Enter a name and description for the reference node. The code is automatically populated based on the name.
3. Select an error handler node to define failure handling.
4. From the Reference Block list, select the reference block that you want to use.
5. Add the input values required by the reference block.
6. Leave the output schema blank so that the node can use the output of the reference block.
7. Publish the workflow.

When the workflow runs, the reference node runs the selected reference block and returns the output to the calling node.

Add Multi Agent Nodes in AI Agent Studio

You can use Multi Agent nodes in which a supervisor agent routes requests to specialized worker agents.

Configure each worker agent with a specific responsibility and use the supervisor prompt and output specification to control routing and response consistency. In the node's properties, you can define how the supervisor agent evaluates incoming requests and selects the appropriate worker agent.

1. Go to AI Agent Studio and open the **Workflows** tab.
2. Edit a workflow and add a **Multi Agent** node to it.
3. Enter a name and description for the node. The code is automatically populated based on the name.
4. Select an error handler node to define failure handling.
5. Specify the maximum number of interactions the node can run.
6. Specify whether the agent must respond in the user’s language. Otherwise, the workflow default language is applied.

7. Specify whether the agent must use the hallucination guardrail prompt every time it runs.
8. Specify whether to include chat history as context for the supervisor and worker agents.
9. In **Agent Persona and Role**, enter the supervisor agent's role, tone, and behavior instructions.
10. In Prompt, use expressions to specify the routing decisions. For more information, see *Expressions in AI Agent Studio*.
11. Select and add topics to define the areas covered by the Multi Agent node.
12. Specify whether to enable summarization to generate a conversation summary after routing completes. If enabled, use expressions to specify the summarization prompt.
13. In **LLM**, select a language model.
14. Define the output specification for downstream steps and expression tools.
15. Add **Agent** nodes within the Multi Agent node as worker agents.
16. Enter a name. The code is automatically populated based on the name.
17. From the **Agent** list, select an existing agent as worker agent for the Multi Agent node. The properties of the required fields are populated with the values defined in the selected agent.
18. Publish the workflow.

When the workflow runs, the supervisor agent receives the user request, evaluates the intent, and routes the task to the most appropriate worker agent. The selected worker agent performs the task and returns the response. If a request spans multiple topics, the supervisor agent can coordinate across multiple worker agents and return a consolidated response.

Configure a Return Node in a Workflow

Configure a Return node to stop processing a workflow branch and return a value to the calling node.

Return nodes are typically used in reusable workflow components, such as reference blocks. Define the return value directly or use expressions to reference workflow variables and outputs from previous nodes.

1. Go to AI Agent Studio and open the **Workflows** tab.
2. Edit a workflow and add a Return node to it.
3. Enter a name and description for the node. The code is populated automatically based on the name.
4. Select an error handler node to define failure handling.
5. Use expressions to specify the value the Return node must send to the calling node. For more information, see *Expressions in AI Agent Studio*.
6. Publish the workflow.

When the workflow reaches the Return node, the current workflow branch ends, and the configured value is returned to the calling node. The returned value can then be used by downstream workflow nodes, applications, agents, or other workflow components.

Expressions in AI Agent Studio

You can use expressions to capture, store, and reuse values instead of deriving them each time, across various steps in your agent team, from AI Agent Studio. For example, you can use expressions to reuse extracted fields such as `personId`, intermediate node outputs, and so on. Expressions are different from static values. Static values can be anything you specify, whereas expressions are represented in the `{{any expression}}` format. You can directly enter expressions in the

required format, or you can add it from the **Insert Expression** menu that's displayed with the fields. Here are the types of values allowed in expressions:

- String
- Date
- Integer
- Number
- Boolean
- Object
- Array

Variables Specific to Workflow Agent Teams

Expressions contain different types of variables that provide the necessary context. They've a defined set of scopes in workflow agent teams.

Scope	Description
Global	Values that can be referenced anywhere in the workflow. For example, `Current Date`, `Trigger Type`, and variables defined at the workflow level.
Node	Values such as the node input variable's value and the output of each node in the workflow.
Trigger	Every workflow can have multiple triggers and each trigger will have its own set of inputs.

Here are the types of variables available in expressions that are specific to agent teams of type Workflow:

- System Variables
- Workflow Variables
- User Variables
- Trigger Variables
- User-defined Variables
- Node Variables

System Variables

System Variables are predefined runtime read-only values provided by the application. These variables can be referenced in inputs, agent mappings, and prompts using the `{{$context.$system...}}` syntax. For example, `{{$context.$system.$currentDate}}`, `{{$context.$system.$inputMessage}}`.

Scope	Expression	Description
Current Date	`{{$context.$system.$currentDate}}`	Current system date (ISO format)
Current Date and Time	`{{$context.$system.$currentDateTime}}`	Current system time stamp

Scope	Expression	Description
Trigger Type	`{{$context.$system.$triggerType}}`	Trigger that started the workflow
Input Message	`{{$context.$system.$inputMessage}}`	Raw user chat input
User Name	`{{$context.$system.$chatHistory}}`	Full conversation history for the current session (prior user and assistant turns)

Workflow Variables

Workflow variables are workflow-scoped runtime values that are created and managed within a workflow process.

Scope	Expression	Description
Workflow Name	`{{$context.$workflow.$name}}`	The ID of the workflow being run. The variable is always available at runtime, and remains stable across runs unless the workflow is renamed.
Trace ID	`{{$context.$workflow.$traceId}}`	Unique identifier for a specific workflow run instance that's generated per run. This variable is critical for debugging and support.
Conversation ID	`{{$context.$workflow.$conversationId}}`	Identifier for the conversation or session associated with the workflow run.

User Variables

User variables are predefined runtime values that describe the user currently interacting with the application or chat. The user's details are available using the `{{$context.$user.}}` syntax. For example, user name can be retrieved using the expression `{{$context.$user.$name}}`.

Scope	Expression	Description
User Name	`{{$context.$user.$name}}`	ID of the authenticated user who initiated the workflow execution

Trigger Variables

When a workflow agent team is called by an email trigger or REST trigger, trigger variables are available for use.

- REST Trigger

Scope	Expression	Description
Triggers > **REST** > **Input** > **field**	`{{$context.$triggers.REST.$input.<field>}}`	Specific request field

- Email Trigger

Scope	Expression	Description
Triggers > **Input** > **subject**	`{{$context.$triggers.EMAIL.$input.subject}}`	Subject line of the incoming email that triggered the workflow
Triggers > **Input** > **fromAddress**	`{{$context.$triggers.EMAIL.$input.fromAddress}}`	Email address of the sender
Triggers > **Input** > **content**	`{{$context.$triggers.EMAIL.$input.content}}`	Body of the email
Triggers > **Input** > **headers**	`{{$context.$triggers.EMAIL.$input.headers}}`	Full set of email headers
Triggers > **Input** > **attachments**	`{{$context.$triggers.EMAIL.$input.attachments}}`	Attachments included in the email

User-defined Variables

In addition to the system provided variables, you can also add variables that can be shared across nodes in an agent team of type Workflow. You can define these variables when creating the workflow agent team, and they're available throughout your agent team's flow. You can use the defined user variables in user questions or conversations.

To add user defined variables, do these steps:

1. When creating or editing a workflow agent team, select the Variables tab.
2. Select **Add** and enter the details:
    - Name: Enter the name for the variable
    - Type: Choose the type, for example, string.
    - Scope: Choose the scope as **User Question** or **Conversation**.
3. Apply the changes and save.

You can now reference the variable using `{{$context.$variables.<variableName>}}`.

To assign a value for the variable you created, do these steps:

1. Edit the workflow agent team in which you want to add the user-defined variable.
2. Add a node of type **Logic** > **Set Variables** and enter the details to create the variable node.
    - From the Variables section, select **Add**.
    - From the **Name** field, choose the user defined variable you had created.
    - Enter the value for the variable. You can enter a static value, or a dynamic value that's derived from the runtime context.
3. Save your changes.

Scope	Expression	Description
Variables > **<name>**	`{{$context.$variables.<name>}}`	Workflow-level variable

Node Variables

- **Node Inputs**

  Node inputs are variables used to pass input to the node in the workflow agent team.

Scope	Expression	Description
Entire input	`{{$context.$nodes.<nodeCode>.$input}}`	Returns the complete input object passed to the specified node.
Specific input field	`{{$context.$nodes.<nodeCode>.$input.<field>}}`	Returns a single field from a node's input object.

- **Node Outputs**

  Node outputs are variables that contain the output of the node in the workflow agent team.

Scope	Expression	Description
Entire output	`{{$context.$nodes.<nodeCode>.$output}}`	Returns the complete output object produced by the specified node (schema-based if defined).
Specific output field	`{{$context.$nodes.<nodeCode>.$output.<field>}}`	Returns a single field from a node's output object.

    - Use entire input or output objects when flexibility is needed or when passing data to an LLM for reasoning.
    - Use specific fields when doing these actions:
        - Writing conditions
        - Assigning workflow variables
        - Populating email subjects or concise prompts
        - Referencing a field that doesn't exist at runtime resolves to empty or null

- **Node Error Details**

  Node error details are variables that are available when a node in the workflow agent team fails and the Error Handler is called. They're scoped to the node that encountered the error.

Scope	Expression	Description
Error information	`{{$context.$nodes.<nodeCode>.$error.$info}}`	Detailed error information returned by the failed node

Scope	Expression	Description
Error code	`{{$context.$nodes.<nodeCode>.$error.$code}}`	Machine-readable error code associated with the node failure

- **Node Execution Status**

  Node execution status variable contains the status of the node in the workflow agent team.

Scope	Expression	Description
Node status	`{{$context.$nodes.<nodeCode>.$status}}`	Processing status of the node

Variables Specific to Supervisor Agent Teams

Here are the types of variables available in expressions that are specific to agent teams of type Supervisor:

- System variables
- Input variables
- Self or agent scoped input variables

System Variables

These variables are runtime, read-only variables.

- `{{$context.$system.$currentDate}}`: The current date at runtime.
- `{{$context.$system.$currentDateTime}}`: The current date and time at runtime.
- `{{$context.$system.$inputMessage}}`: The user's latest message that triggered the agent run.
- `{{$context.$system.$chatHistory}}`: The conversation history available at runtime.
- `{{$context.$system.$availableAgents}}`: The list of agents available for the supervisor to route work to.

Input Variables

These variables are the inputs configured for the supervisor agent team. To pass values to the supervisor agent team as configured inputs, use the `{{$context.$input.<name>}}` syntax.

To configure an input variable, do these steps:

1. When creating or editing a supervisor agent team, select the Input Variables tab.
2. Select **Add** and enter the details:
   - Name: Enter a unique name for the variable. The name is the reference key. For example, name is used in the `{{$context.$input.<name>}}` format.
   - Choose the type, for example, string.

3. Enter the value for the variable. You can enter a static value, or select **Insert Expression**  to add a context expression.
4. Add a description and save your changes.

Self or Agent Scoped Input Variables

Self variables (`$agents.$self`) provide the current agent's own context, that is, its configured inputs and added topics. To reference these variables, use the appropriate format:

- For agent input : `{{$context.$agents.$self.$input.<name>}}`
- For specific topic: `{{$context.$agents.$self.$topics.<name>}}`

6 Edit and Test AI Agents

Edit an Agent Team

To edit an agent team, do these steps:

1. From the Agent Teams tab of AI Agent Studio, search for and select the agent team you want to edit.
2. Select **Edit** to modify the artifacts such as tools, topics, and external tools in the agent team.

   > **Note:** If you see **View** instead of **Edit** , it indicates that the agent team is a preconfigured one, which can't be edited or directly modified. For more information, see *Can I edit a preconfigured agent team?*

3. To edit the agent team details such as the large language model (LLM), chat experience, questions and agent team security, select **Agent Team Settings** . For descriptions of the fields on these tabs, see *Create Custom AI Agents of Type Supervisor* and *Create Custom AI Agents of Type Workflow*.
4. If needed, use **Debug** to test the agent team.
5. Publish your agent team.

Edit and Test AI Agents Iteratively in Playground

You can use **Edit in Playground** to edit and test your AI agents and large language model (LLM) nodes. This option enables iterative, real-time testing and refinement of agent instructions and parameters, without deploying changes to the production environment. It bridges the gap between high-level intent and step-by-step execution, so you can safely test, adjust, and fine-tune your agent's behavior before deploying it to production.

To edit and test AI agents:

1. Go to **AI Agent Studio**.
2. Choose the appropriate tab based on your project:
   - Agent Teams: For collaborative or multi-agent flows
   - Agents: For individual agent configurations
3. Search for the agent or agent team and then select **Edit** to modify their settings.
4. From the editing area, select **Edit in Playground** to edit and test your AI agents.

5. Edit the agent details to adjust the prompt logic and model parameters. To edit and view real-time results in a dual-pane layout, you can select **View Results** .

Prompts Tab

You can design templates and instructions that guide the agent's behavior.

> **Note:** The system prompt and summarization prompts are specific to supervisor type of agents.

Field	Description
System Prompt	Edit the system prompt to define your agent's identity, job description, and rules.
Summarization Prompt	Select the summarization mode, and edit the summarization prompts to include only essential instructions relevant to your specific use case.

> **Note:** You can add expressions to fields using **Insert Expression** . For more information, see *Expressions in AI Agent Studio*.

LLM Tab

Choose the type of model and change the custom model properties if needed. You can either use the default model or choose a model.

Field	Description
Provider	Choose to use the default model or select a model. When using a custom model, specify the model properties.

Input Tab

Enter the specific test data, variables, or context required for an agent or LLM node to run. You can simulate real-world triggers and verify that your agent processes incoming information correctly before saving your changes.

Field	Description
User Input	Enter your queries or actionable commands that initiates the agent's logic.
Evaluation	Use evaluation sets to assess your agent's performance. An evaluation set contains one or more test questions, the expected agent responses, and the metrics to be measured. For more information, see *Evaluate Agents*.
Variables	Configure variables, making them accessible to all nodes within the Workflow type of agent.

Field	Description
Additional Variables	Add additional variables to the prompt. For example, to add the current system date to your prompt, select the **Current Date and Time** option using **Insert Expression** .

Output Tab

Configure the overall structure of the agent's output using JSON schema, to specify the exact output.

Field	Description
Specification Mode	Select this mode to directly modify the JSON schema for the output.
Simple Mode	Select this mode to define the output values and types. The corresponding JSON schema will be generated automatically and displayed in the specification mode for any further changes.

6. Select **Save and Test** to verify the changes you made and test the agent or the LLM node. While the test is running, you can monitor the results and performance in these sections:
   - Input: Review the messages you entered in the Input Message field on the Input tab. To view the JSON schema, select **JSON**. To view the final response, select **Human**.
   - Output: Select a session to open the detailed trace view. The trace shows a step-by-step timeline of the run, including which tools were called, how long each step took, and metrics for each step. Select the color-coded trace lines to view details, such as latency and input or output token usage. To view the final response, select **Human**.
7. Select **Run History** to view, compare, and reuse previous test runs. You can view a detailed record of every run and track your agent's performance.
   - Select **Edit** to add a comment in the change log.
   - Select **Apply** to load a proven configuration into the panel and resume iterative testing from a high-quality baseline. This action only affects the testing environment and won't update your production agent until deployed.
   - Use **Expand** to view change log details of the test.
   - You can compare any two runs by selecting them. Select **Compare** to view a pane displaying the details of each run and highlighting their differences.
   - Select **Save and Test** to save your changes to the agent, and run the agent with the test configuration.
8. Select **Done**.
9. After testing your changes, publish your agent team.

7 Monitor and Evaluate AI Agents

Monitor and Evaluate AI Agents

Monitor and gain insights into how your AI agents are performing, and also evaluate the agents for accuracy. You can also track the interactions with your agents, understand real-world usage patterns, identify common errors, and measure overall performance.

- Monitoring: Monitoring tracks performance and provides insight on how your agents behave in production. Monitor agents to ensure that your quality bars for response time and token counts are maintained over time. You can also see any errors logged here.
- Evaluation: Evaluate agents before you deploy them, to ensure that they're ready for production. Test your agents for response correctness, response time, and token usage to meet your quality standards. You can also check the quality of answers generated through the document tool to assess how effectively agents utilize the retrieved context from the retrieval-augmented generation (RAG) metrics. After making any changes to your agent, or after a model update, rerun evaluations to confirm that your agent continues to perform as expected. This proactive approach helps you maintain high-quality experiences for your users.

Watch video

This table summarizes some key metrics, their descriptions, and their availability for monitoring or evaluation.

Metric	Description	Available to Evaluate	Available to Monitor
Error Rate	Percentage of user sessions that ended in an error.	Yes	Yes
Error Count	Total number of errors recorded.	Yes	Yes
Session Count	Total number of conversations initiated with agents.	Yes	Yes
P99 Latency	The maximum wait time in milliseconds for 99% of users, revealing any areas where you should review and optimize the prompts or structure of the agent.	Yes	Yes
P50 Latency	The maximum wait time in milliseconds for 50% of users, helping identify performance issues.	Yes	Yes

Metric	Description	Available to Evaluate	Available to Monitor
	You can view this metric in the details of the monitoring or evaluation results.		
Total Tokens	Cumulative number of tokens used by all agents.	Yes	Yes
Input Token Count	Total tokens sent to the LLM for requests. This includes system prompts, user messages, retrieved or context data, chat history, and tool or function definitions.	No	Yes
Output Token Count	Total tokens generated by the LLM for requests sent to it.	Yes	Yes
Median Correctness	The 50th percentile of correctness scores across evaluation runs. Each score (0–1) is computed by comparing the agent's answer to the reference answer provided in the evaluation set.	Yes	No
Session Count	The number of unique conversational sessions between a user and an AI agent. One session can include multiple messages or evaluation runs.	Yes	Yes
Groundedness	Alignment of the generated answer with the retrieved source content, indicating how faithfully the response reflects the supporting information.	Yes	No
Answer Relevance	Degree to which the answer directly addresses the user's question, measuring how fully and precisely the content covers the required subject matter.	Yes	No
Context Relevance	Quality and appropriateness of the retrieved information, assessing whether it meets the necessary standards to be considered reliable evidence.	Yes	No

Prerequisites

Aggregate the metrics that are displayed on the Monitoring and Evaluation tab in AI Agent Studio.

1. Go to **Navigator** > **Tools** > **Scheduled Processes**.
2. In Scheduled Processes, click **Schedule New Process**.
3. Leave the type as **Job**.
4. Search for and select **Aggregate AI Agent Usage and Metrics**.
5. Run the **Aggregate AI Agent Usage and Metrics** scheduled process.

   You can schedule this process to run on a recurring basis, for example, once a day.

The process aggregates the metrics that are displayed in the Monitoring and Evaluation tab of AI Agent Studio.

Monitor Agents

1. Go to **Navigator** > **Tools** > **AI Agent Studio**.
2. From the Monitoring and Evaluation tab, go to the Monitoring subtab.
   - The Monitoring subtab displays aggregated metrics of all the agent runs over the selected time frame.
   - All agent runs are monitored, including agents in draft status.
3. Select the agent to view additional details.

   Each row represents a single session and displays the number of turns (back-and-forth messages), the session's completion status (successful or error), and the number of tokens used.
4. Select any session in the list to open the detailed trace view. This view displays a step-by-step timeline of the entire conversation, showing exactly which tools were called, the duration of each step, and the metrics for each step.

Evaluate Agents

Use evaluation sets to assess your agents' performance. An evaluation set contains one or more test questions, the expected agent responses, and the metrics to be measured. Evaluation sets are specific to each agent, and an agent can have multiple evaluation sets. For more information, see *Best Practices for Creating Evaluation Sets*.

1. Create an evaluation set for an agent.
   a. Go to **AI Agent Studio**.
   b. Select the Monitoring and Evaluation tab.

      All evaluations run on the agents are displayed in the Evaluation subtab.
   c. To create an evaluation set, click **Manage Evaluations** and select  .
   d. Enter a name, code, and description for the evaluation set, and select the agent team to be evaluated.
   e. Choose the run mode.

Sequential: Runs the questions in the exact order you define them. Use this if one question depends on the context of the previous one.

Random: Runs the questions in a random order.

f. Select **Enable document tool evaluation** if you want to view RAG metrics in the evaluation report.
g. From the Questions subtab, add common questions that users are likely to ask the agent and the answers you would like the agent to deliver. Ensure both the questions and answers are concise, user-friendly, and reflect best practices.

   You can either add your questions and the expected answers individually, or upload a CSV file with the questions in the first column and the expected answers in the second column.
h. From the **Metrics** subtab, edit each metric to set the pass and fail criteria. For example, to indicate that the test fails if the correctness score is less than 0.7, choose **<** as the threshold condition and enter 0.7 as the threshold value.
i. Select **Create** to save the evaluation set.

2. Run the evaluation set.
   a. On the Manage Evaluations page, select the **Initiate Evaluation Run** action for your evaluation set.
   b. Choose the version of the agent team to evaluate and run the evaluation.
3. Analyze the results.
   a. Click the evaluation set to view the Evaluation Runs page.
   b. Select the evaluation run and select the **View Run Results** action.

Subtab	Information Displayed
Response performance	- Comparison of the expected response versus the actual response from the agent for each question, along with the metrics for each question in the evaluation. - Trace provides information about the detailed timeline for each question in the evaluation.
Correctness	Detailed breakdown of the correctness score. The LLM provides an initial score and feedback, and you can add your own feedback for record keeping in the **Correctness Score by Human** column.

Compare Evaluation Runs

You can see a side-by-side comparison of two different runs of the same evaluation, and easily spot regressions or improvements in latency, correctness, and token usage. Doing so, you can understand how an agent's performance changes over time, especially after you've made modifications.

1. From the Evaluation subtab, select the evaluation.
2. Select any two runs and click **Compare**.
   - The Summary subtab displays a high-level overview of the performance differences between the runs.
   - The Details subtab provides a granular, question-by-question breakdown of the runs. For each question in the evaluation set, you can directly compare the actual response from Run 1 against the actual

response from Run 2. You can also compare the specific latency, tokens used, and trace links for each question, making it easy to pinpoint exactly where and why performance or accuracy has changed.

View the Measured Value of Agents

Use the **Value** dashboard to understand the business impact of your AI agent teams. This visual dashboard provides a consolidated view of your AI fleet, performance by Oracle Fusion Cloud Applications products, and metrics for individual agent teams. These metrics help you estimate and visualize the tangible ROI of your AI initiatives. You can track deployment, usage, and estimated time and cost savings for your published agent teams in one view.

Configure Performance Metrics for an Agent Team

To configure performance metrics for a specific agent team:

1. Go to **AI Agent Studio**.
2. From the Agent Teams tab, search for the agent team you want to edit, and select **Edit** .
3. Select **Agent Team Settings** to edit the agent team details.
4. On the Details tab, scroll to the **Capture estimated Cost and Time savings** section.
5. Configure the estimated values for ROI tracking metrics by setting **Time saved each agent run** and **Cost saved each agent run**. If you don't enter any values, the default values apply.
6. Select **Update** to save your changes.
7. Publish your agent team.

You can now view the measured value of the agent team using the Value subtab of the Monitoring and Evaluation tab.

Monitor AI Agent ROI and Key Efficiency Indicators

1. Go to **AI Agent Studio**.
2. From the Monitoring and Evaluation tab, select the Value subtab. You'll see a dashboard that displays real-time ROI per agent and team, allowing you to filter by Family or Product to monitor key metrics such as **Total Time Savings**, **Total Cost Savings**, **Total Usage**, and **Total Agents**.
3. Select any item on the dashboard to see the total time saved (hours) and total cost saved (USD) across agent teams.
4. You can also configure the estimated ROI metric values for an agent team. Select the **Agent Teams** link, then search for the agent team and select **Edit** to update the settings. Select **Apply**, and then save your changes.

Best Practices for Creating Evaluation Sets

Creating a good evaluation set helps in improving the efficiency of your evaluation. Here are some best practices for creating evaluation data sets and testing protocols for agent teams.

The foundation of a successful evaluation is a high-quality data set. This data set consists of paired questions or inputs, and expected responses. The expected response must be factually correct and grounded in the source context.

Example Data Set

Question	Expected Response
Is aromatherapy covered?	No, aromatherapy isn't covered. According to the provided context, aromatherapy is listed under "Alternative Treatments" which aren't covered by United Healthcare Medical Plans.
Do you pay for thermometers?	Based on the provided context, thermometers aren't covered. The document "MEDICAL SUPPLIES AND APPLIANCES" lists thermometers as excluded supplies.
Is laser surgery for eyes covered?	Based on the provided context, laser surgery for eyes isn't covered. The "VISION" section states that surgery to correct nearsightedness, including laser surgery, is listed under plan exclusions.

General Evaluation Guidelines

These principles apply to all agent teams, to ensure comprehensive testing.

- Safety and grounding: Validate that the agent adheres to safety policies. It must avoid hallucinations, that is, if no relevant information is found, it should state that, rather than guessing.
- Ambiguity and multipart queries: Evaluate the agent's ability to handle vague queries, implicit reasoning, or single turns containing multiple distinct questions.
- Negative testing: Introduce out-of-scope or unanswerable questions, for example, topics not covered by the available tools, to ensure the agent responds with an appropriate "I don't know" or redirect.
- Multiturn dialogue: If the agent is conversational, design tests that require maintaining context and history across several turns, for example, answering a follow-up question based on the previous answer.
- Latency and performance: Measure response time and efficiency, particularly in scenarios involving calls to multiple tools or complex reasoning.

Evaluation Guidelines for Supervisor Agent Teams

Agent teams of type Supervisor are evaluated based on the tools they use.

Tool	Guidelines
Document Tool (RAG)	Questions designed for Retrieval-Augmented Generation (RAG) must test the agent's ability to handle complexity, not just simple keyword retrieval. • Long Range Context: Test if the agent can resolve dependencies scattered across distant sections or several pages of a document. • Distributed Context: Ensure the agent can aggregate information from multiple noncontiguous parts of the document to answer comprehensively. • Concealed Context: Test the ability to find and extract specific, obscure details deep within the text. • Reasoning: Check if the agent can apply reasoning or logic to the retrieved information to provide a correct answer. • Table-Sourced: Test the ability to interpret and pull accurate data from tables within the document.
Business Object	• Function Coverage: Ensure evaluations test all business functions available in the business objects. • Parameter Variation: Test the same business function from different angles and with different parameters. For example, if a BO creates an object, test it with various input types.

Tool	Guidelines
REST	• Endpoint Coverage: Ensure evaluations test all functions available in the REST tool. • Scenario Variation: Test the same REST tool for different scenarios and parameters, for example, handling different payload structures or update types.
Deep Link	Validation: Confirm that deep links are generated correctly and for the appropriate scenarios.
Model Context Protocol (MCP)	• Tool selection: Validate that the correct MCP tool is called. • Function accuracy: Validate that the correct functions are called within the MCP tool.

Evaluation Guidelines for Workflow Agent Teams

Evaluations for Workflow agents must test the overall logic flow and the robustness of individual nodes.

Workflow Structure and Logic

- Path coverage: If the workflow has multiple paths or branches, ensure the evaluation tests all paths.
- Scenario depth: Test multiple distinct scenarios for the same path to ensure consistency.
- Unsupported scenarios: Include tests that identify scenarios the workflow doesn't support, to verify graceful failure or error handling.

Workflow Nodes

A workflow includes multiple node types, and the evaluation must cover the key scenarios applicable to each type.

Node	Scenario
LLM	Test the LLM prompt to identify if it can handle all types of questions and formatting instructions.
Code	• Robustness: Include questions that test the stability of the code. • Edge Cases: Test for scenarios that cover different edge cases to ensure the code doesn't break the flow.
• Business Object • REST	Apply the same guidelines as specified for agent teams of type Supervisor. Test functions from different angles and with varied parameters.
RAG Document Tool	Apply the same guidelines as specified for agent teams of type Supervisor.
Document Processor	Format Handling: Test the node for various attachment types (PDF, txt, and so on) to ensure consistent text extraction.
Vector DB Reader	Retrieval Accuracy: Test if the node retrieves the most semantically relevant chunks based on the input query.

For the other supported nodes, make sure evaluation tests cover other nodes including Tools, Vector DB Reader, and Vector DB Writer.

8 Migrate RAG Agents to AI Agent Studio

Migrate Document Tools of RAG Agents

If you've created any RAG agents in Oracle Fusion Cloud Applications previously, we recommend that you replace your existing agent with an agent you create in AI Agent Studio.

You can migrate the Document tool you created for your existing agent to AI Agent Studio.

1. Go to the **Configure RAG Agents** work area.

   > **Tip:** You can use global search to get to the work area.

2. From the Tools tab, click  to migrate your tool.
3. Enter a unique tool name and code. That will help you easily find your migrated tool in AI Agent Studio.

After migrating your tool, create a new agent in AI Agent Studio using an appropriate template, add your migrated tool, and publish your agent. After testing and verifying your new agent, you can delete the original agent in the **Configure RAG Agents** work area.

9 Promote Published Agents

Migrate AI Agents from One Instance to Another

You can migrate an individual agent team from one environment to another using the import and export options available in AI Agent Studio. For example, you can export from the test or staging environment and import into the production environment.

After importing the agent team, you must publish it in the new environment to make it available for use. Also, if any of your agents use the following tools, make sure you complete the corresponding post-import tasks for each tool:

- Document tool: Upload and republish the document to ensure its content is correctly processed in the new environment.
- Email tool: Reconfigure the email tool to make sure necessary alerts are generated properly in the new environment.
- External REST tool: Reconfigure authentication settings for external REST API tools.

To migrate an agent team:

1. From the source environment, go to **AI Agent Studio**.
2. From the Agent Teams tab, locate the agent team you want to export and click the icon.
3. Enter the export details and click **Export** to download the JSON file to your local system.
4. From the target environment, go to **AI Agent Studio** and select the Agent Teams tab.
5. Click **Import** and upload the previously downloaded JSON file.
6. Use the **Draft** button to edit the imported agent team.
7. Publish the agent team.

10 View Channels in AI Agent Studio

Overview

You can create channels in AI Agent Studio to integrate external applications. To view the Microsoft Teams and Slack channels created before release 26C, do these steps:

1. Go to AI Agent Studio.
2. Open the Credentials tab, and then open the Channels subtab.

To use these channels in release 26C, create them from the Connectors tab. You can create further connectors from the Connectors tab. For more information, see *How do I use connectors in AI Agent Studio?*

11 Build Agentic Apps Using Agents from AI Agent Studio

Overview of Agentic Apps

Agentic apps help in combining your agents to form active, intelligent apps, enabling their use as your decision-making partners. You can consider the agents as workers and the agentic app is the productized solution that bundles those workers with the integrations, knowledge, and controls that are needed. The apps can include multiple agents, each of which is purpose-built for a specific category or domain.

Instead of asking one agent about everything from sales performance to inventory management, you can build specialized agents, for example, a Sales agent, an Inventory agent, a Finance agent; each with deep expertise in their domain. These agents can be used to create your agentic app. Doing so provides better insights and actions because each agent can maintain focused context, domain-specific reasoning, and tailored prompting for its area of responsibility. You can also scale it up by adding additional agents as needed.

The best agentic apps don't simply observe and report; they analyze, prioritize, and recommend actions that drive business forward.

- Proactive alerting over passive monitoring: Agents identify anomalies, opportunities, and risks in real-time, surfacing them with context and urgency. Instead of making users hunt for problems, the agents bring critical issues directly to their attention when they matter.
- Decision support, not data dumps: Apps filter signals from noise by analyzing situations and presenting only actionable decisions. Instead of providing users with every data point, provides synthesized insights with clear recommendations and trade offs.
- Action-oriented intelligence: Every agent interaction leads toward concrete actions, such as, approve a proposal, adjust a strategy, resolve a blocker. If it's purely informational, we can use an API or dashboard instead of agentic apps.
- Context-aware prioritization: Apps understand business goals and user workflows to determine what's urgent versus what can wait. Not all alerts are equal; agents should respect user's time and attention.

Here are the main pillars of agentic apps:

- **Information Displays**

  Shows you data. These are agent-generated visualizations, such as lists, charts, tables, that show relevant data in real-time using predefined patterns. Agents built for agentic apps answer questions by creating information displays, except when intent clarification, confirmation, or a brief response is more appropriate. Each information display is generated with a topic key that captures the context of the user request. As conversation topics shift, the page template may hide or remove displays that are no longer relevant.

  Agents must be configured to support specific information display types, granting them the capability to generate the displays. The implementation details are provided using prompting. At application startup, each agent may generate an initial information display that's shown to the user immediately. The agent determines whether to show an initial display and which display to show, and the host application can override either decision through prompting.

- **Advisors**

Answers your questions. Advisors are agents that answer user questions either independently or in coordination with other agents.

- Single agent mode: Users focus on a specific agent as their advisor
- Multi agent mode: Questions are routed to two or more capable agents, and outputs are consolidated into a unified response

Agents typically respond with both answers and supporting information displays. By default, information displays prioritize showing detailed data within the visualization itself, with accompanying text summarizing key insights. When a single agent acts as advisor, only that single agent produces an information display. In multi-agent scenarios, only selected agents generate displays, with the final answer summarizing their collective output.

- **Actions**

  Helps you make decisions. Actions are agent-generated decision points that require human approval. Here are the artifacts in actions:

  - Title describing the topic
  - Description of what the action performs
  - Instructions for the agent to execute if the action is called (including commands and parameters)

  This configuration is agent-generated and not modifiable by users. Actions are generally associated with a single agent, and when the action is called, that same agent processes the action. The agent either performs the action immediately or engages the user if further clarification is needed.

  Agents must be explicitly configured to generate actions. The generation logic and invocation handling are defined in the agent's prompting. At application startup, each agent can produce an initial list of priority actions that represent immediate decision points an app user may make. When answering questions, agents can include one or more actions in their response, which are then presented to the user.

  Information displays may also expose actions directly within their pattern. They work identically and their presentation is defined by the associated pattern.

- **Communications**

  Helps you reach out. Communications are agent-suggested outbound messages, for example, emails or Slack messages that follow consistent predefined templates. Here are the fields in Communications:

  - Title and description explaining what the communication does and who it targets
  - Template parameters used to compose the message

  Communications always initiate outbound messages; they never intercept or receive incoming messages. The communication suggestions are routed to a specific agent for delivery. This agent doesn't need to be an advisory agent. It can be a generic, reusable agent shared across applications. The designated agent then processes the communication request, performs any additional work, and delivers the message using the appropriate tool, such as email, Slack, and Microsoft Teams.

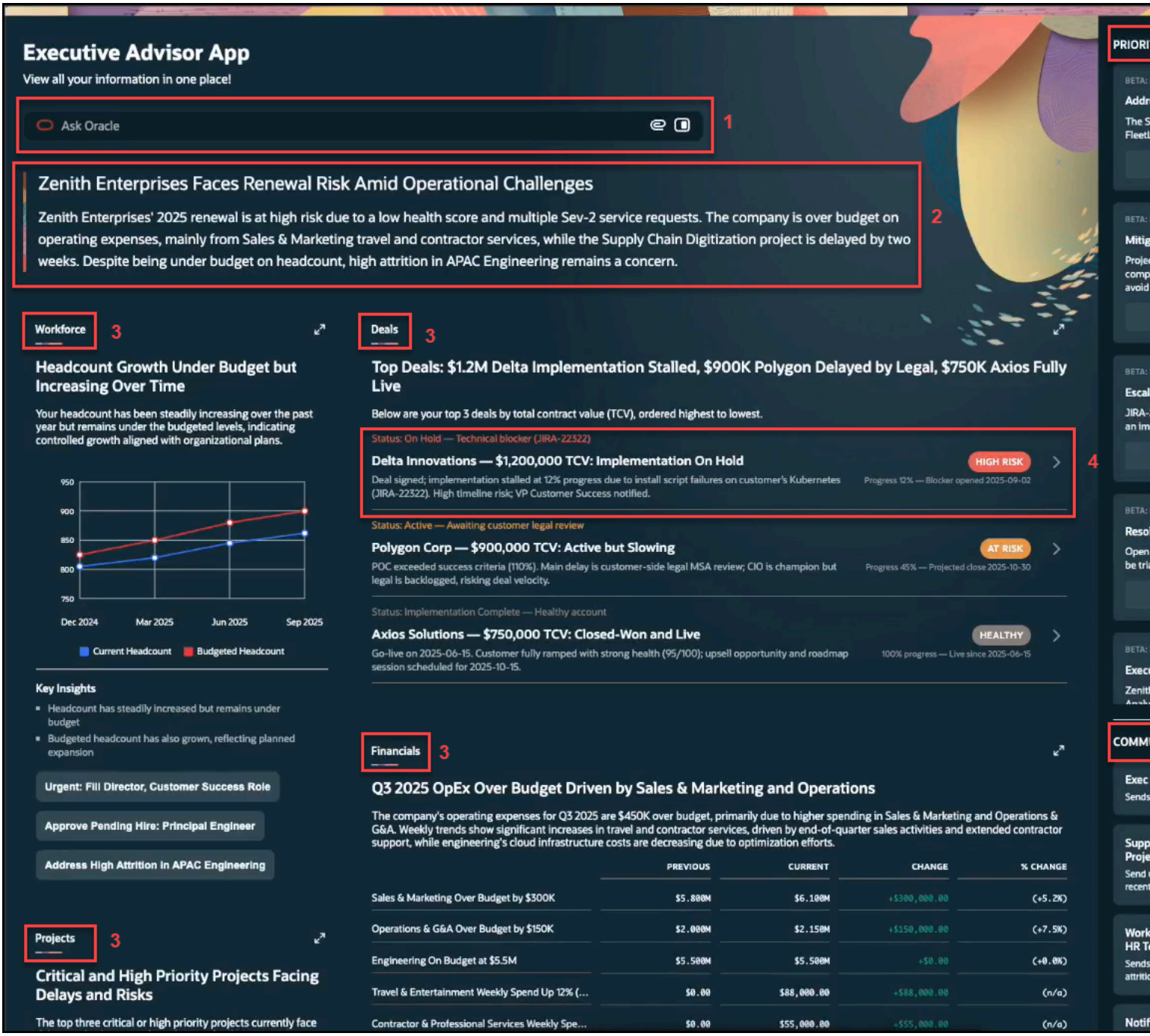
  All communications adhere to the templated patterns for consistency and uniformity. Messages aren't free-form; they conform to specific templates with defined parameter sets. Agents can produce communication suggestions when responding to inquiries and at app initialization. They can prefill template parameters based

on their knowledge and prompting. Communication suggestions can also be app-configured and associated with specific agents or all available agents.

When a communication item is selected, the associated agents attempt to populate any unfilled template parameters. The app user can then modify these values before the message is sent to the template engine for final processing.

Anatomy of an Agentic App

Here's an example of an Executive Advisor agentic app that consists of four agents.



Callout Number	Name	Description
1	Ask Oracle Advisor	Lets you ask natural language questions against the set of agents included in the app.
2	Summary Section	Displays the most critical information across all the domains in the app's agents. The summary

Callout Number	Name	Description
		is typically the first content users read when opening the app.
3	Agents	The example agentic app includes four agents, each represented by a tile on the dashboard. An agent can provide its own headline to convey the most important information for the app user. Users can zoom in on any agent to keep focus on that agent. Any subsequent questions, actions, and communications are then scoped to the selected agent.
4	Widget	A widget generated by an agent from the widget library. Each widget has space for the agent to provide its own insights, and capabilities that the user can choose from. Widgets can also have agent-specific actions that the user can act upon.
5	Priority Actions	App-specific critical information that require the user's immediate attention and provide options for the user to take action.
6	Communications	Outbound message suggestions that may be predefined by the app or generated by the LLM. It can start a flow allowing the user to construct the content and send the communication.

Create and Configure Workflow Agent Teams for Use in Agentic Apps

To build agentic apps with agent teams, first configure Workflow type agent teams for optimal results. Only published workflows can be used in agentic apps.

Best Practices

- Use an agent team of type Workflow. Workflow agent teams enable structured inputs and predictable flow.
- Make sure that the description of your agent team clearly defines the agent team's capabilities and specialization. When multiple agent teams are engaged, this description is used to help identify which team is best suited to handle a given query.
- Design smaller, topic-focused agent teams. You can combine multiple agent teams as needed within the agentic app.

- Keep workflows lean to reduce latency. Limit the number of LLM and agent nodes where possible to reduce perceived delay when resources are requested from the agent team.
- Pass user input and history explicitly. In workflows, conversation history and user input aren't automatically passed to internal LLM, agent, or workflow nodes. Use a code node or variable assignment to pass what you need (for example, using `$context.$system.$inputMessage)`, and then reference those variables in subsequent LLM or agent calls.
- When possible, end your workflow with an agent or LLM node to help prevent output streaming.

Create and Enable Workflows for Using in Apps

An agentic app is built using agent teams of type Workflow. These agent teams provide the response when the agentic app requests for displays, actions, communications, summary text, subtitle text, or user-query answers. The recommended method is for the agent team to receive the request from the agentic app using the `oraMessageHint` webhook variable. Create your agent team structure to receive the webhook message, branch the workflow, and then provide the needed output to the agentic app.

To use workflows in agentic apps, you must first enable the workflow agent teams from AI Agent Studio.

1. Go to AI Agent Studio and open the Agent Teams tab.
2. Edit the workflow agent team you want to use in agentic apps.
3. Select **Agent Team Settings** .
4. From the Details tab, select **Expose to agentic apps**.
5. Save your changes.

`OraMessageHint` Variable

`OraMessageHint` is the primary app-aware workflow variable, and its value can be retrieved using the expression `$context.$app.$OraMessageHint`. It tells your workflow why the app is calling it. In prompt templates, use expressions such as `{{ $context.$app.$OraMessageHint }}`. In Code nodes, use the value as `$context.$app.$OraMessageHint`.

Here are some common values for `OraMessageHint`:

Value	Description
`Summary`	Indicates that the app is asking for summary text. Return short summary text for that agent's contribution.
`InitSubtitle`	Indicates that the app is asking for the dynamic subtitle. Return the subtitle text itself.
`InitDisplay`	Indicates that the app is asking for startup displays. Return one or more `oraInfoDisplay` blocks, and it may also include actions and communication suggestions when that improves the startup experience.
`InitActions`	Indicates that the app is asking for startup actions. Return zero or more action suggestions.
`InitCommunications`	Indicates that the app is asking for startup communication suggestions. Return zero or more `oraComms` blocks.
`Query`	Indicates that the user asked something through the app. Use the input message and answer the user. This response can also include displays, actions, and communications when appropriate.

Value	Description
`InvokeAction`	Indicates that a UI action sent work back to the workflow. Use this for follow-up handling after a widget command triggers an action path that sends a command to an agent.
`AdditionalContent`	Indicates that the app is loading an extra panel under a focused agent. Use `OraPanelName` to tell which panel is being requested.
`FillParameters`	Indicates that the app is asking one or more agents to fill communication parameters before generating or sending a response.
`SendCommunication`	Indicates that the app is asking a target workflow to send or complete a communication flow.

Expressions Useful for Agentic Apps

Here are some expressions that you can use in your workflow agent teams:

Expression	Description
`{{ $context.$app.$OraMessageHint }}`	App stage that's calling the workflow
`{{ $context.$system.$inputMessage }}`	Current user message or app-supplied input for the current stage
`{{ $context.$app.$OraAppContext }}`	Current app context value
`{{ $context.$app.$OraUserContext.fullName }}`	Current user's complete name
`{{ $context.$app.$OraUserContext.userId }}`	Current user's ID
`{{ $context.$app.$OraAttachments }}`	Attachments available to the current app query flow
`{{ $context.$app.$OraAction }}`	Action payload or command supplied to the workflow action path
`{{ $context.$app.$OraCommParamsToFill }}`	Communication parameters that must be filled
`{{ $context.$app.$OraCommNonTemplateParamList }}`	Non template communication parameters for sender-style flows
`{{ $context.$app.$OraPanelName }}`	Additional panel that's requesting content

Here's an example user context object:

```
{
 "fullName": "John Doe",
 "userId": "jdoe1212",
 "userName": "John.Doe@example.com",
```

```
 "guId": "sfsdf332323fdsf"
}
```

Variables and Their Values

Multiple variables can be used in workflows created for agentic apps.

Variable Name	How the Value is Populated
`OraMessageHint`	Set by the app runtime based on what the app is doing right now: startup display load, startup actions, communications, query, subtitle, and so on.
`inputMessage`	The current message for that request. For a user query, it's the user prompt. For some action and communication flows, it's the command or content being sent into the workflow.
`OraAppContext`	Passed from the current app context that may come from the app URL, from app-to-app navigation, or from an action that switches context.
`OraUserContext`	Passed from the signed-in user and is available to app-aware workflows.
`OraAttachments`	Populated for query flows when the conversation includes attachments.
`OraPanelName`	Populated only when the app is loading additional panel content for a focused agent.
`OraCommParamsToFill`	Populated when the app is asking agents to fill in communication parameters.
`OraCommNonTemplateParamList`	Available for communication sender flows that need non template parameter information.

Enable Workflow Agent Teams to Generate Widgets

In your workflow, the LLM and agent nodes can generate widgets. Initial graphics and insight displays are produced by agent output blocks that carry a `patternId` and a widget-specific `config`. In App Builder, the display prompt tells the runtime agent what widget output to generate.

Enable Workflow Agent Teams

To enable workflow agent teams to generate widgets, do these steps:

1. Go to **AI Agent Studio** open the **Agent Teams** tab.
2. Select **Edit**  to modify the workflow.
3. Go to the LLM or agent node that should generate widgets.
4. Edit the node and open the **App Experience** tab.
5. Select the options to enable widgets for actions and communications as needed.
6. Select the types of widget that are to be generated by the workflow agent team.
7. Add instructions for the widget.

Types of Widgets

Here are the types of widgets an LLM or agent node can generate:

Type of Widget	Sample Prompt	Details	Example

Card	Use a cardWidget to present one important update as a clean, scannable card. Include a short subject, an optional subtitle for context, a brief summary, and if helpful a status line or time stamp, so the update feels informative without being crowded.	• **Use it for:** a single alert, update, or status card
		• **Required:** `subject`
		• **Common fields:** `summary`, `timestamp`, `badgeText`, `variant`
		• **Actions:** `link` and optional `additionalLink`, each with `text` and `action`
		• **Variant values:** `error`, `warning`, `info`, or omitted for neutral
		Use commands such as `ora.Invoke(...)`.
			`<oraInfoDisplay key="maintenance-card">`
			`{`
			`"patternId": "cardWidget",`
			`"config": {`
			`"subject": "Maintenance Complete",`
			`"summary": "Scheduled database maintenance finished successfully.",`
			`"timestamp": "1 hour ago",`
			`"badgeText": "Info",`
			`"variant": "info",`
			`"link": {`
			`"text": "View Details",`
			`"action": "ora.Invoke(\"viewMaintenanceSummar\")"`
			`}`
			`}`
			`}`
			`</oraInfoDisplay>`

Messages List	Use a messageListWidget to show a short list of recent messages, alerts, or notable events. Each item should have a clear title, optional supporting context like a subtitle or summary, and a time stamp when useful, so the list feels friendly, current, and easy to scan.	• **Use it for:** alerts, activity feeds, and small collections of highlight items
		• **Required:** `items[]` with at least `title` per item
		• **Item fields:** `subtitle`, `summary`, `status`, `badgeText`, `priority`, `timestamp`, `image`
		• **Interactivity:** item `action` and item `additionalAction` with `text` + `command`
		Runtime behavior is considered. `additionalAction` is only shown when the widget is in a focused panel. Also, the runtime supports success as a priority in addition to alert, warning, and medium.
			`<oraInfoDisplay key="system-alerts">`
			`{`
			`"patternId": "messageListWidget",`
			`"config": {`
			`"subtitle": "Real-time metrics",`
			`"items": [`
			`{`
			`"title": "Memory Usage Spike Detected",`
			`"subtitle": "82% confidence",`
			`"summary": "Memory consumption increased by 40% in the last 5 minutes",`
			`"status": "Investigating root cause",`
			`"badgeText": "Medium",`
			`"priority": "medium",`
			`"timestamp": "3 min ago",`
			`"action": "ora.Invoke(\"openMemoryIncident\", { \"metric\": \"memory\" })",`
			`"additionalAction": {`
			`"text": "View Metrics",`
			`"command": "ora.Invoke(\"viewMetrics`

Type of Widget	Sample Prompt	Details	Example
			\", { \"metric\": \"memory \" })" }, { "title": "Network Latency Increase", "summary": "Average response time increased from 120ms to 180ms.", "status": "Under investigation", "badgeText": "Warning", "priority": "warning", "timestamp": "16 min ago" }] } } </oraInfoDisplay>
Change List	Use a changeListWidget to compare a few key metrics between a previous value and a current value. Add a short subtitle to explain what's being measured, keep the rows easy to read, and only include a message if there's a meaningful insight or anomaly worth calling out.	• **Use it for:** current-versus-previous metric comparisons • **Required:** `displayType`, `items` • **displayType:** `percentage`, `raw`, or `currency` • **items[]:** `title`, `currentValue`, `previousValue` • **Optional:** `subtitle`, `messages` Values should stay numeric. Formatting comes from displayType, not from embedding percent signs or currency symbols inside the numbers.	<oraInfoDisplay key="campaign-change-list"> { "patternId": "changeListWidget", "config": { "subtitle": "Campaign performance comparison", "displayType": "percentage", "items": [{ "title": "Email Open Rate", "currentValue": 28.2, "previousValue": 24.5 }, { "title": "Click-through Rate", "currentValue": 4.6, "previousValue": 3.8 }, { "title": "Conversion Rate", "currentValue": 1.9, "previousValue": 2.1 }], "messages": ["Conversion rate drop may indicate landing page issues"] } } </oraInfoDisplay>
Chart	Use a chartWidget to visualize a simple trend, comparison, or proportion in a way that's easy to understand at a glance. Choose the chart type that fits the story, use clear labels, and include one or two	• **Use it for:** trends, comparisons, or proportions • **Required:** `type`, `data` • **type:** `line`, `bar`, or `pie`	<oraInfoDisplay key="sales-trend-chart"> { "patternId": "chartWidget", "config": {

Type of Widget	Sample Prompt	Details	Example
	short insights underneath to help explain what the chart is showing.	• **data.labels[]:** x-axis labels or categories • **data.datasets[]:** each dataset needs `label` and numeric `data[]` • **Optional:** `insights[]`	`"type": "line", "data": { "labels": ["January", "February", "March", "April", "May"], "datasets": [ { "label": "Sales Revenue", "data": [10000, 25000, 15000, 40000, 30000] } ] }, "insights": [ "Revenue peaked in April", "Steady growth trend overall" ] } } </oraInfoDisplay>`
Record	Use a recordWidget to show a structured record as a simple form that can either be reviewed or edited. Include clearly labeled fields, choose field types that match the data, and keep the layout straightforward so it feels comfortable for someone filling in or checking details.	• **Use it for:** forms and read-only detail views • **Required:** `id`, `fields[]` • **Optional:** `readOnly` • **Field types:** `text`, `textarea`, `number`, `date`, `select`, `system` • **Field fields:** `id`, `type`, `value`, plus `label` and `options` when applicable In editable mode, the runtime automatically submits the form through an `ora.Agent(<oraFormSubmit ..` command. You don't configure a separate submit command in the widget itself.	`<oraInfoDisplay key="employee-form"> { "patternId": "recordWidget", "config": { "id": "employee_form", "readOnly": false, "fields": [{ "id": "full_name", "type": "text", "label": "Full Name", "value": "John Doe" }, { "id": "employee_id", "type": "number", "label": "Employee ID", "value": 12345 }, { "id": "department", "type": "select", "label": "Department", "value": "engineering", "options": [{ "value": "engineering", "label": "Engineering" }, { "value": "sales", "label": "Sales" }] }, { "id": "session_context", "type": "system", "value": { "conversation_id": "abc123" } }`

Type of Widget	Sample Prompt	Details	Example
			] } } </oraInfoDisplay>

Multi Record	Use a multiRecordWidget to present structured information in a compact table with clear columns and rows. Keep the values short and scannable, use badge-style cells for statuses when helpful, and only add row actions if the experience should explicitly support taking action from the table.	• **Use it for:** tabular record lists
		• **Required:** `cols[]` and `rows[]`
		• **Column forms:** plain string, or object with `label` and optional `showOnExpand`
		• **Row cells:** plain string, or badge object with `type`, `text`, and `priority`
		• **Optional row actions:** `action` button and `drillDownAction` on the first column
		• **Optional:** `subtitle` for table context and accessibility
		The `showOnExpand` columns only appear when the widget is focused. This is useful for secondary columns that should stay hidden in compact mode.
			<oraInfoDisplay
			key="pending-approvals-
			table">
			{
			"patternId":
			"multiRecordWidget",
			"config": {
			"subtitle": "Pending
			approvals",
			"cols": ["Request ID",
			"Type", "Status"],
			"rows": [
			{
			"cells": [
			"REQ-101",
			"Budget",
			{ "type": "badge", "text":
			"PENDING", "priority":
			"warning" }
			],
			"action": {
			"text": "Review",
			"command":
			"ora.Invoke(\"reviewRequest
			\", { \"id\":
			\"REQ-101\" })"
			},
			"drillDownAction":
			"ora.Invoke(\"openRequest
			\", { \"id\":
			\"REQ-101\" })"
			},
			{
			"cells": [
			"REQ-102",
			"Travel",
			{ "type": "badge", "text":
			"PENDING", "priority":
			"warning" }
			],
			"action": {
			"text": "Review",
			"command":
			"ora.Invoke(\"reviewRequest
			\", { \"id\":
			\"REQ-102\" })"
			}
			}
			]
			}
			}
			</oraInfoDisplay>

Sankey	Use a sankeyWidget to show how volume or activity flows from one stage to another across a process.	• **Use it for:** flows between stages or states	<oraInfoDisplay key="support-flow-sankey"> {

Type of Widget	Sample Prompt	Details	Example
	Define clear node names, keep the flow easy to follow from left to right, and use realistic values so the visualization tells a simple story about where things are going.	• **Required:** `nodes[]`, `edges[]` • **nodes[]:** `id`, `name` • **edges[]:** `source`, `target`, `value` Node IDs are numeric and edges point to those IDs. Keep values positive and keep the flow easy to follow.	`"patternId": "sankeyWidget", "config": { "nodes": [{ "id": 0, "name": "Web Visitor" }, { "id": 1, "name": "Web Chat" }, { "id": 2, "name": "AI Chat Bot" }, { "id": 3, "name": "Resolved by Bot" }, { "id": 4, "name": "Transfer to Human" }, { "id": 5, "name": "Resolved by Human" }], "edges": [{ "source": 0, "target": 1, "value": 1000 }, { "source": 1, "target": 2, "value": 1000 }, { "source": 2, "target": 3, "value": 700 }, { "source": 2, "target": 4, "value": 300 }, { "source": 4, "target": 5, "value": 300 }] } }` `</oraInfoDisplay>`

Interactive widgets

For interactive widgets, use `ora.Invoke("actionCode", payload)`, which represents the app-defined action path.

Best Practices for Nodes in Workflows

Here are some best practices related to creating nodes for workflows, to be used in agentic apps:

- Start by deciding which app stages your workflow should handle.
- Branch early on `oraMessageHint` so each stage stays simple.
- For `InitDisplay`, return display output and optionally include actions or `oraComms` when that improves the startup experience. For `InitCommunications`, return `oraComms`. For `InitActions`, return actions.
- Use `OraAppContext` and `OraUserContext` whenever the app experience depends on who the user is or what object the app is currently showing.
- If a node needs the current question or a derived prompt, pass it explicitly instead of assuming a downstream node will reconstruct it for you.
- Test the exact stage you're writing before including that workflow into the app.

Build Agentic Apps Using App Builder

You can create agentic apps from the Apps tab in AI Agent Studio. To build agentic apps, do these steps:

1. Go to AI Agent Studio and open the Apps tab.
2. Select **Add** and enter the name, code, and description for your agentic app.
3. Select **Update** to navigate to the App Builder.
4. Create an agentic app using any of these options:
    a. Choose one of the example agentic apps and use its framework to build your own.
    b. Enter your app requirements in the **Ask Oracle** search bar to explore available apps and agent teams that can assist you.
    c. Select **Start from scratch** to build your agentic app from the ground level. Select one of the example agentic apps displayed and use its framework to edit further and build your agentic app.
5. If you've chosen to use an example app or take help from Ask Oracle, you see the app's framework that you can further edit.
6. If you've chosen to build your app from scratch, you can start adding agent teams and other required artifacts to build your agentic app.
7. Enter the values in App Settings.

Add Agent Teams

Agent teams are one of the primary components of your agentic app.

1. In the App Builder select **Add Section** and search for the agent team to include in your agentic app.

    > **Note:** Only the published workflow agent teams that are enabled for using in apps are available for selection.

2. You can also assign custom agent teams for **Ask Oracle** and **Summary** sections.

    If you're selecting a custom agent team for Summary, enter the **Summary Prompt**.

Add Communications

Communications enable your app to send emails and messages according to predefined templates.

Prerequisite

Define the templates for communication.

1. In the App Builder select and add a document template.
2. Select the format for the template.

    You can add templates of type PowerPoint, PDF, email, or text.
    - PowerPoint: Add a PowerPoint template here. You can also define each slide separately. While creating the slide, specify the title for each slide, or select **Allow agent to choose my title**.
    - Email: Select **Allow agent to generate recipients**, **Allow agent to generate the subject** and add sections which can be filled by the agent when you select **Allow agent to fill in this section**. You can also include additional information in the email template.

- PDF: Select **Allow agent to generate the title** and **Allow agent to fill in this section** to add the title and content for the PDF template.
- Text: Select **Allow agent to fill in this section** to add content for the template of type Text.

Add the communication

1. In the App Builder select **Add Communication** and add the title and description for the communication.
2. Select the agent the communication is applicable for.
   a. Select the template that the agent must use when sending messages.
   b. Enter the action to be performed in **Action Text**. For example, Send Email, Send Alert, Send to Agent.

      Optionally, you can also select an agent to send the communication to in **Target Agent**.

Add Actions

Create actions that are to be processed in the course of app flow, including requiring human approval for any step. Actions and commands work in a simple sequence.

1. A widget or another UI element triggers a command.
2. The command identifies which action to run.
3. The action starts.
4. The action runs its steps in order.
5. Those steps perform the real behavior.

Here, the command is the trigger, action is the specified workflow, and the steps are the ordered instructions within the workflow.

1. Select  and add a new action.
2. Enter the action code, display name, and description.
3. Select **Add Step** and choose the action.

Step Name	What it's for
Keep Action Available in UI	Keeps the original action available on UI even after it runs, instead of treating it as a one-time action.
Navigate to App	Opens another app. - Enter the code of the app to navigate to. - Select **Pass Payload to Context** to send the payload. - Enter the context to send the payload to.
Send Agent Command	Sends follow-up work to an agent. Use this when the action should ask an agent to do something next. - Select **Pass Payload to Context** to send the payload. - Enter the command.

Step Name	What it's for
Refresh Agents	Refreshes one or more agent displays. Use this when the user should immediately see the updated content. Select the agent that's to be refreshed.
Switch App Context	Changes the current app context, optionally refreshing the app afterwards. - Select **Use Payload as Context** if needed. - Enter the context. - Select **Reload App** to refresh.

Publish the Agentic App

After adding the relevant artifacts and testing your app in the App Builder, publish your app. Users can view the published agents from the AI Agents page. To access this page, go to Me → Quick Actions → Show More → AI Agent Studio → AI Agents. If you're using **Home with Ask Oracle** as your home page layout, search for the **AI Agents** page.

> **Note:** Make sure that your users have access to interact with the AI agents. For information, see *How can I give users access to AI agents?*

12 Allow External Access to Fusion Applications Agents

Enable Applications to Access Fusion Applications Agents

You can access the agents of Oracle Fusion Cloud Applications from other Oracle applications and from external applications, using the `/invokeAsync` API. The access is based on the role assigned to you, and whether that role can access agent teams. The roles that can access an agent team are specified in the Security tab when creating or editing the agent team in AI Agent Studio.

Access from Other Oracle Applications

To use `/invokeAsync` API, you must authenticate through Oracle Cloud Infrastructure Identity and Access Management (OCI IAM) using an OAuth 2.0 bearer token. For agent operations to function, integration must be bidirectional. Both systems must securely access each other's services.

As an administrator, you create the confidential application in Fusion Applications IAM, representing the other Oracle application service. Then create a matching confidential application in the other application service, representing the AI Agent Studio's service. These two applications form the trusted link for the two-way communication.

Set Up Two-Way Trust Between Applications

1. Create a confidential application in Fusion Applications IAM:

   This application acts as a representation of the external Oracle application in Fusion Applications IAM. After creating the confidential application, enable access to the AI Agent Studio service so the application can call the `/invokeAsync` API exposed by this service.
   - Identify the identity domain (authorization server) where you'll create your confidential application.
   - Specify OAuth grant types. For example, a 2-legged OAuth.
   - Define the access scope.
   - Generate the access token and use it to make REST API calls.

   For more information, see *Configure OAuth Using the Fusion Applications Identity Domain*.
2. Create a confidential application in the external Oracle application's IAM:

   Similar to the previous confidential application you created, create another confidential application that acts as a representation of the Fusion Application service in the external Oracle application's IAM. After creating the confidential application, configure it to allow AI Agent Studio to call the appropriate service endpoints.

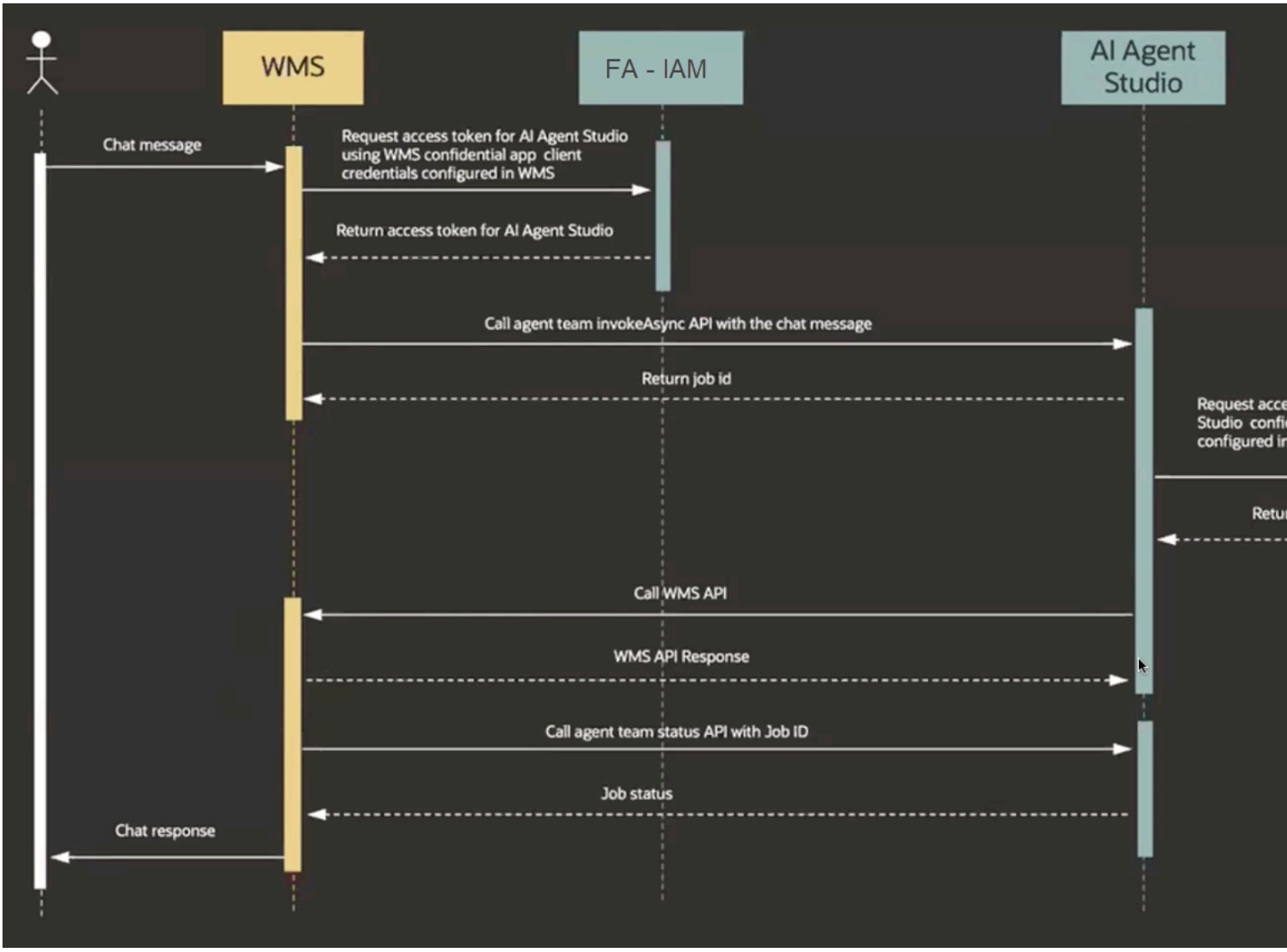
   For more information, see *Configure OAuth Using the Fusion Applications Identity Domain*.

3. Register the confidential application created in the external Oracle application's IAM (from step 2) in AI Agent Studio:
    a. Go to **AI Agent Studio** and open the Credentials tab.
    b. Go to the Data Source Applications subtab and add the details of the external confidential application.
        - Base URL: Enter the base URL of the external API
        - IDCS URL: Enter the URL of the Fusion Applications IAM, to enable AI Agent Studio to get the OAuth token
        - Scope: Specify what access needs to be given whenever that OAuth token is retrieved.
        - Public and Private keys: Enter the keys for signing authentication requests. Used by IDCS to verify that the request is from a trusted and authorized source.
4. Create a business object to access the external Oracle application:
    a. In **AI Agent Studio**, open the Business Object tab.
    b. Add a new business object with the **Resource Type** as **Other Data Source Application**.
    c. From the **Oracle Data Source Application** list, select the data source application that you've registered in step 3.

Access Flow During Runtime

At runtime, the external Oracle application obtains an access token from Fusion Applications using the configured credentials. The application then uses this token to call the `/invokeAsync` API in AI Agent Studio, enabling access to the agent team.

This illustration shows the interactions between Warehouse Management System (WMS), which is an Oracle application outside Fusion Applications, Fusion Applications IAM, AI Agent Studio, and WMS IAM.



- The user interacts with the WMS application and sends a chat message.
- WMS requests an access token from Fusion Applications IAM using its configured credentials.
- WMS calls the `/invokeAsync` API in AI Agent Studio, passing the user's chat message and access token.
- AI Agent Studio returns a Job ID to WMS, which WMS uses to track progress and get the response.
- To call WMS API, AI Agent Studio requests an access token from WMS IAM.
- AI Agent Studio calls the WMS API using the access token.
- WMS returns the API response and requests for the job status.
- AI Agent Studio returns the job status.
- Final chat response is processed and delivered to the user.

Access from External Applications

Use `/invokeAsync` API to access the agents of Fusion Applications. For more information, see *Agent Team REST Endpoints*.

13 Collaborate with AI Agents Across Platforms Using Agent2Agent (A2A) Protocol

Collaborate with AI Agents Across Platforms using Agent2Agent (A2A) Protocol

The published AI agents in Oracle Fusion Cloud Applications can collaborate with AI agents in other platforms using the A2A protocol. A2A is an open, industry-backed protocol that enables AI agents to discover, understand, and call other agents in a standardized way. In AI Agent Studio, A2A serves as the foundation for seamless cross-agent collaboration across Fusion Applications and external ecosystems, eliminating the need for custom point-to-point integrations.

Here are the key features of A2A:

- A standardized way to describe agents through an Agent Card
- A standard request-and-response model for agent interactions
- A shared lifecycle framework for managing long-running and asynchronous tasks

Here's an example scenario for using A2A. If your organization uses an Oracle AI agent for sales managers, that agent can reach out to a third-party agent to get travel options and cost quotes when considering the sales calls. Similarly, third-party platforms can call Oracle AI agents using A2A. Another example scenario for using A2A is when your organization has a general purpose AI chat assistant for employees. You can reach out to subject matter expert agents in Oracle Fusion Applications when customers ask about HR Sickness and Absence Policy or Vacation Bookings. Here's how the interaction using A2A works:

1. Each agent publishes an Agent Card that describes the agent to the third-party platform.
2. Third-party platforms can then call the Fusion Applications agent in a consistent and secure way.

The Fusion Applications A2A API provides clients a practical way to discover, call, poll, and stream workflow-backed agents using A2A-style endpoints and method names. As an authenticated, client-facing API, it offers the following key properties:

- A workflow-backed agent model
- A task-oriented asynchronous interaction model
- A pragmatic subset of broader A2A capabilities

Components in A2A

Here are the core components of A2A interactions:

- Agent Catalog or Agent Registry
- Agent Card

Agent Catalog (Registry)

The Agent Catalog, also referred to as the Agent Registry, acts as a searchable directory of available agents. It doesn't call agents or manage their internal logic.

Here are the key responsibilities of the agent catalog:

- Maintain the list of available agents.
- Index agent metadata for search.
- Enforce access rules.
- Return agent details, that is, Agent Cards.

Agents can be discovered in two main ways:

- Listing Available Agents: By passing an empty query, a client can retrieve a list of all agents it's allowed to access.
- Keyword Search: The agent registry supports keyword-based search across agent metadata. Typical search scenarios include finding an agent by name or description. The search results return agent references, that is, each result points to the agent's Agent Card.

Agent Card

An Agent Card is a machine-readable JSON document that describes an agent. It's the core building block that makes discovery and interoperability possible in A2A. You can consider an Agent Card as an agent's digital profile or business card.

An Agent Card allows a client to answer the following questions without prior knowledge of the agent:

- What's this agent?
- What capabilities or skills does it provide?
- How do I call it?
- What authentication is required?

Call Agents

After an agent is discovered and its Agent Card is retrieved, calling the agent follows a consistent pattern - using asynchronous calls or streaming. Streaming includes `message/send` and `message/stream` methods.

1. Discover the agent using the registry.
2. Retrieve the agent card.
3. Authenticate according to the agent card's requirements.
4. Call the agent using the A2A invocation API.

Authentication and Access for A2A

When using A2A, all endpoints require an authenticated caller. Here are some key points about the Fusion Applications A2A surface:

- It's externally exposed to authorized clients and integrations.
- It's not anonymous or publicly open without authentication.
- It's authenticated for discovery, invocation, status retrieval, and streaming.

The agent card advertises the `Bearer` authentication.

Authorization Behavior

All endpoints require authentication, but clients must also understand the distinction between discovery and execution behavior. A client might be able to discover an agent entry, but actual processing or status access might still depend on user-specific authorization. Status retrieval is user-scoped, and authorization to process is enforced at processing time.

Clients must therefore treat discovery as a way to find candidate agents, and processing as the point where final access is confirmed.

Auth, OAuth, Bearer Tokens, and Authorization

Authentication means proving the identity of the caller. Here `oAuth` is the standard token-based mechanism used to obtain access credentials. A `bearer` token is the token the client sends on the request, usually in the `Authorization` header. Authorization means determining what the authenticated caller is allowed to access or do.

A high-level overview of the authentication process includes the following:

- Clients authenticate by presenting a bearer token.
- That bearer token is obtained through an OAuth flow.
- Authentication gives access to the API.
- Authorization determines whether the client can run a workflow or retrieve a task.

A2A API Overview

The API consists of three endpoints:

1. `GET /orchestrator/a2a/v1/{workflowCode}/.well-known/agent.json`
2. `POST /orchestrator/a2a/v1/agents/search`
3. `POST /orchestrator/a2a/v1/{workflowCode}`

Together, these endpoints support the following actions:

- Agent specific discovery
- Registry-style agent search
- Streaming and nonstreaming invocation and task retrieval

Endpoint	When to use	Request	Response and how to use it	Notes
Agent Card Discovery `GET /orchestrator/a2a/v1/{workflowCode}/.well-known/agent.json`	Use this endpoint when the client already knows the target workflow code or when registry search has already returned a workflow-specific agent-card URL.	The request is identified by: `workflowCode`: the published workflow to discover No request body is required.	The response is an agent card for the specified workflow-backed agent. Clients must expect the card to provide the following details: • Agent identity information • Version information • Provider information • Authentication requirements • Capabilities such as streaming support • Default input and output modes • Skills associated with the agent **How to use the response** Clients must use the card to do these actions: 1. Confirm they have the correct agent. 2. Inspect the advertised authentication and capability model. 3. Determine the invocation path for subsequent calls.	• The card is agent-specific rather than host-wide. • The card model is intentionally narrower than a richer full-specification A2A card.
Registry Search `POST /orchestrator/a2a/v1/agents/search`	Use this endpoint when the client needs discovery rather than direct lookup. Here are some typical scenarios: • Browsing available agents • Searching by name or related query text • Paging through results	The request body supports the following options: • `query`: optional search text • `skip`: optional pagination offset • `top`: optional page size limit	The response is a registry-style result set containing these details: • Version metadata for the registry response • A collection of matching resources • A total count Each resource points to an agent card. **How to use the response** Here's the recommended client flow: 1. Call registry search. 2. Select a matching result.	• This is a Fusion-specific registry-style endpoint, not a generic standards-level registry service. • Results are agent-backed. • Search success must not be treated as a guarantee that subsequent execution will be authorized in every case.

Endpoint	When to use	Request	Response and how to use it	Notes
			3. Fetch that result's agent card. 4. Call the selected agent.	
Nonstreaming Invocation **`POST /orchestrator/a2a/v1/{workflowCode}`**	Use this endpoint when the client wants a standard request/response interaction rather than a Server-Sent Event (SSE) stream. This endpoint supports the following operations: • Starting asynchronous work from a message • Starting or resuming task-oriented work • Polling the latest state of an existing task	The request body supports the following request patterns: • **`message/send`** • **`message/stream`** • **`tasks/send`** • **`tasks/get`** **Request patterns** **`message/send`** Use **`message/send`** to begin asynchronous work from a conversational message. Typical request content includes these details: • A message object • Message parts • Optional context information • Optional metadata **`message/stream`** The response is returned as an SSE stream. **`tasks/send`** Use **`tasks/send`** to start or continue task-oriented execution. Typical request content includes these details: • A task identifier when continuing or resuming work • A message payload • Optional context information • Optional metadata **`tasks/get`** Use **`tasks/get`** to retrieve the latest state of an existing task.	The endpoint returns a JSON-RPC-style response whose result is task-oriented. Clients must expect task-like responses representing these states: • **`Submitted`** • **`Working`** • **`Input-required`** • **`Completed`** • **`Failed`** **`Completed`** and **`waiting`** tasks might include output artifacts. For **`message/stream`**, clients must expect these responses: • An initial event representing the initial task snapshot. • Additional events as task state changes. • Completion of the stream when execution finishes. **How to use the response** Here's a common client pattern: 1. Call **`message/send`** or **`tasks/send`**. 2. Receive a task-shaped response 3. Continue polling with **`tasks/get`** until the task reaches a terminal state. Use **`message/stream`** when streaming is preferred over polling. Prefer nonstreaming invocation plus **`tasks/get`** for these situations:	• Even **`message/send`** is task-oriented in practice. • Clients must expect asynchronous behavior rather than an immediate final answer. • The **`message/stream`** behavior can be considered as repeated task snapshots over SSE. • Clients must not assume a richer event-native A2A stream contract beyond the task-states.

Endpoint	When to use	Request	Response and how to use it	Notes
		Typical request content includes these details: • The task identifier • Optional history preferences	• Polling is simpler for the client. • The integration doesn't need live incremental updates.	

API Task Lifecycle

The API is asynchronous by design. The typical client lifecycle includes these steps:

1. Identify the target agent through direct card lookup or registry search.
2. Call API using `message/send`, `tasks/send`, or `message/stream`.
3. Receive a task-oriented result.
4. Continue through polling or streaming until the task completes.
5. Read final output from the resulting task state and artifacts.

Here are the available task states:

- `submitted`
- `working`
- `input-required`
- `completed`
- `failed`

Clients must be prepared for a task to enter the `input-required` state before reaching completion.

Input and Output Expectations

Here are the input and output expectations for the API.

Inputs

The API is primarily text-oriented. Clients must assume the following inputs:

- Text is the primary supported input mode
- The most important request content is the message text
- Metadata might be included when additional execution context is needed

Outputs

The API returns task-oriented outputs. Clients must expect the following outputs:

- Task identifiers
- Task state
- Timestamps
- Output artifacts when work completes or pauses for input

- Optional history when requested

Usage Guidance

For most client integrations, here's the recommended usage pattern:

1. Use registry search when discovery is needed.
2. Fetch the agent card for the selected agent.
3. Use `message/send` for standard asynchronous invocation.
4. Use `message/stream` when live progress updates are preferred.
5. Use `tasks/get` for polling-based status retrieval.

Error Handling and Client Expectations

Clients must handle the following classes of errors:

- Authentication failures
- Workflow-not-found cases
- Invalid method usage
- Task lookup failures
- Authorization failures during processing or status retrieval

Clients must also assume that invalid or incomplete requests might fail when validating the request. Task retrieval might fail if the task doesn't belong to the specified agent or accessible user context.

Scope and Constraints

The API is intentionally narrower than the broader A2A protocol. Here are some of the important constraints:

- The API is workflow-backed
- The primary response model is task-oriented
- Streaming is snapshot-based over SSE
- Discovery is implementation-specific
- Richer nontext and multipart input handling is limited
- The broader A2A feature set is only partially implemented

14 AI Agent Studio FAQs

Which large language models (LLMs) are currently used or supported for AI agents?

Oracle Fusion Cloud Applications support several LLMs from providers such as, Cohere, Meta, and OpenAI, for powering AI features including content authoring, summarization, and interactive Q&A. For the most current list of supported models, see *What Large Language Models (LLM) are in use or supported by Fusion SaaS Applications? (KB796229)*.

Which scheduled processes do I need to run to make sure the features of AI Agent Studio work as expected?

Here are the scheduled processes required to ensure AI Agent Studio features work as expected.

Feature	Scheduled Process	Purpose	More Details
Access Requirements for AI Agent Studio	Import Resource Application Security Data Import User and Role Application Security Data	Imports resources and data from LDAP, and transfers the necessary information into the security tables of Fusion Applications.	*Access Requirements for AI Agent Studio*
AI Agent Studio Assistant	Index AI Agent Studio Assistant Documents Index AI Agent Studio Assistant Objects and Attributes	Helps you get answers to questions about existing agents, search for agents, tools, and topics using natural language, and receive AI-powered suggestions for relevant resources to use in your agents.	*Access Requirements for AI Agent Studio*
Monitor and Evaluate AI Agents	Aggregate AI Agent Usage and Metrics	Aggregates the metrics that are displayed in the Monitoring and Evaluation tab of AI Agent Studio.	*Monitor and Evaluate AI Agents*
Connectors	Content Intelligence SharePoint Connector	Connects your agents to accurate, context-aware, up-to-date source information. For example, using a SharePoint connector, you can import SharePoint files into articles as attachments, or a WebCrawl	*Schedule Content Intelligence Synchronization Process*

Feature	Scheduled Process	Purpose	More Details
		connector to contact selected sites to update the search index.	

Can I use any agent in my agent team or workflow?

No, you can't use just any agent in your agent team or workflow. You can use predefined agents and reuse agents created from the Agents tab in AI Agent Studio in other agent teams and workflows. However, if you create an agent directly within an agent team or workflow, that agent can't be reused and will only be available in the specific agent team or workflow where it was created.

When should I use a single-agent flow vs a multiagent process?

- Use a single-agent process for straightforward business requirements, such as retrieving answers from a single policy document.
- Opt for a multiagent process when dealing with more complex scenarios. For example, when multiple documents need to be parsed, or when the answers vary based on the user asking the question. In these cases, multiple agents are used, and the user's question is directed to the appropriate worker agent for processing.

What's the difference between the Copy Template and Use Template options in AI Agent Studio?

In AI Agent Studio, you can create agents from preconfigured templates using **Copy Template** or **Use Template**. The main difference between the two options is the order in which you rename the agent team, and update editable artifacts.

- Copy Template: Before creating the agent team, you can add a suffix to its name. The same suffix is also added to other editable artifacts in the agent team. After you add the suffix, the canvas opens with the agent team components, where you can modify editable artifacts. You can also edit the agent team details using **Agent Team Settings** ( ).
- Use Template: The Use Template option initially displays the general settings where you can rename the agent team and make other edits. After you create the agent team, the canvas opens with the agent team components, and you can modify the editable artifacts as needed.

For information about modifying artifacts in agents created from preconfigured templates, see *Can I edit a preconfigured agent team?*

Can I edit a preconfigured agent team?

You can't directly modify the components in a preconfigured agent team template. However, you can create copies of individual components (such as agents, tools, and topics), edit them, and then add the edited versions to your agent team after removing the original components. For example, to edit a topic that's part of a preconfigured agent team template you're using, do these steps.

1. Go to **AI Agent Studio**, and select the preconfigured agent team template you need, either using **Copy Template** or **Use Template**.
2. From the selected template, remove the topic you want to edit.
3. Go to the Topics tab, and search for the topic you removed.

   > **Note:** If you've removed an agent, go to the Agents tab. Similarly, for tools, use the Tools tab.

4. Select **Duplicate** ( ) and create a copy of this topic.
5. Edit the copied topic as needed and add it back to your agent team.
6. Save and publish your agent team.

Can I migrate any agent team from one environment to another?

Yes. As an administrator, you can migrate (import) any individual agent team from one environment to another. However, if you've imported an agent team that belongs to a product family you don't have access to, you can't view the imported agent team in the new environment. For example, you're an Oracle Fusion Cloud Human Capital Management (HCM) administrator and you import an agent team that belongs to Oracle Fusion Cloud Supply Chain & Manufacturing (SCM). If your user doesn't have access to SCM product family in AI Agent Studio, you won't be able to view the imported agent team in the target environment.

For more information about migrating agent teams, see *Migrate AI Agents from One Instance to Another*.

How do I display a welcome message in a chatbot agent?

You can configure your welcome message and any interactive starter questions for your chatbot agent in AI Agent Studio. These messages are displayed during the user's first interaction. You can also define follow-up questions, which are perfect for keeping the conversation going by suggesting logical next steps after an initial inquiry is resolved.

These settings can be configured while creating an agent, or editing it later.

- If you're creating a new agent, add the agent team from the **Agent Teams** tab, and go to the **Questions** subtab to configure the welcome message. For more information, see *Create Custom AI Agents of Type Supervisor* and *Create Custom AI Agents of Type Workflow*.
- If you're editing an existing agent team, go to the **Agent Teams** tab and select **Edit** to modify the agent team. Select **Agent Team Settings** and then open the **Questions** subtab to configure the welcome message. For more information, see *Edit an Agent Team*

> **Note:** If you see **View** instead of **Edit** , it indicates that the agent team is a preconfigured one, which can't be edited or directly modified. For more information, see *Can I edit a preconfigured agent team?*

How do I make agents respond faster?

AI agent response time is directly linked to the combined length of input and output text, measured in tokens for each transaction. For example, 100 tokens is roughly 75 words, though this ratio varies by model, writing style, and language. By being deliberate with your instructions, you ensure that your agents reason clearly and avoid unnecessary processing.

Here are some recommendations to help your agents respond faster.

- Edit summarization prompts to include only essential instructions relevant to your specific use case. The default prompts for supervisor and workflow agents are broadly written to cover many scenarios, so remove any general or redundant sections to streamline processing and improve response speed.
- Minimize the use of input and output tokens with concise prompts. Use retrieval-augmented generation (RAG) instead of overloading the context window, and set specific output length limits.
- Specify response length to avoid overly verbose answers.
- Use smaller, specialized agents that work in parallel and cache static instructions.
- Include only relevant information. Remove unnecessary details from context and documents to streamline processing.
- Consider a multi-agent approach, where specialized agents handle distinct tasks, such as research, coding, or Q&A, instead of loading all information into a single prompt.

By carefully selecting relevant context and breaking complex tasks into smaller, focused agents, you can improve both the speed and efficiency of your AI agents.

How do I send an email using AI Agent Studio?

You can send emails using Supervisor and Workflow agent teams in AI Agent Studio.

- To enable email functionality in a supervisor agent, configure an email tool. For more information, see *Add Email Tool*.
- To send emails from a workflow agent, use an email node.

How do I track the usage of AI Agents created in AI Agent Studio?

You can use the Monitoring and Evaluation tab within the AI Agent Studio to track real-time interactions, analyze usage patterns, and identify errors to improve agent accuracy and performance. For more information, see *Monitor and Evaluate AI Agents*.

How can I best reduce the risk of my agent showing hallucinations?

Large language models (LLMs) can unintentionally introduce extra details when they lack sufficient information.

To mitigate this issue, consider these strategies:

- Supply the necessary knowledge: Integrate relevant sources such as business objects, REST APIs, or RAG documents, ensuring the LLM has access to accurate and up-to-date information.
- Use explicit prompt instructions: Clearly instruct the LLM not to add or infer information beyond what's provided, reducing the risk of embellishment or fabrication.

How do I hide or restrict access to AI Agent Studio for specific users?

To restrict a user's access to AI Agent Studio, remove any related custom roles from the user's account.

1. Go to the Security Console, open the Users tab and search for the specific user.
2. From the search results, select the user, and edit the user account details.
3. In the Roles section of the Edit User Account page, select the custom roles associated with AI Agent Studio, and remove them using the × icon. These roles were added to the user account when access was provided. For a list of these roles, see *Provide Access to Configure AI Agents in all Products*.
4. Save your changes.

How can I provide access to an AI Agent for all employees?

To enable access to an AI agent for all employees, create or edit a role that includes the required privileges to access that AI agent, and assign that role to a common employee job role.

1. Go to the Security Console.
2. To use a new custom role, create it. To use an existing custom role, search for the custom role and edit it.

   Note: Make sure that permission groups are enabled for the custom role.

3. Go to the Function Security Policies page and select **Add Function Security Policy**.
4. Add the Access Intelligent Agent Chat (HRC_ACCESS_AI_AGENT_CHAT_PRIV) privilege to the role and save it.
5. Go to the Role Hierarchy page and open the **Roles and Permission Groups** tab.
6. Add the FAI GenAI Agent Runtime Duty (ORA_DR_FAI_GENERATIVE_AI_AGENT_RUNTIME_DUTY) role.
7. Save the custom role and assign this custom role to the appropriate job roles.

All employees who have been assigned the job role can now access the AI agent.

Why can't I find the standard Employee role in the Security tab of Agent Team Settings?

The standard Employee role isn't available in the Security tab because it's a predefined role and can't be modified to add new privileges or roles. Although permission groups can be enabled for predefined roles, you can't edit those roles to add additional privileges.

If you see an Employee-like role available in the Security tab, it's typically a custom role created by copying the predefined Employee role. Predefined roles generally have names that begin with the ORA_ prefix.

Can I rename the AI chat label to something other than Ask Oracle, or change the design of the chat panel?

No, you can't change the background image or rename 'Ask Oracle' within the active chat window or AI Agent Studio preview.

What's the difference between editing a topic versus editing a system prompt?

When you update a topic, the changes are available across agents because you can reuse topics across agents. When you're updating a system prompt, the changes are available only for the agent being edited.

Tip: We recommend updating topics instead of prompts, so that you can reuse the topics.

What file types, limits, and processing capabilities are supported in AI Agent Studio?

AI Agent Studio provides several options for processing files, including chat uploads, Document Tool ingestion, and background processing with Document Processor.

You can upload these types of files:

- Documents: PDF, DOCX, PPTX
- Text and structured text: TXT, HTML, MARKDOWN, JSON, XML
- Spreadsheets and tabular data: XLSX, CSV
- Images: JPG, PNG
- Multimedia: WAV, MP3, M4A, AAC, OGG, Opus, WebM, FLAC, MP4
- Archives: ZIP

  Note: ZIP files are supported only when using RAG Document Tool, and multimedia files aren't supported for RAG Document Tool.

This table outlines the recommended usage patterns for chat uploads, Document Tool, and the Document Processor. By understanding these capabilities, limitations, and best practices, you can optimize file processing performance and ensure reliable handling of uploaded content across different processing flows.

Recommended patterns

Goal	Option to use	Best suited for	Key limitations	Can you choose LLM?
Work with a file during a chat conversation	Chat file upload	Quick, one-time tasks such as summarizing a document, extracting a few values, reviewing an image, or analyzing a small set of files.	A maximum of five files can be uploaded per conversation, with a combined size limit of 50 MB across all uploaded files.Additionally, each	No, the default LLMs are used for image and multimedia processing.

Goal	Option to use	Best suited for	Key limitations	Can you choose LLM?
			individual multimedia file mustn't exceed 25 MB.	
Build reusable knowledge for retrieval	Document Tool	Guides, FAQs, handbooks, procedures, and other reference content that should be searchable and reusable by the agent over time.	A single file size limit of 25 MB.	No, the default LLMs are used for image and multimedia processing.
Process business object attachments in the background	Document Processor	Asynchronous processing of files attached to business objects, automated ingestion, workflow-driven document handling, scanned PDFs, image-heavy PDFs, long DOCX files, and dense text files.	A single file size limit of 25 MB.	• No, LLM can only be configured when LLM-based data extraction with Document Schema is enabled. • When it isn't enabled, the default LLMs are used for image and multimedia processing.
Extract structured fields from files	Document Processor with Document Schema	Deterministic field extraction, structured output, and cases where only specific values need to be extracted from a document.	• A Document Schema is required. • A single file size limit of 25 MB.	• Yes, when LLM-based data extraction with Document Schema is enabled. • For more information, see *Add a Document Schema to a Document Processing Node of Workflows*.

Best Practices

Follow these best practices to improve reliability, reduce processing failures, and choose the right file-handling approach for your use case.

- Keep files small and focused

  Large files can fail even when they are within the recommended size limits. File size is not the only constraint; extracted text, embedded images, OCR output, tables, and token expansion can also affect processing. Split large or dense files into smaller documents, especially long reports, large spreadsheets, scanned PDFs, and image-heavy PDFs.

- Use clear, readable source files

  Fully scanned PDFs and image-heavy PDFs can be harder to process. Use text-based PDFs when possible. For scanned documents, make sure pages are upright, readable, high contrast, and not blurry or skewed.

- Use chat uploads for lightweight tasks

  Use chat uploads for short, one-time tasks such as summarizing a small document, extracting a few values, reading an image, or reviewing a small set of attachments. Use another processing option when the content must be reused, searched, or processed as part of a background workflow.

- Use Document Tool for reusable knowledge

  Use Document Tool when content should be searchable and reusable over time, such as guides, FAQs, handbooks, procedures, and reference documentation. Keep source documents focused by topic so retrieval returns more relevant results.
- Use Document Processor for background processing and extraction

  Use Document Processor when files are attached to business objects, need asynchronous processing, or require structured extraction. For field extraction, provide a Document Schema and clear instructions so the processor returns only the required values.
- Avoid sending full extracted content to the LLM

  Extracted content can be much larger than the original file and may increase cost, slow processing, or exceed model context limits. Send only the relevant sections, fields, pages, or chunks needed for the task. For large files, use retrieval, filtering, summarization, or a document schema.
- Simplify structured files before upload

  Large spreadsheets, CSV, JSON, and XML files can contain hidden, unused, or deeply nested content that increases processing complexity. Remove unused columns, empty rows, hidden sheets, unnecessary formulas, formatting, and irrelevant records before processing.
- Review archive contents before upload

  ZIP file processing depends on the files inside the archive. Include only supported and relevant files, and avoid unnecessary nested folders, duplicate files, or archives that expand into very large content.

Estimated Content Capacity

The following estimates provide a general guideline for the amount of content that can typically be processed within a 25 MB file size limit. Actual capacity may vary depending on factors such as document complexity, formatting, embedded media, and file structure.

Estimators

File type	Estimated Capacity for a 25 MB file
PDF, text-based	300 to 1000+ pages.
PDF, image-heavy or scanned	50 to 200 pages.
DOCX	100 to 500+ pages.
PPTX	50 to 150 slides.
TXT / Markdown / HTML	5 to 15 million words.
CSV / JSON / XML	Highly variable and best kept smaller if dense or deeply structured.
XLSX	Tens of thousands to hundreds of thousands of cells, depending on formulas, formatting, and sheets.
Images	10 to 100 images, depending on resolution and compression.
Multimedia	About 15 to 60 minutes, depending on format and bit rate.

File type	Estimated Capacity for a 25 MB file
ZIP archives	Depends on the files inside and extracted content still needs to be processed.

Why does the external REST tool only support HTTPS transactions?

The tool is limited to only HTTPS to maintain data privacy in transit.

Why is the Provider list in the LLM subtab empty?

If the Provider list is empty when creating an agent team, it's likely due to missing permissions. To resolve this, make sure the required permission groups with the **AllRowsAllFields** security view are added to the custom role assigned to the user, and run the necessary scheduled processes to synchronize security data. For more information, see *LLM Provider LOV Empty when Creating a new Agent in Ai Agent Studio (KB863374)*.

Why can't I see all agent teams in the Agent Teams tab of AI Agent Studio?

That's because only published agent teams are displayed by default. Use the **Draft** button to view agents that aren't published yet.

Why is my agent not fetching data from the document added to it?

Check if you've marked your document as published. If not, publish your document. To do so, from the Tools tab in AI Agent Studio, set the status of your document as **Ready to Publish**. Changing the status initiates the **Process Agent Documents** scheduled process. After the scheduled process completes, your document status is automatically changed to **Published**.

If your document status hasn't changed to **Published** even after five minutes, that means the scheduled process might have failed to trigger or complete. To resolve, do these steps:

1. Make sure your user has the **oraCommonFusionAISecurityAuthorization_create_Rag_OraResource** permission assigned.
2. From the Tools tab in AI Agent Studio, edit your document tool and change the status of your document from **Ready to Publish** to **Draft**.
3. Save the document tool.
4. Now, again change the status of your document back to **Ready to Publish**.
5. Save again to trigger the scheduled process.

For information about creating documents to be uploaded to the Document tool, see *Create Documents to Add to Document Tool*.

Why can't I see the imported agent teams in the target environment?

You might not have access to the product family the imported agent team belongs to. For more information, see *Provide Access to Configure AI Agents in all Products*.

How can I move the AI Agents link to the top of the Quick Actions list?

For information about reordering links in Quick Actions, see *Reorder Navigation Entries*.

Why is the Roles list in the Security tab empty?

That's probably because permission groups aren't enabled for your duty role. Security administrators can enable permission groups for the duty role assigned to you, in Security Console. For details, see *Access Requirements for AI Agent Studio*.