

Oracle®

MicroTx Workflows User's Guide



Release 26.1

G54906-01

May 2026

ORACLE®

Oracle MicroTx Workflows User's Guide, Release 26.1

G54906-01

Copyright © 2026, Oracle and/or its affiliates.

Primary Author: Sylaja Kannan

Contributing Authors: Tulika Das

Contributors: Deepak Goel, Brijesh Kumar Deo, Janaki Ram, Bharath MC, Pruthvithej R, Atul Dhiman, Tushar Shaily, Chandrashekar Venkatachar, Himanshu Gaur, Shivanshu Singh

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

1 About MicroTx Workflows

1.1	Fundamental Features of MicroTx Workflows	2
1.2	How MicroTx Workflows Empowers Your Projects	3
1.3	Traceability and Monitoring	3
1.4	About Security and Trust	4

2 Workflows

2.1	Manage Workflow Definitions	1
2.1.1	Specify Values for Workflow Definition Options	2
2.1.2	Define a Workflow Using Workflow Builder	4
2.1.3	Define a Workflow Using JSON	8
2.1.4	View a Workflow Definition	11
2.1.5	Update a Workflow Definition	13
2.1.6	Clone a Workflow Definition	14
2.1.7	Delete a Workflow Definition	15
2.2	Manage Workflow Executions	15
2.2.1	Run or Execute a Workflow	16
2.2.2	About Workflow Status	18
2.2.3	About Idempotent Workflows and Tasks	20
2.2.4	Update and Rerun a Workflow Execution	22
2.2.5	View Details of a Workflow Execution	24
2.2.6	Search and View Executed Workflows	27
2.2.7	Perform Actions on Multiple Executed Workflows	29
2.2.8	Handle Workflow Execution Failures	30
2.2.8.1	Identify Failed Tasks in a Workflow Execution	31
2.2.8.2	Specify a Failure Workflow	33
2.2.8.3	Handling Task Timeout Issues	34
2.3	Manage Schedules to Run Workflows	34
2.3.1	Create a Schedule Definition	35
2.3.2	Update a Schedule Definition	36
2.3.3	View a Schedule Definition	36
2.3.4	Delete a Schedule Definition	37
2.4	Manage Event Handlers	37

2.4.1	Create an Event Handler	37
2.4.2	Update an Event Handler	39
2.4.3	View an Event Handler	39
2.4.4	Delete an Event Handler	40
2.5	View and Approve Workflow Notifications	40

3 Tasks

3.1	About Tasks	1
3.1.1	About System Tasks	2
3.1.2	About Operator Tasks	3
3.1.3	About Simple Tasks or Worker Tasks	3
3.1.4	About Task Status	4
3.2	Create System Tasks and Operator Tasks	4
3.2.1	Access the Task Tab in Workflow Builder	6
3.2.2	Agentic Task	8
3.2.3	Agentic Planner Task	11
3.2.4	GenAI Task	17
3.2.5	GenAI Ingestion Task	19
3.2.6	GenAI Retrieve Task	22
3.2.7	gRPC Task	25
3.2.8	SFTP Task	27
3.2.9	HTTP Task	30
3.2.10	JSON RPC Task	32
3.2.11	SQL Task	33
3.2.12	Transaction Task	36
3.2.13	TX EventQ Publish Task	38
3.2.14	Human Task	40
3.2.15	Inline Task	41
3.2.16	Wait Task	41
3.2.17	Kafka Publish Task	42
3.2.18	Start Workflow Task	42
3.2.19	JSON JQ Transform Task	42
3.3	Manage Task Definitions for Simple Tasks	42
3.3.1	Create a Task Definition	43
3.3.2	Import a Task Definition	43
3.3.3	Add a Task Definition to a Workflow	44
3.3.4	Configure Timeout for Tasks	47
3.3.5	Edit a Task Definition	48
3.3.6	View a Task Definition	49
3.3.7	View Task Queues	49
3.3.8	Delete a Task Definition	50

3.4	Manage Executed Tasks	50
3.4.1	Search and View Details of Executed Tasks	50
3.4.2	Perform an Action on Multiple Tasks	52

4 Connectors

4.1	Manage LLM Connector	2
4.1.1	Create an LLM Definition	3
4.1.2	Update an LLM Definition	5
4.1.3	Test an LLM Definition	5
4.1.4	View an LLM Definition	6
4.1.5	Delete an LLM Definition	6
4.2	Manage Database Profiles	7
4.2.1	Create a Database Profile	7
4.2.2	Update a Database Profile	9
4.2.3	Test a Database Profile	9
4.2.4	View a Database Profile	10
4.2.5	Delete a Database Profile	10
4.3	Manage Internal Tool Configurations	10
4.3.1	Create an Internal Tool Configuration	11
4.3.2	Update an Internal Tool Configuration	13
4.3.3	View an Internal Tool Configuration	13
4.3.4	Delete an Internal Tool Configuration	13
4.4	Manage MCP Server Details	14
4.4.1	Provide MCP Server Details	14
4.4.2	Update an MCP Server Details	16
4.4.3	Test MCP Server Details	17
4.4.4	View MCP Server Details	17
4.4.5	Delete MCP Server Details	17
4.5	Manage File Storage	18
4.5.1	Upload Files to Storage	18
4.5.2	Update a File in Storage	20
4.5.3	Delete a File From Storage	20
4.5.4	Delete all Files From Storage	20
4.6	Manage Cloud Profiles	21
4.6.1	Create a Cloud Profile	21
4.6.2	Update a Cloud Profile	22
4.6.3	View a Cloud Profile	22
4.6.4	Delete a Cloud Profile	23
4.7	Manage SFTP Profiles	23
4.7.1	Create an SFTP Profile	23
4.7.2	Update an SFTP Profile	24

4.7.3	Test an SFTP Profile	25
4.7.4	View an SFTP Profile	25
4.7.5	Delete an SFTP Profile	25

5 Agentic AI

5.1	About Conversational Agent Chat	2
5.1.1	Set Up Conversational Agent Chat	4
5.2	Manage Prompt Template Definitions	5
5.2.1	Create a Prompt Template Definition	6
5.2.2	Update a Prompt Definition	7
5.2.3	View a Prompt Definition	8
5.2.4	Delete a Prompt Definition	8
5.3	Manage Agent Profiles	8
5.3.1	Create an Agent Profile	9
5.3.2	Update an Agent Profile	12
5.3.3	View an Agent Profile	13
5.3.4	Delete an Agent Profile	13

6 Known Issues in MicroTx Workflows

About MicroTx Workflows

Oracle Transaction Manager for Microservices - Workflows (MicroTx Workflows) is an enterprise-grade orchestration engine that enables teams to build and operationalize AI-powered business workflows without writing any code. It provides ready-to-use AI building blocks, that include LLM access, agentic tasks, reusable AI agents, tool integrations, RAG retrieval, and planner-based reasoning, so that enterprises can incorporate intelligent capabilities directly into their existing processes.

As agentic artificial intelligence (AI) reshapes enterprise automation and decision-making, MicroTx Workflows offers an enterprise-ready, no-code orchestration engine that enables embedding AI capabilities and AI agents into existing and new workflows. Leveraging Conductor open source, a cloud-native architecture, and robust integration with large language models (LLMs), databases, and orchestration tools, MicroTx Workflows enables organizations to rapidly develop, scale, and govern intelligent agent-driven solutions, unlocking new levels of automation, resilience, and insight.

MicroTx Workflows supports all three categories of modern workflow automation:

- **Automated workflows** – traditional deterministic flows driven by microservices and rules.
- **Non-agentic AI workflows** – deterministic flows augmented with AI tasks (e.g., extraction, summarization, classification).
- **Agentic workflows** – adaptive, non-deterministic flows where AI can plan, execute tools, reflect, and re-plan within enterprise guardrails.

By providing no-code modeling, AI connectors, and governed execution, MicroTx Workflows makes it easy to create, extend, and transition between automated, non-agentic, and agentic workflows, all while reusing existing enterprise assets and integrations.

Topics

- [Fundamental Features of MicroTx Workflows](#)
MicroTx Workflows helps you design, run, monitor, and manage durable workflows across microservices, AI services, human approvals, and external systems. You can model workflows visually or with JSON, add reusable tasks and agents, and reduce the amount of custom orchestration code required to build enterprise business processes.
- [How MicroTx Workflows Empowers Your Projects](#)
MicroTx Workflows helps you build enterprise workflow and agentic AI applications faster by reducing custom orchestration code, centralizing workflow logic, and providing durable execution out of the box.
- [Traceability and Monitoring](#)
MicroTx Workflows provides traceability for Agentic AI workflows through workflow execution details, task-level execution views, workflow logs, and server logs.
- [About Security and Trust](#)
MicroTx Workflows provides security and governance through authentication, role-based access control, and configurable guardrails for AI-powered execution.

1.1 Fundamental Features of MicroTx Workflows

MicroTx Workflows helps you design, run, monitor, and manage durable workflows across microservices, AI services, human approvals, and external systems. You can model workflows visually or with JSON, add reusable tasks and agents, and reduce the amount of custom orchestration code required to build enterprise business processes.

- **Visual and JSON workflow modeling:** Design workflows using Workflow Builder or define them directly in JSON. Use tasks, operators, schedules, event handlers, and input/output mappings to model business logic with little custom orchestration code.
- **Agentic and GenAI capabilities:** Add AI-powered steps to workflows using GenAI Tasks, Agentic Tasks, Agentic Planner, GenAI Ingestion, and GenAI Retrieve tasks. Use these capabilities for summarization, classification, extraction, reasoning, planning, RAG, and adaptive automation.
- **Dynamic orchestration with Agentic Planner:** Use Agentic Planner to evaluate runtime context, tools, tasks, and prompts, and determine the next action dynamically. This helps workflows adapt to changing inputs and task outcomes.
- **Reusable AI agents:** Define agent profiles once and invoke them from multiple workflows using Agentic Tasks. Centralize agent instructions, LLM configuration, tools, MCP servers, prompt variables, and guardrails.
- **Tool and MCP integration:** Extend agent capabilities using built-in tools and authenticated Model Context Protocol (MCP) servers. Agents can interact with external systems and services through controlled integrations.
- **Durable and resilient execution:** Run long-lived workflows with persisted state, retries, timeouts, pause and resume operations, restart and retry actions, and execution history.
- **Data consistency and idempotency:** Use idempotency and transaction-aware tasks to reduce duplicate side effects and improve reliability across distributed workflows and microservices.
- **Microservice integration:** Connect workflows to REST APIs, gRPC services, databases, messaging systems, worker services, and external applications. Existing services can participate as reusable workflow tasks.
- **Human-in-the-loop support:** Add human tasks to pause a workflow until an authorized user reviews, approves, or rejects a request. This helps add trust and control to sensitive automated or AI-assisted decisions.
- **Traceability and monitoring:** View workflow execution status, task input and output, logs, timing, errors, and status transitions in the UI. Use execution history and server logs to monitor and troubleshoot workflows.
- **Reusable components and templates:** Reuse workflow definitions, task definitions, prompt templates, agent profiles, tools, and MCP server definitions across workflows.
- **Enterprise security:** Protect workflow APIs and UI access using OAuth 2.0 access tokens from supported identity providers, such as Keycloak and Oracle IAM. Use RBAC to control access to workflow and Agentic AI resources, and guardrails to evaluate AI input and output before and after LLM calls.
- **UI and API access:** Create, manage, execute, and monitor workflows through the MicroTx Workflows UI or REST APIs, enabling both interactive development and programmatic integration.

1.2 How MicroTx Workflows Empowers Your Projects

MicroTx Workflows helps you build enterprise workflow and agentic AI applications faster by reducing custom orchestration code, centralizing workflow logic, and providing durable execution out of the box.

Instead of embedding complex sequencing, retries, state tracking, and integration logic across multiple services, you can model the process as a workflow and let MicroTx Workflows manage execution.

MicroTx Workflows is especially useful for projects that need to coordinate microservices, AI tasks, human approvals, database operations, messaging systems, and external tools. You can compose workflows visually or with JSON, reuse common tasks and prompt templates, invoke reusable agent profiles, and monitor execution from a single place.

Unlike solutions that require developers to build orchestration, state management, retry logic, and monitoring separately, MicroTx Workflows provides these capabilities as part of the platform. This helps teams move faster while still maintaining control, traceability, and reliability for production workloads.

- **Rapidly compose agentic applications:** Combine existing microservices, system tasks, worker tasks, AI agents, GenAI tasks, tools, and MCP servers in a single workflow.
- **Reduce custom orchestration code:** Move process flow, task sequencing, retries, timeouts, and status handling into workflow definitions instead of spreading that logic across application code.
- **Adapt workflows dynamically:** Use Agentic Planner to evaluate runtime context and decide the next action based on goals, task results, tools, and available workflow inputs.
- **Reuse components across projects:** Reuse workflow definitions, task definitions, prompt templates, agent profiles, tools, and MCP server definitions to accelerate new use cases.
- **Integrate human decision points:** Add human tasks when a workflow needs approval, exception handling, compliance review, or validation of AI-assisted decisions.
- **Improve reliability:** Use durable execution, persisted workflow state, retries, idempotency, timeout handling, and restart or retry operations to support long-running and failure-prone processes.
- **Support enterprise integration:** Connect workflows with REST APIs, gRPC services, databases, queues, worker services, AI models, and external tools.
- **Increase transparency:** View workflow execution details, task inputs and outputs, logs, status transitions, and failure reasons to troubleshoot and audit business processes.
- **Strengthen security and governance:** Use OAuth 2.0 access tokens from supported identity providers, RBAC, and AI guardrails to control access and reduce risk in AI-powered workflows.

By combining workflow orchestration, agentic AI, durable execution, human-in-the-loop control, and operational traceability, MicroTx Workflows helps you deliver intelligent business processes that are easier to build, evolve, and operate in production.

1.3 Traceability and Monitoring

MicroTx Workflows provides traceability for Agentic AI workflows through workflow execution details, task-level execution views, workflow logs, and server logs.

After a workflow runs, you can open the workflow execution details page to view the overall execution status and inspect each task in the workflow. For AI-powered workflows, this includes Agentic Tasks, Agentic Planner, GenAI Tasks, and any tools, human tasks, or system tasks used during execution.

The workflow execution view helps you monitor task status, input, output, timing, and failure information. The Logs section provides additional execution details that can help trace how an AI workflow progressed, including agent activity, planner steps, tool usage, and task-level results where available.

MicroTx Workflows also records server-side logs. Error logs are captured by default, and DEBUG or TRACE logging can be enabled through configuration for deeper troubleshooting in development or staging environments. Use these logs to investigate workflow execution issues, task failures, prompt resolution, tool invocation, model calls, retries, and other runtime behavior.

1.4 About Security and Trust

MicroTx Workflows provides security and governance through authentication, role-based access control, and configurable guardrails for AI-powered execution.

The workflow server is protected by bearer tokens issued by supported identity providers, and you can control access to workflow and Agentic AI metadata by using roles. Guardrails can evaluate prompts and responses outside the LLM call path, helping reduce unsafe input, prompt injection risks, and undesired generated output. You can use RBAC to control who can create, update, view, and use Agentic AI metadata, including prompt templates, agent profiles, tools, LLM definitions, and MCP server definitions.

Guardrails help protect AI-powered workflow steps from unsafe or undesired input and output. You can configure guardrails to evaluate prompts before they are sent to the LLM and responses after they are returned by the LLM. This allows MicroTx Workflows to block or mask configured content outside the LLM itself, helping reduce risks such as prompt injection or unsafe generated responses.

2

Workflows

Workflows in MicroTx Workflows provide blueprints for defining, running, tracking, and managing business process logic.

MicroTx Workflows provides a durable execution model for defining and executing workflows at scale. You can model workflows visually by using Workflow Builder or define them directly in JSON. A workflow can include system tasks, worker tasks, operators, human tasks, schedules, event handlers, and AI-powered tasks. MicroTx Workflows stores workflow execution history, task state, input and output data, logs, and status transitions so that you can monitor, troubleshoot, retry, restart, or audit workflow executions.

Topics

- [Manage Workflow Definitions](#)
Create, validate, update, retrieve, search, and delete workflow definitions. You can retrieve a specific definition by name and version, list all definitions, get only the latest versions, or view workflow names and versions without loading full definition bodies.
- [Manage Workflow Executions](#)
Start workflows asynchronously or synchronously, pass input data, set a version, correlation ID, priority, or idempotency key, and track the returned workflow ID. You can view execution status, include task details, find workflows by correlation ID, list running workflows, search workflow history, pause or resume an execution, skip a task, rerun, restart, retry, terminate, or remove workflow executions.
- [Manage Schedules to Run Workflows](#)
Create, update, retrieve, search, and delete scheduled workflow definitions. Use schedules to run workflows automatically based on a configured schedule, such as daily, weekly, monthly, or another supported schedule expression. Scheduling APIs are available only when the scheduler feature is enabled.
- [Manage Event Handlers](#)
Create, update, retrieve, search, and delete event handlers. Event handlers in MicroTx Workflows listen for events from supported internal or external message brokers and perform one or more actions on workflow or task when matching conditions are met. This feature also enables loose coupling between workflows.
- [View and Approve Workflow Notifications](#)
View pending or in-progress human tasks generated by workflows. Human task notifications help you find workflow steps that require manual action, such as review or approval. The notification API lists human tasks; completing, approving, rejecting, or otherwise acting on the task is handled by updating the corresponding task status and output.

2.1 Manage Workflow Definitions

Create, validate, update, retrieve, search, and delete workflow definitions. You can retrieve a specific definition by name and version, list all definitions, get only the latest versions, or view workflow names and versions without loading full definition bodies.

A workflow definition is the blueprint for a workflow. It defines the workflow name, version, tasks, task reference names, inputs, transitions, and other metadata that determine the

execution behavior. A workflow execution is a running or completed instance of a workflow definition.

You can create workflow definitions visually with Workflow Builder or by providing JSON, validate definitions before saving them, view existing definitions, update definitions, clone definitions, and delete a specific version when it is no longer required.

Topics

- [Specify Values for Workflow Definition Options](#)
Specify the workflow definition options before you add tasks and save the workflow.
- [Define a Workflow Using Workflow Builder](#)
Use Workflow Builder to create a workflow definition visually. You can add tasks, arrange the flow, configure task properties, and define how tasks interact without writing the full JSON manually.
- [Define a Workflow Using JSON](#)
Define a workflow by providing its JSON definition directly. Use this option when you want precise control over workflow metadata, task configuration, versions, inputs, and flow behavior. You can validate the JSON definition before saving the workflow definition.
- [View a Workflow Definition](#)
View a workflow definition by name and, optionally, by version. You can also list all workflow definitions, search definitions, view only the latest versions, or retrieve workflow names and versions without loading the full definition body.
- [Update a Workflow Definition](#)
Update an existing workflow definition when you need to change its task configuration, metadata, inputs, or flow behavior. The update operation supports one or more workflow definitions. To rename a workflow, create a new workflow definition with the new name instead of changing the existing workflow name.
- [Clone a Workflow Definition](#)
Clone a workflow definition when you want to create a new definition based on an existing one. Cloning starts from the selected definition's JSON or visual structure, then saves it as a separate workflow definition, typically with a new name or version.
- [Delete a Workflow Definition](#)
Delete a workflow definition version when it is no longer required. Deletion is performed by workflow name and version. Deleting a workflow definition does not delete workflow executions that were already created from that definition.

2.1.1 Specify Values for Workflow Definition Options

Specify the workflow definition options before you add tasks and save the workflow.

A workflow definition describes how a workflow behaves when it runs. It includes the workflow name, version, task list, task reference names, input parameters, output parameters, timeout configuration, failure handling, owner details, and other execution metadata.

A workflow definition must include at least one task. If the workflow definition includes a collection of tasks, the definition determines the order in which the workflow executes tasks. A workflow definition may also define how to pass data and state from one task to another using input and output parameters.

You can create a workflow definition visually with Workflow Builder or by providing JSON. MicroTx Workflows validates workflow names, task names, and task reference names before creating or updating a definition.

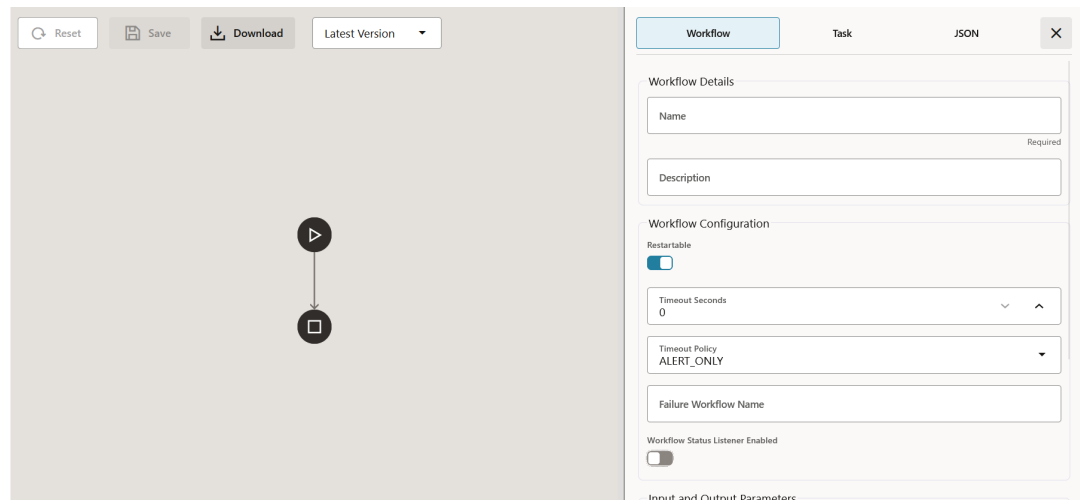
To specify the workflow definition options before you add tasks and save the workflow:

1. Open the navigation menu and click **Definitions**.
2. Click the **Workflows** tab.

The **Workflows** list page opens and displays the workflow definitions that have been created.

3. Click  (Add New Workflow).

The Workflow Builder opens and shows a visual depiction of all the components of the workflow. The **Workflow** tab displays the workflow definition options.



4. In the **Workflow Details** group, enter the following information.
 - **Name:** Mandatory. Enter a unique name for the workflow definition. The name can be up to 128-characters long. Use only letters, numbers, underscores (_), and hyphens (-). Spaces and other special characters are not supported. You cannot change this name later.
 - **Description:** Optional. Enter a description for the workflow definition. The description can be up to 2048-characters long.
5. In the **Workflow Configuration** group, enter the following information.
 - **Restartable:** Optional. Select this option to allow the workflow execution to be restarted. This option is enabled by default. When enabled, you can start restart workflows in COMPLETED, TERMINATED, FAILED, or TIMED_OUT state. When disabled, you cannot restart workflows.
 - **Timeout Seconds:** Optional. Enter a time period in seconds. The default value is 0, which means that there is no timeout and the workflow runs as long it takes to reach a final state, which may be days, weeks, or months.
 - **Timeout Policy:** Optional. Select the action performed when the workflow does not move to a final state within the specified timeout.
 - **TIME_OUT_WF:** Terminates the workflow execution and changes the workflow status to TIMED_OUT. For example, let's consider that you specify the timeout value as 30s. If the workflow does not reach its final state in 30s, the workflow status changes to TIMED_OUT and the associated task gets canceled.
 - **ALERT_ONLY:** Records the timeout condition for monitoring but does not terminate the workflow execution. This is the default timeout policy.

- **Failure Workflow Name:** Optional. Enter the name of the workflow that you want to start if this workflow fails. Use failure workflows for cleanup, compensation, notification, or other failure-handling actions.
 - **Workflow Status Listener Enabled:** Optional. Enable this option only when your environment uses a workflow status listener. When enabled, MicroTx Workflows invokes the listener for workflow-lifecycle events such as started, paused, resumed, retried, restarted, completed, terminated, and finalized.
6. In the **Input and Output Parameters** group, click **Add Parameter** to add an input or an output parameter.
- **Input Parameters:** Optional. Add the names of input values that callers can pass when starting the workflow. Tasks can reference these values using workflow input expressions.
 - **Output Parameters:** Optional. Add output mappings that define the values returned or exposed by the workflow. Output values can reference workflow inputs, workflow variables, or task outputs.

Next: A workflow must have at least one task. Add a task to the workflow and save it. See [Define a Workflow Using Workflow Builder](#).

2.1.2 Define a Workflow Using Workflow Builder

Use Workflow Builder to create a workflow definition visually. You can add tasks, arrange the flow, configure task properties, and define how tasks interact without writing the full JSON manually.

The builder generates the workflow definition in JSON. Review the generated workflow JSON before saving the workflow definition. After creating and saving the workflow definition, you can use it to start workflow executions.

A workflow definition contains the workflow metadata and the list of tasks that make up the workflow. Each task that you add to the canvas becomes part of the workflow definition. Worker tasks are based on task definitions, while system tasks and operators use built-in task types.

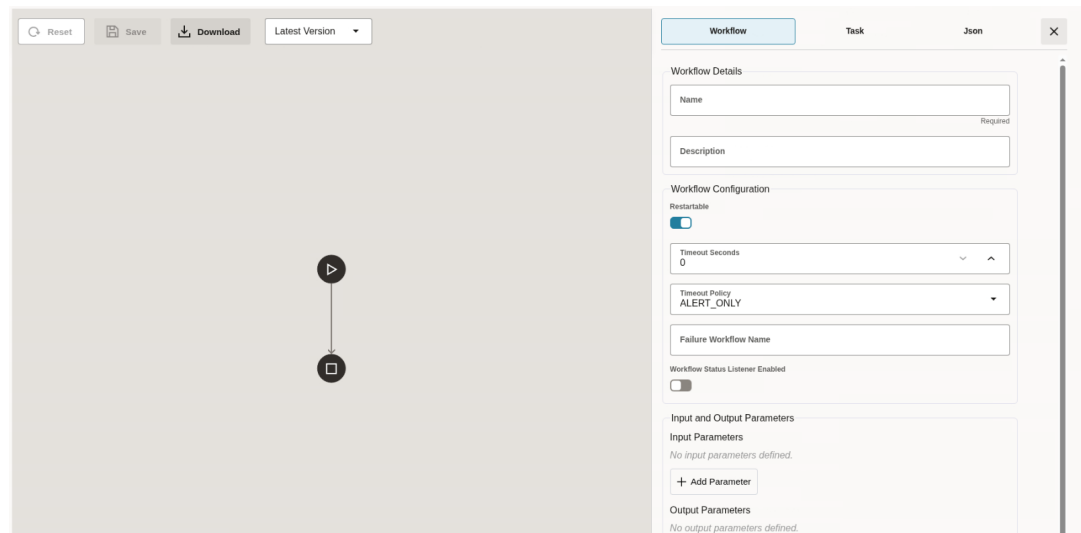
To define a workflow using Workflow Builder:

1. Open the navigation menu and click **Definitions**.
2. Click the **Workflows** tab.

The **Workflows** list page opens and displays the workflow definitions that have been created.

3. Click .

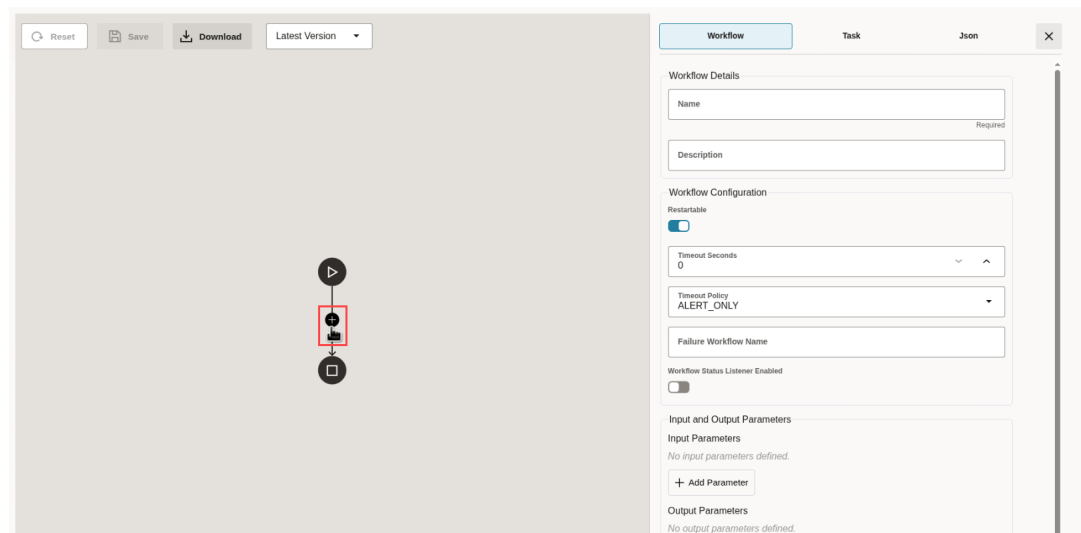
The Workflow Builder is displayed. On the right pane, the **Workflow** tab displays all the details of the workflow as shown in the following figure. On the left pane, the components of the workflow are depicted visually.



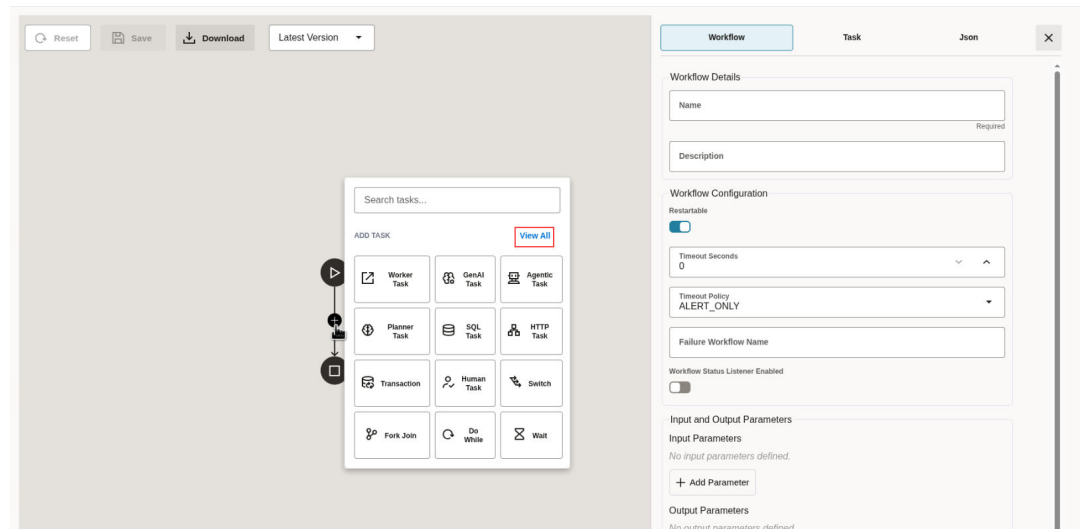
4. In the **Workflow Details** group, you must enter a unique name for the workflow definition. The name can be up to 128-characters long. Use only letters, numbers, underscores (_), and hyphens (-). Spaces and other special characters are not supported. You cannot change the workflow name after the workflow definition is created.

You can also specify optional workflow definition values, such as description, version, timeout configuration, failure workflow, input parameters, and output parameters. See [Specify Values for Workflow Definition Options](#).

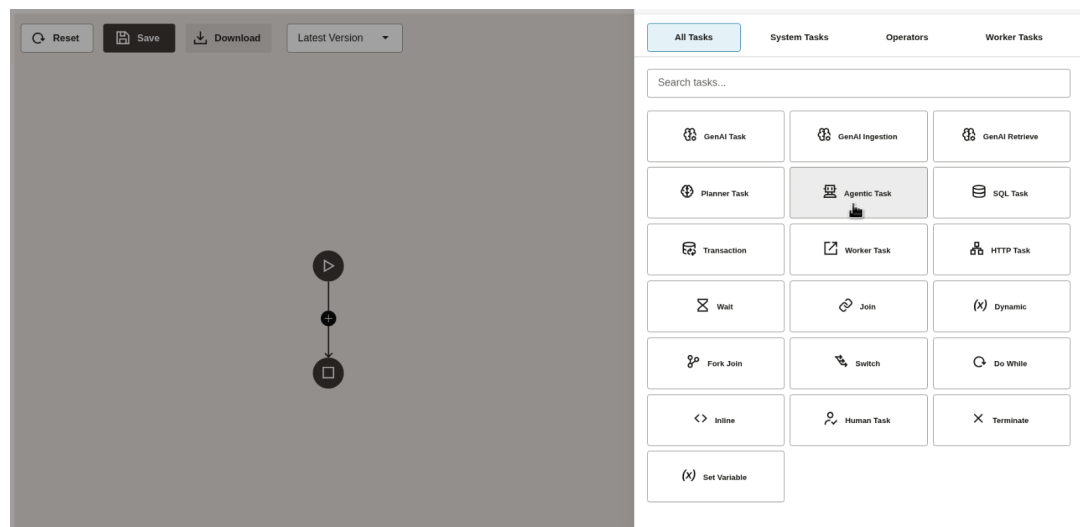
5. To add a task in your workflow using the builder, place your cursor on the arrow mark. A plus sign appears as shown in the following image.



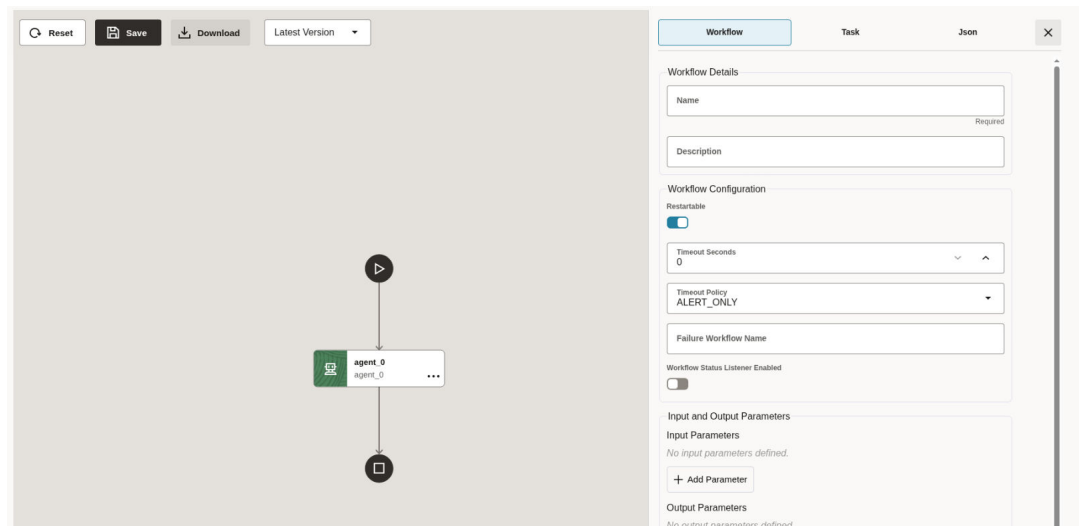
6. Click the plus icon. Many tasks are displayed.



7. Click **View All** to view all the tasks that you can add using the Workflow Builder.

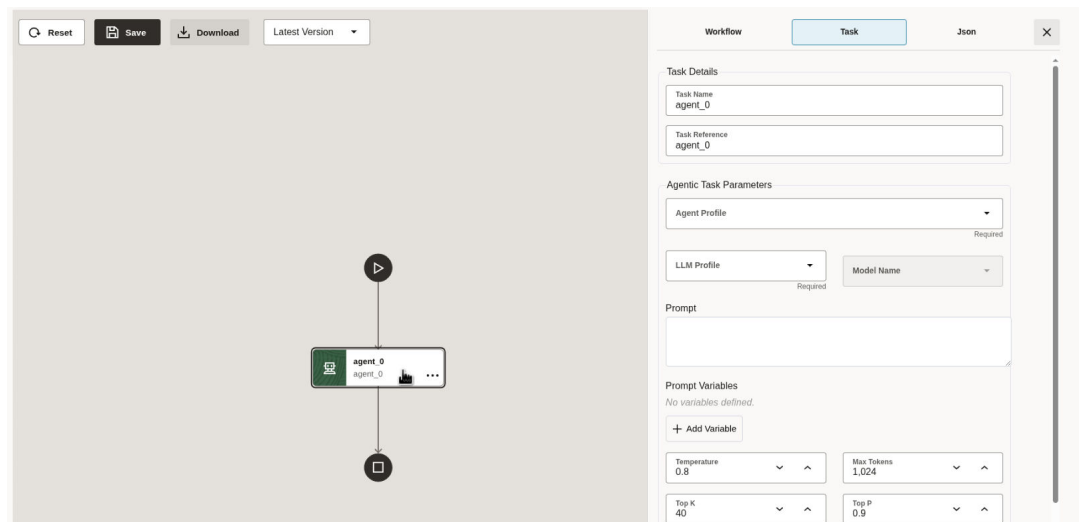


8. Search for a task or browse the available task groups.
 - **All Tasks:** Displays all task types that you can add to the workflow.
 - **System Tasks:** Displays built-in system tasks.
 - **Operators:** Displays control-flow and operator tasks, such as branching, joining, waiting, terminating, or looping tasks.
 - **Worker Tasks:** Displays reusable worker task definitions that have been created as task definitions. These task definitions are stored separately and can be reused across workflow definitions.
9. Click a task that you want to add to the workflow. The following image shows an agentic task is added to the workflow.



10. Click the task on the canvas and enter the task details in the **Task** tab.

Configure values such as the task name, task reference name, input parameters, optional behavior, timeout settings, and other task-specific parameters. The task reference name must be unique within a workflow definition. For information about each task and its parameters, see [Create System Tasks and Operator Tasks](#).



11. Click **Save**.
12. MicroTx Workflows displays the changes in JSON code. Review all the changes.
13. Click **Confirm Save** to save the changes.

If you do not want to save the changes, click **Cancel**, and then click **Reset** to undo all the changes that you have made since the workflow was last saved.

14. Optional. To download the JSON of the workflow, click **Download**.

The new workflow is displayed on the Workflows list page.

2.1.3 Define a Workflow Using JSON

Define a workflow by providing its JSON definition directly. Use this option when you want precise control over workflow metadata, task configuration, versions, inputs, and flow behavior. You can validate the JSON definition before saving the workflow definition.

A workflow definition must include a unique workflow name and at least one task. You can also specify optional values such as description, version, input parameters, output parameters, timeout configuration, failure workflow, variables, input template, and metadata. For information about optional parameters of a workflow definition, see [Specify Values for Workflow Definition Options](#).

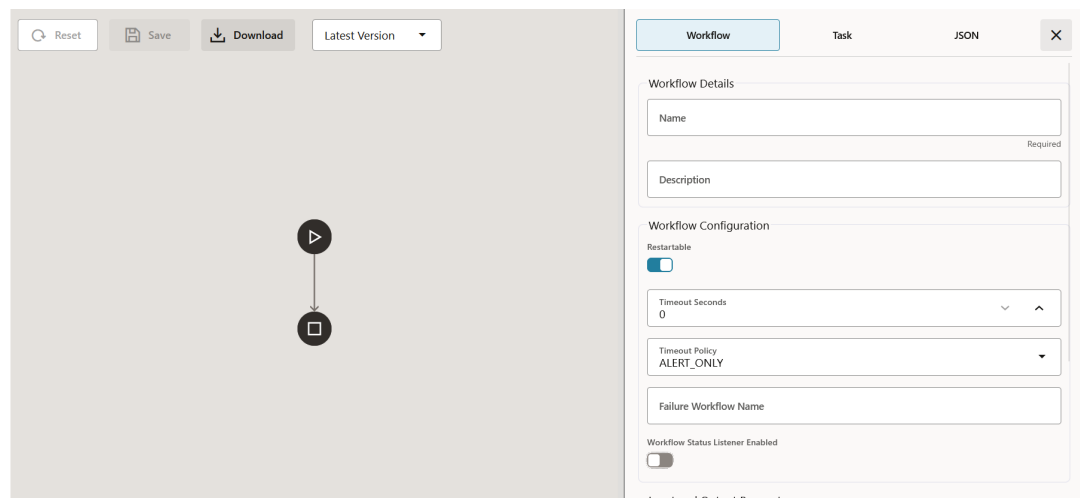
To define a workflow using JSON:

1. Open the navigation menu and click **Definitions**.
2. Click the **Workflows** tab.

The **Workflows** list page opens and displays the workflow definitions that have been created.

3. Click .

The Workflow Builder is displayed. On the right pane, the **Workflow** tab displays all the details of the workflow as shown in the following figure. On the left pane, the components of the workflow are depicted visually.



4. Click the **JSON** tab.
5. Delete the default JSON, if required, and paste the JSON code for your workflow definition. A sample workflow code is provided below for your reference.

Example JSON code

This sample workflow summarizes the text provided by a user. The workflow contains one GenAI task. The GenAI task uses the `filePath` value from the workflow input by referencing `${workflow.input.filePath}`.

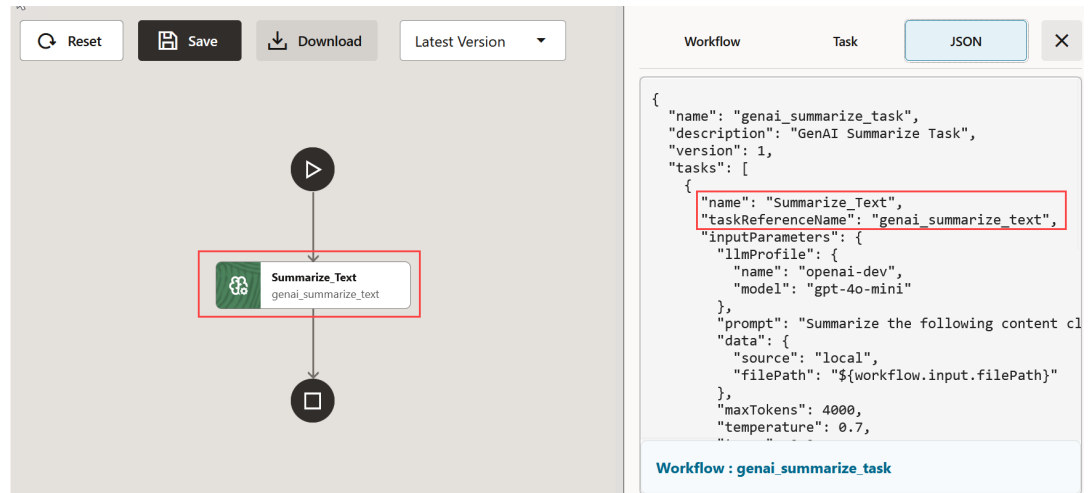
```
{
  "name": "genai_summarize_task",
  "description": "GenAI Summarize Task",
```

```

"version": 1,
"tasks": [
  {
    "name": "Summarize_Text",
    "taskReferenceName": "genai_summarize_text",
    "inputParameters": {
      "llmProfile": {
        "name": "openai-dev",
        "model": "gpt-4o-mini"
      },
      "prompt": "Summarize the following content clearly and concisely,
highlighting the key points, main ideas, and any important conclusions or
actions. Avoid unnecessary details, but preserve the original meaning and
intent.",
      "data": {
        "source": "local",
        "filePath": "${workflow.input.filePath}"
      },
      "maxTokens": 4000,
      "temperature": 0.7,
      "top_p": 0.9,
      "top_k": 50
    },
    "type": "GENAI_TASK"
  }
],
"inputParameters": [],
"outputParameters": {},
"schemaVersion": 2,
"restartable": true,
"workflowStatusListenerEnabled": false,
"ownerEmail": "example@email.com",
"timeoutPolicy": "ALERT_ONLY",
"timeoutSeconds": 0,
"variables": {},
"inputTemplate": {},
"enforceSchema": true,
"metadata": {}
}

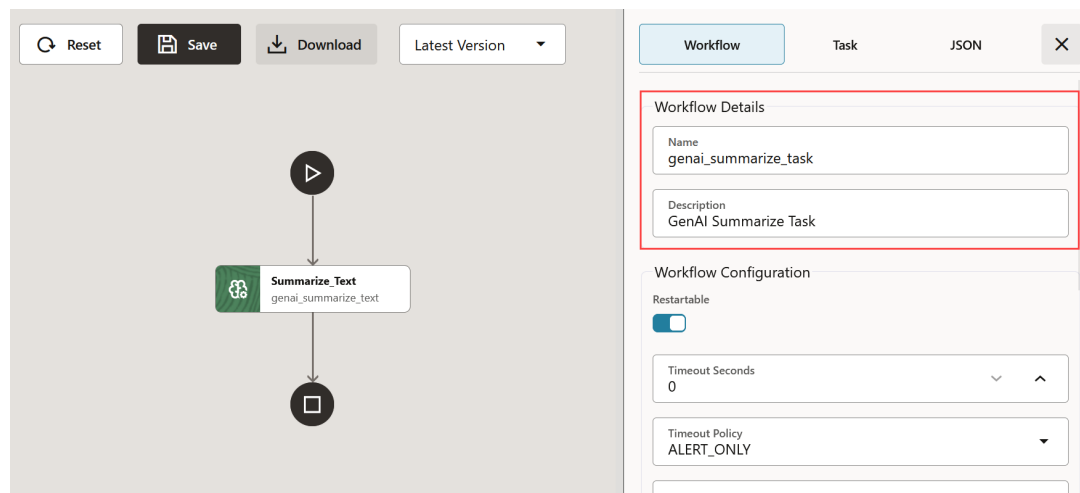
```

MicroTx Workflows validates the JSON before rendering the workflow visually on the canvas, as shown in the following image, or saving the workflow definition. The workflow name is required, the workflow must contain at least one task, and each task must include a task name and a unique task reference name. If the JSON contains invalid syntax, missing required fields, an invalid workflow name, or invalid task configuration, correct the errors before saving. Since the sample workflow has a single task, the visual canvas displays one task block.

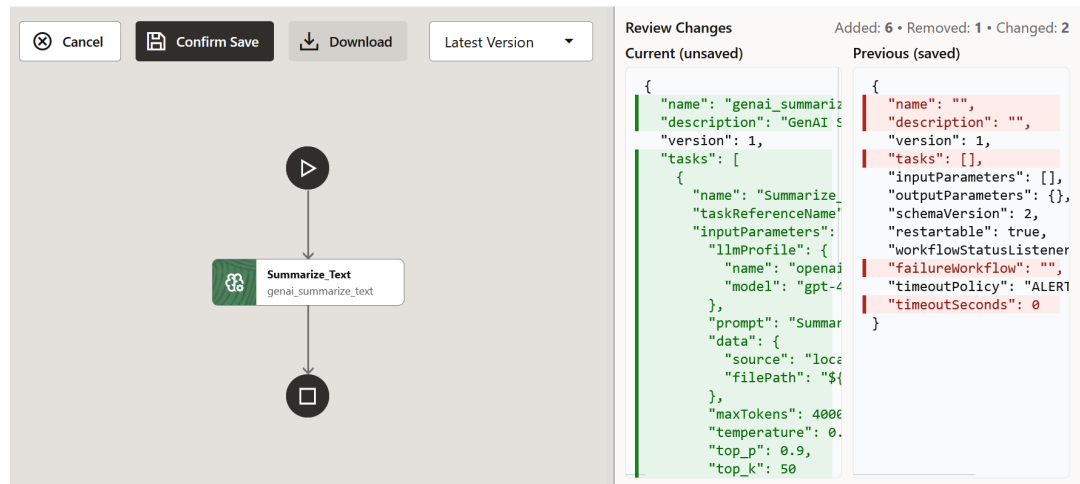


6. Click the **Workflow** tab.

As shown in the following image, MicroTx Workflows automatically populates the details in the Workflow tab from the JSON code that you enter.



7. Click **Save**.
8. MicroTx Workflows displays the changes in JSON code as shown in the following figure. Review all the changes.



9. Click **Confirm Save** to save the changes.

The new workflow is saved successfully. You can view the new workflow on the Workflows list page.

Next: Execute the workflow. See [Run or Execute a Workflow](#).

2.1.4 View a Workflow Definition

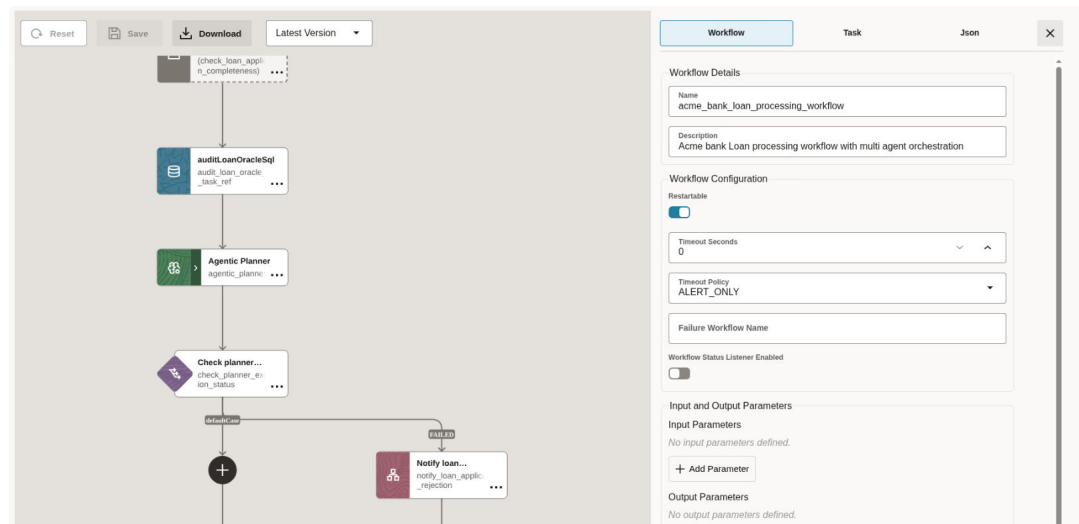
View a workflow definition by name and, optionally, by version. You can also list all workflow definitions, search definitions, view only the latest versions, or retrieve workflow names and versions without loading the full definition body.

The definition can be viewed as JSON and, where supported by the UI, as a visual workflow diagram. The visual diagram shows the tasks, operators, decision paths, and other building blocks that make up the workflow. Use this view to understand the workflow structure, task order, and how tasks are connected.

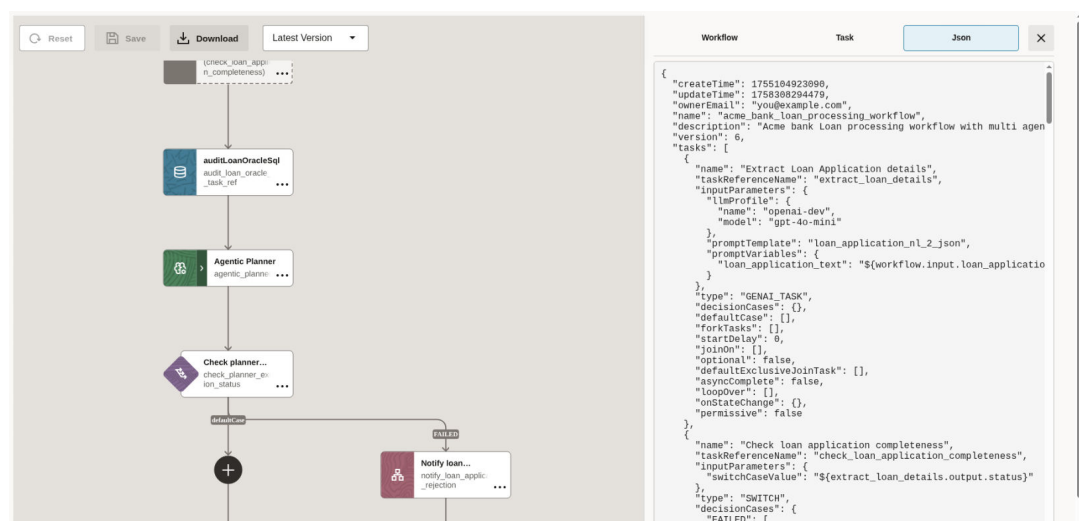
To view a workflow definition:

1. Open the navigation menu and click **Definitions**.
2. Click the **Workflows** tab.
The **Workflows** list page opens and displays the workflow definitions that have been created.
3. Optional. Use the search bar to search for a workflow using its name or description.
4. Optional. You can customize the columns displayed in your workflow list. Click **Columns**, select all the details you want to view as mentioned in the following list, and then click **Close**.
 - **Name:** Name of a workflow definition.
 - **Description:** Description for a workflow definition.
 - **Version:** Version number of the workflow definition.
 - **Created Time:** Date and time when the workflow definition was created.
 - **Owner:** Email of the user who created the workflow definition.
 - **Created By:** Name of the user who created the workflow definition.
 - **Task Count:** Displays the number of tasks associated with the workflow definition.

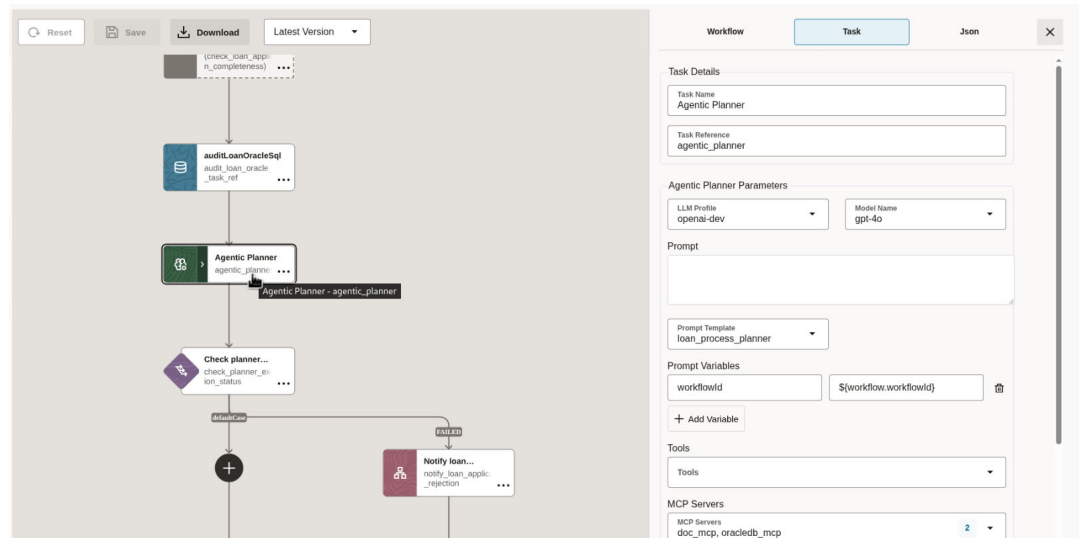
- **Updated Time:** The most recent date and time when any changes were made to the workflow definition.
 - **Updated By:** Name of the user who performed the most recent update.
5. Identify the workflow that you want to view.
 6. Click **Workflow Info** (📘) under **Actions**.
The Workflow Builder is displayed.
 7. As shown in the following image, the Workflow Builder visually depicts all the components of the workflow, such as tasks, operators, and decision boxes in the left pane. Scroll to view all the components of the workflow to understand the different components of the workflow and how the workflow is executed. The **Workflow** tab displays all the details of the workflow.



8. Click the **JSON** tab to view the workflow definition in JSON format as shown in the following image. Scroll to view the entire JSON.



9. Click a task in the left pane to view more details about the task. The following figure shows the details of an Agentic Planner task in the **Task** tab in the right pane.



10. You can view different versions of the workflow. By default, the Workflow Builder displays the latest version of the workflow. To view a previous version of a workflow, click the **Latest Version** dialog box, and select the version that you want to view.

2.1.5 Update a Workflow Definition

Update an existing workflow definition when you need to change its task configuration, metadata, inputs, or flow behavior. The update operation supports one or more workflow definitions. To rename a workflow, create a new workflow definition with the new name instead of changing the existing workflow name.

Before you begin, decide if you want to update an existing version of a workflow definition or create a new version. You can create a new version of a workflow definition when you don't want to disrupt ongoing executions that use the current workflow definition. For more information about the impact of creating a new version or updating a current version of workflow definition, see [Versioning Workflows](#) in open-source Conductor documentation.

1. Open the navigation menu and click **Definitions**.

2. Click the **Workflows** tab.

The **Workflows** list page opens and displays the workflow definitions that have been created.

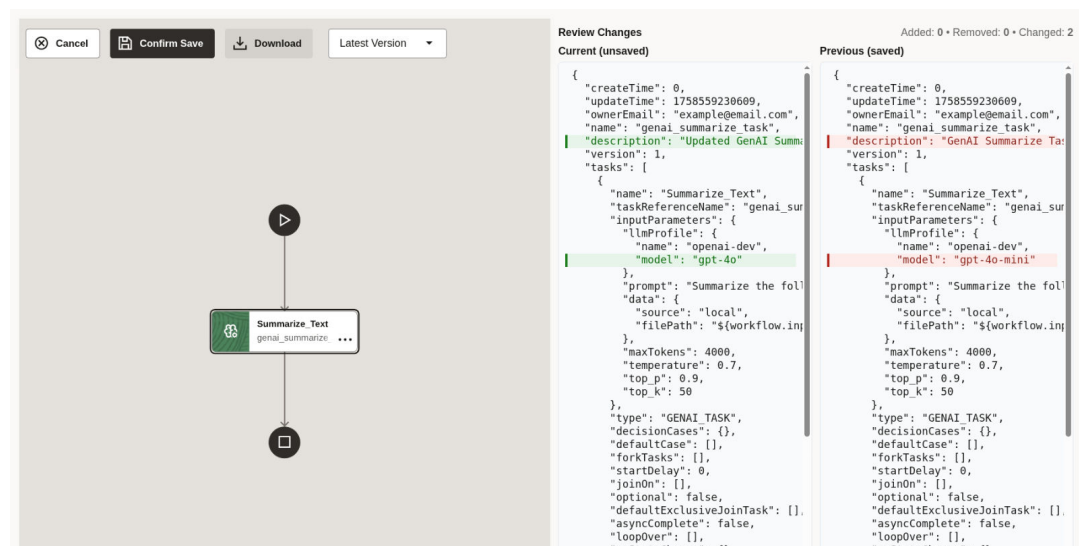
3. Identify the workflow definition that you want to update.

4. Click (**Edit**).

5. Select the version of the workflow definition that you want to update from the **Version** drop-down list. By default the latest version of the workflow definition is displayed.

6. You can update all the configuration values you specified when creating a workflow definition, except the workflow name. To update a value, perform one or more of the following tasks depending on the value that you want to update.
 - To edit the workflow definition properties, click the **Workflow** tab, and then update the value.
 - To edit the details for a task, click a task in the left pane. Details of the selected task appear in the right pane. Click the **Task** tab, and then update a value.

- To directly edit the JSON code, click the **JSON** tab and then update the code.
7. Skip this step if you do not want to save the updated workflow definition as a new version and you want to overwrite the selected version of the workflow definition. To save the updated workflow definition as a new version, you must explicitly update the value of the version field in the **JSON** tab with the new version number. For example, if you update version 5 of a workflow definition, all changes that you make overwrite version 5. However, if you want to retain the current code of version 5 and save the new changes in another version, change the value in the version field to 6 to save the changes in version 6 of a workflow definition.
 8. Click **Save**.
 9. MicroTx Workflows displays the changes in JSON code as shown in the following figure. Review all the changes.



10. Click **Confirm Save** to save the changes.

If you do not want to save the changes, click **Cancel**, and then click **Reset** to undo all the changes that you have made since the workflow definition was last saved.

2.1.6 Clone a Workflow Definition

Clone a workflow definition when you want to create a new definition based on an existing one. Cloning starts from the selected definition's JSON or visual structure, then saves it as a separate workflow definition, typically with a new name or version.

1. Open the navigation menu and click **Definitions**.
2. Click the **Workflows** tab.
The **Workflows** list page opens and displays the workflow definitions that have been created.
3. (Optional.) Use the search bar to search for a workflow using its name or description.
4. Identify the workflow that you want to clone, and then click  (**Clone Workflow**) under **Actions**.

The **Clone Workflow** dialog box is displayed.

5. Enter a new unique name to identify a workflow. The name can be up to 128-characters long. Use only letters, numbers, underscores (_), and hyphens (-). Spaces and other special characters are not supported. You cannot change this name later.
6. Enter a natural number as the version of the workflow.
7. Click **Clone**.

The new workflow is displayed in the Workflow Builder.

2.1.7 Delete a Workflow Definition

Delete a workflow definition version when it is no longer required. Deletion is performed by workflow name and version. Deleting a workflow definition does not delete workflow executions that were already created from that definition.

If you delete all versions of a workflow definition, MicroTx Workflows also deletes all tasks associated with the workflow.

1. Open the navigation menu and click **Definitions**.
2. Click the **Workflows** tab.

The **Workflows** list page opens and displays the workflow definitions that have been created.

3. Identify the workflow definition that you want to delete.
4. Click  (**Delete**).
5. Select the version of the workflow definition that you want to delete.
6. When prompted, confirm the deletion.

2.2 Manage Workflow Executions

Start workflows asynchronously or synchronously, pass input data, set a version, correlation ID, priority, or idempotency key, and track the returned workflow ID. You can view execution status, include task details, find workflows by correlation ID, list running workflows, search workflow history, pause or resume an execution, skip a task, rerun, restart, retry, terminate, or remove workflow executions.

A workflow execution is a running or completed instance of a workflow definition. MicroTx Workflows stores execution details so you can track workflow progress, inspect task status, review execution history, and troubleshoot failures.

You can start a workflow asynchronously or synchronously, pass workflow input, specify a workflow version, use correlation IDs to group related executions, and use idempotency keys to avoid duplicate execution requests. After a workflow starts, you can view its status by workflow ID, include task details, search execution history, find executions by correlation ID, and list running workflows.

You can also manage active or completed executions by pausing, resuming, retrying, rerunning, restarting, skipping tasks, terminating, or removing workflow executions.

Topics

- [Run or Execute a Workflow](#)
Use Workbench to run a workflow from the MicroTx Workflows console. It is useful for testing workflow definitions, validating input, and inspecting execution results.

- [About Workflow Status](#)
A workflow status indicates the current state of a workflow execution. Understanding workflow statuses helps you determine which actions are available, whether the workflow is still active, and how to troubleshoot failures.
- [About Idempotent Workflows and Tasks](#)
Idempotency ensures that repeated attempts to process the same logical request do not create duplicate workflow executions or duplicate task side effects. This is useful in distributed systems where clients, schedulers, workers, or AI agents may retry an operation after a timeout, network failure, or partial failure.
- [Update and Rerun a Workflow Execution](#)
Workbench stores the details of the executed workflows in your browser's local storage.
- [View Details of a Workflow Execution](#)
- [Search and View Executed Workflows](#)
You can search for workflow executions and view execution details such as status, start time, end time, failed tasks, input and output payload paths, and the reason for failure or termination.
- [Perform Actions on Multiple Executed Workflows](#)
You can perform the same action on multiple workflow executions at the same time. Bulk actions are applied independently to each selected workflow. If an action succeeds for some workflows and fails for others, MicroTx Workflows displays the successful workflow IDs and the reason for each failure.
- [Handle Workflow Execution Failures](#)
Learn about the issues you may encounter, how to identify their causes, and how to work around them.

2.2.1 Run or Execute a Workflow

Use Workbench to run a workflow from the MicroTx Workflows console. It is useful for testing workflow definitions, validating input, and inspecting execution results.

You can also use Workbench to run production workflows, although production applications typically invoke workflows through REST APIs or SDKs. Workflow executions started from Workbench, REST APIs, or SDKs are stored and can be viewed in workflow execution history.

Before running a workflow, review the workflow definition and identify the input parameters that the workflow requires.

To run a workflow from Workbench:

1. Open the navigation menu and click **Workbench**.
2. Select the name of the workflow that you want to run from the **Workflow Name** drop-down list.
3. Select the version of the workflow that you want to run in the **Workflow Version** drop-down list.

If you do not select a version, MicroTx Workflows uses the configured default or the latest version of the selected workflow.

4. In the **Input Parameters** group, provide values for the required input parameters of the workflow. You can either provide all the input parameters in JSON format or select **Form** to provide input parameters as a key-value pair.

- Select **JSON** to enter a valid JSON object that contains the input values required by the workflow. For example, if the workflow definition references `$ {workflow.input.filePath}`, include `filePath` in the input JSON.
 - Select **Form**, and then click **Add Parameter**. Enter each input parameter as a key-value pair and specify the data type, such as string or number.
5. Optional. In the **Correlation ID** field, enter an ID to logically group related workflow executions. For example, you can use correlation ID to group executions related to the same business request, order, transaction, or customer.

Unlike the workflow ID, which uniquely identifies one execution, you can use the same correlation ID across multiple workflow executions.

6. Optional. In the **Idempotency Key** field, enter a unique key for the workflow execution request.

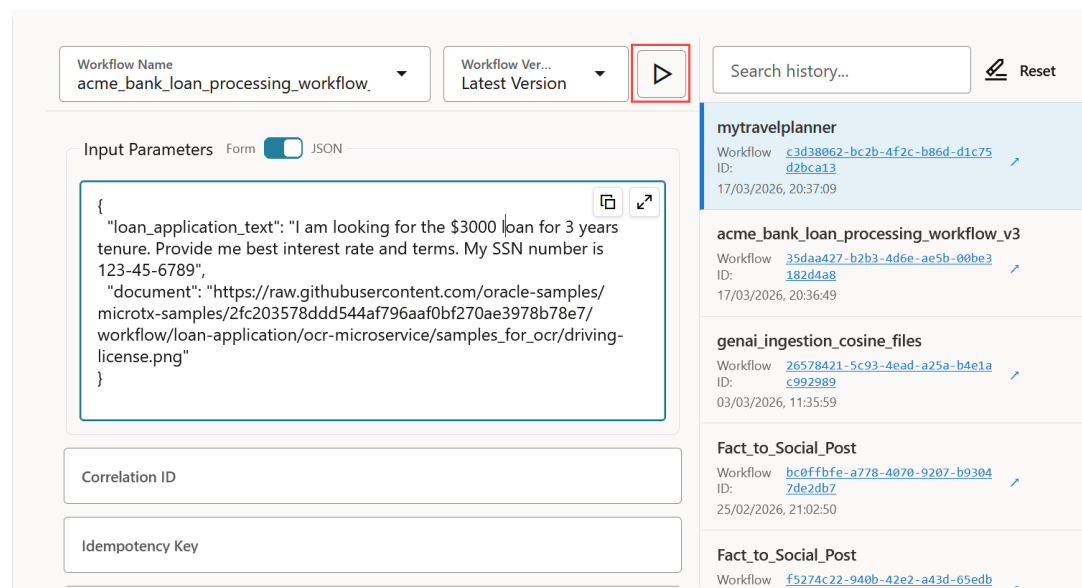
Use an idempotency key to prevent duplicate workflow executions when the same request is submitted more than once, such as during retries caused by network failures or client timeouts. If another execution already exists for the same workflow and idempotency key, MicroTx Workflows uses that key to detect the duplicate request instead of starting an unintended duplicate execution.

If you do not enter a value, MicroTx Workflows creates a unique value because every workflow execution is mapped to a unique idempotent key.

7. Optional. In the **Task to Domain (JSON)** field, enter the names of the domains in which you want to run a task.

For more information, see [Task Domains](#).

8. Click  (**Execute**) to run the selected workflow. The following image displays sample data as input parameters for a selected workflow.



The screenshot shows the MicroTx Workflows interface. At the top, there are dropdowns for 'Workflow Name' (set to 'acme_bank_loan_processing_workflow') and 'Workflow Ver...' (set to 'Latest Version'). To the right is a search bar and a 'Reset' button. Below these is a large text area for 'Input Parameters' with a toggle for 'Form' and 'JSON' (currently 'JSON' is selected). The JSON input contains a sample loan application. Below the input area are fields for 'Correlation ID' and 'Idempotency Key'. On the right side, a list of workflow executions is displayed, including 'mytravelplanner', 'acme_bank_loan_processing_workflow_v3', 'genai_ingestion_cosine_files', and 'Fact_to_Social_Post'. Each entry shows the workflow name, ID, and a timestamp.

In the pane on the right side of the window, a new workflow execution ID is displayed.

Workflow Name: **acme_bank_loan_processing_workflow_v3**

Workflow Ver...: **Latest Version**

Search history... **Reset**

Input Parameters Form ☒ JSON

```
{
  "loan_application_text": "I am looking for the $3000 loan for 3 years
  tenure. Provide me best interest rate and terms. My SSN number is
  123-45-6789",
  "document": "https://raw.githubusercontent.com/oracle-samples/
  microtx-samples/2fc203578ddd544af796aaf0bf270ae3978b78e7/
  workflow/loan-application/ocr-microservice/samples_for_ocr/driving-
  license.png"
}
```

Correlation ID

Idempotency Key

acme_bank_loan_processing_workflow_v3
 Workflow: [2ea980da-9274-428d-81e4-1e4ed](#)
 ID: [1dba477](#)
 17/03/2026, 20:57:26

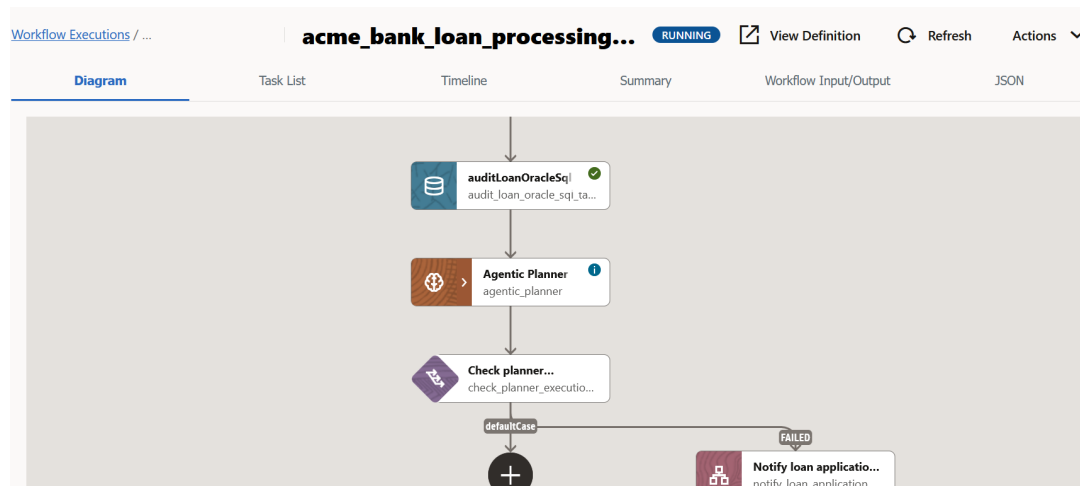
mytravelplanner
 Workflow: [c3d38062-bc2b-4f2c-b86d-d1c75](#)
 ID: [d2bca13](#)
 17/03/2026, 20:37:09

acme_bank_loan_processing_workflow_v3
 Workflow: [35daa427-b2b3-4d6e-ae5b-00be3](#)
 ID: [182d4a8](#)
 17/03/2026, 20:36:49

genai_ingestion_cosine_files
 Workflow: [26578421-5c93-4ead-a25a-b4e1a](#)
 ID: [c992989](#)
 03/03/2026, 11:35:59

Fact_to_Social_Post
 Workflow: [bc0ffbf6-a778-4070-9207-b9304](#)

- Click the workflow execution ID. The status of the workflow execution is displayed as shown in the following image for a sample workflow. Click the **Diagram** tab see a visual depiction of all the components of a workflow.



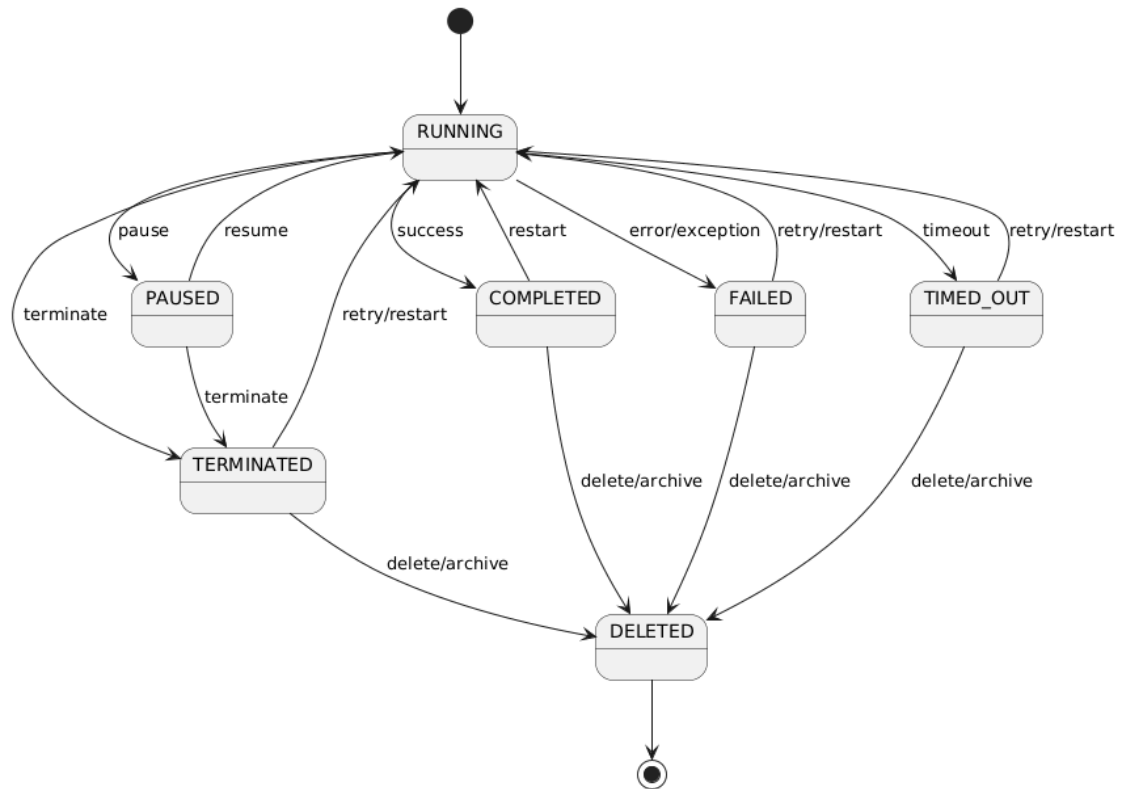
- Click **Refresh** to view the updated status of the workflow after a few seconds. The time taken to run a workflow depends on the complexity of the workflow. A green check mark indicates that a task has been executed successfully. For more details, see [View Details of a Workflow Execution](#).
- When the workflow is executed successfully, the status of the workflow changes to **Completed**.

2.2.2 About Workflow Status

A workflow status indicates the current state of a workflow execution. Understanding workflow statuses helps you determine which actions are available, whether the workflow is still active, and how to troubleshoot failures.

A workflow execution starts in the **RUNNING** status. It can then complete successfully, fail, time out, be paused, be resumed, or be terminated. Some statuses are terminal, which means the workflow execution has ended and will not continue unless you retry or restart it.

The following image shows how the workflow moves from one status to another.



To check the current status of a workflow, see [View Details of a Workflow Execution](#).

The following table describes the different workflow statuses and lists the permitted actions that you can perform in each status. Actions attempted outside their allowed status do not result in a valid status change.

Workflow Status	Description	Permitted Actions
RUNNING	The workflow is currently active and tasks are being scheduled or executed	Pause, terminate
PAUSED	The workflow execution is paused. It does not continue until it is resumed. A user or program can resume or terminate the workflow.	Resume, terminate
COMPLETED	The workflow execution finished successfully and all required tasks were completed. This is a terminal status.	Delete, archive, restart
FAILED	The workflow execution failed because one or more tasks failed or an error occurred during execution. This is a terminal status.	Delete, archive, retry, restart

Workflow Status	Description	Permitted Actions
TERMINATED	The workflow execution was stopped before it completed, usually by a user, API call, event, or another workflow. This is a terminal status.	Delete, archive, retry, restart
TIMED_OUT	The workflow execution exceeded a configured workflow or task timeout. This is a terminal status.	Delete, archive, retry, restart
DELETED	Workflow has been deleted and is no longer available.	None

At any time during the **RUNNING** or **PAUSED** status, you can terminate a workflow, which moves it to the **TERMINATED** status.

During normal execution, if the workflow finishes successfully, it transitions from **RUNNING** to **COMPLETED**. If it encounters an error or exception, it moves from **RUNNING** to **FAILED**. If it exceeds a designated time limit, it transitions from **RUNNING** to **TIMED_OUT**. These three statuses—**COMPLETED**, **FAILED**, and **TIMED_OUT**—are considered end or terminal states under normal circumstances.

From the **COMPLETED**, **FAILED**, **TERMINATED**, or **TIMED_OUT** statuses, you can either delete or archive a workflow—both actions result in the workflow moving to a **DELETED** (or archived) status.

Additionally, you can reprocess a workflow after it has finished or failed. If a workflow is in the **TERMINATED**, **FAILED** or **TIMED_OUT** statuses, you can retry or restart the workflow, which sends it back to the **RUNNING** status and begins execution anew. If the workflow is in the **COMPLETED** status, you restart the workflow, but you cannot retry it. Restarting a workflow also brings the workflow back to the **RUNNING** status. Restart starts the workflow execution again from the beginning. Restart is available only for terminal workflows and depends on whether the workflow definition is restartable.

A workflow begins in **RUNNING** status. While it is running, you can pause it, which changes the status to **PAUSED**. A paused workflow can be resumed, which returns it to **RUNNING** and continues execution, or it can be terminated.

If the workflow finishes all required tasks successfully, it changes to **COMPLETED**. If execution fails, it changes to **FAILED**. If the workflow or a task exceeds its configured timeout, it changes to **TIMED_OUT**. If a user, API call, event, or another workflow stops the execution, it changes to **TERMINATED**.

COMPLETED, **FAILED**, **TERMINATED**, and **TIMED_OUT** are terminal statuses. Workflows in a terminal status do not continue automatically. Depending on the status and workflow definition, you can retry or restart the workflow. Retry resumes execution from **FAILED**, **TIMED_OUT**, or **CANCELED** tasks where retry is possible. Retry is not available for **COMPLETED** workflows. You can also remove terminal workflows, and optionally archive them during removal.

When you delete a workflow execution, MicroTx Workflows removes it from the data store. When archive is enabled, terminal workflow executions can be archived before removal.

2.2.3 About Idempotent Workflows and Tasks

Idempotency ensures that repeated attempts to process the same logical request do not create duplicate workflow executions or duplicate task side effects. This is useful in distributed

systems where clients, schedulers, workers, or AI agents may retry an operation after a timeout, network failure, or partial failure.

MicroTx Workflows applies idempotency at two levels: workflow level and task level.

Workflow-level idempotency

Workflow-level idempotency prevents the same workflow from being started more than once for the same logical request. MicroTx Workflows identifies an idempotent workflow request by using a combination of idempotency key, workflow name, and workflow version.

When you run a workflow, you can provide an idempotency key. If you do not provide one, MicroTx Workflows automatically generates a unique idempotency key for that execution. The generated key is included in the workflow output so that you can track the execution.

If you submit another request with the same idempotency key, workflow name, and workflow version, MicroTx Workflows handles the request based on the status of the existing workflow execution:

Existing Workflow Status	Behavior
Running	MicroTx Workflows rejects the duplicate request because the workflow is already in progress.
Completed	MicroTx Workflows returns the existing workflow execution instead of starting a new one.
Failed, terminated, or timed out	MicroTx Workflows rejects the request and reports that a previous workflow with the same idempotency key already exists in that status.

If there is no matching execution, MicroTx Workflows starts a new workflow execution.

To make client-side retries idempotent, use the same idempotency key when retrying the same logical request. If you leave the idempotency key empty each time, MicroTx Workflows generates a new key for each request, and each request is treated as a new workflow execution.

Let's understand how MicroTx Workflows provides idempotency at the workflow level with an example. Consider a workflow that contains a SQL Task to update the account balance in a table. The workflow has already been executed successfully once. If you submit another request to run the workflow again with the *same* idempotency key which was used in the previous execution, MicroTx Workflows checks if the account balance was already updated. If the account balance was already updated, MicroTx Workflows retrieves the balance stored in the database and returns the result without performing the operation again. In such a scenario, idempotency at the workflow level ensures that the account balance is not updated a second time and the workflow is not executed again.

Task-level idempotency

Task-level idempotency prevents supported tasks from applying the same side effects more than once during retries, restarts, reruns, or agent-driven re-inocations. MicroTx Workflows scopes task idempotency under the workflow idempotency context and the task reference name.

For supported task implementations, the effective task identity is based on the workflow idempotency context and the task reference name. For database-backed task idempotency, MicroTx Workflows uses a fenced token lock to ensure that only one executor can own and complete the task result for a given idempotency identity.

Agentic task and Agentic Planner task idempotency

Agentic task and Agentic Planner task use fenced token lock idempotency for every execution. They store the task execution state by using the workflow idempotency key and the task reference name. If a matching completed task result already exists, MicroTx Workflows restores the stored output instead of invoking the agent or planner again.

This prevents duplicate LLM calls, repeated tool invocations, and repeated MCP or external tool side effects when an agentic task is retried or resumed.

SQL Task, GenAI Ingestion Task, and TxEventQ Task idempotency

SQL task, GenAI Ingestion task, and TxEventQ task support task-level idempotency through task input options. To enable task-level idempotency for these tasks, set the `enableIdempotency` property to `true`.

You can also provide the name of an idempotent table to specify the database table used for the fenced idempotency lock. If you do not provide a table name, MicroTx Workflows uses the default table name, `fenced_task_idempotency_lock`.

For SQL task and GenAI Ingestion task, the idempotency table must exist in the customer database where the task performs its database operation. MicroTx Workflows uses the table to acquire a fenced lock and store the task outcome. The database operation and idempotency result are committed together. If the operation fails, the transaction is rolled back so that partial changes are not committed.

For SQL Task, you can enable idempotency when the task is enlisted in an XA transaction.

Simple tasks and worker tasks

MicroTx Workflows does not automatically enforce task-level idempotency for custom simple tasks or worker tasks. If your worker task performs side effects, such as updating a database, calling an external API, or publishing a message, you must provide custom code to implement idempotency in the worker or downstream service.

Advantages

- Prevents duplicate workflow starts for the same logical request.
- Prevents duplicate side effects for supported system tasks.
- Makes retries safer when clients, schedulers, workers, or agents re-submit the same request.
- Reduces operational risk from duplicate inserts, repeated notifications, repeated message publishing, double charging, or repeated AI/tool calls.
- Provides a consistent idempotency model for workflow execution and supported task execution.

2.2.4 Update and Rerun a Workflow Execution

Workbench stores the details of the executed workflows in your browser's local storage.

After you run a workflow, a new workflow execution ID is displayed in the pane on the right side of the window. You can search for a specific workflow by providing its ID or name. Click a specific entry to view the parameters used to run that workflow. You can edit the parameters, if required, and run the workflow again.

1. Open the navigation menu and click **Workbench**.

2. In the **Search history** field in the pane on the right side of the window, enter the partial or complete workflow name or workflow ID.

All the workflows that match the search criterion are displayed.

3. Click a workflow ID.

All the details, such as input parameters, that you had provided when you ran the workflow earlier are displayed on the screen as shown in the following image.

Workflow Name: **acme_bank_loan_processing_workflow_v3** | Workflow Ver...: **Latest Version** | Search history... | Reset

Input Parameters Form ☒ JSON

```
{
  "loan_application_text": "I am looking for the $3000 loan for 3 years
  tenure. Provide me best interest rate and terms. My SSN number is
  123-45-6789",
  "document": "https://raw.githubusercontent.com/oracle-samples/
  microtx-samples/2fc203578ddd544af796aaf0bf270ae3978b78e7/
  workflow/loan-application/ocr-microservice/samples_for_ocr/driving-
  license.png"
}
```

Correlation ID

Idempotency Key

acme_bank_loan_processing_workflow_v3
Workflow: [2ea980da-9274-428d-81e4-1e4ed](#)
ID: [1dba477](#)
17/03/2026, 20:57:26

mytravelplanner
Workflow: [c3d38062-bc2b-4f2c-b86d-d1c75](#)
ID: [d2bca13](#)
17/03/2026, 20:37:09

acme_bank_loan_processing_workflow_v3
Workflow: [35daa427-b2b3-4d6e-ae5b-00be3](#)
ID: [182d4a8](#)
17/03/2026, 20:36:49

genai_ingestion_cosine_files
Workflow: [26578421-5c93-4ead-a25a-b4e1a](#)
ID: [c992989](#)
03/03/2026, 11:35:59

Fact_to_Social_Post
Workflow: [bc0ffbf-e-a778-4070-9207-b9304](#)

4. View all the details, such as workflow version, input parameters, and the domains in which you want to run the task. If required, update the details. You can update all workflow execution parameters, except the name of the workflow.

The following image displays the updated values of input parameters for the selected workflow.

Workflow Name: **acme_bank_loan_processing_workflow_v3** | Workflow Ver...: **Latest Version** | Search history... | Reset

Input Parameters Form ☒ JSON

```
{
  "loan_application_text": "I am looking for the $5000 loan for 5 years
  tenure. Provide me best interest rate and terms. My SSN number is
  123-45-6789",
  "document": "https://raw.githubusercontent.com/oracle-samples/
  microtx-samples/2fc203578ddd544af796aaf0bf270ae3978b78e7/
  workflow/loan-application/ocr-microservice/samples_for_ocr/driving-
  license.png"
}
```

Correlation ID

Idempotency Key

acme_bank_loan_processing_workflow_v3
Workflow: [2ea980da-9274-428d-81e4-1e4ed](#)
ID: [1dba477](#)
17/03/2026, 20:57:26

mytravelplanner
Workflow: [c3d38062-bc2b-4f2c-b86d-d1c75](#)
ID: [d2bca13](#)
17/03/2026, 20:37:09

acme_bank_loan_processing_workflow_v3
Workflow: [35daa427-b2b3-4d6e-ae5b-00be3](#)
ID: [182d4a8](#)
17/03/2026, 20:36:49

genai_ingestion_cosine_files
Workflow: [26578421-5c93-4ead-a25a-b4e1a](#)
ID: [c992989](#)
03/03/2026, 11:35:59

Fact_to_Social_Post
Workflow: [bc0ffbf-e-a778-4070-9207-b9304](#)

- Optional. In the **Correlation ID** field, enter an ID to logically group related workflow executions. For example, you can use correlation ID to group executions related to the same business request, order, transaction, or customer.

Unlike the workflow ID, which uniquely identifies one execution, you can use the same correlation ID across multiple workflow executions.

- Optional. In the **Idempotency Key** field, enter an idempotency key to prevent duplicate workflow executions. See [About Idempotent Workflows and Tasks](#).

If you do not enter a value, MicroTx Workflows creates a unique value because every workflow execution is mapped to a unique idempotent key.

- Click **(Execute)** to run the workflow.

A new workflow ID is created every time you run a workflow. In the pane on the right side of the window, a new workflow execution ID is displayed.

The screenshot displays the MicroTx Workflows interface. On the left, the 'Workflow Name' is 'acme_bank_loan_processing_workflow' and the 'Workflow Ver...' is 'Latest Version'. Below this, the 'Input Parameters' section shows a JSON object with fields for 'loan_application_text' and 'document'. The 'Correlation ID' and 'Idempotency Key' fields are empty. On the right, a list of workflow executions is shown, with the top entry highlighted in blue and a red box around it. This entry is for 'acme_bank_loan_processing_workflow_v3' with Workflow ID 'b3ce5e1f-de9f-4df8-8a62-1869c' and Execution ID '936ae2e', dated 18/03/2026, 12:38:29. Other entries include 'acme_bank_loan_processing_workflow_v3' (dated 17/03/2026, 20:57:26), 'mytravelplanner' (dated 17/03/2026, 20:37:09), 'acme_bank_loan_processing_workflow_v3' (dated 17/03/2026, 20:36:49), and 'genai_ingestion_cosine_files'.

- Click the workflow execution ID.

The status of the workflow execution is displayed. See [View Details of a Workflow Execution](#).

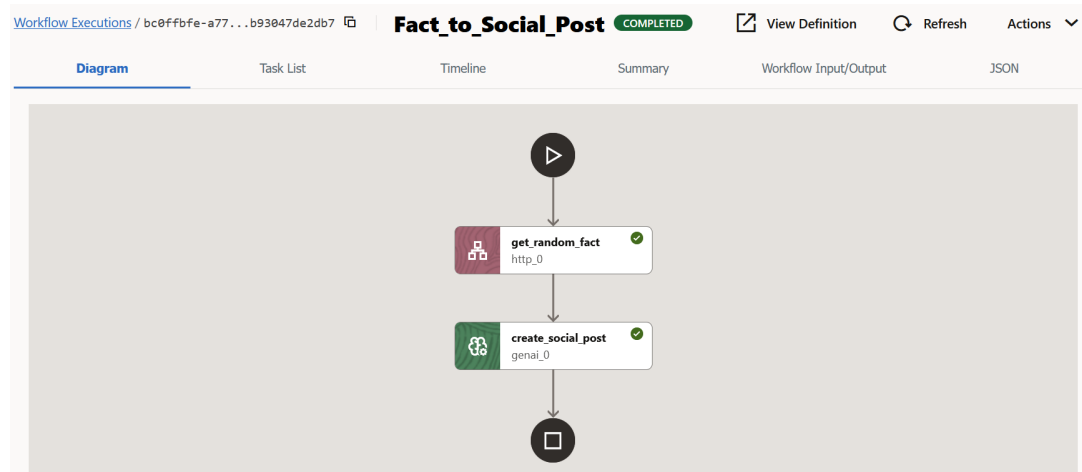
If you want to clear all the details of the executed workflows in your browser's local storage, click **Reset**.

2.2.5 View Details of a Workflow Execution

- Open the navigation menu and click **Executions**.
- Click the **Workflows** tab.
Details of all executed workflows are displayed.
- Optional. Apply search criteria to filter and identify the workflow execution that you want to view. See [Search and View Executed Workflows](#).
- Click the Workflow ID that you want to view.

The status of the workflow execution is displayed as shown in the following images for a sample workflow that generates a random fact. Click one or more of the following tabs to view more details.

5. The **Diagram** tab visually depicts all the components of a workflow. A green check mark indicates that a task has been executed successfully.



6. Click the **Task List** tab to view a list of all tasks and their details as shown in the following image.

The screenshot shows the 'Task List' tab of the Oracle Workflow Execution interface. The table lists two tasks: 'get_random_fact' (http_0) and 'create_social_post' (genai_0). Both tasks are marked as 'COMPLETED'.

Seq	Task Id	Task Ref	Status	Attempt	Type	Start Time	End Time
1	92c0...f2...	http_0	COMPLETED	1	HTTP	25/02/2026, 21:02:56	25/02/2026, 21:02:56
2	03c4...b7...	genai_0	COMPLETED	1	GENAI_TASK	25/02/2026, 21:02:57	25/02/2026, 21:03:00

- **Seq:** Sequence in which the tasks were executed.
- **Task ID:** A unique value. Inside a workflow execution, task ID is created as a unique identifier for each task that runs inside a workflow execution boundary.
- **Task Ref:** A unique value that is used to refer to the task within a workflow.
- **Status:** Status of the task.
- **Attempt:** Number of retries, if any. 1 indicates that the task was executed successfully in the first attempt.
- **Type:** The type of task, such as Agentic Planner or GenAI.
- **Start Time:** The time at which the task execution was started in the format, dd/MM/yyyy, HH:mm:ss. For example, 29/11/2025 17:40:56.
- **End Time:** The time at which the task execution was completed in the format, dd/MM/yyyy, HH:mm:ss. For example, 30/11/2025 13:07:03.

Workflow Executions / bc0ffbfef-a77...b93047de2db7 **Fact_to_Social_Post** COMPLETED View Definition Refresh Actions

Diagram Task List Timeline Summary Workflow Input/Output JSON

Table View Structured View Collapse all

- http_0 COMPLETED

Task Id	Type	Ref	Start Time	End Time
92c0383e-3389-4b95-a37e-f8f75799f273	HTTP	http_0	25/02/2026, 21:02:56	25/02/2026, 21:02:56

 Status: COMPLETED
 Execution Payload: Object(1)
 response: Object(4)
- genai_0 COMPLETED

Task Id	Type	Ref	Start Time	End Time

7. Click the **Timeline** tab to see a visual representation on a timeline about the time period taken to execute a workflow, when each task was executed, how much time it took to run the task, and for how long the task had to wait in queue. It also provides information about whether tasks were executed in parallel or in sequence. Tasks are highlighted in different colors to indicate whether the tasks were executed successfully.

Workflow Executions / bc0ffbfef-a77...b93047de2db7 **Fact_to_Social_Post** COMPLETED View Definition Refresh Actions

Diagram Task List Timeline Summary Workflow Input/Output JSON

9:03 PM

http_0 5s 55ms

genai_0 3s 190ms

Workflow Start: 25/02/2026, 21:02:51
 Workflow End: 25/02/2026, 21:03:00
 Total Duration: 9s 731ms
 Tasks Run: 2
 Total Queue Wait: 5s 957ms

QUEUED COMPLETED

8. Click the **Summary** tab to view details about the workflow. If a workflow fails, refer to the information provided in the **Reason** field to troubleshoot the workflow.

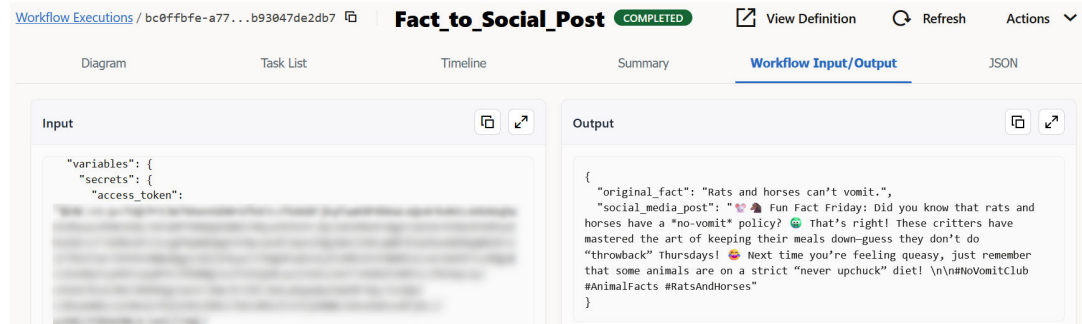
Workflow Executions / bc0ffbfef-a77...b93047de2db7 **Fact_to_Social_Post** COMPLETED View Definition Refresh Actions

Diagram Task List Timeline Summary Workflow Input/Output JSON

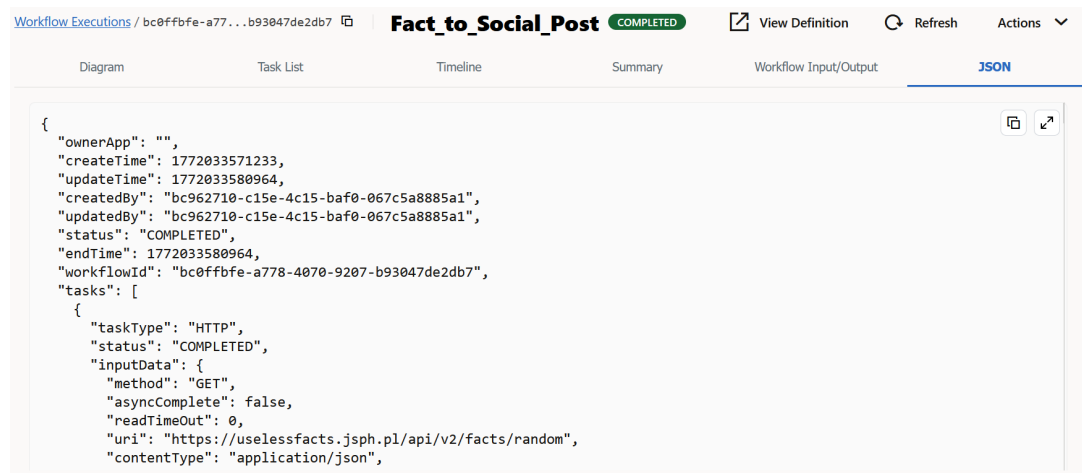
Execution Summary COMPLETED

Workflow ID	bc0ffbfef-a778-4070-9207-b93047de2db7
Workflow Name	Fact_to_Social_Post
Version	1
Started	25/02/2026, 21:02:51
Ended	25/02/2026, 21:03:00
Duration	9s

- Click the **Workflow Input/Output** tab to view the input and output parameters of a workflow in JSON format.



- Click the **JSON** tab to view the JSON code of the entire workflow execution.



2.2.6 Search and View Executed Workflows

You can search for workflow executions and view execution details such as status, start time, end time, failed tasks, input and output payload paths, and the reason for failure or termination.

- Open the navigation menu and click **Executions**.
- Click the **Workflows** tab.
Details of all executed workflows are displayed.
- Use one or more of the following options to filter the executed workflows.
 - Workflow Name:** Select the name of the workflow you want to search.
 - Workflow ID:** Enter the unique identifier to view details of a specific workflow. A unique Workflow ID is generated every time you run a workflow.
 - Status:** Filter workflows by their execution status, such as running, failed, completed, or paused.
 - Start Time From:** Select a date and time range to narrow down searches based on the duration from when workflows were executed. Use this option with the **Start Time To** field to retrieve workflows that were started within a specific duration.

- **Start Time To:** Select a date and time range to narrow down searches based on the duration until when workflows were executed. Use this option with the **Start Time From** field to retrieve workflows that were started within a specific duration.
4. Click **Search** to apply your search criteria to display matching workflow executions.
 5. Optional. You can customize the columns displayed in your workflow list. Click **Columns**, select all the details you want to view as mentioned in the following list, and then click **Close**.
 - **Start Time:** The date and time when the workflow execution began.
 - **Workflow ID:** The unique identifier assigned to each workflow instance.
 - **Workflow Name:** The descriptive name of the workflow definition being executed.
 - **Created By:** The user or service that initiated the workflow instance.
 - **End Time:** Date and time when the workflow execution reached a terminal state. This value is empty for workflows that are still running.
 - **Status:** Current status of the workflow execution.
 - **Version:** The version number of the workflow definition used for this execution.
 - **Correlation ID:** Identifier used to associate related workflow executions. You can use this value to group executions related to the same business request, order, transaction, or customer.
 - **Update Time:** The most recent date and time when any changes occurred to the workflow instance.
 - **Reason:** Reason why the workflow did not complete successfully. This value is populated by MicroTx Workflows when a workflow fails, times out, or is terminated. It may include an error message from a failed task, a system-generated reason, or a termination reason provided by the caller.
 - **Execution Time:** Total time taken by the workflow execution. For completed workflows, this is calculated from start time to end time.
 - **Failed Ref Tasks:** Task reference names of the tasks that failed in the workflow execution. Task reference names are the unique task identifiers used within a workflow definition.
 - **Event:** Event that triggered the workflow execution, if the workflow was started by an event handler.
 - **Input Payload Path:** External storage path for the workflow input payload, if the input payload was stored externally.
 - **Output Payload Path:** External storage path for the workflow output payload, if the output payload was stored externally.
 - **Priority:** Execution priority assigned when the workflow was started. Priority is a numeric value from 0 to 99, where a higher value gives the workflow higher scheduling priority.

The selected details of the executed workflows are displayed. To reset the selected columns, click **Reset to default**.

You can continue to filter the search results or click **Refresh** to reload results without any search criteria.

2.2.7 Perform Actions on Multiple Executed Workflows

You can perform the same action on multiple workflow executions at the same time. Bulk actions are applied independently to each selected workflow. If an action succeeds for some workflows and fails for others, MicroTx Workflows displays the successful workflow IDs and the reason for each failure.

1. Open the navigation menu and click **Executions**.
2. Click the **Workflows** tab.
Details of all executed workflows are displayed.
3. Optional. Apply search criteria to filter the search results and view the executed workflows on which you want to perform an action. See [Search and View Executed Workflows](#).
4. Select one or more workflows.
5. Click **Bulk Actions**, and then click one of the following actions.
 - **Pause**: Pauses the selected workflow executions that are not in a terminal state. A paused workflow stops further processing until it is resumed. You cannot pause a workflow that is the completed, failed, timed out, or terminated state.
 - **Resume**: Resumes selected workflow executions that are in the paused state. The workflow status changes back to running and processing continues from the point where the workflow was paused.
 - **Terminate**: Stops selected workflow executions that are still active. The workflow status changes to terminated, and pending tasks are removed from processing queues. A terminated workflow cannot be resumed, but it can be retried or restarted.
 - **Terminate With Reason**: Stops selected workflow executions and records the reason that you enter. The reason is stored with the workflow execution and can be used for auditing and troubleshooting.
 - **Delete**: Permanently removes the selected workflow executions and their associated task records from the active workflow store. Deleted workflows cannot be recovered.

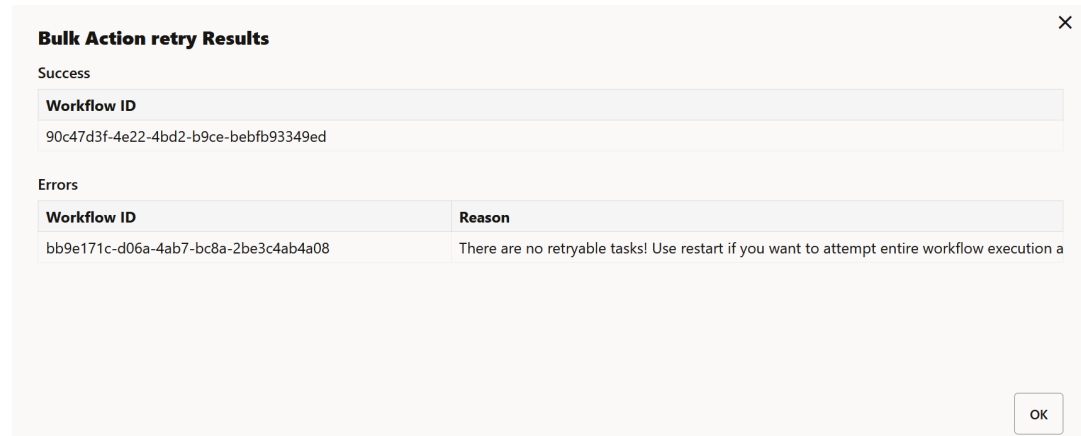
Caution

Use **Delete** only when the workflow execution history is no longer required. If you need to retain execution history for troubleshooting or audit purposes, use **Archive** instead.

- **Archive**: Removes the selected workflow executions from the active workflow store but retains an archived execution record for historical reference. Only workflows in a terminal state, such as completed, failed, terminated, or timed out can be archived.
- **Retry**: Retries selected workflows that are in a terminal state. MicroTx Workflows identifies the last unsuccessful task instances, such as failed, timed out, or canceled tasks, and reschedules those tasks. The workflow status changes back to running. If there are no retryable tasks, use restart instead.
- **Restart with current def**: Restarts selected workflows from the beginning by using the workflow definition associated with the original execution. Existing task execution state is cleared, the workflow status changes to running, and the workflow starts again.

- **Restart with latest def:** Restarts selected workflows from the beginning by using the latest available version of the workflow definition. Use this option when you want the restarted execution to pick up changes made after the original workflow execution.

The selected action is performed on the workflows based on the current state of the workflow. If an action cannot be performed on a selected workflow, the reason is displayed as shown in the following figure.



6. Click **OK**.

2.2.8 Handle Workflow Execution Failures

Learn about the issues you may encounter, how to identify their causes, and how to work around them.

MicroTx Workflows is designed for enterprise orchestration scenarios where reliability and traceability are important. It supports long-running workflows, retries, task timeouts, workflow status management, idempotency, workflow execution search, and detailed execution views. You can inspect each workflow execution, view task-level details, review logs, and understand how data moved through the workflow. This helps teams diagnose failures, validate business outcomes, and maintain operational visibility across distributed applications.

For additional information, see [Handling Workflow Errors](#) in open-source Conductor documentation.

Topics

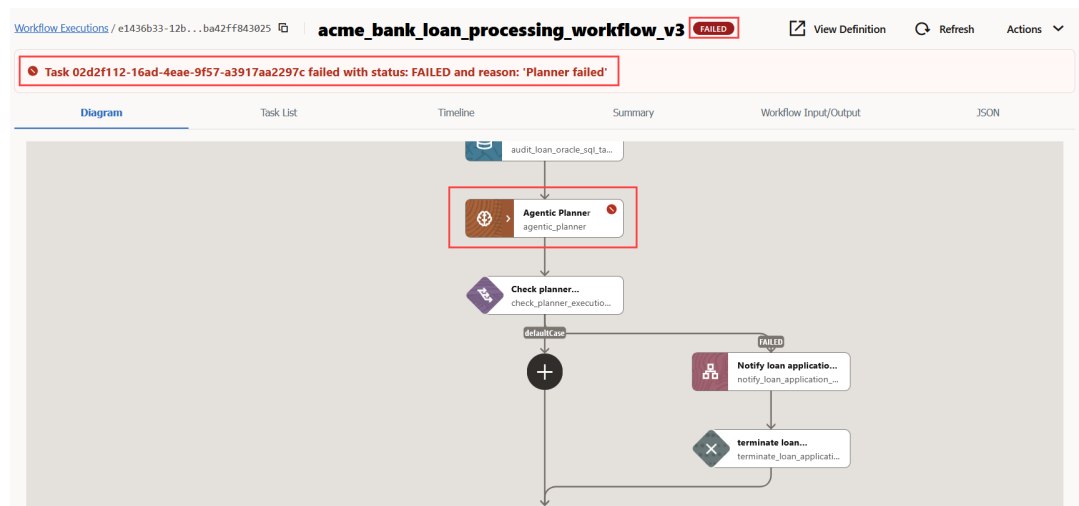
- [Identify Failed Tasks in a Workflow Execution](#)
If a workflow execution fails, review the **Timeline** tab and status indicators to quickly locate failed tasks, then refer to the **Summary** or **Task List** tabs for more troubleshooting details.
- [Specify a Failure Workflow](#)
Update the workflow definition to enter the name of the workflow that you want to trigger in case of any failure. Use failure workflows to perform clean up or other actions on failure.
- [Handling Task Timeout Issues](#)
If a task times out, review the workflow execution details to identify the timed-out task and the reason for incompleteness.

2.2.8.1 Identify Failed Tasks in a Workflow Execution

If a workflow execution fails, review the **Timeline** tab and status indicators to quickly locate failed tasks, then refer to the **Summary** or **Task List** tabs for more troubleshooting details.

1. Open the navigation menu and click **Executions**.
2. Click the **Workflows** tab.
Details of all executed workflows are displayed.
3. Optional. Apply search criteria to filter and identify the workflow executions which are in the FAILED or TIMED_OUT status. See [Search and View Executed Workflows](#).
4. Click the Workflow ID that you want to view.

The status of the workflow execution and error message is displayed as shown in the following image for a sample workflow that validates documents and processes a bank loan. Note the status of the workflow execution and the error message. In the **Diagram** tab, you can identify the task that failed.



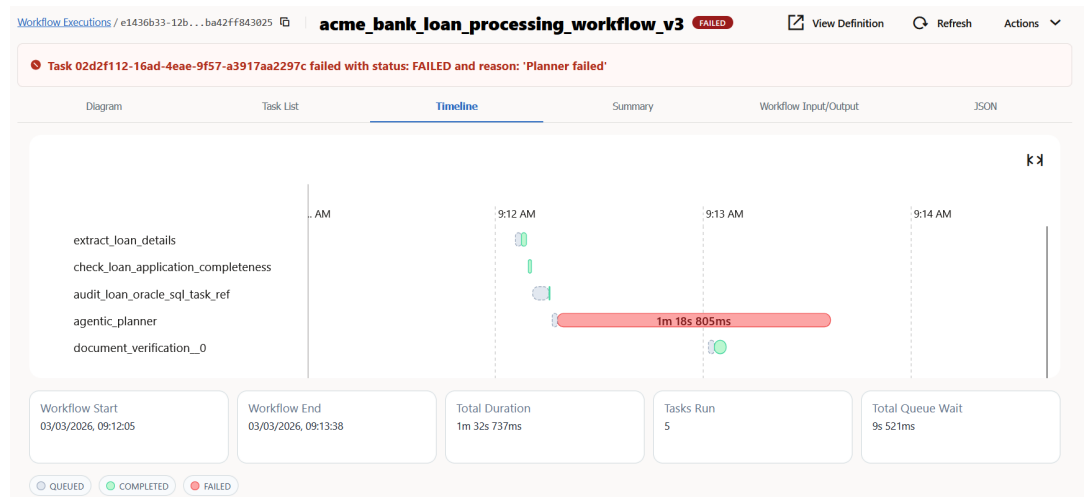
5. Click the **Summary** tab. The **Reason** field describes the cause for failure.

The screenshot shows the 'Summary' tab of the same workflow execution. It displays the following information:

- Execution Summary** (with a 'FAILED' status indicator)
- Workflow ID:** e1436b33-12b2-4c49-94d7-ba42ff843025
- Workflow Name:** acme_bank_loan_processing_workflow_v3
- Version:** 1
- Started:** 03/03/2026, 09:12:05
- Ended:** 03/03/2026, 09:13:38
- Duration:** 1m 32s
- Reason:** Task 02d2f112-16ad-4eae-9f57-a3917aa2297c failed with status: FAILED and reason: 'Planner failed'

A 'Copy reason' link is available below the reason field.

6. Click the **Timeline** tab to see a visual representation of the failed tasks on a timeline as shown in the following image.



7. Click the **Task List** tab to identify tasks in the failed status as shown in the following image.

Workflow Executions / e1436b33-12b...ba42ff843025 **acme_bank_loan_processing_workflow_v3** FAILED View Definition Refresh Actions

Task 02d2f112-16ad-4eae-9f57-a3917aa2297c failed with status: FAILED and reason: 'Planner failed'

Diagram **Task List** Timeline Summary Workflow Input/Output JSON

Table View Structured View

Seq	Task Id	Task Ref	Status	Attempt	Type	Start Time	End Time
1	0722...3cdc	extract_loan_details	COMPLETED	1	GENAI_TASK	03/03/2026, 09:12:07	03/03/2026, 09:12:09
2	c2d9...5369	check_loan_application_completeness	COMPLETED	1	SWITCH	03/03/2026, 09:12:09	03/03/2026, 09:12:10
3	b25a...deea	audit_loan_oracle_sql_task_ref	COMPLETED	1	SQL	03/03/2026, 09:12:15	03/03/2026, 09:12:15
4	02d2...297c	agentic_planner	FAILED	1	AGENTIC_PLANNER	03/03/2026, 09:12:17	03/03/2026, 09:13:36
5	631a...6861	document_verification_0	COMPLETED	1	AGENTIC_TASK	03/03/2026, 09:13:03	03/03/2026, 09:13:06

8. Click the task reference for the failed task.

Details of the task are displayed in the right pane as shown in the following image.

Workflow Executions / e1436b33-12b...ba42ff843025 **acme_bank_loan_processing_workflow_v3** FAILED View Definition Refresh Actions

Task 02d2f112-16ad-4eae-9f57-a3917aa2297c failed with status: FAILED and reason: 'Planner failed'

Diagram **Task List** Timeline Summary Workflow Input/Output JSON

Table View Structured View

Seq	Task Id	Task Ref	Status	Attempt	Type	Start Time	End Time
1	0722...	extract_loan_det...	COMP	1	GENA...	03/03/20...	03/03/20...
2	c2d...	check_loan_appl...	COMP	1	SWIT...	03/03/20...	03/03/20...
3	b25...	audit_loan_oracl...	COMP	1	SQL	03/03/20...	03/03/20...
4	02d...	agentic_planner	FAIL	1	AGEN...	03/03/20...	03/03/20...
5	631...	document_verific...	COMP	1	AGEN...	03/03/20...	03/03/20...

agentic_planner FAILED 02d2f112-16a...a3917aa2297c

Summary Input Output Logs JSON Definition

Task type: AGENTIC_PLANNER
Status: FAILED
Task name: Agentic Planner
Task reference: agentic_planner
Task execution id: 02d2f112-16ad-4eae-9f57-a3917aa2297c
Retry count: 0
Scheduled time: 03/03/2026, 09:12:16
Start time: 03/03/2026, 09:12:17
End time: 03/03/2026, 09:13:36
Queue time: 14d 19h 14m 38s
Duration: 1m 18s

- Click the **Logs** tab in the right pane to view additional details that can help you troubleshoot as shown in the following image.

The screenshot displays the Oracle Workflow Executions interface for a workflow named 'acme_bank_loan_processing_workflow_v3'. A red banner at the top indicates that 'Task 02d2f112-16ad-4eae-9f57-a3917aa2297c failed with status: FAILED and reason: 'Planner failed''. Below this, the 'Task List' tab is active, showing a table of tasks. Task 4, 'agentic_planner', is highlighted with a red 'FAILED' status. To the right, the 'Logs' tab for this task is open, showing a series of log entries. The final log entry states 'Planner failed. null'.

Seq	Task Id	Task Ref	Stat	At	Typ	Start Time	End Time
1	072...	extract_loan_det...	COMP	1	GENA...	03/03/20...	03/...
2	c2d...	check_loan_appli...	COMP	1	SWIT...	03/03/20...	03/...
3	b25...	audit_loan_oracl...	COMP	1	SQL	03/03/20...	03/...
4	02d...	agentic_planner	FAILED	1	AGEN...	03/03/20...	03/...
5	631...	document_verific...	COMP	1	AGEN...	03/03/20...	03/...

Log Entries for 'agentic_planner':

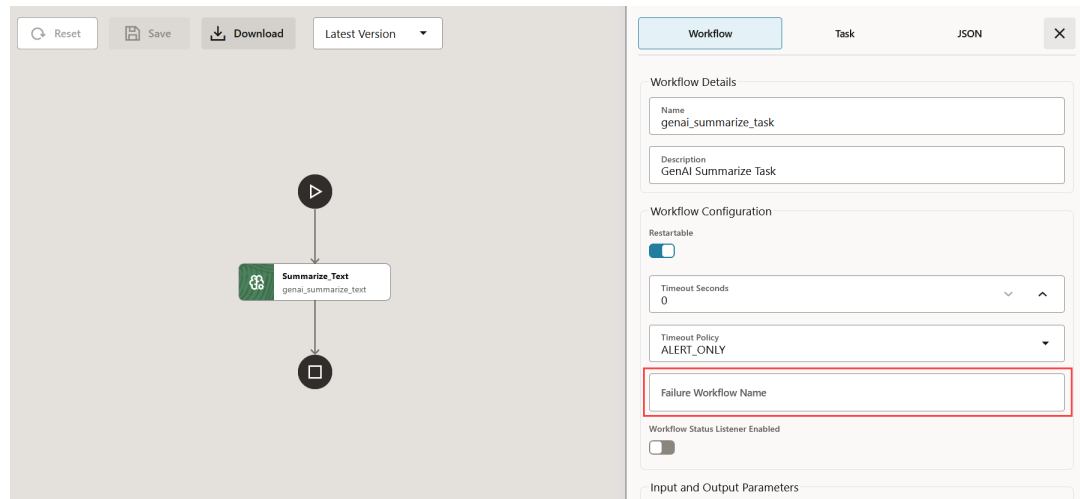
- 18/03/2026, 04:52:11: Scheduling the tasks for execution. Tasks ->
- 18/03/2026, 04:52:11: Scheduled Task : document_verification
- 18/03/2026, 04:52:11: Planned task(document_verification) execution finished with status COMPLETED.
- 18/03/2026, 04:52:11: Identified 3 tool(s) to be included in the planner prompt
- 18/03/2026, 04:52:11: Executing Agentic Planner prompt
- 18/03/2026, 04:52:11: { "plan": [], "data": { "status": "FAILED" } }
- 18/03/2026, 04:52:11: Planner failed. null

2.2.8.2 Specify a Failure Workflow

Update the workflow definition to enter the name of the workflow that you want to trigger in case of any failure. Use failure workflows to perform clean up or other actions on failure.

Before you begin, ensure that you have created a failure workflow.

- Open the navigation menu and click **Definitions**.
- Click the **Workflows** tab.
The **Workflows** list page opens and displays the workflow definitions that have been created.
- Identify the workflow definition that you want to update.
- Click **(Edit)**.
- Select the version of the workflow definition that you want to update from the **Version** drop-down list. By default the latest version of the workflow definition is displayed.
- In the **Failure Workflow Name** field, specify the name of the workflow that is executed in case of any failures.



7. Click **Save**.
8. MicroTx Workflows displays the changes in JSON code as shown in the following figure. Review all the changes.
9. Click **Confirm Save** to save the changes.

2.2.8.3 Handling Task Timeout Issues

If a task times out, review the workflow execution details to identify the timed-out task and the reason for incompleteness.

To handle task timeout issues, you can take one or more of the following actions:

- Increase poll timeout if tasks are timing out before workers pick them up.
- Add or scale worker instances if poll timeouts indicate insufficient worker capacity.
- Increase response timeout if the worker needs more time before sending progress updates.
- Update the worker to send IN_PROGRESS updates for long-running tasks.
- Increase timeout if the task legitimately needs more time to finish.
- Configure `retryCount`, `retryLogic`, and `retryDelaySeconds` so transient failures can be retried.
- Retry or restart the workflow after fixing the worker, timeout, or downstream service issue.

2.3 Manage Schedules to Run Workflows

Create, update, retrieve, search, and delete scheduled workflow definitions. Use schedules to run workflows automatically based on a configured schedule, such as daily, weekly, monthly, or another supported schedule expression. Scheduling APIs are available only when the scheduler feature is enabled.

MicroTx Workflows runs workflows automatically based on the defined schedule, such as daily, weekly, or monthly.

For example, you may want to run a workflow to generate and analyze reports about resource usage at different times of a weekday. To automate this process, create a schedule definition to specify the frequency at which a workflow is executed.

Topics

- [Create a Schedule Definition](#)
Define the frequency at which you want to execute a workflow, such as weekly, monthly, or daily.
- [Update a Schedule Definition](#)
You can update all the field values you specified when creating a schedule definition, except for the name.
- [View a Schedule Definition](#)
View information about a schedule definition.
- [Delete a Schedule Definition](#)
You can delete a schedule definition when it is no longer required.

2.3.1 Create a Schedule Definition

Define the frequency at which you want to execute a workflow, such as weekly, monthly, or daily.

Before you create a schedule definition, ensure that `scheduler.enabled` is set to `true` in the `values.yaml` file. See *MicroTx Workflows Server Configuration* in *MicroTx Installation and Configuration Guide*.

To create a schedule definition:

1. Open the navigation menu and click **Definitions**.
2. Click the **Schedules** tab.

The **Schedules** list page opens. All the schedules that you have defined are displayed in a table.

3. Click  (Add New Schedule).

The New Schedule Definition dialog box is displayed.

4. Enter the following information to schedule a workflow execution.
 - **Name:** Mandatory. Enter a unique name to identify the schedule. The name can be up to 128-characters long. Use only letters, numbers, underscores (`_`), and hyphens (`-`). Spaces and other special characters are not supported. You cannot change this name later.
 - **Workflow Name:** Select a workflow.
 - **Workflow Version:** Select the version of the workflow.
 - **Status:** Enable a schedule to run workflows as per the defined schedule. By default, the schedule is disabled.
 - **Timezone:** Select the time zone that the server uses to schedule the workflow.
 - **Repeats:** Select one of the following to specify the frequency at which the workflow is executed.
 - **Repeat by minute:** Specify the number of minutes after which the workflow is executed in the **Every _ minute(s)** field.
 - **Repeat by hour:** Specify the number of hours after which the workflow is executed in the **Every _ hour(s)** field.

- **Repeat daily:** Specify the number of days after which the workflow is executed in the **Every _ day(s)** field when you select the daily mode as **Repeat every X days**. If you select **Only on weekdays (MON-FRI)** as the daily mode, it runs the workflow daily on weekdays.
 - **Repeat weekly:** Select the days of the week when the workflow is executed in the **Select day(s) of week** field.
 - **Repeat monthly:** Specify additional details about the day of the month or a weekday when the workflow is executed.
 - **Start Time:** Select the time at which you want the workflow to be executed. The service stores time in UTC format.
 - **Cron Expression:** When you click **Generate** in the Schedule group, a CRON expression is automatically populated based on selected options in the Schedule group.
 - **Input (JSON):** Provide values for the input parameters of the workflow, if any, in JSON format.
5. Click **Save**.

Details of the schedule that you have created are displayed on the **Schedules** list page.

2.3.2 Update a Schedule Definition

You can update all the field values you specified when creating a schedule definition, except for the name.

1. Open the navigation menu and click **Definitions**.
2. Click the **Schedules** tab.

The **Schedules** list page opens. All the schedules that you have defined are displayed in a table.
3. Identify the schedule definition that you want to update.
4. Click  (**Edit**).
5. You can update all the parameter values you specified while creating a schedule definition, except for the name.
6. Click **Submit**.

2.3.3 View a Schedule Definition

View information about a schedule definition.

You can only view schedule definitions that are enabled, that is the **Status** is set to true for the schedule definition.

1. Open the navigation menu and click **Definitions**.
2. Click the **Schedules** tab.

The **Schedules** list page opens. All the schedules that you have defined are displayed in a table.
3. Identify the schedule definition that you want to view.
4. Click  (**Show Details**).

5. Click **Close** after viewing the details.

2.3.4 Delete a Schedule Definition

You can delete a schedule definition when it is no longer required.

To disable a schedule while retaining the schedule definition in the database, set the status as false. See [Update a Schedule Definition](#).

Before you delete a schedule definition, ensure that it is not being used by any task. The task execution fails if it can't find a specified schedule definition.

1. Open the navigation menu and click **Definitions**.
2. Click the **Schedules** tab.

The **Schedules** list page opens. All the schedules that you have defined are displayed in a table.

3. Identify the schedule that you want to delete.
4. Click  (**Delete**).
5. When prompted, confirm the deletion.

2.4 Manage Event Handlers

Create, update, retrieve, search, and delete event handlers. Event handlers in MicroTx Workflows listen for events from supported internal or external message brokers and perform one or more actions on workflow or task when matching conditions are met. This feature also enables loose coupling between workflows.

Topics

- [Create an Event Handler](#)
Create event handlers in MicroTx Workflows to listen to messages sent by Transactional Event Queues (TxEventQ) within MicroTx Workflows or to listen to external message brokers, such as Kafka.
- [Update an Event Handler](#)
You can update all the field values you specified when creating an event handler, except for the name.
- [View an Event Handler](#)
View details of an event handler.
- [Delete an Event Handler](#)
You can delete an event handler when it is no longer required.

2.4.1 Create an Event Handler

Create event handlers in MicroTx Workflows to listen to messages sent by Transactional Event Queues (TxEventQ) within MicroTx Workflows or to listen to external message brokers, such as Kafka.

When the message broker receives a message, event handlers can perform one or more of the following actions based on the specified conditions.

- Start running a workflow
- Terminate a workflow

- Complete a task
- Fail a task

Prerequisites

Depending on the type of message broker, you must set `txeventq.enabled` or `kafka.enabled` to `true` in the `values.yaml` file. This enables event handlers to listen to the message brokers. For Kafka message broker, specify a comma-separated list of addresses of bootstrap servers to access Kafka brokers. See *Event Handler Configuration in MicroTx Installation and Configuration Guide*.

To create an event handler:

1. Open the navigation menu and click **Definitions**.
2. Click the **Event Handlers** tab.

The **Event Handlers** list page opens. All the event handlers that you have defined are displayed in a table.

3. Click  (Add Event Handler).

The New Event Handler dialog box is displayed.

4. Enter the following information to create an event handler.
 - **Name:** Mandatory. Enter a unique name to identify the event handler. The name can be up to 128-characters long. Use only letters, numbers, underscores (`_`), and hyphens (`-`). Spaces and other special characters are not supported. You cannot change this name later.
 - **Queue Type:** Select **txeventq** if the event handler listens to messages sent by Transactional Event Queues (TxEventQ) within MicroTx Workflows or **kafka** to listen to external Kafka events.
 - **Queue Name:** The name of the queue or topic to which the message is pushed.
 - **Condition:** Define an expression that determines what triggers the event handler. The expression must return a boolean value. The actions specified in the event handler are triggered only when the result of the condition expression is true. Enter the expression in a JavaScript-like syntax which evaluates the result of the condition based on the payload.
 - **Evaluator Type:** Enter **JavaScript** as the type of evaluator for the condition. This is the only supported value.
 - **Active:** Select this option to enable the event handler.
5. If you select **txeventq** as the **Queue Type**, provide the following additional information.
 - **Publisher Name:** Enter the name or identifier of a publisher agent, producer application, entity, or task that enqueues messages.
 - **DB Profile:** Select the database profile which contains the connection details of the Oracle Database, to which you want to publish messages.
6. Add one or more of the following types of actions in the **Actions** group to determine the activities performed by the event handler when it is triggered.
 - **Start Workflow.** Starts the specified workflow. Provide information for the following fields.
 - **Workflow Name:** Select a workflow.
 - **Workflow Version:** Select a version of the workflow.

- **Correlation ID:** Enter an ID to logically group and monitor several related workflow executions.
 - **Input (JSON):** Provide input parameters for the workflow, if any, in JSON format.
 - **Task to Domain (JSON):** Enter the names of the domains in which you want to run a task.
 - **Complete Task** to complete executing a task or **Fail Task** to fail the task execution. This changes the status of a task from in progress to completed or failed based on your selection. Provide information for the following fields.
 - **Workflow ID:** Enter the workflow execution ID. This ID is unique to a workflow execution.
 - **Task Ref Name:** Enter the unique reference name for the task. This is used to reference the task within a workflow.
 - **Task ID:** Enter a unique identifier to retrieve details of a specific task.
 - **Output (JSON):** Enter the message that you want to display in JSON format.
 - **Terminate Workflow:** Terminates the specified workflow. Provide information for the following fields.
 - **Workflow ID:** Enter the workflow execution ID. This ID is unique to a workflow execution.
 - **Termination Reason:** Enter the cause for terminating the workflow execution.
7. Optional. Click **Add Action** to add another action that the event handler performs when it is triggered.
 8. Click **Submit** to save the changes.

Details of the event handler that you have created are displayed on the **Event Handlers** list page.

2.4.2 Update an Event Handler

You can update all the field values you specified when creating an event handler, except for the name.

1. Open the navigation menu and click **Definitions**.
2. Click the **Event Handlers** tab.

The **Event Handlers** list page opens. All the event handlers that you have defined are displayed in a table.
3. Identify the event handler that you want to update.
4. Click  (**Edit**).
5. You can update all the parameter values you specified while creating an event handler, except for the name.
6. Click **Submit**.

2.4.3 View an Event Handler

View details of an event handler.

1. Open the navigation menu and click **Definitions**.

2. Click the **Event Handlers** tab.

The **Event Handlers** list page opens. All the event handlers that you have defined are displayed in a table.

3. Identify the event handler that you want to view.
4. Click  (**Show Details**).
5. Click **Close** after viewing the JSON code of the event.

2.4.4 Delete an Event Handler

You can delete an event handler when it is no longer required.

Before you delete an event handler, ensure that it is not being used by any task. The task execution fails if it can't find a specified event handler.

1. Open the navigation menu and click **Definitions**.
2. Click the **Event Handlers** tab.

The **Event Handlers** list page opens. All the event handlers that you have defined are displayed in a table.

3. Identify the event handler that you want to delete.
4. Click  (**Delete**).
5. When prompted, confirm the deletion.

2.5 View and Approve Workflow Notifications

View pending or in-progress human tasks generated by workflows. Human task notifications help you find workflow steps that require manual action, such as review or approval. The notification API lists human tasks; completing, approving, rejecting, or otherwise acting on the task is handled by updating the corresponding task status and output.

Workflow notifications are created for human tasks. Use human tasks when a workflow requires manual review, approval, rejection, or other human intervention before the workflow can continue.

When a workflow reaches a human task, MicroTx Workflows creates a notification for that task and keeps the task in progress until a user takes action. The workflow waits at that step and resumes only after the human task is updated. This is useful for approval flows, exception handling, compliance checks, and agentic workflows where an AI-generated recommendation or action must be reviewed before execution continues.

Before you begin, note the workflow instance ID of the workflow execution. You can use this ID to find the related workflow notification and review the workflow execution details before approving or rejecting the task.

To take an action on a workflow notification:

1. Open the navigation menu and click **Workflow Notification**.
All pending and in-progress human tasks that require action are displayed in a table.
2. Use the search bar to filter notifications. You can search by values such as workflow name, task name, task ID, or workflow instance ID.
3. Identify the workflow notification on which you want to act.

4. Optional. Open the related workflow execution details to review the workflow input, output, task history, and current execution state before taking action.
5. Click  (**Act**) under **Actions**.
The **Take Action on Task** dialog box appears.
6. In the **Status** drop-down list, select one of the following values:
 - **Completed**: Approves the notification and marks the human task as completed. The workflow continues to the next step based on the workflow definition.
 - **Failed**: Rejects the notification and marks the human task as failed. If you select this option, enter a reason in the **Failure Reason** field. The workflow then continues based on the failure handling logic defined in the workflow.
7. Click **Submit**.
A message is displayed that the task was updated successfully.
8. Click **OK**.

After you take action, the notification no longer appears on the **Workflow Notification** page because the human task is no longer pending or in progress. MicroTx Workflows resumes processing the workflow based on the status you selected and the workflow's business logic.

3

Tasks

Tasks are the core building blocks of a workflow. Each task represents a unit of work, such as invoking a service, running a database operation, evaluating a condition, waiting for an event, requesting human approval, or calling an AI model or agent.

A workflow definition uses tasks to describe what must happen, in what order, and how data moves from one step to another.

Topics

- [About Tasks](#)
Tasks are the building blocks of a workflow. A workflow definition must contain at least one task.
- [Create System Tasks and Operator Tasks](#)
- [Manage Task Definitions for Simple Tasks](#)
This section provides information about how you can create or import a new task definition, and then add it to a workflow. It also provides information about managing task definitions for simple tasks.
- [Manage Executed Tasks](#)
MicroTx Workflows stores details of all the tasks that are executed as part of a workflow execution.

3.1 About Tasks

Tasks are the building blocks of a workflow. A workflow definition must contain at least one task.

A workflow definition describes how MicroTx Workflows executes business logic. It defines the tasks to run, the order in which they run, how each task receives input, and how task output is passed to later tasks or returned as workflow output. Tasks can run sequentially, in parallel, conditionally, or in response to events, depending on how the workflow is modeled.

MicroTx Workflows supports most of the task types provided by open-source Conductor and also provides custom tasks for database operations, events, human intervention, and AI or agentic workflows.

Types of Tasks

MicroTx Workflows supports different types of tasks that you can use to build workflow logic. Each task type fulfills a different purpose in a workflow.

- System tasks are built-in tasks that MicroTx Workflows manages and executes. MicroTx Workflows also includes agentic and AI-related tasks that let you integrate LLM profiles, agents, planners, tools, knowledge sources, and GenAI operations into workflow execution. Use system tasks for common operations such as invoking HTTP endpoints, running SQL, publishing events, waiting for signals, requesting human approval, using GenAI capabilities, or invoking agentic tasks.

- Operator tasks are built-in tasks that control workflow execution logic. Use operators to model branching, joins, loops, parallel execution, dynamic tasks, and other orchestration patterns.
- Simple tasks or worker tasks let you implement custom business logic in your own services outside MicroTx Workflows. External workers poll MicroTx Workflows tasks for work, execute custom business logic, and return results to MicroTx Workflows.

This chapter describes the custom tasks available in MicroTx Workflows. For standard Conductor task behavior, see the [Conductor documentation](#).

Topics

- [About System Tasks](#)
System tasks are built-in tasks that MicroTx Workflows manages and executes within its server environment. Use system tasks to perform common workflow operations without creating and running custom worker services.
- [About Operator Tasks](#)
Operator tasks control how a workflow moves from one step to another. Use operators to model branching, looping, joins, parallel execution, dynamic task selection, sub-workflows, and other orchestration patterns.
- [About Simple Tasks or Worker Tasks](#)
Simple tasks, also called worker tasks, let you run custom business logic in your own services outside MicroTx Workflows.
- [About Task Status](#)
Understanding how a task moves from one status to another helps you to execute the workflow successfully and troubleshoot any issues.

3.1.1 About System Tasks

System tasks are built-in tasks that MicroTx Workflows manages and executes within its server environment. Use system tasks to perform common workflow operations without creating and running custom worker services.

MicroTx Workflows supports system tasks from open-source Conductor and provides additional custom system tasks for AI, agentic orchestration, database access, events, file transfer, transactions, and human intervention. This section describes the custom system tasks that you can add from the MicroTx Workflows user interface. For information about general system tasks, see [System Tasks](#) in open-source Conductor documentation.

You can leverage system tasks to handle common operations without writing custom code as described in the following table.

Task Name	Description
Agentic Task	With an Agentic task, you can seamlessly integrate Agent Profiles that you have defined in MicroTx Workflows directly into workflows. An Agentic task brings AI-driven reasoning into workflow execution while keeping the agent configuration centralized in the agent profile.
Agentic Planner Task	Uses an LLM-enabled planner to evaluate runtime context, prompt goals, tools, and available tasks, and then dynamically determine the next actions in the workflow. Use this operator to build adaptive, agentic workflows that can plan and schedule follow-up steps based on runtime inputs.
GenAI Task	Invokes an LLM by using a single prompt and input data. Use this task for summarization, classification, extraction, generation, and other single-step GenAI operations.

Task Name	Description
GenAI Ingestion Task	Large Language Models (LLMs) can produce inaccurate responses when they do not have access to relevant and current information. Use a GenAI Ingestion task to prepare your business data for Retrieval Augmented Generation (RAG), so that later tasks can retrieve relevant context from a vector store before generating a response.
GenAI Retrieve Task	Retrieves relevant context from a vector database and uses it to generate a natural language response to a user query. Use this task for RAG-based question answering and knowledge retrieval.
gRPC Task	Calls a remote gRPC service from a workflow. Use this task when the target service exposes methods through gRPC and protocol buffer contracts.
SFTP Task	Transfers files to and from remote SFTP servers. Use this task to upload, download, list, or manage files without writing a custom worker task.
HTTP Task	Calls external REST APIs from a workflow. Use this task to integrate workflows with HTTP services.
SQL Task	Executes SQL statements against a configured database. Use this task to query, insert, update, or otherwise interact with database records from a workflow.
Transaction Task	Uses MicroTx Coordinator to manage XA transactions across multiple tasks in a workflow. Use this task when several operations must participate in the same distributed transaction.
TX EventQ Publish Task	Publishes a message to TxEventQ in Oracle Database, where the database acts as the message broker.
Human Task	Pauses workflow execution until a user approves, rejects, or otherwise completes the task. Use this task for manual review, approval workflows, exception handling, and human-in-the-loop agentic flows.

3.1.2 About Operator Tasks

Operator tasks control how a workflow moves from one step to another. Use operators to model branching, looping, joins, parallel execution, dynamic task selection, sub-workflows, and other orchestration patterns.

Operators do not usually perform business logic directly. Instead, they control the execution path of other tasks, similar to programming constructs such as if-else, switch-case, loops, and parallel blocks. For example, a switch operator can choose a path based on input data, a fork and join can run tasks in parallel and wait for them to finish, and a sub-workflow operator can run another workflow as part of a larger workflow.

MicroTx Workflows supports most of the operators available in open-source Conductor. It also includes the Agentic Planner operator, which uses an LLM-enabled planner to evaluate runtime context, prompt goals, tools, and available tasks, and then dynamically determine the next actions in a workflow. Use this operator to build adaptive agentic workflows that can plan and schedule follow-up steps based on runtime inputs.

For more information about standard operators, see [Operators](#) in open-source Conductor.

3.1.3 About Simple Tasks or Worker Tasks

Simple tasks, also called worker tasks, let you run custom business logic in your own services outside MicroTx Workflows.

Use simple tasks when a workflow must call logic that is too custom, complex, long-running, or environment-specific to model directly in the workflow definition. Your worker application polls

MicroTx Workflows for pending tasks, performs the work, and sends the task result back to MicroTx Workflows by using the REST API or SDK.

Simple tasks are language independent. You can implement workers in Java, Python, JavaScript, Go, or any other technology that can call the MicroTx Workflows APIs. You can also scale workers independently from the workflow server based on workload and performance needs.

Simple tasks also provide a scalable extension model for MicroTx Workflows. Because the worker runs outside the MicroTx Workflows server, you can scale the worker service independently based on workload, resource needs, or deployment topology. This is useful for compute-heavy processing, long-running business logic, integrations that require customer-managed infrastructure, and workloads that should not consume workflow server resources. You can also use simple tasks to integrate externally developed agentic systems or AI services by exposing them through a worker that polls MicroTx Workflows and returns the result.

Examples of simple tasks include:

- Order validation task: A Java microservice validates order details and returns the validation result.
- Inventory check task: A Python worker checks inventory levels and updates the task status.
- Image processing task: A microservice processes an image by using an AI model hosted in your own infrastructure.
- External system integration task: A worker task connects to an external system or third-party service that requires custom authentication, libraries, or business rules.

Use a simple task when you need full programming flexibility, external dependencies, custom error handling, or direct access to systems that are not exposed through built-in system tasks.

For lightweight logic that can run directly inside the workflow, use an inline task instead of a simple task. Inline tasks are useful for small JavaScript-based operations such as transforming input data, calculating derived values, formatting strings or dates, or performing simple condition checks. For example, you can use an inline task to convert a JSON date string to another format or add two numbers from workflow input for use by a downstream task. See [Inline Task](#).

For more information about simple tasks or worker tasks, see [Workers](#) in open-source Conductor documentation.

3.1.4 About Task Status

Understanding how a task moves from one status to another helps you to execute the workflow successfully and troubleshoot any issues.

See [Task Lifecycle](#) in open-source Conductor documentation.

3.2 Create System Tasks and Operator Tasks

Create a workflow using Workflow Builder, and then add system tasks or operator tasks directly to the workflow.

Topics

- [Access the Task Tab in Workflow Builder](#)
You can add a task to a workflow in the Workflow Builder.

- [Agentic Task](#)
Use an Agentic task to invoke an Agent Profile from a workflow. An Agentic task brings AI-driven reasoning into workflow execution while keeping the agent configuration centralized in the agent profile.
- [Agentic Planner Task](#)
The Agentic Planner is an operator task that uses an LLM to dynamically orchestrate workflow tasks and tools at runtime. Unlike an Agentic Task, which runs a specific agent, the Agentic Planner acts as a dynamic orchestrator or master planning node. It evaluates the goal, runtime context, available tasks, internal tools, and MCP server tools, and then decides which task or tool to run next.
- [GenAI Task](#)
A GenAI Task lets you call a large language model (LLM) from a workflow for single-prompt use cases. Use this task when you want the workflow to send a prompt, optional runtime variables, and optional supporting data to an LLM and use the generated response in downstream workflow steps.
- [GenAI Ingestion Task](#)
Large Language Models (LLMs) can produce inaccurate responses when they do not have access to relevant and current information. Use a GenAI Ingestion task to prepare your business data for Retrieval Augmented Generation (RAG), so that later tasks can retrieve relevant context from a vector store to generate more accurate AI responses.
- [GenAI Retrieve Task](#)
Use a GenAI Retrieve Task to generate natural language responses by using retrieval augmented generation (RAG). The task retrieves relevant context from a vector table in Oracle Database and sends that context, along with the user query, to a large language model (LLM) to generate the response.
- [gRPC Task](#)
Use a gRPC task to invoke a unary gRPC method from a workflow. The task uses gRPC server reflection to discover the service, method, request message, and response message descriptors at runtime. The target gRPC server must have server reflection enabled.
- [SFTP Task](#)
The SFTP system task allows workflows to securely transfer files—including uploading, downloading, and listing—directly to and from remote SFTP servers, eliminating the need to develop custom simple tasks or worker tasks.
- [HTTP Task](#)
Use an HTTP task to make REST API calls to external services.
- [JSON RPC Task](#)
Use the JSON RPC task to invoke JSON-RPC 2.0 services over HTTP(S) using POST requests with application/json payloads. It supports single-call mode for one RPC invocation and batched requests (array) for multiple calls in a single HTTP request.
- [SQL Task](#)
The SQL task allows workflows to retrieve or update data based on the SQL statements that you define. It enables workflows to interact with SQL databases to execute queries or store information.
- [Transaction Task](#)
The transaction task enables you to use MicroTx Distributed Transactions coordinator to manage XA transactions that span across multiple tasks in MicroTx Workflows.
- [TX EventQ Publish Task](#)
Use the TX EventQ Publish task to publish a message to a queue in Oracle Database, which serves as a message broker.

- [Human Task](#)
Include human tasks in Agentic AI workflows to add manual review, approval, rejection, or other human intervention before a workflow continues. This is useful when an AI-generated decision, recommendation, or action requires human validation before it affects a business process.
- [Inline Task](#)
Use an Inline task when you need to run lightweight, simple logic directly within your workflow.
- [Wait Task](#)
Use a wait task to pause a workflow.
- [Kafka Publish Task](#)
Use this task to send messages to another microservice through a Kafka topic with customizable headers and serializers.
- [Start Workflow Task](#)
Use this task to run another workflow asynchronously from a workflow that is currently being executed.
- [JSON JQ Transform Task](#)
Use this task to process JSON data using jq.

3.2.1 Access the Task Tab in Workflow Builder

You can add a task to a workflow in the Workflow Builder.

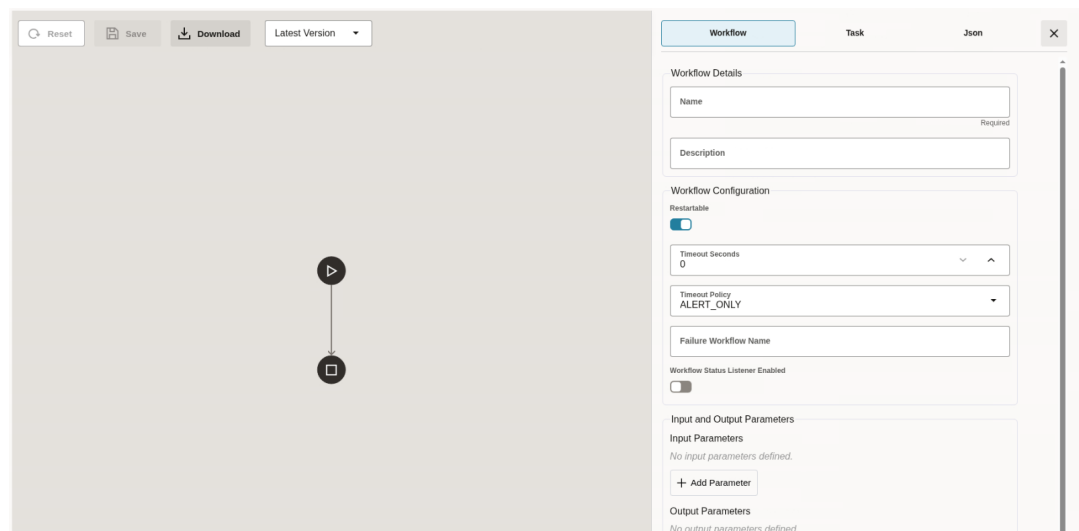
To access the **Task** tab in Workflow Builder:

1. Open the navigation menu and click **Definitions**.
2. Click the **Workflows** tab.

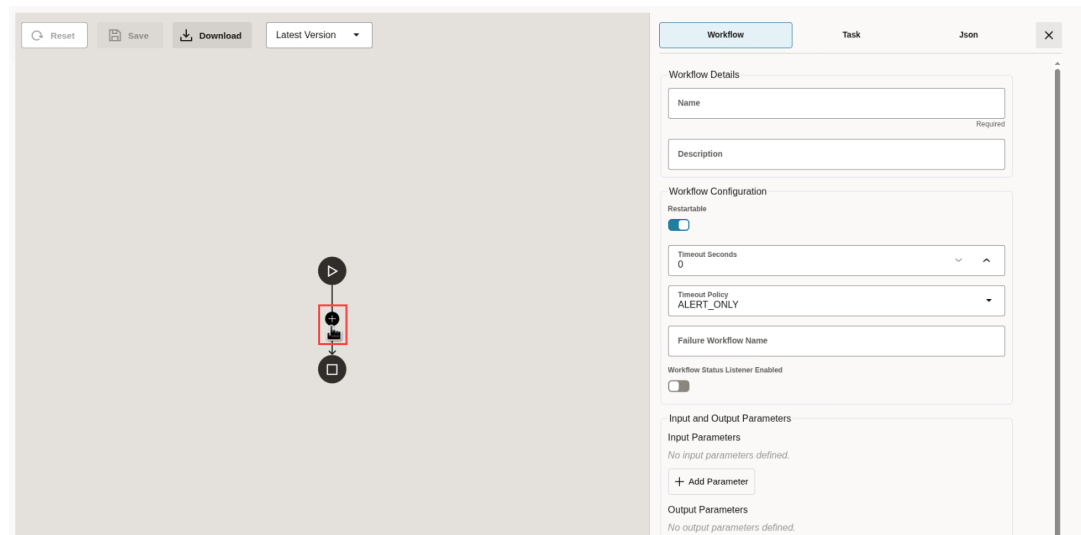
The **Workflows** list page opens and displays the workflow definitions that have been created.

3. Click .

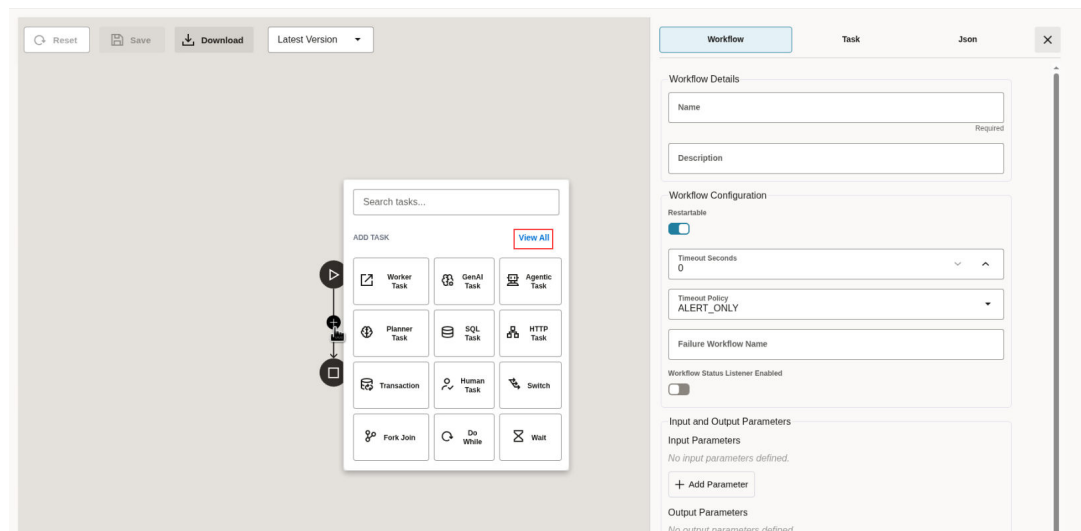
The Workflow Builder is displayed. On the right pane, the **Workflow** tab displays all the details of the workflow as shown in the following figure. On the left pane, the components of the workflow are depicted visually.



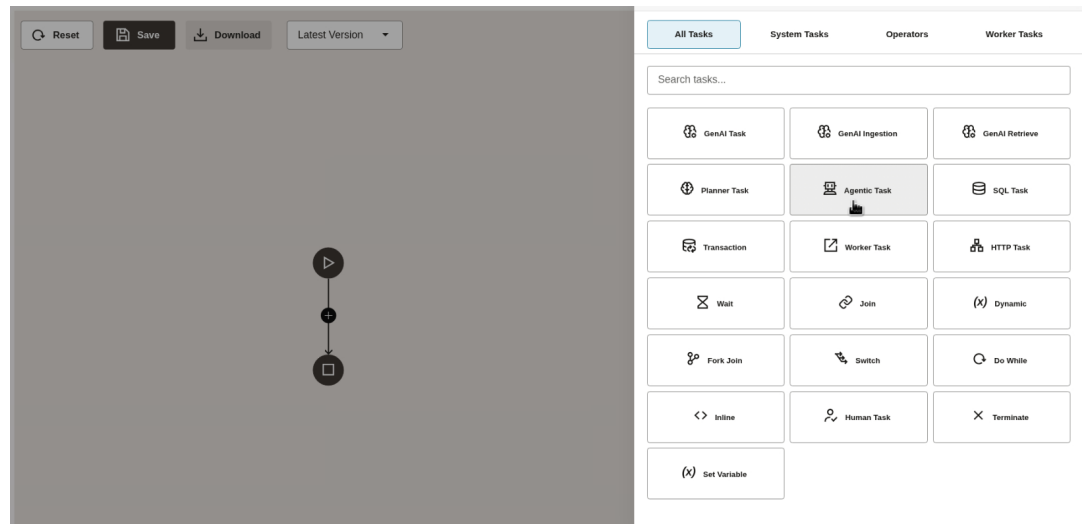
- To add a task in your workflow using the builder, place your cursor on the arrow mark. A plus sign appears as shown in the following image.



- Click the plus icon. Many tasks are displayed.



- Click **View All** to view all the tasks that you can add using the Workflow Builder.



7. In the **More Tasks** dialog box, search for a task using the search pane or navigate through the tabs to identify a task that you want to add to the workflow.
 - **All Tasks** lists all the tasks that you can add using the UI.
 - **System Tasks** lists all the system tasks that you can add using the UI.
 - **Operators** lists all the operators that you can add using the UI.
 - **Worker Tasks** lists all the system tasks or worker tasks that you have created as a task definition.

3.2.2 Agentic Task

Use an Agentic task to invoke an Agent Profile from a workflow. An Agentic task brings AI-driven reasoning into workflow execution while keeping the agent configuration centralized in the agent profile.

An agent profile defines the agent's role, instructions, LLM profile, model, tools, MCP servers, memory settings, prompt variables, LLM parameters, and guardrails. In the workflow, the agentic task only needs to reference the agent profile and provide workflow-specific context, such as additional prompt text, prompt variable values, input data, or task-level guardrails.

Use agentic tasks when you want a workflow to run an AI agent in a durable and orchestrated manner. MicroTx Workflows records the agent execution as part of the workflow execution, including task logs, input, output, tool activity, and status. Agentic tasks also use task-level idempotency by default, which helps prevent duplicate side effects when a task is retried or replayed.

Agentic tasks are useful for integrating agents created in MicroTx Workflows, and they also help bring agents designed in other frameworks into MicroTx Workflows orchestration by representing the agent as a reusable agent profile.

Data size is ultimately limited by document extraction, workflow payload size, and the LLM model context window. For large document collections, use GenAI Ingestion and GenAI Retrieve tasks rather than passing all content directly to an Agentic Task.

Prerequisites

Before you begin, ensure that you have completed the following tasks.

- Create an agent profile. See [Create an Agent Profile](#). Ensure that the agent profile has the **Workflow** capability enabled.
- Create an LLM definition. See [Create an LLM Definition](#).
- Configure any tools, MCP servers, prompt variables, guardrails, or model parameters in the agent profile as required.

To add an Agentic Task

1. Navigate to the **Task** tab in a workflow and view all the tasks that you can add using the Workflow Builder. See [Access the Task Tab in Workflow Builder](#).
2. In the **More Tasks** dialog box, click **Agentic Task** to add it to the workflow.
3. Click the task that you have added in the left pane. The **Task** tab in the right pane displays details about the task, such as its name and parameters. Next, let's provide details for the task.
4. In the **Task Details** group, enter the following information.
 - **Task Name:** Mandatory. Enter a unique name for the task. The name must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (`_`) and hyphen (`-`).
 - **Task Reference:** Mandatory. Enter a value to refer to the task within a workflow definition. This value must be unique within a workflow. The task reference value must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (`_`) and hyphen (`-`).
5. In the **Agentic Task Parameters** group, provide the following information.
 - **Agent Profile:** Mandatory. Select the agent profile to run. The agent profile supplies the LLM profile, model, tools, MCP servers, role, instructions, memory settings, and LLM parameters.
 - **Prompt:** Optional. Enter additional prompt text for this workflow step. MicroTx Workflows appends this prompt to the instruction defined in the agent profile. Use this field for workflow-specific guidance, not for repeating the complete agent instruction.
 - **Prompt Variables:** Optional. Provide values for placeholders used in the agent profile instruction. For example, if the agent instruction contains `${customerName}`, provide `customerName` as a prompt variable. If the same variable is defined in both the agent profile and the agentic task, the value provided in the agent profile takes precedence.
 - **Idempotency Timeout (ms):** Optional. Specifies how long MicroTx Workflows holds the task-level idempotency lock for this task. The lock prevents duplicate concurrent execution during retries or replays. The default is 600000 milliseconds, or 10 minutes.
 - **Guardrails:** Optional. Provide task-specific guardrails such as stop words. If guardrails are defined in both the agent profile and the agentic task, MicroTx Workflows merges them, and task-level guardrail values override matching agent profile guardrail values.
 - **Add Data Source:** Optional. Select this check box when you want to provide any additional data that is required to execute the task, and then select one of the following options in the **Data Source** drop-down list. If no data is provided, the agent runs using the agent profile instruction, prompt variables, optional task prompt, and workflow context.
 - **WEB:** Provide a URL. MicroTx Workflows retrieves content from the URL and adds it to the agent prompt context.
 - **OCI:** Provide a URL to content available in Oracle Cloud Infrastructure (OCI).
 - **LOCAL:** Provide a file path for a file uploaded to MicroTx Workflows local storage.

- **TEXT:** Provide plain text directly in the task input.
- 6. Click **Save** to save the changes to the workflow.
MicroTx Workflows displays the changes in JSON code.
- 7. Review all the changes, and then click **Confirm Save** to save the changes.
If you do not want to save the changes, click **Cancel**, and then click **Reset** to undo all the changes that you have made since the workflow was last saved.

Example

When you enter information in the **Task** tab, the corresponding code of the task is updated in the **JSON** tab. The following sample code displays the JSON code for an Agentic task with sample values. The sample code invokes a loan document verification agent. The Agent Profile contains the LLM profile, model, tools, instructions, and model parameters. The task provides only runtime values and document input.

```
{
  "name": "loan_document_verification",
  "taskReferenceName": "document_verification",
  "type": "AGENTIC_TASK",
  "inputParameters": {
    "agentProfile": "loan_document_verification_agent",
    "prompt": "Review the applicant documents and return a verification decision with reasons.",
    "promptVariables": {
      "applicantId": "${workflow.input.applicantId}",
      "loanType": "${workflow.input.loanType}"
    },
    "data": {
      "source": "local",
      "filePath": "${workflow.input.documentPath}"
    }
  }
}
```

After execution, open the workflow execution details and select the Agentic task to view task input, output, logs, and tool execution details. The task output contains the agent response and can be used by downstream workflow tasks. See [View Details of a Workflow Execution](#).

Practical Example

Let's consider a loan approval process. You can create an agentic task to invoke a document verification agent by sending the latest applicant data and memory of the previous document checks. The agent dynamically retrieves the data from the documents, analyzes the risks, accesses APIs through its LLM profile, and returns its decision—all orchestrated within MicroTx Workflows.

Use agentic task to support advanced use cases such as:

- AI-assisted data analysis and recommendations.
- Enterprise knowledge search with tools or MCP servers.
- Human-in-the-loop approval preparation.
- Document verification and risk review.
- Agentic workflows that call external tools and return structured decisions.

3.2.3 Agentic Planner Task

The Agentic Planner is an operator task that uses an LLM to dynamically orchestrate workflow tasks and tools at runtime. Unlike an Agentic Task, which runs a specific agent, the Agentic Planner acts as a dynamic orchestrator or master planning node. It evaluates the goal, runtime context, available tasks, internal tools, and MCP server tools, and then decides which task or tool to run next.

Use the Agentic Planner when the workflow path cannot be fully determined at design time, or when the workflow must make adaptive decisions based on intermediate task results. A workflow can include one or more Agentic Planner operators, depending on how much of the workflow needs dynamic orchestration.

The Agentic Planner can:

- Plan and schedule workflow tasks dynamically.
- Invoke internal tools and MCP server tools.
- Evaluate results from previously executed planned tasks.
- Continue planning in iterations until the goal is completed or failed.
- Coordinate agentic tasks, HTTP tasks, simple tasks, human tasks, and other supported workflow tasks.
- Record planner iterations, tool calls, task decisions, and outputs in workflow execution details.

Agentic Planner execution is durable. MicroTx Workflows records the planner history, raw planner responses, tool execution details, task scheduling decisions, and output as part of workflow execution. The planner also uses fenced idempotency handling so that repeated execution does not duplicate completed planner work.

Execution Flow

At runtime, the Agentic Planner performs the following steps:

1. Receives a goal from a prompt or prompt template.
2. Builds a planning context using the available workflow tasks, internal tools, MCP tools, optional data, and previous execution results.
3. Calls the configured LLM to generate a plan.
4. Executes planned tools immediately or schedules planned workflow tasks.
5. Waits for scheduled tasks to complete, and then evaluates their output.
6. Repeats planning iterations until the planner returns a completed or failed status.
7. Stores planner output, history, and execution details in the workflow execution.

You can view the planner's iterations, task decisions, tool execution details, logs, input, and output from the workflow execution details page under the Agentic Planner task.

Prerequisites

Before you add an Agentic Planner, create or identify the following resources:

- An LLM definition. See [Create an LLM Definition](#).
- A prompt or prompt template that describes the planner goal. See [Create a Prompt Template Definition](#).

- Internal tool configuration for every tool, if any, that the agentic planner task uses. See [Create an Internal Tool Configuration](#).
- MCP server definitions, if the planner uses MCP tools. See [Provide MCP Server Details](#).
- Identify and define all the tasks that you want to run as part of the agentic planner task execution.
- Workflow tasks that the planner can schedule, such as Agentic Task, HTTP, Simple task, Human task, or other supported tasks. The Agentic Planner must have at least one available execution target, such as a workflow task, internal tool, or MCP server tool.

To add an Agentic Planner Task

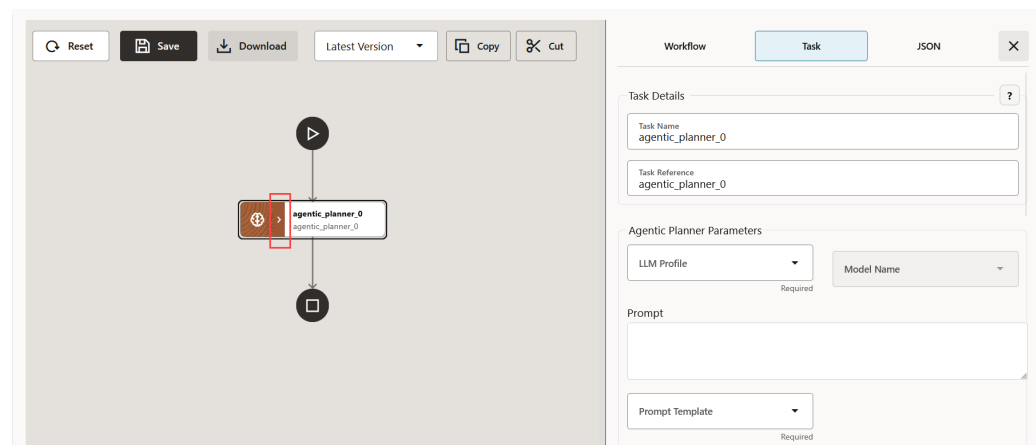
1. Navigate to the **Task** tab in a workflow and view all the tasks that you can add using the Workflow Builder. See [Access the Task Tab in Workflow Builder](#).
2. In the **More Tasks** dialog box, click **Planner Task** to add it to the workflow.
3. Click the task that you have added in the left pane. The **Task** tab in the right pane displays details about the task, such as its name and parameters. Next, let's provide details for the task.
4. In the **Task Details** group, enter the following information.
 - **Task Name:** Mandatory. Enter a unique name for the task. The name must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (`_`) and hyphen (`-`).
 - **Task Reference:** Mandatory. Enter a value to refer to the task within a workflow definition. This value must be unique within a workflow. The task reference value must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (`_`) and hyphen (`-`).
5. In the **Agentic Planner Task Parameters** group, provide the following information.
 - **LLM Profile:** Mandatory. Select the LLM definition that the planner uses for reasoning.
 - **Model Name:** Mandatory. Select the model to use from the LLM definition.
 - **Prompt** or **Prompt Template:** Mandatory. Enter a prompt or select a prompt template that describes the planner goal. At least one of these values is required.
 - **Prompt Variables:** Optional. Provide values for variables, if any, used in the prompt template.
 - **Idempotency Timeout (ms):** Optional. Specifies how long MicroTx Workflows holds the task-level idempotency lock for this task. The lock prevents duplicate concurrent execution during retries or replays. The default is 180000 milliseconds or 3 minutes.
 - **Tools:** Optional. Select internal tool configurations that the planner can call directly.
 - **MCP Servers:** Optional. Select MCP server definitions. The planner can discover and call tools exposed by these MCP servers.
 - **Guardrails:** Optional. Provide task-specific guardrails such as stop words.
 - **Temperature:** Optional. Controls randomness of the generated response. Lower values generate more deterministic responses. Higher values generate more varied responses. The default value is 0.2.
 - **Max Tokens:** Optional. Sets the maximum number of tokens that the model can generate. The default value is 1024.
 - **Top K:** Optional. Number of highest-probability tokens considered during generation. The default value is 40.

- **Top P:** Optional. Controls nucleus sampling value. The default value is 0.9.
- **Add Data Source:** Optional. Select this check box when you want to provide any additional data that is required to execute the task, and then select one of the following options in the **Data Source** drop-down list.
 - **WEB:** Select this option when you want to access data from the web, and then enter the complete URL to the data in the **URL** field.
 - **OCI:** Select this option when you want to access data that is available in Oracle Cloud Infrastructure (OCI), and then enter the complete URL to the data in the **URL** field.
 - **LOCAL:** Select this option when you want to use a file that you have uploaded to the local storage, and then select or provide a file name in the **File Path** drop-down list.
 - **TEXT:** Select this option when you want to provide the text as supporting data in the **TEXT** field.

For the Agentic Planner, provide data as a JSON object and include it in the planning context. For example:

```
{
  "data":
  {
    "loanApplicationText": "${
workflow.input.loan_application_text}",
    "document": "${workflow.input.document}"
  }
}
```

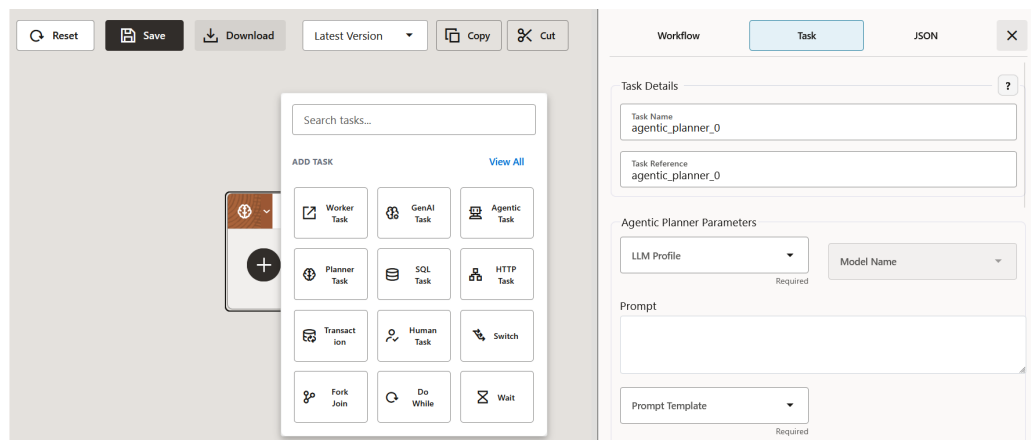
- Optional. Add all the tasks that you want to run as part of the agentic planner task execution. You must provide at least one task, tool, or MCP server.
 - Click  to expand the Agentic Planner task.



- Click .



Many tasks are displayed.



- c. Select the workflow tasks that the planner is allowed to schedule. These tasks act as available actions for the planner.
7. Click **Save** to save the changes to the workflow.
MicroTx Workflows displays the changes in JSON code.
8. Review all the changes, and then click **Confirm Save** to save the changes.

If you do not want to save the changes, click **Cancel**, and then click **Reset** to undo all the changes that you have made since the workflow was last saved.

Example

The following example shows an Agentic Planner that orchestrates a loan-processing flow. The planner can schedule document verification, compliance check, loan processing, notification, and human verification tasks. It can also use tools exposed by the doc_mcp MCP server.

```

{
  "name": "Agentic Planner",
  "taskReferenceName": "agentic_planner",
  "inputParameters": {
    "llmProfile": {
      "name": "openai-dev-llm-profile",
      "model": "gpt-4o"
    },
    "promptTemplate": "loan_process_planner",
    "promptVariables": {
      "document": "${workflow.input.document}"
    },
    "data": {
      "loanApplicationText": "${workflow.input.loan_application_text}",
      "document": "${workflow.input.document}"
    },
    "mcpServers": [
      "doc_mcp"
    ],
    "tasks": [
      {
        "name": "Loan Document Verification Task",
        "taskReferenceName": "document_verification",
        "inputParameters": {
          "agentProfile": "loan_document_verification_agent",
          "promptVariables": {
            "document": "${workflow.input.document}"
          }
        },
        "type": "AGENTIC_TASK"
      },
      {
        "name": "Compliance Agent",
        "taskReferenceName": "compliance_agent",
        "type": "HTTP",
        "inputParameters": {
          "http_request": {
            "method": "POST",
            "uri": "http://localhost:8001/api/compliance/check",
            "headers": {
              "Content-Type": "application/json"
            },
            "body": {
              "socialSecurityNumber": "${
extract_loan_details.output.ssn}"
            }
          }
        },
        "type": "HTTP"
      }
    ]
  }
}

```

```

    {
      "name": "loan_processing_agent_task",
      "taskReferenceName": "loan_processing_agent_task",
      "inputParameters": {
        "applicantId": "${workflow.input.applicantId}"
      },
      "type": "SIMPLE"
    },
    {
      "name": "Notify AML verification failure",
      "taskReferenceName": "notify_aml_check_failure_to_admin",
      "type": "HTTP",
      "inputParameters": {
        "http_request": {
          "method": "POST",
          "uri": "http://localhost:8085/email/notify",
          "headers": {
            "Content-Type": "application/json"
          },
          "body": {
            "text": "Loan application requires AML
reverification. Request: ${workflow.input.loan_application_text}"
          }
        }
      },
      "type": "HUMAN",
      "inputParameters": {
        "applicant": "${extract_loan_details.output}"
      }
    }
  ],
  "type": "AGENTIC_PLANNER"
}

```

Practical Example

In a loan-processing workflow, you can use the Agentic Planner after a GenAI task extracts applicant details from natural language input. The planner can then decide how to proceed: verify documents with an Agentic Task, call a credit or compliance HTTP service, invoke MCP tools for AML or income verification, request human approval when needed, and continue planning based on the results.

Use the Agentic Planner for use cases such as:

- Loan processing and approval orchestration.
- Customer support ticket triage.
- DevOps incident response.
- Research workflows that coordinate multiple data-gathering and validation steps.
- Multi-agent workflows that combine Agentic Tasks, external services, tools, and human approval.

3.2.4 GenAI Task

A GenAI Task lets you call a large language model (LLM) from a workflow for single-prompt use cases. Use this task when you want the workflow to send a prompt, optional runtime variables, and optional supporting data to an LLM and use the generated response in downstream workflow steps.

GenAI tasks are useful for operations such as summarization, translation, classification, content generation, text extraction, and response generation. Unlike an Agentic Task or Agentic Planner, a GenAI Task does not run an autonomous agent or dynamically orchestrate other tasks. It performs one direct LLM interaction based on the prompt and input that you provide.

At runtime, MicroTx Workflows resolves the prompt or prompt template, substitutes prompt variables, loads any configured data source, appends the data content to the prompt, invokes the selected model, and stores the generated response as the task output.

Prerequisites

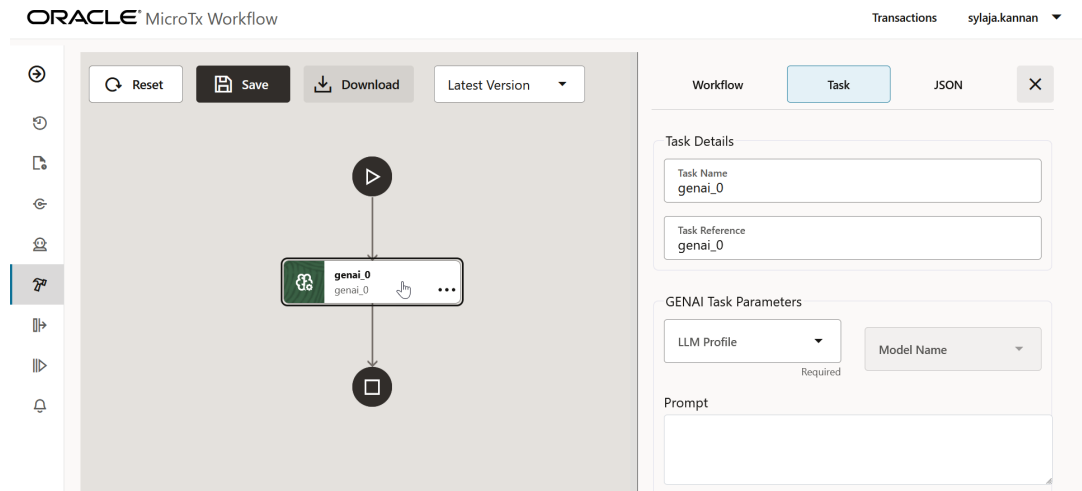
Before you begin, ensure that you have completed the following tasks.

- Create an LLM definition. See [Create an LLM Definition](#).
- Create a prompt definition. See [Create a Prompt Template Definition](#).
- Identify if you want to provide any additional data to execute the task and the location of the file. If you want to use a file that is available on your local system, upload it to the storage. See [Upload Files to Storage](#).

To add a GenAI Task

You must define a custom prompt tailored to the specific context of your workflow. Additionally, you can fine-tune the task's behavior, by using configurable parameters such as temperature, stop_words, top_p, top_k, and maxTokens. This allows precise control over the generation process, improving relevance and accuracy for different business scenarios.

1. Navigate to the **Task** tab in a workflow and view all the tasks that you can add using the Workflow Builder. See [Access the Task Tab in Workflow Builder](#).
2. In the **More Tasks** dialog box, click **GenAI Task** to add it to the workflow.
3. Click the task that you have added in the left pane. The **Task** tab in the right pane displays details about the task, such as its name and parameters. Next, let's provide details for the task.



4. In the **Task Details** group, enter the following information.
 - **Task Name:** Mandatory. Enter a unique name for the task. The name must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (`_`) and hyphen (`-`).
 - **Task Reference:** Mandatory. Enter a value to refer to the task within a workflow definition. This value must be unique within a workflow. The task reference value must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (`_`) and hyphen (`-`).
5. In the **GenAI Task Parameters** group, provide the following information.
 - **LLM Profile:** Mandatory. Select the LLM definition that the task uses to connect to the LLM provider.
 - **Model Name:** Mandatory. Select the model to use from the LLM definition.
 - **Prompt** or **Prompt Template:** Mandatory. Enter a prompt directly or select a prompt template definition. You must provide at least one of these values. If you provide both, MicroTx Workflows combines the inline prompt and the resolved prompt template.
 - **Prompt Variables:** Optional. If the prompt template contains variables, specify the values that replace those variables at runtime. You can use workflow input expressions, such as `workflow.input.customerMessage`.
 - **Temperature:** Optional. Controls randomness of the generated response. Lower values generate more deterministic responses. Higher values generate more varied responses. The default value is 0.2.
 - **Max Tokens:** Optional. Sets the maximum number of tokens that the model can generate. The default value is 1024.
 - **Top K:** Optional. Sets the number of highest-probability tokens considered during generation. The default value is 40.
 - **Top P:** Optional. Sets the cumulative probability threshold for nucleus sampling. The default value is 0.9.
 - **Guardrails:** Optional. Configure stop words to mask or reject input or output text based on the guardrail action.
 - **Add Data Source:** Optional. Select this option when the prompt needs additional supporting data. You can provide data from one of the following sources:

- **WEB:** Select this option when you want to access data from the web, and then enter the complete URL to the data in the **URL** field.
 - **OCI:** Select this option when you want to access data that is available in Oracle Cloud Infrastructure (OCI), and then enter the complete URL to the data in the **URL** field.
 - **LOCAL:** Select this option when you want to use a file that you have uploaded to the local storage, and then select or provide a file name in the **File Path** drop-down list.
 - **TEXT:** Select this option when you want to provide the text as supporting data in the **TEXT** field.
6. Click **Save** to save the changes to the workflow.
MicroTx Workflows displays the changes in JSON code.
 7. Review all the changes, and then click **Confirm Save** to save the changes.
If you do not want to save the changes, click **Cancel**, and then click **Reset** to undo all the changes that you have made since the workflow was last saved.

Example

When you enter information in the **Task** tab, the corresponding code of the task is updated in the **JSON** tab. The following example displays the JSON code for a GenAI task with sample values.

```
{
  "name": "Summarize_Text",
  "taskReferenceName": "genai_summarize_text",
  "inputParameters": {
    "llmProfile": {
      "name": "openai-dev",
      "model": "gpt-4o-mini"
    },
    "prompt": "Summarize the following content clearly and concisely,
highlighting the key points, main ideas, and any important conclusions or
actions. Avoid unnecessary details, but preserve the original meaning and
intent.",
    "data": {
      "source": "local",
      "filePath": "${workflow.input.filePath}"
    },
    "maxTokens": 4000,
    "temperature": 0.7,
    "top_p": 0.9,
    "top_k": 50
  },
  "type": "GENAI_TASK"
}
```

3.2.5 GenAI Ingestion Task

Large Language Models (LLMs) can produce inaccurate responses when they do not have access to relevant and current information. Use a GenAI Ingestion task to prepare your

business data for Retrieval Augmented Generation (RAG), so that later tasks can retrieve relevant context from a vector store to generate more accurate AI responses.

The GenAI Ingestion task reads content from a configured data source, splits the content into smaller chunks, generates vector embeddings by using the selected embedding model, and stores those embeddings in an Oracle Database vector table. The ingested data can then be queried by a GenAI Retrieve task or other RAG-based components.

Use this task when you want to ingest documents, text, or file content into a vector database as part of a workflow. For example, you can ingest policy documents, support articles, product manuals, loan documents, or customer records, and then use that data later to answer natural language questions.

Prerequisites

Oracle AI Vector Search, which is available in Oracle Database 26ai, is designed for Artificial Intelligence (AI) workloads and allows you to query data based on semantics, rather than keywords.

Before you begin, ensure that you have completed the following tasks.

1. Create a table using the VECTOR data type in Oracle Database 26ai as shown in the following example. The table must be compatible with the vector store schema used by MicroTx Workflows. Note down the name of the table as you will provide this name while creating the GenAI Ingestion task. This is the table where the GenAI Ingestion task stores the vector embeddings.

```
CREATE TABLE TEST_VECTORS (  
    id VARCHAR2(36) PRIMARY KEY,  
    content CLOB,  
    metadata JSON,  
    embedding VECTOR  
);
```

For more information about creating a table, see [Create Tables Using the VECTOR Data Type](#) in *Oracle AI Vector Search User's Guide*.

2. Create a database profile for the Oracle Database instance that contains the vector table. Note down this name as you will provide this name later. See [Create a Database Profile](#).
3. Create an LLM definition that uses an embedding model. See [Create an LLM Definition](#).
4. Identify the data source that you want to ingest. If you want to ingest a local file, upload the file to storage before using it in the task. See [Upload Files to Storage](#).

To add a GenAI Ingestion Task

1. Navigate to the **Task** tab in a workflow and view all the tasks that you can add using the Workflow Builder. See [Access the Task Tab in Workflow Builder](#).
2. In the **More Tasks** dialog box, click **GenAI Ingestion Task** to add it to the workflow.
3. Click the task that you have added in the left pane. The **Task** tab in the right pane displays details about the task, such as its name and parameters. Next, let's provide details for the task.
4. In the **Task Details** group, enter the following information.
 - **Task Name:** Mandatory. Enter a unique name for the task. The name must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (_) and hyphen (-).

- **Task Reference:** Mandatory. Enter a value to refer to the task within a workflow definition. This value must be unique within a workflow. The task reference value must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (_) and hyphen (-).
5. In the **GenAI Ingestion Parameters** group, provide the following information.
- **Embedding Profile Name:** Mandatory. Select the LLM definition that contains the embedding model configuration.
 - **Embedding Model:** Mandatory. Select the embedding model to use for generating vector embeddings. The selected model must exist in the embedding profile.
 - **Data Store Profile:** Mandatory. Select the database profile that connects to the Oracle Database instance where the vector table exists.
 - **Table Name:** Mandatory. Enter the name of the vector table where MicroTx Workflows stores the generated embeddings.
 - **Chunk Size:** Optional. Enter the target chunk size used when splitting input content. The default value is 512. Smaller chunks can improve retrieval precision but create more rows and embedding operations. Larger chunks preserve more context but can reduce retrieval precision.
 - **Dimensions:** Optional. Enter the number of dimensions for the generated embedding vectors. The default value is 512. This value must match the embedding model output and the vector column definition in the database table.
 - **Min Chunk Size (chars):** Optional. Enter the minimum chunk size in characters. The default value is 80 characters.
 - **Min Chunk Length to Embed:** Optional. Enter the minimum length a chunk must have before MicroTx Workflows generates an embedding for it. The default value is 5.
 - **Max Num Chunks:** Optional. Enter the maximum number of chunks to generate from the input content. The default value is 10000.
 - **Index Type:** Optional. Select the type of vector index. Supported values are HNSW for Hierarchical Navigable Small World (HNSW), IVF for Inverted File Flat (IVF), and NONE. The default value is HNSW.
 - **Distance Type:** Optional. Select the distance metric used for vector similarity. Supported values are COSINE, DOT, EUCLIDEAN, MANHATTAN, and EUCLIDEAN_SQUARED. The default value is COSINE. For information about the different functions, see [Vector Distance Metrics](#) in *Oracle AI Vector Search User's Guide*.
 - **Keep Separator:** Optional. Select this option to retain text separators, such as newline characters, during chunking. The default value is true.
 - **Data Source:** Mandatory. Select the content that you want to ingest.
 - **WEB:** Retrieve content from a web URL.
 - **OCI:** Retrieve content from an OCI URL.
 - **LOCAL:** Retrieve content from a file uploaded to local storage.
 - **TEXT:** Provide inline text to ingest.
 - **Enable Idempotency:** Optional. Enable this option when you want the ingestion task to avoid duplicate ingestion for the same workflow idempotency key and task idempotency key. When enabled, MicroTx Workflows performs the vector store update and idempotency result update in a single database transaction. If ingestion fails, the transaction is rolled back. Idempotency is disabled by default.

- **Idempotent Table Name:** Optional. Enter the table name used to store task idempotency lock information. If you do not provide a value, MicroTx Workflows uses `fenced_task_idempotency_lock`.
 - **Idempotency Timeout (ms):** Optional. Specifies how long MicroTx Workflows holds the task-level idempotency lock for this task. The lock prevents duplicate concurrent execution during retries or replays. The default is 600000 milliseconds, or 10 minutes.
6. Click **Save** to save the changes to the workflow.
MicroTx Workflows displays the changes in JSON code.
 7. Review all the changes, and then click **Confirm Save** to save the changes.
If you do not want to save the changes, click **Cancel**, and then click **Reset** to undo all the changes that you have made since the workflow was last saved.

Example

When you enter information in the **Task** tab, the corresponding code of the task is updated in the **JSON** tab. The following sample code displays the JSON code for a GenAI Ingestion task with sample values.

```
{
  "name": "sample_genai_ingest",
  "taskReferenceName": "genai_ingestion_ref",
  "inputParameters": {
    "embeddingModelProfile": {
      "name": "oci-cohere-embedding",
      "model": "cohere.embed-multilingual-image-v3.0"
    },
    "dataStoreProfile": "oracle-atp-123",
    "tableName": "test_vectors",
    "data": {
      "source": "local",
      "filePath": "oracle-transaction-manager-microservices-developer-
guide.pdf"
    },
    "dimensions": "512",
    "chunkSize": "512",
    "minChunkSizeChars": "80",
    "minChunkLengthToEmbed": "5",
    "maxNumChunks": "10000",
    "keepSeparator": true,
    "indexType": "HNSW",
    "distanceType": "cosine"
  },
  "type": "GENAI_INGESTION"
}
```

3.2.6 GenAI Retrieve Task

Use a GenAI Retrieve Task to generate natural language responses by using retrieval augmented generation (RAG). The task retrieves relevant context from a vector table in Oracle Database and sends that context, along with the user query, to a large language model (LLM) to generate the response.

Retrieval Augmented Generation (RAG) is an approach developed to address the limitations of LLMs. RAG combines the strengths of pretrained language models with the ability to retrieve recent and accurate information from a dataset or database in real-time during the generation of responses. For more information about RAG, see [About Retrieval Augmented Generation](#) in *Oracle AI Vector Search User's Guide*.

A GenAI Retrieve Task uses two model configurations:

- An LLM profile and model to generate the final natural language response.
- An embedding profile and embedding model to convert the query into a vector and search the vector table.

Prerequisites

Before you begin, complete the following tasks:

- Create an LLM definition for response generation. See [Create an LLM Definition](#).
- Create an embedding model LLM definition. This should match the embedding model used when the data was ingested.
- Create a database profile for the Oracle Database instance that contains the vector table. See [Create a Database Profile](#).
- Use this task after you have ingested content into a vector table, typically by using a GenAI Ingestion task. For example, you can ingest policy documents, product manuals, or customer records, and then use a GenAI Retrieve task to answer questions based on that ingested content. This task must use the same embedding model, dimensions, index type, and distance type as the ingestion setup for the vector table.

To add a GenAI Retrieve Task

1. Navigate to the **Task** tab in a workflow and view all the tasks that you can add using the Workflow Builder. See [Access the Task Tab in Workflow Builder](#).
2. In the **More Tasks** dialog box, click **GenAI Retrieve Task** to add it to the workflow.
3. Click the task that you have added in the left pane. The **Task** tab in the right pane displays details about the task, such as its name and parameters. Next, let's provide details for the task.
4. In the **Task Details** group, enter the following information.
 - **Task Name:** Mandatory. Enter a unique name for the task. The name must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (`_`) and hyphen (`-`).
 - **Task Reference:** Mandatory. Enter a value to refer to the task within a workflow definition. This value must be unique within a workflow. The task reference value must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (`_`) and hyphen (`-`).
5. In the **GenAI Retrieve Parameters** group, provide the following information.
 - **LLM Profile:** Mandatory. Select the LLM definition used to generate the final response.
 - **Model Name:** Mandatory. Select the model to use to generate response.
 - **Embedding Profile Name:** Mandatory. Select the LLM definition that contains the embedding model. Use the same embedding profile that was used to ingest the data, unless the new model produces compatible embeddings.
 - **Embedding Model:** Mandatory. Select the embedding model used to convert the query into a vector. This should match the model used during ingestion.

- **Data Store Profile:** Mandatory. Select the database profile for the Oracle Database instance that contains the vector table.
 - **Table Name:** Mandatory. Enter the table name that contains the vector embeddings.
 - **Query:** Mandatory. Enter the natural language query. You can enter static text or use workflow expressions, such as `${workflow.input.query}`.
 - **Temperature:** Optional. Controls randomness of the generated response. Lower values generate more deterministic responses. Higher values generate more varied responses. The default value is 0.2.
 - **Max Tokens:** Optional. Sets the maximum number of tokens that the model can generate. The default value is 1024.
 - **TopK (LLM):** Optional. Controls token sampling for the response generation model. In JSON, use `top_k`. The default value is 40.
 - **Dimensions:** Optional. Enter the vector dimension count. The default value is 512. This value must match the vector table definition and the embeddings generated during ingestion.
 - **TopK (RAG):** Optional. Sets the number of matching chunks retrieved from the vector table and passed as context to the LLM. In JSON, use `topK`. The default value is 5.
 - **RAG Type:** Optional. Select the retrieval mode. Supported values are **Naive** and **Advanced**. The default value is **naive**. The advanced mode rewrites the query and compresses retrieved documents before generating the response.
 - **Index Type:** Optional. Select the vector index type. Use the same index configuration as the vector table. Select **HNSW** for Hierarchical Navigable Small World (HNSW), **IVF** for Inverted File Flat (IVF), or **NONE**. The default value is HNSW.
 - **Distance Type:** Optional. Select the distance metric used for vector similarity search. Use the same distance metric used during ingestion. Supported values are **COSINE**, **DOT**, **EUCLIDEAN**, **MANHATTAN**, and **EUCLIDEAN_SQUARED**. The default value is cosine.
6. Click **Save** to save the changes to the workflow.
MicroTx Workflows displays the changes in JSON code.
 7. Review all the changes, and then click **Confirm Save** to save the changes.
If you do not want to save the changes, click **Cancel**, and then click **Reset** to undo all the changes that you have made since the workflow was last saved.

Example

When you enter information in the **Task** tab, the corresponding code of the task is updated in the **JSON** tab. The following sample code displays the JSON code for a GenAI Retrieve task with sample values.

```
{
  "name": "genai_retrieve",
  "taskReferenceName": "genai_retrieve_ref",
  "inputParameters": {
    "llmProfile": {
      "name": "openai-dev",
      "model": "gpt-4o-mini"
    },
    "maxTokens": 4000,
    "embeddingModelProfile": {
```

```

        "name": "oci-cohere-embedding",
        "model": "cohere.embed-multilingual-image-v3.0"
    },
    "dataStoreProfile": "oracle-atp",
    "tableName": "test_vectors",
    "topK": 5,
    "query": "${workflow.input.query}",
    "ragType": "naive",
    "dimensions": "512",
    "indexType": "HNSW",
    "distanceType": "COSINE"
},
"type": "GENAI_RETRIEVE"
}

```

3.2.7 gRPC Task

Use a gRPC task to invoke a unary gRPC method from a workflow. The task uses gRPC server reflection to discover the service, method, request message, and response message descriptors at runtime. The target gRPC server must have server reflection enabled.

Currently, only REFLECTION mode is supported. PROTO file-based invocation is not supported from the Workflow Builder.

To add a gRPC Task

1. Navigate to the **Task** tab in a workflow and view all the tasks that you can add using the Workflow Builder. See [Access the Task Tab in Workflow Builder](#).
2. In the **More Tasks** dialog box, click **gRPC Task** to add it to the workflow.
3. Click the task that you have added in the left pane. The **Task** tab in the right pane displays details about the task, such as its name and parameters. Next, let's provide details for the task.
4. In the **Task Details** group, enter the following information.
 - **Task Name:** Mandatory. Enter a unique name for the task. The name must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (`_`) and hyphen (`-`).
 - **Task Reference:** Mandatory. Enter a value to refer to the task within a workflow definition. This value must be unique within a workflow. The task reference value must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (`_`) and hyphen (`-`).
5. In the **gRPC Task Parameters** group, provide the following information.
 - **Host:** Mandatory. The network address of the gRPC server to call. For example, `localhost` or `my-grpc-service.example.com`.
 - **Port :** Mandatory. The port on which the target gRPC server is listening.
 - **Service Name:** Mandatory. The fully qualified gRPC service name that contains the RPC method you want to invoke. For example, `example.echo.EchoService`.
 - **Method Name:** Mandatory. The unary RPC method to invoke on the specified service. For example, `Echo`.

- **Mode:** The descriptor resolution mode for the gRPC call. Currently, only REFLECTION mode is supported. In this mode, MicroTx Workflows uses server reflection to discover services and methods without requiring PROTO files.
 - **Metadata:** Optional. Task metadata in the workflow definition. Do not use this field for gRPC request headers unless metadata header support is available in your environment.
 - **Payload:** The request message body for the gRPC method, expressed as JSON. The JSON structure must match the protobuf request message expected by the target method.
6. Click **Save** to save the changes to the workflow.
MicroTx Workflows displays the changes in JSON code.
 7. Review all the changes, and then click **Confirm Save** to save the changes.
If you do not want to save the changes, click **Cancel**, and then click **Reset** to undo all the changes that you have made since the workflow was last saved.

Example

The following example invokes the unary Echo method on the example.echo.EchoService service using server reflection. In this example, the target gRPC server must expose the example.echo.EchoService service through server reflection, and the Echo method must be a unary method whose request message contains a message field.

```
{
  "name": "grpc_reflection_example",
  "description": "Invokes a unary gRPC method using server reflection",
  "version": 1,
  "tasks": [
    {
      "name": "grpc_0",
      "taskReferenceName": "grpc_0",
      "type": "GRPC",
      "inputParameters": {
        "host": "localhost",
        "port": 50051,
        "serviceName": "example.echo.EchoService",
        "methodName": "Echo",
        "payload": {
          "message": "hello"
        }
      },
      "mode": "REFLECTION"
    }
  ],
  "inputParameters": [],
  "outputParameters": {},
  "schemaVersion": 2,
  "restartable": true,
  "workflowStatusListenerEnabled": false,
  "failureWorkflow": "",
  "timeoutPolicy": "ALERT_ONLY",
  "timeoutSeconds": 0
}
```

3.2.8 SFTP Task

The SFTP system task allows workflows to securely transfer files—including uploading, downloading, and listing—directly to and from remote SFTP servers, eliminating the need to develop custom simple tasks or worker tasks.

This capability supports a widespread enterprise integration pattern commonly found in industries such as finance, healthcare, government, retail, and supply chain, where batch file exchanges are a standard process. By providing a straightforward, no-code, declarative interface for SFTP operations, this task streamlines integration, minimizes development effort, and leverages existing SFTP connector profiles in MicroTx Workflows for simplified configuration. This results in faster implementation, enhanced data security, and easier maintenance of file-based data exchanges.

Note

The maximum size of a file that can be transferred is 50 MB.

The SFTP task maintains high watermark (an internal timestamp) for each workflow version. When the task runs again, it processes only the files that were added or modified after the last recorded timestamp. For example, let's consider that the SFTP task runs at 5 PM and processes all existing files. During the next execution, the SFTP task considers only files added or updated after 5 PM within the same workflow version.

To add an SFTP Task

1. Navigate to the **Task** tab in a workflow and view all the tasks that you can add using the Workflow Builder. See [Access the Task Tab in Workflow Builder](#).
2. In the **More Tasks** dialog box, click **SFTP Task** to add it to the workflow.
3. Click the task that you have added in the left pane. The **Task** tab in the right pane displays details about the task, such as its name and parameters. Next, let's provide details for the task.
4. In the **Task Details** group, enter the following information.
 - **Task Name:** Mandatory. Enter a unique name for the task. The name must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (_) and hyphen (-).
 - **Task Reference:** Mandatory. Enter a value to refer to the task within a workflow definition. This value must be unique within a workflow. The task reference value must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (_) and hyphen (-).
5. In the **SFTP Task Parameters** group, in the **Operation** drop-down list select one of the following tasks that you want to perform with the SFTP server.
 - **Upload:** Transfers files to the specified remote SFTP directory. Reads files from the local file system storage repository, filters them by pattern, and securely uploads them to a designated remote SFTP directory with MD5 checksum validation.
 - **Download:** Retrieves files from a remote SFTP directory and saves the files to your chosen destination, such as local file system or Oracle Cloud Infrastructure (OCI) Object Storage. Additionally, based on the settings, it filters the retrieved results by file name pattern, tracks when files were last processed, and calculates MD5 checksum for verification.

- **List:** Displays names of all the files available in the specified remote SFTP directory.
 - **Exists:** Verifies the presence of a file in the specified remote SFTP directory.
6. If you select the **Upload** operation, provide the following information and then skip to the last step to save the changes.
- **Source Type:** Select one of the following as the source, and then provide the connection details.
 - **File Resource:** Takes files from the file system repository, and then stores the files in the specified destination SFTP server. When you select this option, the default base path of the file storage is the value specified in the `conductor.connector.storage-path` property in the `application.properties` file. When you select this source type, optionally you can provide a Java regular expression to filter file names in the specified remote directory in the **File Name Pattern** field. Case-insensitive matching is supported. If you do not enter a value, all executable files are filtered by default.
 - **OCI Bucket:** Takes files in the specified OCI Object Storage bucket in the provided namespace, and then stores the files in the specified destination SFTP server. SFTP task uses the details provided in the selected Cloud profile to establish a connection with the destination Object Storage bucket.
 - * **Cloud Profile:** Select the Cloud profile containing the credentials and configuration required to access OCI Object Storage.
 - * **OCI Namespace:** Enter the Object Storage namespace associated with your OCI Object Storage bucket.
 - * **OCI Bucket Name:** Enter the name of the existing bucket in your OCI Object Storage where files should be stored. Buckets are logical containers for storing objects.
 - * **OCI Object Prefix:** Optional. Specify a prefix to be added to all objects uploaded to the bucket.
 - **Overwrite:** Optional. Select this option to always overwrite any existing file with the new file during upload. This option is disabled by default.
 - **Maximum File Transfer:** Optional. Enter a natural number to specify the maximum number of files to transfer in a single task execution. The default value is 10. This restricts the number of files for better control over large batch transfers.
 - **Destination SFTP Profile:** Mandatory. Select an existing SFTP connector profile that contains the connection settings for the destination SFTP server to which you want to upload files.
 - **Destination Directory:** Mandatory. Enter the path of the directory in SFTP server to which you want to upload files.
7. If you select the **Download** operation, provide the following information and then skip to the last step to save the changes.
- **Source SFTP Profile:** Mandatory. Select an existing SFTP connector profile that contains the connection settings for the source SFTP server from which you want to download files.
 - **Source Directory:** Mandatory. Enter the path of the directory in SFTP server from which you want to download files.
 - **File Name Pattern:** Optional. Provide a Java regular expression to filter file names in the specified remote directory. Case-insensitive matching is supported. If you do not enter a value, all executable files are filtered by default.

- **Destination Type:** Select one of the following as the destination, and then provide the connection details.
 - **File Resource:** Stores files received from the specified source SFTP server to the MicroTx Workflows file storage.
 - **OCI Bucket:** Stores files in the specified OCI Object Storage bucket in the provided namespace and uses the details provided in the Cloud profile selected in the **Cloud Profile** to establish a connection with the destination Object Storage bucket.
 - * **Cloud Profile:** Select the Cloud profile containing the credentials and configuration required to access OCI Object Storage.
 - * **OCI Namespace:** Enter the Object Storage namespace associated with your OCI object storage bucket.
 - * **OCI Bucket Name:** Enter the name of the existing bucket in your OCI Object Storage where files should be stored. Buckets are logical containers for storing objects.
 - * **OCI Object Prefix:** Optional. Specify a prefix to be added to all objects uploaded to the bucket.
 - **SFTP:** Transfers files to the SFTP directory specified in the **Destination Directory** field and uses the details provided in the SFTP profile selected in the **Destination SFTP Profile** to establish a connection with the destination SFTP server.
 - **Idempotency Strategy:** Select one of the following methods to handle existing files on the destination server to ensure safe, repeatable operations.
 - **none:** Always perform the operation, without checking for existing files.
 - **skipIfExists:** Skip the operation if the file already exists on the destination.
 - **overwrite:** Always overwrite any existing file with the new file.
 - **checksum:** Compare files using MD5 checksum; only transfer if local and remote files differ.
 - **Maximum File Transfer:** Enter a natural number to specify the maximum number of files to transfer in a single task execution. The default value is 10. This restricts the number of files for better control over large batch transfers.
8. If you select the **List** operation, provide the following information and then skip to the last step to save the changes.
- **Source SFTP Profile:** Mandatory. Select an existing SFTP connector profile that contains the connection settings for the source SFTP server from which you want to list file names.
 - **Source Directory:** Mandatory. Enter the path of the directory in SFTP server from which you want to download files.
 - **File Name Pattern:** Optional. Provide a Java regular expression to filter file names in the specified remote directory. Case-insensitive matching is supported. If you do not enter a value, all executable files are filtered by default.
9. If you select the **Exists** operation, provide the following information and then skip to the last step to save the changes.
- **Source SFTP Profile:** Mandatory. Select an existing SFTP connector profile that contains the connection settings for the source SFTP server from which you want to download files.

- **Source File:** Mandatory. Enter the path of the source file on the SFTP server to verify if the file exists.
10. Click **Save** to save the changes to the workflow.
MicroTx Workflows displays the changes in JSON code.
 11. Review all the changes, and then click **Confirm Save** to save the changes.
If you do not want to save the changes, click **Cancel**, and then click **Reset** to undo all the changes that you have made since the workflow was last saved.

Example

When you enter information in the **Task** tab, the corresponding code of the task is updated in the **JSON** tab. The following example displays the JSON code for an SFTP task which downloads files from a specified SFTP server and uploads them to the provided OCI Object Storage bucket.

```
{
  "name": "Download files from SFTP Server to OCI Object Storage Bucket",
  "taskReferenceName": "download_files_SFTP_server",
  "inputParameters": {
    "operation": "DOWNLOAD",
    "sourceSftpProfileName": "sftp-profile-name",
    "sourceDirectory": "/home/ubuntu/inbound",
    "fileNamePattern": "^.*\\.xml$",
    "destinationType": "OCI_BUCKET",
    "ociNamespace": "oabcs1",
    "ociBucketName": "microtx",
    "ociProfile": "phoenix0",
    "idempotencyStrategy": "skipIfExists",
    "maximumFileTransfer": 5
  },
  "type": "SFTP"
}
```

3.2.9 HTTP Task

Use an HTTP task to make REST API calls to external services.

To add an HTTP Task

1. Navigate to the **Task** tab in a workflow and view all the tasks that you can add using the Workflow Builder. See [Access the Task Tab in Workflow Builder](#).
2. In the **More Tasks** dialog box, click **HTTP Task** to add it to the workflow.
3. Click the task that you have added in the left pane. The **Task** tab in the right pane displays details about the task, such as its name and parameters. Next, let's provide details for the task.
4. In the **Task Details** group, enter the following information.
 - **Task Name:** Mandatory. Enter a unique name for the task. The name must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (_) and hyphen (-).
 - **Task Reference:** Mandatory. Enter a value to refer to the task within a workflow definition. This value must be unique within a workflow. The task reference value must

be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (_) and hyphen (-).

5. In the **HTTP Task Parameters** group, provide the following information.
 - **Request URI:** Mandatory. Enter the URI of the object on which you want to perform API calls. Each object is identified uniquely by its URI.
 - **HTTP Method:** Mandatory. Specify the operation that you want to perform using the standard HTTP method verbs.
 - POST creates a request using the raw data in the request body. This is the default value.
 - GET retrieves data.
 - PUT updates data.
 - DELETE removes data.
 - **Accept:** Specify the accepted media types in the Accept header of the request.
 - **Content-Type:** Select the expected version of the content in an HTTP request body sent to the API server and the response.
 - **Connection Timeout (ms):** Maximum permissible time, in milliseconds (ms), to establish a TCP connection. Default value is 5000 ms.
 - **Read Timeout (ms):** Maximum permissible time, in milliseconds (ms), to read full response after connection. The default value is 60000 ms. You can tune this for large payloads, such as transaction logs.
 - **Enlist in transaction:** Optional. You must select this option only if you want to enlist the HTTP task as a participant service in a global XA transaction. This is disabled by default.
 - **Trust self-signed certificate:** Optional. Select this option when you want to use HTTPS protocol.
 - **Add Header:** Add headers that you want to pass in the header section of the HTTP request or response. If a server called by HTTP task supports idempotency, pass the idempotency key in the HTTP request header.
 - **Body:** The request body, which is required for POST and PUT HTTP methods.
6. Click **Save** to save the changes to the workflow.
MicroTx Workflows displays the changes in JSON code.
7. Review all the changes, and then click **Confirm Save** to save the changes.
If you do not want to save the changes, click **Cancel**, and then click **Reset** to undo all the changes that you have made since the workflow was last saved.

Example

When you enter information in the **Task** tab, the corresponding code of the task is updated in the **JSON** tab. The following example displays the JSON code for an HTTP task that makes REST API calls to an email notification service.

```
{
  "name": "Send Email notification",
  "taskReferenceName": "email_notify",
  "inputParameters": {
    "method": "POST",
    "uri": "http://notification-service:8085/email-service/sendMail",
```

```

        "headers": {
            "Content-Type": "application/json"
        },
        "body": {
            "from": "microtx.user@localhost",
            "to": "microtx.user@microtx.com",
            "cc": "",
            "subject": "Loan approval request!",
            "body": "Please approve loan req ${workflow.workflowId}",
            "isEmailBodyText": true
        }
    },
    "type": "HTTP"
}

```

3.2.10 JSON RPC Task

Use the JSON RPC task to invoke JSON-RPC 2.0 services over HTTP(S) using POST requests with application/json payloads. It supports single-call mode for one RPC invocation and batched requests (array) for multiple calls in a single HTTP request.

When to use this task

Use this task to:

- Call JSON-RPC services. For example, blockchain nodes, such as Ethereum methods or custom RPC APIs as part of your workflow.
- Batch multiple JSON-RPC calls in a single HTTP request for efficiency.
- Fire-and-forget notification calls by omitting the `id` field (server does not reply for those items).

This task constructs standardized JSON-RPC 2.0 payloads from UI inputs, sends them to the endpoint, and captures responses for workflow processing.

Do not use this task for calls to HTTP server and use it only to call JSON-RPC compliant APIs.

To add a JSON RPC Task

1. Navigate to the **Task** tab in a workflow and view all the tasks that you can add using the Workflow Builder. See [Access the Task Tab in Workflow Builder](#).
2. In the **More Tasks** dialog box, click **JSON RPC Task** to add it to the workflow.
3. Click the task that you have added in the left pane. The **Task** tab in the right pane displays details about the task, such as its name and parameters. Next, let's provide details for the task.
4. In the **Task Details** group, enter the following information.
 - **Task Name:** Mandatory. Enter a unique name for the task. The name must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (`_`) and hyphen (`-`).
 - **Task Reference:** Mandatory. Enter a value to refer to the task within a workflow definition. This value must be unique within a workflow. The task reference value must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (`_`) and hyphen (`-`).
5. In the **JSON_RPC Task Parameters** group, provide the following information.

- **Endpoint URI:** Mandatory. Target HTTPS or HTTP URL for the JSON-RPC POST request.
 - **Batch Request:** Select this option to enable batch mode for multiple RPC calls in one payload. By default, this option is not selected.
 - **Accept:** Select the accepted media types in the accept header of the request.
 - **Content-Type:** Select the supported media type. If you select application/json as the media type, all the content in the request body must be encoded in JavaScript Object Notation (JSON).
 - **RPC Method:** Mandatory only for single-call mode. Enter the JSON-RPC method name. The name is case sensitive. Maps to method in payload and it is ignored in the batch mode.
 - **Request ID:** Optional. It correlates the request and response. This option is used only in only for single-call mode and is ignored in the batch mode. Omit for JSON-RPC notifications where server MUST NOT reply for a request.
 - **Params (JSON object or array):** Specify the JSON-RPC parameters. In the single call mode, enter an array or a map of named objects. In the batch mode, enter an array of BatchItem objects which provides details for the following.
 - rpcMethod: Required.
 - params: Optional. Enter a map of named objects.
 - id: Optional. It correlates the request and response. Omit for JSON-RPC notifications where server MUST NOT reply for a request.
 - **Connection Timeout (ms):** Maximum permissible time, in milliseconds (ms), to establish a TCP connection. Default value is 5000 ms.
 - **Read Timeout (ms):** Maximum permissible time, in milliseconds (ms), to read full response after connection. The default value is 60000 ms. You can tune this for large payloads, such as transaction logs.
 - **Add Header:** Add custom HTTP headers, as key-value pairs, that you want to pass in the header section of a request or response.
6. Click **Save** to save the changes to the workflow.
- MicroTx Workflows displays the changes in JSON code.
7. Review all the changes, and then click **Confirm Save** to save the changes.
- If you do not want to save the changes, click **Cancel**, and then click **Reset** to undo all the changes that you have made since the workflow was last saved.

3.2.11 SQL Task

The SQL task allows workflows to retrieve or update data based on the SQL statements that you define. It enables workflows to interact with SQL databases to execute queries or store information.

Prerequisites

Before you begin, you must complete the following tasks:

- Create a database profile with capability specified as RELATIONAL. See [Create a Database Profile](#).

- Decide if you want the SQL Task to participate in a global XA transaction within MicroTx Workflows. In such scenarios, MicroTx Workflows starts the XA transaction. SQL Tasks automatically join the active global transaction when it runs and executes its SQL without committing locally. Within an XA transaction, MicroTx Workflows commits multiple SQL operations and other XA actions in case of success. In case of any failure, MicroTx Workflows rolls back all the actions. This ensures that SQL operations remain consistent with other tasks in a workflow, such as microservice calls or other database updates and succeed or fail together as a single, atomic unit.
- - To use a SQL task in an XA Transaction you need to use it between a transaction task with action set to BEGIN and another transaction task with action set to COMMIT. The Transaction task with action set to BEGIN starts the global transaction. See [Transaction Task](#). When used between BEGIN and COMMIT Transaction tasks, the SQL task participates in workflow-defined transaction boundaries instead of controlling its own transactions.
 - Enter the URL of the participant application in the `conductor.tasks.microtx.participantCallbackUrl` property in the Helm chart. This is the URL that the MicroTx Distributed Transaction coordinator uses to communicate with MicroTx Workflows.

Supported Data Sources for SQL Task

You can execute SQL stored procedures that are saved in any Oracle Database and PostgreSQL database to which you can connect using a JDBC connection string.

If you want the SQL Task to participate in a global XA transaction, this is supported only when using an Oracle Database.

Supported SQL Data Types

Variables defined in SQL stored procedures are exposed as parameters in the SQL tasks that you create or run.

The following SQL data types are supported for input and output parameters when you use stored procedures in SQL tasks:

- VARCHAR
- NUMERIC
- BOOLEAN

To add a SQL Task

1. Navigate to the **Task** tab in a workflow and view all the tasks that you can add using the Workflow Builder. See [Access the Task Tab in Workflow Builder](#).
2. In the **More Tasks** dialog box, click **SQL Task** to add it to the workflow.
3. Click the task that you have added in the left pane. The **Task** tab in the right pane displays details about the task, such as its name and parameters. Next, let's provide details for the task.
4. In the **Task Details** group, enter the following information.
 - **Task Name:** Mandatory. Enter a unique name for the task. The name must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (`_`) and hyphen (`-`).
 - **Task Reference:** Mandatory. Enter a value to refer to the task within a workflow definition. This value must be unique within a workflow. The task reference value must

be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (_) and hyphen (-).

5. In the **SQL Task Parameters** group, provide the following information.
 - **Database Profile:** Mandatory. Select the relational database profile which contains the database connection parameters.
 - **SQL Type:** Mandatory. Select one of the following type of SQL statements that you want to run.
 - **SELECT:** Retrieves data from the database.
 - **UPDATE:** Adds new data and modifies existing data in the database.
 - **DELETE:** Removes existing data from the database.
 - **PLSQL:** Helps you create and perform SQL operations from PL/SQL in the SQL task.
 - **SQL Statement:** Enter the SQL statements that you want to execute every time this task runs. Refer to the Usage Notes section for more details.
 - **SQL Parameters** or **PLSQL Parameters:** Optional. Specify the parameters that will replace any wildcards in the SQL statement that you provide.
 - **Enlist in transaction:** Optional. You must select this option only if you want to enlist the SQL task as a participant service in a global XA transaction. This is disabled by default.
 - **Enable Idempotency:** Optional. Enable this option when you want the SQL task to avoid duplicate execution for the same workflow idempotency key and task idempotency key. If you enable idempotency, you may also provide values for the following fields:
 - **Idempotency Table Name:** Optional. Enter the table name used to store task idempotency lock information. If you do not provide a value, MicroTx Workflows uses `fenced_task_idempotency_lock`.
 - **Idempotency Timeout:** Optional. Specifies how long MicroTx Workflows holds the task-level idempotency lock for this task. The lock prevents duplicate concurrent execution during retries or replays. The default is 600000 milliseconds, or 10 minutes.
6. Click **Save** to save the changes to the workflow.

MicroTx Workflows displays the changes in JSON code.
7. Review all the changes, and then click **Confirm Save** to save the changes.

If you do not want to save the changes, click **Cancel**, and then click **Reset** to undo all the changes that you have made since the workflow was last saved.

Usage Notes

Consider the following points when you enter the SQL statements that you want to execute every time the SQL task runs.

- The SQL statements must be idempotent.
- Multiple SQL statements can be delimited by ;
- If you want to enter a SELECT statement, then you can enter only a single statement.
- You can use the wild card ? in the query.

Example

When you enter information in the **Task** tab, the corresponding code of the task is updated in the **JSON** tab. The following example displays the JSON code for a SQL task with sample values.

```

{
  "name": "oracleTask",
  "taskReferenceName": "oracle_refer",
  "inputParameters": {
    "databaseProfile": "MyDatabaseProfile",
    "sqlStatement": "insert into accounts (account_id,name,amount) values
(?, ?, ?)",
    "type": "UPDATE",
    "parameters": [
      "${workflow.input.oracle.accountName}",
      "${workflow.input.oracle.accountName}",
      "${workflow.input.oracle.accountBalance}"
    ]
  },
  "type": "SQL"
}

```

3.2.12 Transaction Task

The transaction task enables you to use MicroTx Distributed Transactions coordinator to manage XA transactions that span across multiple tasks in MicroTx Workflows.

The Transaction Task provides the capability of distributed transaction management as a system task within MicroTx Workflows. This task communicates with MicroTx Distributed Transactions coordinator using HTTP REST APIs and currently supports XA distributed transactions. This allows workflows to automatically initiate, commit, or roll back transactions based on the value provided for the action parameter. When a transaction starts, the transaction context (txCtx) is maintained within the workflow input instance and it is shared across the tasks that are called between the begin and commit actions. At present, txCtx propagation is only implemented for HTTP tasks and SQL tasks; when an HTTP task or SQL task is executed within this transaction context, the link headers related to txCtx are propagated.

Prerequisites

Before you begin, ensure that you have installed and configured the MicroTx Distributed Transactions coordinator. See [Access MicroTx Coordinator](#).

To add a Transaction Task

1. Navigate to the **Task** tab in a workflow and view all the tasks that you can add using the Workflow Builder. See [Access the Task Tab in Workflow Builder](#).
2. In the **More Tasks** dialog box, click **Transaction** to add it to the workflow.
3. Click the task that you have added in the left pane. The **Task** tab in the right pane displays details about the task, such as its name and parameters. Next, let's provide details for the task.
4. In the **Task Details** group, enter the following information.

- **Task Name:** Mandatory. Enter a unique name for the task. The name must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (_) and hyphen (-).
 - **Task Reference:** Mandatory. Enter a value to refer to the task within a workflow definition. This value must be unique within a workflow. The task reference value must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (_) and hyphen (-).
5. In the **Transaction Parameters** group, provide the following information.
- **Coordinator URL:** Mandatory. Enter the details required to create the external URL to access MicroTx from outside the Kubernetes cluster where you have deployed the service. See [Access MicroTx Coordinator](#). For example, `https://192.0.2.1:443/api/v1`.
 - **Transaction Type:** Select XA to specify the distributed transaction protocol used by the microservices.
 - **Action:** Select one of the following values. Accordingly, the Transaction task begins, commits, or rolls back a transaction.
 - BEGIN: starts the XA transaction.
 - COMMIT: commits the XA transaction.
 - ROLLBACK: rolls back the XA transaction.
- When you begin, commit, or roll back a transaction, you must mention a value for **Action**.
- **Transaction Timeout (ms):** Specify the transaction timeout, in milliseconds, only when the action is set to BEGIN. If you have also specified the transaction timeout for the transaction initiator application, that timeout overrides the value you specify here.
 - **Add Header:** Add headers that you want to pass in the header section of the outgoing request.
6. Click **Save** to save the changes to the workflow.
- MicroTx Workflows displays the changes in JSON code.
7. Review all the changes, and then click **Confirm Save** to save the changes.
- If you do not want to save the changes, click **Cancel**, and then click **Reset** to undo all the changes that you have made since the workflow was last saved.

Usage Notes

When you want to perform an XA transaction with a workflow, use transaction tasks within a workflow. In between these two Transaction tasks, you can add other tasks that are part of the XA transaction based on the business logic. You can add non-XA compliant tasks and XA-compliant tasks, such as HTTP task and SQL task. The following example describes how you can call SQL task and HTTP task between two Transaction tasks.

1. Create a transaction task with action set to BEGIN.
2. Create a SQL task to query balance on account for verification.
3. Create an HTTP task to withdraw money from one account and deposit it into another account.
4. Create another transaction task with action set to COMMIT to commit the transaction.
5. Create a SQL task to query balance on accounts after withdraw and deposit.

Example

```
{
  "name": "begin_tx",
  "taskReferenceName": "begin_tx",
  "inputParameters": {
    "transaction_request": {
      "coordinatorUrl": "http://192.0.2.1:9000/api/v1",
      "transactionType": "XA",
      "action": "BEGIN"
    },
    "type": "TRANSACTION"
  }
}
```

3.2.13 TX EventQ Publish Task

Use the TX EventQ Publish task to publish a message to a queue in Oracle Database, which serves as a message broker.

This task uses Transactional Event Queues (TxEventQ) and Advanced Queuing (AQ), message queuing systems integrated with Oracle Database. This robust and feature-rich message queuing system leverages the functions of Oracle Database so that producers and consumers can exchange messages with high throughput, by storing messages persistently and propagating messages between queues on different databases. Producers create messages and publish them to a specific topic. This action enqueues the message into the topic's queue within the database. Consumers subscribe to topics to receive messages. When a message is published to a topic, all subscribed consumers are notified and can dequeue the message for processing. For more information, see Introduction to Transactional Event Queues and Advanced Queuing in *Transactional Event Queues and Advanced Queuing User's Guide*.

Prerequisites

Before you begin, ensure that you have completed the following tasks.

- Create a database profile to define the connection parameters to the Oracle Database where you want to publish messages. This task supports only Oracle AI Database 26ai and Oracle Database 19c as the database engine and does not support other databases, such as PostgreSQL database. See [Create a Database Profile](#). MicroTx Workflows uses the credentials provided in the database profile to push the provided message to a topic in Oracle Database.
- Create a topic or event queue in Oracle Database. In Oracle Database 19c, use the CREATE_SHARDED_QUEUE procedure. See CREATE_SHARDED_QUEUE Procedure in *Oracle AI Database PL/SQL Packages and Types Reference*. To create a topic or event queue in Oracle AI Database 26ai use Java or PL/SQL code. See Java Message Service for Transactional Event Queues and Advanced Queuing or Oracle Database Advanced Queuing Operations Using PL/SQL in *Oracle AI Database Transactional Event Queues and Advanced Queuing User's Guide*.
Note down the name of the topic as you will need to provide it later.
- Enable event handlers for the message brokers. See Event Handler Configuration Properties in *MicroTx Installation and Configuration Guide*.

To add a TX EventQ Publish Task

1. Navigate to the **Task** tab in a workflow and view all the tasks that you can add using the Workflow Builder. See [Access the Task Tab in Workflow Builder](#).
2. In the **More Tasks** dialog box, click **TX EventQ Publish Task** to add it to the workflow.
3. Click the task that you have added in the left pane. The **Task** tab in the right pane displays details about the task, such as its name and parameters. Next, let's provide details for the task.
4. In the **Task Details** group, enter the following information.
 - **Task Name:** Mandatory. Enter a unique name for the task. The name must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (`_`) and hyphen (`-`).
 - **Task Reference:** Mandatory. Enter a value to refer to the task within a workflow definition. This value must be unique within a workflow. The task reference value must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (`_`) and hyphen (`-`).
5. In the **TXEventQ_PUBLISH Parameters** group, provide the following information.
 - **Database Profile:** Mandatory. Specify the name of the database profile which contains the connection parameters to connect to the Oracle Database instance where you want to publish the events.
 - **Topic:** Mandatory. Enter the name of the destination queue or topic in Oracle Database where messages are published.
 - **Value:** Mandatory. Enter the message that you want to send to the topic.
 - **Publisher Agent Name:** Mandatory. Enter the name or identifier of a publisher agent, producer application, entity, or task that enqueues messages into a topic.
 - **Enable Idempotency:** Optional. Enable this option when you want the TX EventQ Publish task to avoid duplicate execution for the same workflow idempotency key and task idempotency key. If you enable idempotency, you may also provide values for the following fields:
 - **Idempotency Table Name:** Optional. Enter the table name used to store task idempotency lock information. If you do not provide a value, MicroTx Workflows uses `fenced_task_idempotency_lock`.
 - **Idempotency Timeout:** Optional. Specifies how long MicroTx Workflows holds the task-level idempotency lock for this task. The lock prevents duplicate concurrent execution during retries or replays. The default is 600000 milliseconds, or 10 minutes.
6. Click **Save** to save the changes to the workflow.

MicroTx Workflows displays the changes in JSON code.
7. Review all the changes, and then click **Confirm Save** to save the changes.

If you do not want to save the changes, click **Cancel**, and then click **Reset** to undo all the changes that you have made since the workflow was last saved.

Example

When you enter information in the **Task** tab, the corresponding code of the task is updated in the **JSON** tab. The following example displays the JSON code for a TX EventQ Publish task with sample values.

```
{
  "name": "txeventq_publish_1234",
  "taskReferenceName": "txeventq_publish_sample",
  "inputParameters": {
    "databaseProfile": "sample_db_profile_name",
    "topic": "sample_queue",
    "value": "This is the message I want to send. A simple hello.",
    "publisherAgentName": "Workflow_Publisher_sampleName",
    "enableIdempotency": true,
    "idempotentTableName": "fenced_task_idempotency_lock"
  },
  "type": "TXEVENTQ_PUBLISH"
}
```

3.2.14 Human Task

Include human tasks in Agentic AI workflows to add manual review, approval, rejection, or other human intervention before a workflow continues. This is useful when an AI-generated decision, recommendation, or action requires human validation before it affects a business process.

When a workflow reaches a human task, MicroTx Workflows pauses the execution and creates a workflow notification. It keeps the task in progress until an authorized user can review the workflow execution details, inspect the AI task outputs, and approve or reject the task. The workflow waits at that step and resumes only after the human task is updated. Based on the response of the authorized user, the workflow continues according to the business logic defined in the workflow. For information about handling notifications, see [View and Approve Workflow Notifications](#).

Human-in-the-loop review helps improve trust in agentic workflows by keeping people in control of sensitive or high-impact decisions, such as approvals, exception handling, compliance checks, customer communications, actions that interact with external systems, or agentic workflows where an AI-generated recommendation or action must be reviewed before execution continues.

To add a human task to a workflow:

1. Navigate to the **Task** tab in a workflow and view all the tasks that you can add using the Workflow Builder. See [Access the Task Tab in Workflow Builder](#).
2. In the **More Tasks** dialog box, click **Human Task** to add it to the workflow.
3. Click the task that you have added in the left pane. The **Task** tab in the right pane displays details about the task, such as its name and parameters. Next, let's provide details for the task.
4. In the **Task Details** group, enter the following information.
 - **Task Name:** Mandatory. Enter a unique name for the task. The name must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (_) and hyphen (-).

- **Task Reference:** Mandatory. Enter a value to refer to the task within a workflow definition. This value must be unique within a workflow. The task reference value must be between 1 to 128 alphanumeric characters in length and cannot contain spaces or any special characters. Optionally, you can use underscore (_) and hyphen (-).
- 5. Optional. In the **Human Task Parameters** group, click **Add Parameter** to specify the values that the task requires as an input.
- 6. Click **Save** to save the changes to the workflow.
MicroTx Workflows displays the changes in JSON code.
- 7. Review all the changes, and then click **Confirm Save** to save the changes.
If you do not want to save the changes, click **Cancel**, and then click **Reset** to undo all the changes that you have made since the workflow was last saved.

Example

When you enter information in the **Task** tab, the corresponding code of the task is updated in the **JSON** tab. The following example displays the JSON code for a Human task that requires verification from a human.

```
{
  "name": "human_aml_verification_task",
  "taskReferenceName": "human_aml_verification",
  "type": "HUMAN",
  "inputParameters": {
    "applicant": "${extract_loan_details.output}"
  }
}
```

3.2.15 Inline Task

Use an Inline task when you need to run lightweight, simple logic directly within your workflow.

Inline tasks are ideal for operations that you can handle with a small JavaScript function—such as data transformations, basic calculations, or simple condition checks. For example, converting a JSON date string to a different format or adding two numbers from the input for downstream use. You can write the code for these operations within the workflow definition, making it quick and easy to maintain for straightforward logic.

When your workflow needs to trigger more complex or extensive custom logic that cannot (or should not) be contained within a workflow definition, use a simple task. For example, tasks that require full programming power, flexibility, robust handling, external dependencies, or access to external systems, such as processing an image with an AI model hosted in your infrastructure. See [About Simple Tasks or Worker Tasks](#).

For more information about this task, see [System Tasks](#) in open-source Conductor documentation.

3.2.16 Wait Task

Use a wait task to pause a workflow.

For more information about this task, see [System Tasks](#) in open-source Conductor documentation.

3.2.17 Kafka Publish Task

Use this task to send messages to another microservice through a Kafka topic with customizable headers and serializers.

For more information about this task, see [System Tasks](#) in open-source Conductor documentation.

3.2.18 Start Workflow Task

Use this task to run another workflow asynchronously from a workflow that is currently being executed.

For more information about this task, see [Operators](#) in open-source Conductor documentation.

3.2.19 JSON JQ Transform Task

Use this task to process JSON data using jq.

For more information about this task, see [System Tasks](#) in open-source Conductor documentation.

3.3 Manage Task Definitions for Simple Tasks

This section provides information about how you can create or import a new task definition, and then add it to a workflow. It also provides information about managing task definitions for simple tasks.

You can also define system tasks and operator tasks using task definitions, but the recommended approach is to create a workflow using Workflow Builder, and then adding tasks directly to the workflow. To add a task using Workflow Builder, see [Define a Workflow Using Workflow Builder](#).

Note that task definitions that you create are visible in the Workflow Builder. However, if you add a task directly to a workflow using the Workflow Builder, you can view information about the task only within a specific workflow.

Topics

- [Create a Task Definition](#)
Create a task definition to define a simple task or worker.
- [Import a Task Definition](#)
You can import multiple task definitions using Postman Collection.
- [Add a Task Definition to a Workflow](#)
- [Configure Timeout for Tasks](#)
You can configure timeouts for worker tasks by specifying timeout options in the task definition. These timeout settings apply to custom worker tasks that are registered as task definitions.
- [Edit a Task Definition](#)
You can update all the field values you specified when creating a task definition, except for the name.
- [View a Task Definition](#)
You can retrieve and view the details of a task definition.

- [View Task Queues](#)
Use the Task Queues page to monitor worker-pollled task queues, such as simple tasks or worker tasks. This page helps you check whether workers are polling for a selected task and whether any task executions are waiting in the queue.
- [Delete a Task Definition](#)
You can delete a task definition when it is no longer required.

3.3.1 Create a Task Definition

Create a task definition to define a simple task or worker.

1. Open the navigation menu and click **Definitions**.
2. Click the **Tasks** tab.

The **Tasks** list page opens. All the tasks that you have defined are displayed in a table.

3. Click  (**Add New Task Definition**).

The New Task Definition dialog box is displayed.

4. Select **Form** to enter values for the fields that are displayed or select **JSON** to paste the task definition in JSON format.

For details about the fields of a simple task or worker, see [Task Definition](#). The following code sample provides the JSON code of the task definition for a simple task that accepts a loan request.

```
{
  "name": "loan_processing_agent_task",
  "taskReferenceName": "loan_processing_agent_task",
  "inputParameters": {
    "applicantId": "12345"
  },
  "type": "SIMPLE"
}
```

5. Click **Submit** to save the changes.

Details of the task that you have created is displayed on the **Tasks** list page.

Next: [Add a Task Definition to a Workflow](#).

3.3.2 Import a Task Definition

You can import multiple task definitions using Postman Collection.

You can export JSON files as Postman Collection directly from Postman. See [Export data from Postman](#) in Postman Docs.

You cannot import certain Postman features, such as references to dynamic variables, REST API calls with file as form data, runtime environment variables, JavaScript and other scripts. If a Postman Collection contains any of these features, MicroTx Workflows does not import these features into the tasks in MicroTx Workflows. For example, if a task definition contains an environment variable in Postman Collection, MicroTx Workflows imports the task definition, but not the environment variable.

1. Open the navigation menu and click **Definitions**.

- Click the **Tasks** tab.

The **Tasks** list page opens. All the tasks that you have defined are displayed in a table.

- Click **Import**.

The **Import Task Definition** dialog box opens.

- Enter a unique name. This name is prefixed to all the task definitions that you are going to import using Postman Collection. This ensures that tasks have unique names in the **Tasks** list page.
- Click **Browse** to search and select the file that contains the task definitions.
- Click **Submit**.

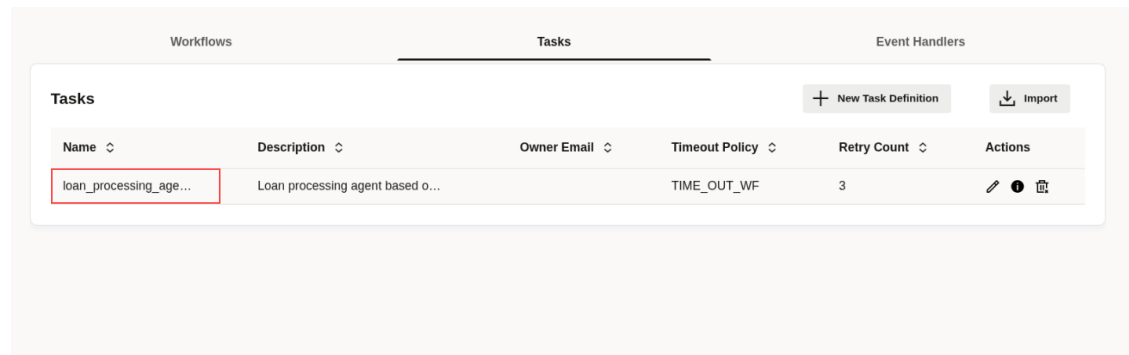
Details of the task that you have imported are displayed in the **Tasks** list page.

Next: [Add a Task Definition to a Workflow](#).

3.3.3 Add a Task Definition to a Workflow

After importing or creating a new task definition, you can add the task definition to a workflow using the Workflow Builder. Let's understand by adding an example worker task or simple task, called `loan_processing_agent_task`, to a new workflow.

Before you begin, ensure that you have created or imported the task definition and it is visible in the **Tasks** list page as shown in the following image.



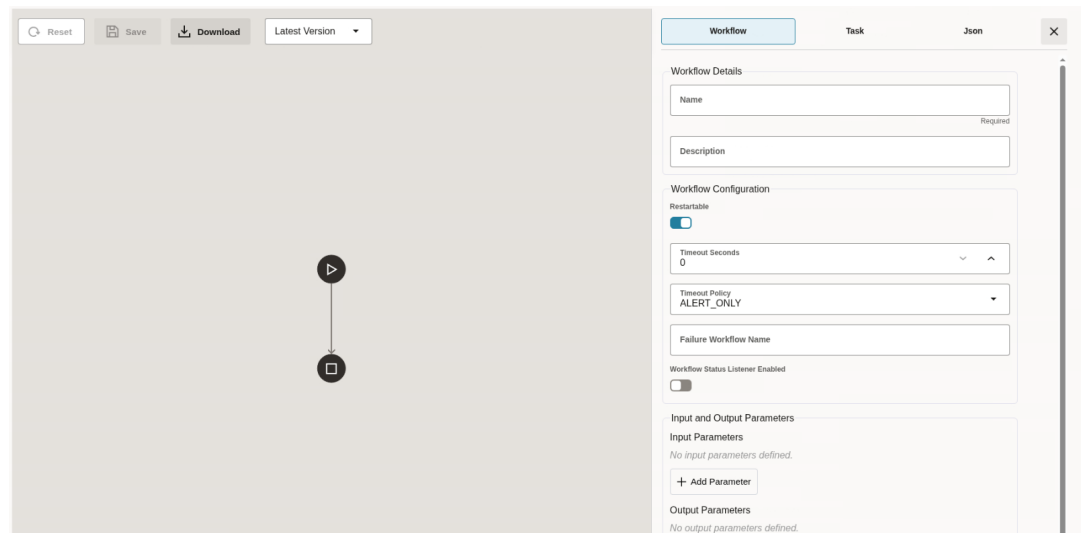
Tasks					
Name	Description	Owner Email	Timeout Policy	Retry Count	Actions
loan_processing_age...	Loan processing agent based o...		TIME_OUT_WF	3	[Edit] [Info] [Delete]

- Open the navigation menu and click **Definitions**.
- Click the **Workflows** tab.

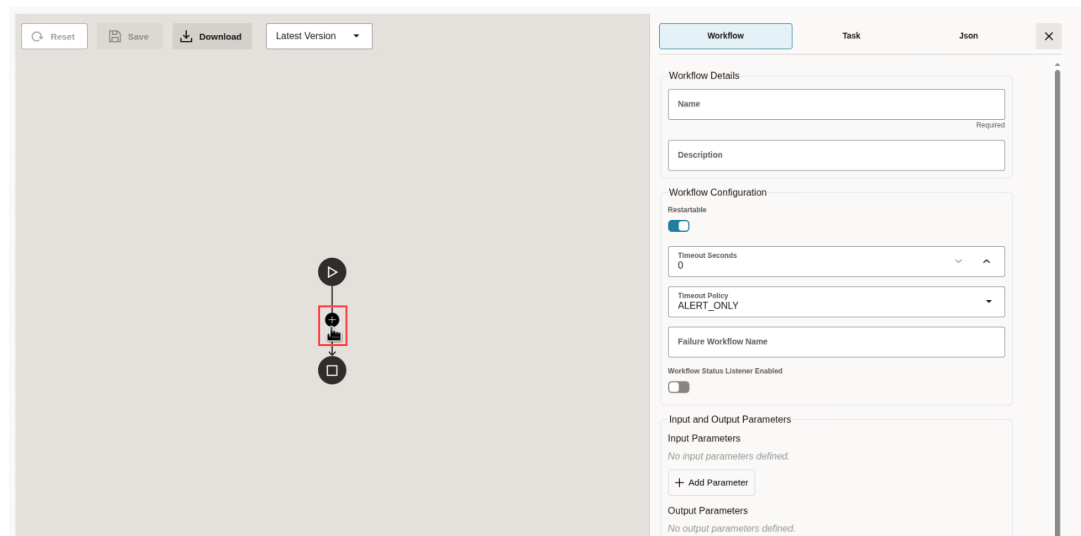
The **Workflows** list page opens and displays the workflow definitions that have been created.

- Click **New Workflow**.

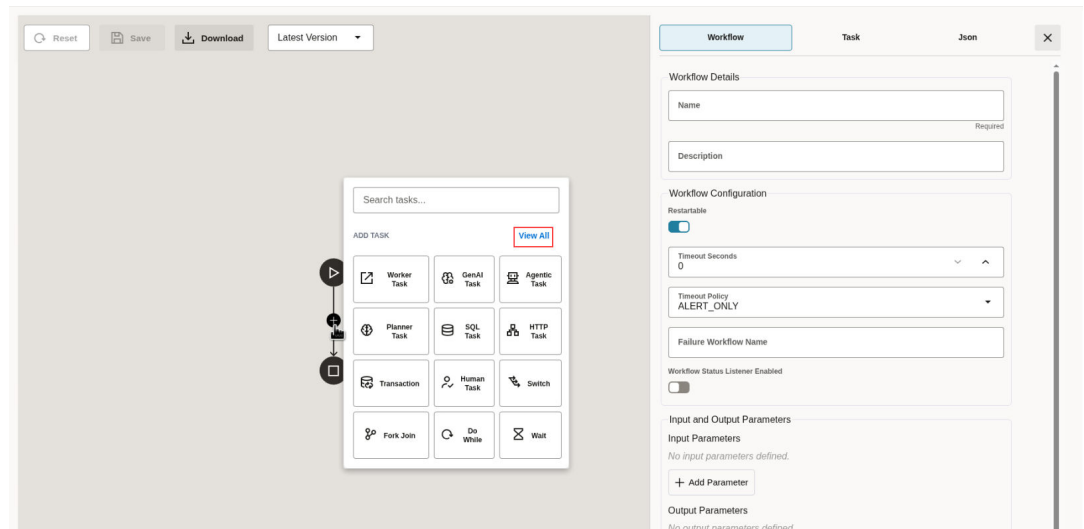
The Workflow Builder is displayed. On the right pane, the **Workflow** tab displays all the details of the workflow as shown in the following figure. On the left pane, the components of the workflow are depicted visually.



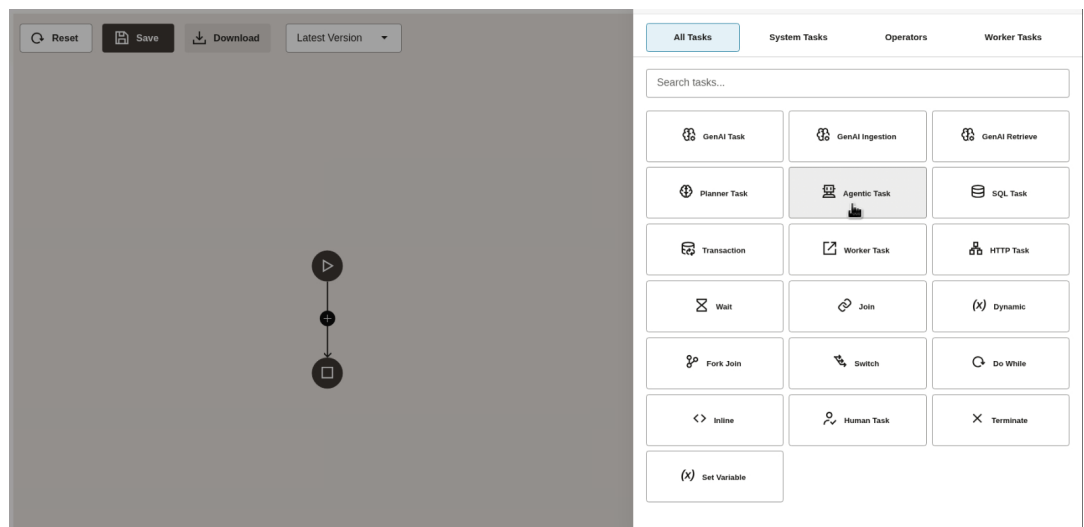
4. Place the cursor on the arrow mark. A plus sign appears.



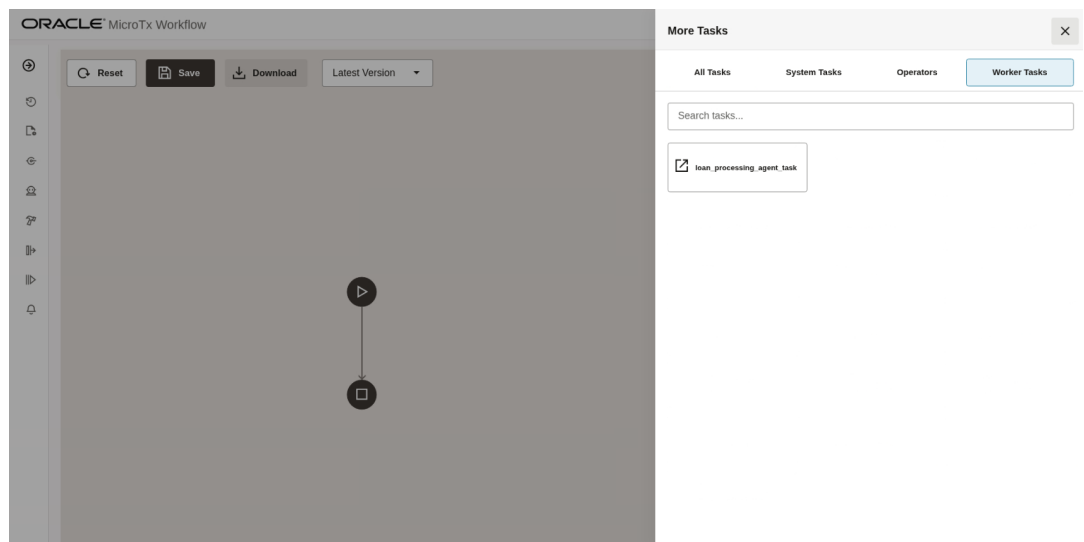
5. Click the plus icon. Many tasks are displayed.



6. Click **View All** to view all the tasks that you can add using the Workflow Builder.

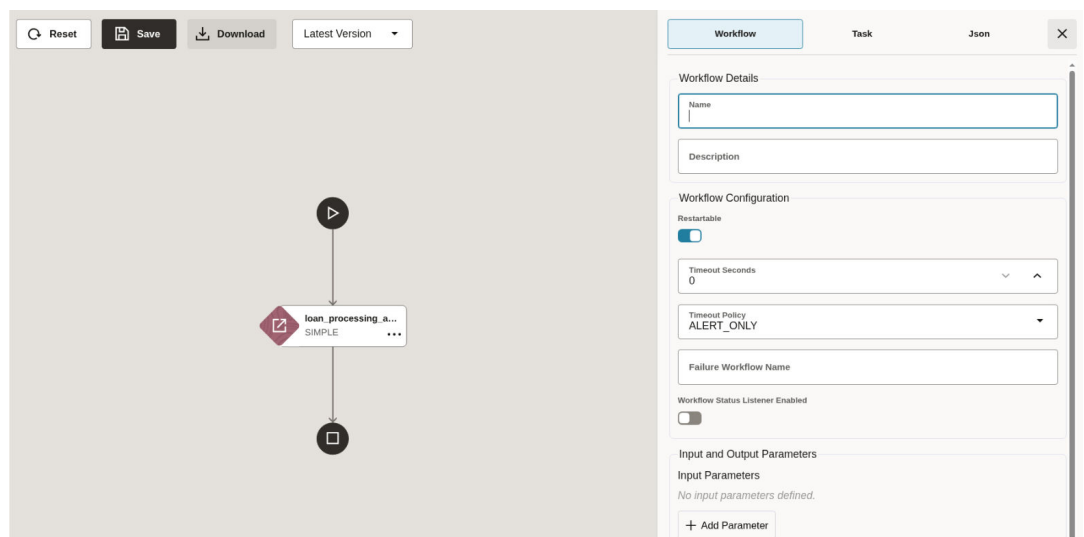


7. You can search for a task using the search pane or navigate through the tabs to identify a task that you want to add to the workflow. Click **Worker Tasks** to view simple tasks that you have defined.
- As shown in the following image, the simple task that you have defined earlier is displayed.



8. Click the task name in the right pane.

MicroTx Workflows adds the task to the workflow as shown in the following image.



9. Based on your business logic, add more tasks and provide details about the workflow.
10. Click **Save**. MicroTx Workflows displays the changes in JSON code. Review all the changes.
11. Click **Confirm Save** to save the changes.

3.3.4 Configure Timeout for Tasks

You can configure timeouts for worker tasks by specifying timeout options in the task definition. These timeout settings apply to custom worker tasks that are registered as task definitions.

They do not apply to MicroTx Workflows system tasks and operators, which are executed by MicroTx Workflows and may expose their own task-specific timeout or wait parameters.

Task definitions are used to register worker task types and define reusable defaults such as retries, timeout settings, input keys, and output keys. See [Create a Task Definition](#).

When you create or update a task definition, configure one or more of the following timeout options:

- **Poll timeout:** The maximum time, in seconds, that a task can remain in the `SCHEDULED` state without being picked up by a worker. If no worker polls the task within this time, the task is marked as `TIMED_OUT`. Use this option to detect worker capacity issues or a backlogged task queue. If set to 0, poll timeout is disabled.
- **Response timeout:** The maximum time, in seconds, that a worker can hold a task without sending an update. If the worker polls the task but does not update the task status within this time, the task is marked as `TIMED_OUT`. Use this option to detect crashed workers, network failures, or workers that stopped responding. If the task needs more time, the worker should send an `IN_PROGRESS` update before the response timeout expires.
- **Timeout:** The maximum time, in seconds, that the task can take to reach a terminal state after it starts execution. If the task does not complete within this time, it is marked as `TIMED_OUT`. Use this option to enforce the overall service-level agreement for a task. If set to 0, task timeout is disabled.
- **Timeout policy:** Defines what MicroTx Workflows does when the task times out.
 - **RETRY:** Retries the task if retry attempts are still available.
 - **TIME_OUT_WF:** Marks the workflow as `TIMED_OUT` and terminates the workflow. This is the default timeout policy.
 - **ALERT_ONLY:** Records the timeout, but does not change the task or workflow execution state.

Ensure that timeout is greater than response timeout. If timeout is configured, MicroTx Workflows validates that the response timeout value is less than the timeout value.

Example task definition timeout configuration

```
{
  "name": "process_payment",
  "retryCount": 3,
  "retryLogic": "FIXED",
  "retryDelaySeconds": 10,
  "pollTimeoutSeconds": 60,
  "responseTimeoutSeconds": 120,
  "timeoutSeconds": 600,
  "timeoutPolicy": "RETRY"
}
```

3.3.5 Edit a Task Definition

You can update all the field values you specified when creating a task definition, except for the name.

1. Open the navigation menu and click **Definitions**.
2. Click the **Tasks** tab.
The **Tasks** list page opens. All the tasks that you have defined are displayed in a table.
3. Identify the task that you want to update.
4. Click  (**Edit**).

5. You can update all the parameter values you specified while creating a task definition, except for the name. For information about the required parameter values for a task, see [Create System Tasks and Operator Tasks](#).
 6. Click **Submit**.
 7. A confirmation message appears that the task has been edited successfully. Click **OK**.
- MicroTx Workflows automatically picks up the updated workflow definition.

3.3.6 View a Task Definition

You can retrieve and view the details of a task definition.

1. Open the navigation menu and click **Definitions**.
2. Click the **Tasks** tab.

The **Tasks** list page opens. All the tasks that you have defined are displayed in a table.
3. Identify the task definition that you want to view.
4. Click ⓘ (**Show Details**).
5. Click **Close** after viewing the details.

3.3.7 View Task Queues

Use the Task Queues page to monitor worker-pollled task queues, such as simple tasks or worker tasks. This page helps you check whether workers are polling for a selected task and whether any task executions are waiting in the queue.

A task queue is created for a task type when task executions are scheduled and must be picked up by workers. Workers poll the queue for a task type, execute the task, and update the task result. If tasks are added faster than workers can poll and complete them, the queue size increases.

You can use this page to:

- Verify whether workers are actively polling a task queue.
- Identify the last worker that polled for a task.
- Check whether pending task executions are accumulating.
- Compare queue backlog across domains, when domains are used.
- Troubleshoot worker availability or capacity issues when queue size remains high or last poll time is stale.

For example, if the queue size for a task is increasing and the last poll time is old, workers may not be running, may not be polling the correct task name or domain, or may not have enough capacity to process the queued work.

To view task queue details:

1. Open the navigation menu and click **Task Queues**.
2. In the **Select a Task Name** drop-down list, select the task whose queue details you want to view. The drop-down list displays task definition names.

The following details are displayed for the selected task.

- **Domain:** The domain in which workers poll for the selected task. If no domain is configured, **(Domain not set)** is displayed.
- **Last Polled Worker:** The worker ID of the most recent worker that polled this task queue for the displayed domain. This is not the task name. It is the worker identifier sent by the worker while polling.
- **Last Poll Time:** The time when the task queue was last polled by a worker for the displayed domain. The time is displayed in the browser's local time zone in the yyyy-MM-dd HH:mm:ss format, followed by a relative time value. For example, 2026-11-30 17:40:00 (less than a minute ago).
- **Queue Size:** The number of task executions currently waiting in the queue for the selected task and domain. A value of 0 means there are no pending executions waiting to be polled for that task queue.

3.3.8 Delete a Task Definition

You can delete a task definition when it is no longer required.

Before you delete a task definition, ensure that it is not being used by any workflow. The workflow execution fails if it can't find a specified task.

1. Open the navigation menu and click **Definitions**.
2. Click the **Tasks** tab.

The **Tasks** list page opens. All the tasks that you have defined are displayed in a table.

3. Identify the task definition that you want to delete.
4. Click  (**Delete**).
5. When prompted, confirm the deletion.

3.4 Manage Executed Tasks

MicroTx Workflows stores details of all the tasks that are executed as part of a workflow execution.

You can view the details of these tasks and take appropriate action on a task based on its status.

Topics

- [Search and View Details of Executed Tasks](#)
You can search for and view information about tasks associated with executed workflows.
- [Perform an Action on Multiple Tasks](#)
You can select multiple tasks and perform an action on all the selected tasks at the same time.

3.4.1 Search and View Details of Executed Tasks

You can search for and view information about tasks associated with executed workflows.

1. Open the navigation menu and click **Executions**.
2. Click the **Tasks** tab.

Details of all tasks associated with executed workflows are displayed.

3. Use one or more of the following options to filter the displayed tasks.
 - **Task Name:** Select or enter the name of a simple task or worker task to search.
 - **Task Type:** Select the type of task to search. Leave this blank for Simple tasks or Worker tasks.
 - **Workflow Name:** Select a workflow for which you want to view the tasks. A workflow can contain many tasks.
 - **Task ID:** Enter the task ID to view details of a specific task.
 - **Status:** Filter tasks by their current status, such as in progress, completed, or scheduled.
 - **Update Time From:** Select a date and time to narrow down searches based on the duration from when tasks were updated.
 - **Update Time To:** Select a date and time to narrow down searches based on the duration till when tasks were updated.
4. Click **Search** to apply your search criteria to display matching tasks.
5. Optional. You can tailor the view to display the most relevant information for your requirements. Click **Columns**, select the column details that you want to view and clear those that you want to hide.
 - **Update Time:** The most recent date and time when the task status or details were updated.
 - **Task ID:** The identifier assigned to each instance of a task within a workflow.
 - **Task Name:** The name of the task as defined in the workflow configuration.
 - **Task Type:** The type of task, such as Agentic task, GenAI task.
 - **Workflow Name:** The name of the workflow to which the task belongs.
 - **Status:** Current status of the task, such as in progress, completed, and failed.
 - **Scheduled Time:** Date and time at which the task was scheduled to begin execution.
 - **Start Time:** Timestamp marking when the task execution started.
 - **End Time:** Timestamp marking when the task was completed or terminated.
 - **Workflow ID:** The unique identifier of the workflow execution that contains this task. Workflow execution ID is specific to the version of a workflow that is triggered.
 - **Execution Time:** The total time taken to execute a task.
 - **Queue Wait Time:** The amount of time a task spent waiting in queue before execution began.
 - **Workflow Priority:** The execution priority of the workflow containing this task.
 - **Input:** The input data or payload provided to the task at the time of execution.
 - **Output:** The output data or result generated by the task upon completion.
 - **Reason for Incompletion:** An explanation or error message describing why the task did not complete successfully.

The search results are immediately updated to display information of the selected columns for each task. To reset the selected columns to default options, click **Reset to default**. Click **Close** to exit the Column list.

You can continue to filter the search results or click **Refresh** to reload results without any search criteria.

3.4.2 Perform an Action on Multiple Tasks

You can select multiple tasks and perform an action on all the selected tasks at the same time.

MicroTx Workflows performs actions on the selected tasks based on the status of the workflow that contains the task. For information about the actions that you can perform on workflows based on their status, see [About Workflow Status](#).

1. Open the navigation menu and click **Executions**.
2. Click the **Tasks** tab.

Details of all tasks associated with executed workflows are displayed.

3. Optional. Apply search criteria to filter the search results and view the executed workflows on which you want to perform an action. See [Search and View Details of Executed Tasks](#).
4. Select one or more tasks.
5. Click **Bulk Actions**, and then click one of the following actions. The
 - **Pause**: Temporarily halts the execution of the workflows that contain the selected tasks. Workflows will stop processing at their current step until you resume the workflow execution.
 - **Resume**: Restarts any paused workflows that contain the selected tasks, allowing them to continue processing from the point they were paused.
 - **Terminate**: Immediately stops the execution of workflows that contain the selected tasks. You can only terminate workflows which are in the running or paused state. You cannot resume terminated workflows, but you may retry or restart the workflow.
 - **Terminate With Reason**: Immediately stops the execution of workflows that contain the selected tasks and prompts you to enter a reason for the terminating the tasks. The reason you specify is logged for auditing and troubleshooting. You can only terminate tasks which are in the running or paused state.
 - **Delete**: Permanently removes the workflows that contain the selected tasks. Only workflows in one of the following states can be deleted: completed, terminated, failed, or timed out. An error message is displayed for workflows in other states.

Caution

You cannot recover deleted workflows.

- **Archive**: Moves workflows that contain the selected tasks to an archived state, removing them from active lists while retaining them for historical reference or compliance.

Workflows in only one of the following states can be archived: completed, terminated, failed, or timed out. An error message is displayed for workflows in other states.

Tasks in one of the following states are archived: completed, terminated, failed, or timed out. An error message is displayed for tasks in other states.
- **Retry**: Reattempts to process the selected tasks, which are in the failed, terminated, or timed out state in multiple workflows. The reattempt starts from the point of failure.
- **Restart with current def**: Restarts the workflows that contain the selected tasks. MicroTx Workflows uses the same workflow definition that was in effect at the time they were originally executed.

You can restart workflows only when they are in one of the following states: completed, terminated, failed, or timed out. An error message is displayed for workflows in other states.

- **Restart with latest def:** Restarts the workflows that contain the selected tasks. MicroTx Workflows picks up any changes or updates made to the workflow since an earlier execution, and then restarts the workflow.

You can restart workflows only when they are in one of the following states: completed, terminated, failed, or timed out. An error message is displayed for workflows in other states.

MicroTx Workflows performs actions on the selected tasks based on the status of the workflow that contains the task. If an action cannot be performed on a selected task, the reason is displayed as shown in the following figure.

Bulk Action pause Results	
Errors	
Workflow ID	Reason
c1c386f7-b4e3-4f6e-987f-52b1ce5221d3	Workflow pytest_fw_create_tables_parallel.1/has ended, status cannot be updated.
e4ceb8c4-af3e-44fb-979d-c668556183b7	Workflow pytest_fw_execute_plsql.1/has ended, status cannot be updated.

6. Click **OK**.

4

Connectors

Connectors provide a centralized way to define and manage integration details for external systems used by MicroTx Workflows. A connector can capture connection information such as endpoint URLs, authentication details, credentials, and other configuration parameters required to access an external service.

By defining connectors once, you can reference the same integration from multiple task definitions and AI agent profiles. This avoids duplicating connection details or credentials across workflows and helps keep integrations consistent, easier to maintain, and easier to update.

Credentials stored in connectors are protected through the MicroTx Workflows secret management mechanism. Sensitive values are stored as encrypted secrets using the encryption keys configured during installation, and are masked when connector details are retrieved through the APIs.

What this means for you as a developer:

- **Centralized integration management:** Manage external service connection details and credentials in one place.
- **Credential reuse without duplication:** Reference configured connectors from workflows, tasks, and AI agent profiles without embedding credentials in each definition.
- **Improved security:** Sensitive connector fields are encrypted at rest and masked in API responses.
- **Simpler workflow development:** Workflows can use existing connector definitions instead of requiring each task to carry its own integration configuration.
- **Easier maintenance:** If an endpoint, credential, or integration parameter changes, update the connector once instead of updating every workflow or task that uses it.

Let's understand this with an example. Suppose a workflow needs to call an AI model or access a database. Instead of placing the service URL, credentials, and related configuration directly in the workflow task, you can define the connection as a connector profile. The workflow or AI agent profile references that connector by name, and MicroTx Workflows uses the stored configuration to perform the integration.

Topics

- [Manage LLM Connector](#)
Define and manage Large Language Model (LLM) provider configurations used by MicroTx Workflows for AI-related workflow and agent operations.
- [Manage Database Profiles](#)
Define and manage database connection details in one place. Use database profiles to access data stores across workflows, tasks, and agent-driven processes.
- [Manage Internal Tool Configurations](#)
In addition to tools integrated through MCP servers, tool configuration provides a centralized way for you to register and manage internal MicroTx Workflows tools. Define tool settings once, and reference them across AI-driven workflow features.

- [Manage MCP Server Details](#)
Define and manage Model Context Protocol (MCP) server details so workflows and AI agents can integrate with external tools and data sources.
- [Manage File Storage](#)
File storage in MicroTx Workflows provides a centralized way to define and manage local shared storage for files accessed during workflow and agent execution.
- [Manage Cloud Profiles](#)
Cloud profiles in MicroTx Workflows provide a centralized way to define and manage connections to Oracle Cloud Infrastructure (OCI) services, such as OCI Object Storage.
- [Manage SFTP Profiles](#)
Define and manage remote SFTP server connection details. Use SFTP profiles to access remote SFTP servers across workflows and tasks.

4.1 Manage LLM Connector

Define and manage Large Language Model (LLM) provider configurations used by MicroTx Workflows for AI-related workflow and agent operations.

An LLM connector stores the common configuration required to access an LLM provider, including provider type, API connection details, and supported model names. You can define the connector once and reference it from workflow tasks, AI agent profiles, and other AI-driven workflow features instead of repeating provider configuration in each place. LLM connectors support both chat or completion model providers and embedding model providers. Supported provider types include:

- OCI
- Ollama
- OpenAI
- OpenAI-compatible providers
- Oracle Database

For example, you can create an LLM connector for an OpenAI-compatible endpoint, an OCI Generative AI model, an Ollama-hosted model, or an Oracle Database embedding model. After the connector is created, workflows and agents can reference it by name when they need to perform AI operations such as generation, reasoning, tool use, or embedding generation.

- [Create an LLM Definition](#)
Create an LLM connector to define the model provider, model names, and connection details used by the AI features for MicroTx Workflows. You can reference the connector from workflow tasks, AI agent profiles, and other AI-driven workflow configurations.
- [Update an LLM Definition](#)
- [Test an LLM Definition](#)
- [View an LLM Definition](#)
- [Delete an LLM Definition](#)

4.1.1 Create an LLM Definition

Create an LLM connector to define the model provider, model names, and connection details used by the AI features for MicroTx Workflows. You can reference the connector from workflow tasks, AI agent profiles, and other AI-driven workflow configurations.

Prerequisites

Before creating an LLM connector, complete the setup required for the model provider that you want to use.

To use OpenAI as the model provider, complete the following prerequisites:

1. Create a new API key in the [API Keys page](#) of the OpenAI Developer Platform or use the [OpenAI API](#).
2. Copy the name and value of the created key and save it safely. You will need to provide this information later.
3. Identify the model names that you want to use.
4. Use `https://api.openai.com/` as the base URL unless your environment requires a different endpoint.

To use OCI as the model provider, complete the following tasks:

- Create or identify the OCI user that will call OCI Generative AI APIs.
- Generate an OCI API signing key pair in PEM format. See [Required Keys and OCIDs](#).
- Upload the public key to the OCI user and note the public key value and fingerprint.
- Keep the private key value available; you will paste it into the LLM connector as the API key.
- Note down the OCID of the user, the tenancy's OCID, OCI region, and the compartment OCID.
- Confirm the serving mode for the model. You can reach the pretrained foundational models in Generative AI through two modes: on-demand and dedicated.

To use Ollama as the model provider, complete the following tasks:

- Ensure the Ollama endpoint is reachable from MicroTx Workflows.
- Identify the model names that you want to use.
- Copy the base URL for the Ollama service. For example `http://localhost:11434` if it is running locally.
- Create and copy the value of an API key if your Ollama endpoint requires authentication.

To use Oracle Database, create a database profile in MicroTx Workflows. Ensure the database has the embedding model available. See [Create a Database Profile](#). Use this provider when you want Oracle Database to generate embeddings. Oracle Database embedding support uses vector embedding models loaded into the database, such as ONNX-format embedding models. You can use this connector in GENAI_INGESTION task. For information about using Oracle Database as embedding model provider, see <https://blogs.oracle.com/machinelearning/use-our-prebuilt-onnx-model-now-available-for-embedding-generation-in-oracle-database-23ai>.

To create an LLM definition:

1. Open the navigation menu and click **Connectors**.
2. Click **LLM**.

The **LLM Definitions** list page opens. All the LLMs that you have defined are displayed in a table.

3. Click .

The **New LLM Definition** dialog box appears.

4. Enter the following information.

- **Name:** Enter a unique and descriptive name to identify this LLM definition in workflows. The name can be up to 128-characters long. Use only letters, numbers, underscores (_), and hyphens (-). Spaces and other special characters are not supported.
- **Model Provider:** Select one of the following models: OCI, OpenAI, Ollama, Oracle Database.
- **Models:** List the model names you intend to use for the selected model provider, separated by commas. For example, you may want to enter gpt-4o, gpt-4o-mini if you select OpenAI as the model provider. Refer to documentation specific to the selected model provider for the list of supported models.
- **Description:** Optional. Enter a description for the LLM definition.

5. Enter the following additional configuration information based on the selected model provider.

- If you select **OCI** as the **Model Provider**, provide the following information to access the selected LLM model which is hosted on OCI.
 - **API Key:** Paste the value of the private RSA key in PEM format.
 - **User ID:** Enter the OCID of the OCI user whose API signing key is used.
 - **Tenant ID:** Enter the tenancy's OCID where the models are hosted.
 - **Region:** Enter the OCI region. For example, us-phoenix-1.
 - **Fingerprint:** Enter the fingerprint of the uploaded public key.
 - **Pass Phrase:** Optional. Enter the pass phrase, if you had provided one, while creating the private key.
 - **Compartment ID:** Specify the OCID of the compartment that contains or has access to the model.
 - **Serving Mode:** Optional. Specify the serving mode for the model, according to your deployment. The default value is on demand. See [On-Demand and Dedicated Modes for OCI Generative AI Models](#) in *Oracle Cloud Infrastructure Documentation*.
- If you select **OpenAI** as the model provider, additionally provide the following configuration information.
 - **API Key:** Paste your OpenAI API key, which authenticates your requests.
 - **Base URL:** Specify a custom base URL to route requests to the OpenAI API endpoint. For example, `https://api.openai.com/`.
- If you select **Ollama** as the model provider, additionally provide the following configuration information.
 - **API Key:** Enter the API key if your Ollama endpoint requires one.
 - **Base URL:** Enter the Ollama service endpoint.

- If you select **ORACLEDB** as the model provider, you must select a database profile that contains the connection details for the Oracle Database used to generate embeddings.

Note

Oracle Database is used as an embedding model provider. It does not function as a general-purpose chat or text-generation LLM in this connector flow.

6. Click **Test** to ensure that you can establish a connection with the provided details.

If you have entered multiple models, test is performed only for the first model that is specified in the comma-separated list.

If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue, and then click **Close**.

7. Click **Test** to ensure that you can establish a connection with the provided details.

If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue, and then click **Close**.

8. Click **Submit**.

Your new definition appears in the list of available LLMs. Reference this name in your workflows or tasks to utilize the specified model and provider.

4.1.2 Update an LLM Definition

You can update all the field values you specified when creating an LLM definition, except for the name.

1. Open the navigation menu and click **Connectors**.

2. Click **LLM**.

The **LLM Definitions** list page opens. All the LLMs that you have defined are displayed in a table.

3. Identify the LLM definition that you want to update.

4. Click  (**Edit**).

5. You can update all the field values you specified when creating an LLM definition, except for the name.

6. Click **Test** to ensure that you can establish a connection with the provided details.

If you have entered multiple models, test is performed only for the first model that is specified in the comma-separated list.

If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue, and then click **Close**.

7. Click **Submit**.

4.1.3 Test an LLM Definition

You can test an LLM definition after creating or editing it.

To test an LLM definition:

1. Open the navigation menu and click **Connectors**.
2. Click **LLM**.

The **LLM Definitions** list page opens. All the LLMs that you have defined are displayed in a table.

3. Identify the LLM definition that you want to test.
4. Click  **(Test)**.

A message is displayed that the test was successful.

If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue.

5. Click **Test** to ensure that you can establish a connection with the provided details.

If you have entered multiple models, test is performed only for the first model that is specified in the comma-separated list.

If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue.

6. Click **OK**.

4.1.4 View an LLM Definition

You can retrieve and view the details of an LLM definition.

To view details of an LLM definition:

1. Open the navigation menu and click **Connectors**.
2. Click **LLM**.

The **LLM Definitions** list page opens. All the LLMs that you have defined are displayed in a table.

3. Identify the LLM definition that you want to view.
4. Click  **(Show Details)**.

5. Click **Test** to ensure that you can establish a connection with the provided details.

If you have entered multiple models, test is performed only for the first model that is specified in the comma-separated list.

If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue, and then click **Close**.

6. Click **Close** when you finish viewing the details.

4.1.5 Delete an LLM Definition

You can delete an LLM definition when it is no longer required.

Before you delete an LLM definition, ensure that it is not being used by any workflow. The workflow execution fails if it can't find a specified LLM definition.

1. Open the navigation menu and click **Connectors**.
2. Click **LLM**.

The **LLM Definitions** list page opens. All the LLMs that you have defined are displayed in a table.

3. Identify the LLM definition that you want to delete.
4. Click  (**Delete**).
5. When prompted, confirm the deletion.

4.2 Manage Database Profiles

Define and manage database connection details in one place. Use database profiles to access data stores across workflows, tasks, and agent-driven processes.

Each database connector specifies essential details such as the database type, connection credentials, and advanced configuration settings, such as connection pooling and secure authentication.

Here are some of the advantages that this feature offers to you as a developer:

- **Centralized management:** All database credentials are maintained in one place, making it easier to update, troubleshoot, and audit.
- **Reusability:** Database profiles can be referenced across multiple workflows, tasks, and agents, reducing duplication of configuration and ensuring consistent access patterns.
- **Security:** Centralized handling of credentials and wallet files strengthens security, enabling best practices for credential storage and retrieval.
- **Flexibility:** Database profiles support different database types, such as relational and vector, giving you the freedom to use the right data store for each use case without complexity.

Topics

- [Create a Database Profile](#)
Create a database profile to define connection parameters, such as the type of database engine, supported capabilities, connection URL, user credentials, and secure wallet options.
- [Update a Database Profile](#)
You can update all the field values you specified when creating a database profile, except for the name.
- [Test a Database Profile](#)
After creating or updating a database profile, you can test it to ensure that you can establish a connection with the provided details.
- [View a Database Profile](#)
You can retrieve and view the details of a database profile.
- [Delete a Database Profile](#)
You can delete a database profile when it is no longer required.

4.2.1 Create a Database Profile

Create a database profile to define connection parameters, such as the type of database engine, supported capabilities, connection URL, user credentials, and secure wallet options.

Prerequisites

To create a database profile that you can use later to generate vector embeddings, you must select embedding generation as a capability. Ensure that an Oracle Database is already configured with an embedding generation LLM model installed in the database. For information about importing and configuring embedding models in Oracle Database, see [Generate Vector Embeddings](#) in *Oracle AI Vector Search User's Guide*. Later, you can select this database profile while creating a new LLM Definition with **OracleDB** selected as the LLM model provider.

1. Open the navigation menu and click **Connectors**.

2. Click **Database**.

The **Database** profile list page opens. All the database profiles that you have defined are displayed in a table.

3. Click .

The **New Database Profile** dialog box appears.

4. Enter the following information.

- **Name:** Enter a unique, descriptive name for your database profile. The name can be up to 128-characters long. Use only letters, numbers, underscores (_), and hyphens (-). Spaces and other special characters are not supported. This helps distinguish it from other profiles, especially if you manage multiple database connections.
- **Description:** Optional. Add details describing the purpose or scope of this database profile, such as which application, workflow, or environment it serves.
- **Engine:** Select the database engine that you want to use. Select **ORACLE** to use Oracle Database or **POSTGRES** to use PostgreSQL database.
- **Username:** Enter the username required to authenticate and connect to the database.
- **Password:** Enter the password for the specified database user name. Click the eye icon to show/hide the value if needed. Ensure you use strong credentials and handle them securely.
- **URL:** Provide the full JDBC connection URL to the database. This typically includes the protocol, hostname, port, and database service identifier. For example, `jdbc:oracle:thin:@host:port/service` for an Oracle Database.
- **Max Connection Pool Size:** Configure the maximum number of parallel database connections that MicroTx Workflows can create using this database profile.

5. Enter the following additional configuration information if you select **ORACLE** as the Database Engine.

- **Capabilities:** Specifies the supported features for the database profile. Select one or more of the following values based on your use case.
 - **RELATIONAL:** Indicates support for traditional relational database operations, such as SQL queries and transactions.
 - **VECTOR:** Indicates the database profile can be used for vector-based operations, such as similarity searches or AI/ML vector storage and retrieval.
 - **EMBEDDING_GENERATION:** Indicates that the database profile is used to generate vector embeddings.
- **Wallet File (optional):** Search and upload the Oracle Wallet file if your database connection requires it. The wallet file contains encrypted credentials and security details for secure authentication.

- **Wallet Password:** Enter the wallet password if you are using an Oracle Wallet for secure credential storage. Click the eye icon to show/hide the value if needed. This is used for decrypting wallet files and securing connections.
6. Click **Test** to ensure that you can establish a connection with the provided details.
If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue, and then click **Close**.
 7. Click **Submit** to create the new database profile and save your configuration.
- You can use the new database profile for secure connections within workflows, tasks, and agents.

4.2.2 Update a Database Profile

You can update all the field values you specified when creating a database profile, except for the name.

1. Open the navigation menu and click **Connectors**.
2. Click **Database**.
The **Database** profile list page opens. All the database profiles that you have defined are displayed in a table.
3. Identify the database profile that you want to update.
4. Click  (**Edit**).
5. You can update all the field values you specified when creating a database profile, except the name. Database user passwords and wallet passwords cannot be viewed, but they can be updated. Be careful while modifying password fields.
6. Click **Test** to ensure that you can establish a connection with the provided details.
If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue, and then click **Close**.
7. Click **Submit**.

4.2.3 Test a Database Profile

After creating or updating a database profile, you can test it to ensure that you can establish a connection with the provided details.

To test a database profile:

1. Open the navigation menu and click **Connectors**.
2. Click **Database**.
The **Database** profile list page opens. All the database profiles that you have defined are displayed in a table.
3. Identify the database profile that you want to test.
4. Click  (**Test**).
A message is displayed that the test was successful.
If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue.
5. Click **OK**.

4.2.4 View a Database Profile

You can retrieve and view the details of a database profile.

To view details of a database profile:

1. Open the navigation menu and click **Connectors**.
2. Click **Database**.

The **Database** profile list page opens. All the database profiles that you have defined are displayed in a table.

3. Identify the database profile that you want to view.
4. Click  (**Show Details**).
5. Click **Test** to ensure that you can establish a connection with the provided details.

If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue, and then click **Close**.

6. Click **Close** when you finish viewing the details.

4.2.5 Delete a Database Profile

You can delete a database profile when it is no longer required.

Before you delete a database profile, ensure that it is not being used by any workflow. The workflow execution fails if it can't find a specified database profile.

1. Open the navigation menu and click **Connectors**.
2. Click **Database**.

The **Database** profile list page opens. All the database profiles that you have defined are displayed in a table.

3. Identify the database profile that you want to delete.
4. Click  (**Delete**).
5. When prompted, confirm the deletion.

4.3 Manage Internal Tool Configurations

In addition to tools integrated through MCP servers, tool configuration provides a centralized way for you to register and manage internal MicroTx Workflows tools. Define tool settings once, and reference them across AI-driven workflow features.

Supported internal tool categories

- Database
- Date time
- Calculator
- File
- Workflow
- Agent

- RAG search which supports naive and advanced as sub types
- API which supports only HTTP as the sub type

With internal tools configuration, you can:

- Register tool connection and execution settings for each tool, such as URL, API key, database profile, RAG settings, allowed workflows.
- Keep integration settings centralized and auditable.
- Improve consistency, auditability, and secure handling of sensitive fields (such as API keys).
- Create multiple configurations for the same tool category, enabling environment- or use-case-specific behavior without changing the agent or workflow logic.
- Reuse tool definitions across multiple agents or workflows.

After creating an internal tool configuration, you can reference it in agentic planner tasks and in agent profiles for agentic task and conversational agent. For this, you must first create or configure the tool with required parameters. Then ensure that the workflow or agent definitions reference the tool by its unique tool configuration name.

Topics

- [Create an Internal Tool Configuration](#)
Create an internal tool configuration to register an internal MicroTx Workflows tool, so that you can reuse the tool in agentic workflows and AI agent profiles.
- [Update an Internal Tool Configuration](#)
You can update all the field values you specified when creating an internal tool configuration, except for the name.
- [View an Internal Tool Configuration](#)
You can retrieve and view the details of an internal tool configuration.
- [Delete an Internal Tool Configuration](#)
You can delete an internal tool configuration when it is no longer required.

4.3.1 Create an Internal Tool Configuration

Create an internal tool configuration to register an internal MicroTx Workflows tool, so that you can reuse the tool in agentic workflows and AI agent profiles.

Each tool configuration stores the runtime settings needed to invoke a tool, such as the tool's endpoint URL, credentials, profile references, or tool-specific options.

Prerequisites

Before you begin creating a tool configuration, ensure the required dependencies for that category are available.

- For Database tool, ensure that either a database profile with the required details already exists or create a new one. You will need to provide the name of the database profile while creating a database tool. See [Create a Database Profile](#).
- For Workflow tool, identify the list of allowed workflow names.
- For Agent tool, define the target agent execution settings in an Agent profile. See [Create an Agent Profile](#).
- For RAG search tool, prepare the required RAG retrieval settings.

- For API (HTTP) tool, ensure that the endpoint URL and required authentication details are available. For example, API key.

To create a tool configuration:

1. Open the navigation menu and click **Connectors**.
2. Click **Internal Tools**.

The **Internal Tools** list page opens. All the internal tool configurations that you have defined are displayed in a table.

3. Click .

The **New Tools Config** dialog box appears.

4. In the **Name** box, enter a unique and descriptive name to identify this tool configuration in workflows and tasks. The name can be up to 128-characters long. Use only letters, numbers, underscores (`_`), and hyphens (`-`). Spaces and other special characters are not supported.
5. Optional. In the **Description** box, provide a brief summary about the intended use of the tool configuration or any relevant details that will help others understand its purpose.
6. In the **Category** drop-down list, select one of the following options to specify an appropriate category for the tool.
 - **API**: Select this option when an agent must call external HTTP endpoints.
 - **Type**: Select **HTTP**.
 - **URL**: Optional. Enter the endpoint or base URL required to access this tool. This is typically the address used for API calls or service requests.
 - **API Key**: Optional. If the tool requires authorization using an API key, provide the bearer token value. MicroTx Workflows encrypts and stores the provided API key. Ensure that you follow your organization's security guidelines for handling and storing sensitive credentials.
 - **Database**: Select this option when agent must run database operations. In the **Database Profile** drop-down list, select a database profile that you have created earlier to securely provide the database connection details.
 - **Agent**: Select this option when one agent or tool should invoke another agent profile. This is mainly used by conversational agents to invoke sub-agents as part of agent chat. In the **Agent Profile** drop-down list, select the target agent profile that you have created earlier. In the **Input Data (JSON)** field, provide the input parameters if the target agent requires any.
 - **Workflow**: Select this option when the tool should start the selected workflows. This is mainly used by conversational agents to invoke workflows through the tool. In the **Workflow Name** field, select the names of the workflows.
 - **DateTime**
 - **Calculator**
 - **File**: When you select this option, files are selected from the local file storage of MicroTx Workflows. The default base path of the local file storage is the value specified in the `conductor.connector.storage-path` property in the `application.properties` file.
 - **RAG search**: Select this option when agent must retrieve knowledge from vector or RAG store. The information for the various parameters is the same as the GENAI Retrieve Parameters. See [GenAI Retrieve Task](#).

7. Click **Submit**.

Your new tool configuration appears in the list of available tools configurations. You can reference this tool by its tool configuration name in agentic planner tasks and in agent profiles for agentic task and conversational agent.

4.3.2 Update an Internal Tool Configuration

You can update all the field values you specified when creating an internal tool configuration, except for the name.

1. Open the navigation menu and click **Connectors**.
2. Click **Internal Tools**.

The **Internal Tools** list page opens. All the internal tool configurations that you have defined are displayed in a table.

3. Identify the internal tool configuration that you want to update.
4. Click  (**Edit**).
5. You can update all the field values you specified when creating an internal tool configuration, except for the name.
6. Click **Submit**.

4.3.3 View an Internal Tool Configuration

You can retrieve and view the details of an internal tool configuration.

To view details of an internal tool configuration:

1. Open the navigation menu and click **Connectors**.
2. Click **Internal Tools**.

The **Internal Tools** list page opens. All the internal tool configurations that you have defined are displayed in a table.

3. Identify the internal tool configuration that you want to view.
4. Click  (**Show Details**).
5. Click **Close** when you finish viewing the details.

4.3.4 Delete an Internal Tool Configuration

You can delete an internal tool configuration when it is no longer required.

Before you delete an internal tool configuration, ensure that it is not used by any Agent Profiles or in Agentic Planner tasks in a workflow. The workflow execution fails if it can't find a specified configuration.

1. Open the navigation menu and click **Connectors**.
2. Click **Internal Tools**.

The **Internal Tools** list page opens. All the internal tool configurations that you have defined are displayed in a table.

3. Identify the internal tools configuration that you want to delete.
4. Click  (**Delete**).

5. When prompted, confirm the deletion.

4.4 Manage MCP Server Details

Define and manage Model Context Protocol (MCP) server details so workflows and AI agents can integrate with external tools and data sources.

To learn more about MCP, see [Model Context Protocol](#).

Topics

- [Provide MCP Server Details](#)
Enter MCP server details to register an external MCP integration in MicroTx Workflows and to use it as a tool during workflow execution.
- [Update an MCP Server Details](#)
You can update all the field values you specified when providing details for an MCP server, except for the name.
- [Test MCP Server Details](#)
After creating or updating MCP server details, you can test it to ensure that you can establish a connection with the provided details.
- [View MCP Server Details](#)
You can retrieve and view the details of an MCP server.
- [Delete MCP Server Details](#)
You can delete MCP server details when it is no longer required.

4.4.1 Provide MCP Server Details

Enter MCP server details to register an external MCP integration in MicroTx Workflows and to use it as a tool during workflow execution.

Later, you can reference the configured MCP servers in agentic planner tasks, agent profiles used by agentic task in workflow execution, and agent profiles used by conversational agent APIs.

At runtime, MicroTx Workflows discovers the tools exposed by the configured MCP servers and makes available for workflow task and agent execution.

Before providing MCP server details, ensure the following points:

- The MCP server endpoint or process command is available and MicroTx Workflows can access it.
- Identify the transport mode and provide the command if you select **STDIO** as the transport mode or provide the URL if you select **SSE**.

Note

Use STDIO transport only when the MCP server can be started as a local process from the MicroTx Workflows server at runtime. In Docker or Kubernetes deployments, the command is executed inside the workflow server container or pod, not on the client machine. Ensure that the executable, runtime dependencies, scripts, environment variables, permissions, and any required files are available in that container. For example, to use an npx-based MCP server, ensure that Node.js, npm, npx and the required MCP package must be installed or packaged in the workflow server image. If the MCP server runs in a separate container, pod, host, or managed service, use SSE or HTTP-based transport instead of STDIO.

- MCP server connections support authentication using API token and token propagation. To authenticate using API key, ensure that a valid API token is available. To authenticate using token propagation, ensure that token propagation is enabled in your environment.
- The MCP server exposes the tools you want to use during agent or planner execution.

1. Open the navigation menu and click **Connectors**.

2. Click **MCP Server**.

The **MCP Definitions** list page opens. All the MCP servers that you have defined are displayed in a table.

3. Click .

The **New MCP Definition** dialog box appears.

4. Enter the following information.

- **Name:** Enter a unique, descriptive name for your MCP server. The name must be between 1 to 128 characters long, must start with an alphabet, and can contain only letters, numbers, underscores (_), and hyphens (-). Spaces and other special characters are not supported. This name helps you identify this MCP server definition in tasks and workflows.
- **Description:** Optional. Add details describing the purpose or scope of this MCP server definition.
- **Transport:** Select one of the following network transport protocols which the MCP server uses to establish a connection.
 - **STDIO:** Select this option when MicroTx Workflows must launch a local MCP process or command and communicate over stdin (standard input) and stdout (standard output).
 - **SSE:** Select this option when MCP server is hosted remotely and can be accessed over HTTP or HTTPS with server-sent events.
- Provide the following configuration information additionally only if you select **SSE** in the **Transport** drop-down list.
 - **Authorization:** Optional. Select one of the following options to set up authorization.
 - * **API Key** to grant access to the MCP server with the provided API key. You will need to include the JWT token, which authorizes workflow execution, as the authorization bearer token with every request to the MCP server.

- * **Token Propagation** to allow MicroTx Workflows to automatically propagate the JWT token which is sent to the MCP server with every request. Select this option when caller token propagation is configured for your environment.
 - * **None** Select this option for public or internal trusted MCP endpoints with no authorization requirement.
- **URL:** Provide the base URL of the MCP server.
- **SSE Endpoint:** Optional. Enter the full endpoint path for server-sent events (SSE). This is required for communicating with the MCP server. For example, /sse/notifications. The default value is /sse.
- **API Key:** Enter the API key which MicroTx Workflows uses to grant access the MCP server. You must provide an API key if you have selected **API Key** as the option to set up authorization. This is a secret credential and should be treated as sensitive data.
- Provide the following configuration information only if you select **STDIO** in the **Transport** drop-down list.
 - **Command:** Required. Enter the executable command used to start the MCP server process, such as binary or script.
 - **Args (comma-separated):** Optional. Enter a comma-separated list of command-line arguments passed to the command when starting the MCP server process.
- 5. Click **Test** to ensure that you can establish a connection with the provided details.
If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue, and then click **Close**.
- 6. Click **Submit** to create a new MCP definition with the details you have provided and save your configuration.

You can use the MCP server details within Agentic Planner task and in the Agent profiles, which are used in Agentic task inside workflow and the Conversational agent chat API.

4.4.2 Update an MCP Server Details

You can update all the field values you specified when providing details for an MCP server, except for the name.

1. Open the navigation menu and click **Connectors**.
2. Click **MCP Server**.
The **MCP Definitions** list page opens. All the MCP servers that you have defined are displayed in a table.
3. Identify the MCP definition that you want to update.
4. Click  (**Edit**).
5. You can update all the field values you specified when creating an MCP definition, except for the name.
6. Click **Test** to ensure that you can establish a connection with the provided details.
If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue, and then click **Close**.
7. Click **Submit**.

4.4.3 Test MCP Server Details

After creating or updating MCP server details, you can test it to ensure that you can establish a connection with the provided details.

To test MCP server details:

1. Open the navigation menu and click **Connectors**.
2. Click **MCP Server**.

The **MCP Definitions** list page opens. All the MCP servers that you have defined are displayed in a table.

3. Identify the MCP server details that you want to test.
4. Click  (**Test**).

A message is displayed that the test was successful.

If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue.

5. Click **Close** when you finish viewing the details.

4.4.4 View MCP Server Details

You can retrieve and view the details of an MCP server.

To view details of an MCP server:

1. Open the navigation menu and click **Connectors**.
2. Click **MCP Server**.

The **MCP Definitions** list page opens. All the MCP servers that you have defined are displayed in a table.

3. Identify the MCP server details that you want to view.
4. Click  (**Show Details**).
5. Click **Test** to ensure that you can establish a connection with the provided details.

If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue, and then click **Close**.

6. Click **Close** when you finish viewing the details.

4.4.5 Delete MCP Server Details

You can delete MCP server details when it is no longer required.

Before you delete MCP server details, ensure that it is not being used by any workflow. The workflow execution fails if it can't find a specified MCP server details.

1. Open the navigation menu and click **Connectors**.
2. Click **MCP Server**.

The **MCP Definitions** list page opens. All the MCP servers that you have defined are displayed in a table.

3. Identify the MCP definition that you want to delete.

4. Click  (**Delete**).
5. When prompted, confirm the deletion.

4.5 Manage File Storage

File storage in MicroTx Workflows provides a centralized way to define and manage local shared storage for files accessed during workflow and agent execution.

Use file storage to upload and reuse files for processing in scenarios such as:

- document ingestion
- agentic planner inputs
- agent/tool tasks that need file-based inputs

MicroTx Workflows stores files under the configured storage path, `conductor.storage.storage-path.location` and maintains file metadata for search and retrieval. The built-in File tool uses this storage path as its default base directory, so agents can read files from the same managed location. This feature provides access to local filesystem-based storage inside MicroTx Workflows.

Note

This feature is not a replacement for cloud object storage, such as Oracle Cloud Infrastructure (OCI) Object Storage.

Topics

- [Upload Files to Storage](#)
Upload one or more files into the local storage of MicroTx Workflows. These files are used for processing later by the File tool and by tasks, such as GenAI Ingestion tasks and Agentic Planner tasks.
- [Update a File in Storage](#)
You can upload an updated version of an existing file in storage.
- [Delete a File From Storage](#)
You can delete a file from storage when it is no longer required.
- [Delete all Files From Storage](#)
Delete all files from storage when you change the file storage location.

4.5.1 Upload Files to Storage

Upload one or more files into the local storage of MicroTx Workflows. These files are used for processing later by the File tool and by tasks, such as GenAI Ingestion tasks and Agentic Planner tasks.

The file storage is configured when MicroTx is installed based on the information provided in the `values.yaml` file. See *File System Storage Properties* in *MicroTx Installation and Configuration Guide*.

Ensure that the MicroTx Workflows runtime user has read and write permissions on this folder.

Upload limits and restrictions

- Multipart upload limit is 32 MB by default.

- Path traversal is blocked; files cannot be stored outside the configured storage file path.
- Upload all types of files as no file-type or extension whitelist is enforced during upload.
- If you use the REST API call to upload files to storage, `POST /api/storage/upload`, you must specify the server context path, such as `/workflow-server/api/storage/upload`.

To upload files to storage:

1. Open the navigation menu and click **Connectors**.
2. Click **Storage**.

All files available in storage are displayed in a table.

3. Click .

The **Upload Files** dialog box appears.

4. Select the files that you want to upload from your computer in one of the following ways:
 - Drag one or more files from your computer and drop them into the **Drag and drop files here or click to browse** box.
 - Click inside the **Drag and drop files here or click to browse** box. A file picker window opens. Navigate to and select one or more files that you want to upload.
5. Ensure that the file names that you have selected are listed.
6. Select **Overwrite existing files** to replace any existing files in the local storage with the latest uploaded files that have the same name.

By default this option is not selected. Files are uploaded to storage with the same name. By default, if a file with the same name already exists in the storage, that file is not uploaded.

7. Click **Upload** to upload the selected files to storage.

Files are uploaded to storage with the same name. If a file with the same name already exists in the storage, that file is overwritten if you select **Overwrite existing files**. By default, if a file with the same name already exists in the storage, that file is not uploaded.

After you upload the files to storage, you can use the uploaded files in tasks. The following sample code demonstrates how you can use `pdfreference1.4.pdf`, an uploaded file, in a GenAI Ingestion task.

```
{
  "name": "genai_ingestion",
  "taskReferenceName": "genai_ingestion_ref",
  "inputParameters": {
    ....
    "data": {
      {
        "source": "local",
        "filePath": "pdfreference1.4.pdf"
      }
      ....
    }
  }
}
```

4.5.2 Update a File in Storage

You can upload an updated version of an existing file in storage.

1. Open the navigation menu and click **Connectors**.
2. Click **Storage**.
All files available in storage are displayed in a table.
3. Identify the file that you want to update.
4. Click  (**Edit**).
5. Select the file from your computer in one of the following ways:
 - Drag one or more files from your computer and drop them into the **Drag and drop files here or click to browse** box.
 - Click inside the **Drag and drop files here or click to browse** box. A file picker window opens. Navigate to and select one or more files that you want to upload.

Note

Ensure that the file you select has the same name as the file that you want to update.

6. Ensure that the file that you have selected is listed.
7. Click **Save** to update the selected file.

4.5.3 Delete a File From Storage

You can delete a file from storage when it is no longer required.

Before you delete a file from storage, ensure that it is not being used in any workflow. The workflow execution fails if it can't find a specified file.

1. Open the navigation menu and click **Connectors**.
2. Click **Storage**.
All files available in storage are displayed in a table.
3. Identify the file that you want to delete from storage.
4. Click  (**Delete**).
5. When prompted, confirm the deletion.

4.5.4 Delete all Files From Storage

Delete all files from storage when you change the file storage location.

If you change the location where the files are stored, for example when you move files from one storage volume to another, then you must use this functionality to delete all entries of the files from MicroTx Workflows. Later, you can upload the files from the new storage location to MicroTx Workflows.

 Caution

Before you delete all files from storage, ensure that you copy all files used by workflows to a new location and retain the same file name. Once you delete all the files, you cannot recover them. Workflow execution fails if it can't locate a specified file.

1. Open the navigation menu and click **Connectors**.
2. Click **Storage**.
All files available in storage are displayed in a table.
3. Click  (**Delete All Files**).
4. Click **Delete** to confirm that you want to delete all the files.

4.6 Manage Cloud Profiles

Cloud profiles in MicroTx Workflows provide a centralized way to define and manage connections to Oracle Cloud Infrastructure (OCI) services, such as OCI Object Storage.

Topics

- [Create a Cloud Profile](#)
Create a cloud profile to define Oracle Cloud Infrastructure (OCI) connection parameters, such as user credentials and host details. You will use the cloud profile later in SFTP tasks to connect to the OCI object storage service to upload files.
- [Update a Cloud Profile](#)
- [View a Cloud Profile](#)
- [Delete a Cloud Profile](#)

4.6.1 Create a Cloud Profile

Create a cloud profile to define Oracle Cloud Infrastructure (OCI) connection parameters, such as user credentials and host details. You will use the cloud profile later in SFTP tasks to connect to the OCI object storage service to upload files.

Prerequisites

Before you begin, complete the following tasks:

- Create or identify the OCI user who will connect to the OCI object storage service.
- Note down the user ID and tenancy ID to access the OCI object storage bucket.
- Generate an OCI API signing key pair in PEM format. See [Required Keys and OCIDs](#).
- Upload the public key to the OCI user and note the public key value and fingerprint.
- Keep the private key value available; you will paste it into the cloud profile as the private key.

To create a cloud profile:

1. Open the navigation menu and click **Connectors**.
2. Click **Cloud**.

The **Cloud** profile list page opens. All the Cloud profiles that you have defined are displayed in a table.

3. Click .

The **New Cloud Profile** dialog box appears.

4. Enter the following information.
 - **Name:** Enter a unique and descriptive name to identify this profile in workflows. The name can be up to 128-characters long. Use only letters, numbers, underscores (_), and hyphens (-). Spaces and other special characters are not supported.
 - **User OCID:** Enter the user's OCID to authenticate and access an SFTP server in OCI.
 - **Tenancy OCID:** Enter the tenancy's OCID where the OCI object storage bucket is available.
 - **Fingerprint:** Enter the fingerprint of the public key.
 - **Private Key:** Paste the private SSH key required for key-based authentication.
 - **Region ID:** Specify your OCI region. For example, us-phoenix-1.
 - **Pass Phrase:** Optional. Pass phrase, if you provided one, while creating the private key. Click the eye icon to show/hide the value if needed.
5. Click **Submit** to create the cloud profile.

The new profile appears in the list of available cloud profiles. After creating a cloud profile, you can use it in SFTP tasks to connect to the OCI object storage service to upload files.

4.6.2 Update a Cloud Profile

You can update all the field values you specified when creating a cloud profile, except the name.

1. Open the navigation menu and click **Connectors**.
2. Click **Cloud**.

The **Cloud** profile list page opens. All the Cloud profiles that you have defined are displayed in a table.
3. Identify the Cloud profile that you want to update.
4. Click  (**Edit**).
5. You can update all the field values you specified when creating a cloud profile, except the name.
6. Click **Save**.

4.6.3 View a Cloud Profile

You can retrieve and view the details of a Cloud profile.

To view details of a Cloud profile:

1. Open the navigation menu and click **Connectors**.
2. Click **Cloud**.

The **Cloud** profile list page opens. All the Cloud profiles that you have defined are displayed in a table.

3. Identify the Cloud profile that you want to view.
4. Click  (**Show Details**).
5. Click **Close** when you finish viewing the details.

4.6.4 Delete a Cloud Profile

You can delete a Cloud profile when it is no longer required.

Before you delete a Cloud profile, ensure that it is not being used. The workflow execution fails if it can't find a specified Cloud profile.

1. Open the navigation menu and click **Connectors**.
2. Click **Cloud**.

The **Cloud** profile list page opens. All the Cloud profiles that you have defined are displayed in a table.

3. Identify the Cloud profile that you want to delete.
4. Click  (**Delete**).
5. When prompted, confirm the deletion.

4.7 Manage SFTP Profiles

Define and manage remote SFTP server connection details. Use SFTP profiles to access remote SFTP servers across workflows and tasks.

Topics

- [Create an SFTP Profile](#)
Create an SFTP profile to define connection parameters, such as user credentials and SFTP server details.
- [Update an SFTP Profile](#)
- [Test an SFTP Profile](#)
After creating or updating an SFTP profile, you can test it to ensure that you can establish a connection with the provided details.
- [View an SFTP Profile](#)
- [Delete an SFTP Profile](#)

4.7.1 Create an SFTP Profile

Create an SFTP profile to define connection parameters, such as user credentials and SFTP server details.

Prerequisites

Before you begin, complete the following tasks:

- Generate an API signing key and copy the private key value.
- Note down the username, host name, and port number to access the SFTP server.

You'll need to provide this information while creating an SFTP profile.

To create an SFTP Profile:

1. Open the navigation menu and click **Connectors**.
2. Click **SFTP**.

The **SFTP** profile list page opens. All the SFTP profiles that you have defined are displayed in a table.

3. Click .

The **New SFTP Profile** dialog box appears.

4. Enter the following information.
 - **Name:** Enter a unique and descriptive name to identify this SFTP profile in workflows. The name can be up to 128-characters long. Use only letters, numbers, underscores (`_`), and hyphens (`-`). Spaces and other special characters are not supported.
 - **Username:** Specify the username used to authenticate and connect to the SFTP server.
 - **Host:** Enter the hostname or IP address of the SFTP server to which you want to connect.
 - **Port:** Enter the network port for the SFTP connection (default is usually 22).
 - **Private Key:** Paste the private SSH key required for key-based authentication with the SFTP server.
 - **Pass Phrase:** Pass phrase for the private key, if any. Click the eye icon to show/hide the value if needed.
5. Click **Test** to ensure that you can establish a connection with the provided details.

If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue, and then click **Close**.
6. Click **Submit** to create the SFTP profile.

The new profile appears in the list of available SFTP profiles. Reference this name in your SFTP tasks to connect to the specified SFTP server.

4.7.2 Update an SFTP Profile

You can update all the field values you specified when creating an SFTP profile, except the name.

1. Open the navigation menu and click **Connectors**.
2. Click **SFTP**.

The **SFTP** profile list page opens. All the SFTP profiles that you have defined are displayed in a table.
3. Identify the SFTP profile that you want to update.
4. Click  (**Edit**).
5. You can update all the field values you specified when creating an SFTP profile, except the name.
6. Click **Test** to ensure that you can establish a connection with the provided details.

If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue, and then click **Close**.

7. Click **Save**.

4.7.3 Test an SFTP Profile

After creating or updating an SFTP profile, you can test it to ensure that you can establish a connection with the provided details.

To test an SFTP profile:

1. Open the navigation menu and click **Connectors**.
2. Click **SFTP**.

The **SFTP** profile list page opens. All the SFTP profiles that you have defined are displayed in a table.

3. Identify the SFTP profile that you want to test.
4. Click  (**Test**).

A message is displayed that the test was successful.

If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue.

5. Click **Close** when you finish viewing the details.

4.7.4 View an SFTP Profile

You can retrieve and view the details of an SFTP profile.

To view details of an SFTP profile:

1. Open the navigation menu and click **Connectors**.
2. Click **SFTP**.

The **SFTP** profile list page opens. All the SFTP profiles that you have defined are displayed in a table.

3. Identify the SFTP profile that you want to view.
4. Click  (**Show Details**).
5. Click **Test** to ensure that you can establish a connection with the provided details.

If there are any issues, an error message is displayed. Note down the error message so that you can troubleshoot the issue, and then click **Close**.

6. Click **Close** when you finish viewing the details.

4.7.5 Delete an SFTP Profile

You can delete an SFTP profile when it is no longer required.

Before you delete an SFTP profile, ensure that it is not being used by any SFTP task. The workflow execution fails if it can't find a specified SFTP profile.

1. Open the navigation menu and click **Connectors**.

2. Click **SFTP**.

The **SFTP** profile list page opens. All the SFTP profiles that you have defined are displayed in a table.

3. Identify the SFTP profile that you want to delete.
4. Click  (**Delete**).
5. When prompted, confirm the deletion.

5

Agentic AI

Agentic AI in MicroTx Workflows makes workflow execution more dynamic and adaptable.

It introduces intelligent orchestration through Agentic Planner, reusable AI-driven execution through Agentic Tasks, single-prompt AI integration through GenAI Tasks, built-in tools for common agent use cases, and secure integration with external tools and services through authenticated Model Context Protocol (MCP) connections.

With these capabilities, MicroTx Workflows enables you to create, configure, test, and manage reusable AI agents that can be invoked directly from workflows. Define these agents using prompts, tools, MCP servers, and related execution settings, allowing workflows to perform context-aware actions, interact with external systems, and support more advanced automation patterns.

For simpler AI use cases, MicroTx Workflows also provides GenAI Tasks, which allow you to integrate single-prompt LLM interactions directly into a workflow step. This is useful for scenarios such as summarization, classification, extraction, translation, or other focused prompt-based tasks that do not require full agent orchestration.

Benefits

Here are some of the benefits that this feature offers to you as a developer:

- **Dynamic orchestration:** Build workflows that can adapt at runtime using Agentic Planner, enabling more intelligent and context-aware execution paths.
- **Reusable agent execution:** Create reusable AI agents through Agentic Tasks and invoke them across workflows for consistent and maintainable automation.
- **Simple prompt-based AI integration:** Use GenAI Tasks to add focused LLM-powered capabilities to workflow steps without requiring full agent configuration.
- **Durable execution:** Run reusable agent-based workflow steps in a durable execution model, improving reliability for long-running and multi-step processes.
- **Built-in extensibility:** Use internal tools for common agent scenarios and extend capabilities further with secure MCP integrations.
- **Enhanced automation:** Enable higher levels of workflow automation by combining AI reasoning, reusable agent behavior, prompt-based AI tasks, and external tool access.
- **Simplified integration:** Invoke AI-powered capabilities directly within MicroTx Workflows with minimal boilerplate, which helps build intelligent workflows more efficiently.

Key Building Blocks

Use Agentic AI in MicroTx Workflows as a key building block in a few tasks and also leverage some other features of MicroTx Workflows.

- **Agentic Tasks**

Agentic Tasks allow you to define reusable AI agents and invoke them directly from workflows by name. These tasks encapsulate agent behavior in a reusable form, making it easier to standardize intelligent workflow steps across multiple business processes. With

durable execution support, Agentic Tasks can participate reliably in long-running workflow executions.

- **GenAI Tasks**

GenAI Tasks provide single-prompt AI integration for workflow steps that require direct interaction with an LLM. They are well suited for focused use cases such as summarization, content generation, classification, data extraction, and translation. GenAI Tasks offer a simple way to add AI capabilities to a workflow without defining a reusable agent.

- **Agentic Planner**

Agentic Planner enables dynamic orchestration within MicroTx Workflows. Instead of relying only on static workflow definitions, it can evaluate runtime context and determine the next action in the workflow. This makes workflows more flexible and better suited for scenarios that require adaptive decision-making.

- **Prompts**

Prompts are natural language instructions that guide agent or task behavior, such as summarizing content, generating plans, extracting information, or making decisions. They define the core behavior of AI-driven workflow steps in a format that is easy to author, maintain, and reuse.

- **Internal Tools**

Internal tools provide built-in capabilities for common agent use cases. These tools expand what agents can do during execution, such as performing calculations, invoking APIs, interacting with services, or carrying out other task-specific operations required by the workflow.

- **Model context protocol (MCP) server**

MCP servers allow agents to securely access external tools and services in a standardized way. MicroTx Workflows supports MCP integration with authentication, enabling developers to extend agent capabilities while maintaining controlled and secure access to external systems.

Topics

- [About Conversational Agent Chat](#)
Conversational Agent Chat lets client applications interact with MicroTx Workflows agent profiles through REST APIs. You can use this feature to build custom chat experiences where users ask questions, receive streamed agent responses, and continue a conversation with retained context.
- [Manage Prompt Template Definitions](#)
MicroTx Workflows enables you to define and manage reusable prompt templates for AI-powered workflow execution in a centralized way. Use prompt templates to drive GenAI Tasks for single-prompt AI use cases and Agentic Planner tasks for dynamic task and tool orchestration.
- [Manage Agent Profiles](#)
An agent profile provides the centralized configuration for reusable AI agents in MicroTx Workflows.

5.1 About Conversational Agent Chat

Conversational Agent Chat lets client applications interact with MicroTx Workflows agent profiles through REST APIs. You can use this feature to build custom chat experiences where

users ask questions, receive streamed agent responses, and continue a conversation with retained context.

MicroTx Workflows does not provide a built-in chat UI for this feature. Instead, it exposes REST APIs that customers can use to build their own web, mobile, or enterprise chat interface. The REST API specification is available through Swagger. A sample conversational agent application, such as a medical history retrieval chat sample, is available in the MicroTx samples repository.

A conversational agent uses the same Agent Profile model used by Agentic Tasks in workflows. To use an agent profile for chat, enable the **Conversational** capability in the agent profile. If you also want to use the same profile inside workflows, enable the **Workflow** capability as well.

The agent profile defines the agent's role, instructions, LLM profile, model, tools, MCP servers, memory setting, prompt variables, model parameters, and guardrails. During chat execution, MicroTx Workflows passes the user input to the agent, loads chat context when memory is enabled, invokes the configured LLM and tools, and streams progress and final response events back to the client using server-sent events (SSE).

Use Conversational Agent Chat for scenarios such as:

- Customer support assistants.
- Enterprise knowledge assistants.
- Medical, financial, or operational inquiry assistants.
- Workflow-driven assistants that can start or check workflows.
- Multi-agent assistants that delegate work to specialized agents.
- RAG-based assistants that retrieve information from vector stores.

Chat Memory

Conversational agent chat supports memory through the agent profile.

When memory is enabled, MicroTx Workflows loads recent chat messages and a running conversation summary for the chat session. This context is added to the prompt so the agent can respond based on the current conversation. The service maintains:

- Short-term memory: Recent chat messages for the session.
- Long-term summary: A compact summary of earlier conversation context.
- Chat history: Persisted user and agent messages that can be retrieved through the chat history APIs.

When memory is disabled on the agent profile, MicroTx Workflows does not load the previous chat context into the prompt. Each request is handled as a fresh interaction, although chat messages may still be stored for history and audit purposes.

Tools for Conversational Agents

Conversational agents can use tools configured in the agent profile. In addition to common tools, MicroTx Workflows supports special agentic tools such as:

- RAG Retrieval Tool: Allows an agent to retrieve relevant context from a vector store and use it in the response.
- Agent Execute Tool: Allows an agent to call another agent profile, enabling nested or multi-agent interactions.

- MicroTx Workflow Tool: Allows an agent to start an allowed workflow and check workflow execution status. This tool uses an allow-list of workflow names to restrict what the agent can invoke.
- MCP Server Tools: Allows agents to access external tools and services through configured MCP servers.

REST APIs for Conversational Agent Chat

MicroTx Workflows exposes APIs to chat with agents and manage chat history.

- POST /api/conversational-agent/chat: Starts or continues a chat session and streams the response using SSE.
- GET /api/conversational-agent/history/{agentName}: Lists chat sessions for an agent.
- GET /api/conversational-agent/history/{agentName}/{chatSessionId}: Retrieves chat history for a specific session.
- DELETE /api/conversational-agent/history/{agentName}: Deletes chat history for an agent.
- DELETE /api/conversational-agent/history/{agentName}/{chatSessionId}: Deletes a specific chat session.

The chat request includes the agent name, chat session ID, and user input.

```
{
  "agentName": "medical_history_agent",
  "chatSessionId": "session-001",
  "userInput": "Summarize the patient history and highlight open risks."
}
```

The chat API streams intermediate events such as memory loading, LLM request start, tool call start, tool call end, heartbeat events, and the final result.

- [Set Up Conversational Agent Chat](#)

5.1.1 Set Up Conversational Agent Chat

Prerequisites

Before setting up the conversational agent chat, complete the following steps:

- Configure the LLM provider and model that the agent uses. See [Create an LLM Definition](#).
- Create an agent profile. See [Create an Agent Profile](#).
 - Define the agent role, instructions, LLM profile, model, prompt variables, tools, MCP servers, guardrails, and model parameters.
 - Enable the **CONVERSATIONAL** capability. If the same profile must also run inside workflows, enable the **WORKFLOW** capability too.
 - Enable or disable memory on the agent profile. Memory is enabled by default. When enabled, MicroTx Workflows includes recent chat messages and a conversation summary in the prompt context.
- Configure tools if required. Create tool configurations for RAG retrieval, agent execution, workflow invocation, APIs, databases, files, or MCP servers, and attach the required tools to the agent profile. See [Create an Internal Tool Configuration](#).

- Secure API access. Ensure the client application obtains an OAuth 2.0 access token from the configured identity provider and passes it when invoking the chat APIs. RBAC permissions control who can execute chat and read or delete chat history.
- Configure streaming and memory settings if needed. Optional runtime properties include chat stream executor settings, SSE timeout and heartbeat settings, chat context cache TTL, maximum context tokens, maximum message tokens, and maximum short-term window messages.
- Plan deployment for multi-pod environments. The current implementation uses local in-memory cache for recent chat context. For multi-pod deployments, configure session affinity so requests for the same chat session are routed consistently. Align SSE heartbeat and timeout settings with load balancer and proxy idle timeouts.

Use the conversational agent through the chat REST APIs:

1. Build or configure a client application to call the chat streaming API.
2. Obtain an OAuth 2.0 access token from the configured identity provider.
3. Invoke POST /api/conversational-agent/chat with the agent name, chat session ID, and user input.
4. Consume the streamed SSE response events from the API, including tool execution events, heartbeat events, and the final agent response.
5. Use the history APIs as needed to list, retrieve, or delete chat sessions.

Example

```
{
  "agentName": "medical_history_agent",
  "chatSessionId": "session-001",
  "userInput": "Summarize the patient history and highlight open risks."
}
```

You can either integrate this API directly into your own application or invoke it directly for testing. You can also refer to a sample Python-based chat application in the GitHub samples repository that demonstrates use of the chat streaming API. See <https://github.com/oracle-samples/microtx-samples/tree/main/workflow/RAG-ingestion-retrieval-workflows>.

5.2 Manage Prompt Template Definitions

MicroTx Workflows enables you to define and manage reusable prompt templates for AI-powered workflow execution in a centralized way. Use prompt templates to drive GenAI Tasks for single-prompt AI use cases and Agentic Planner tasks for dynamic task and tool orchestration.

By defining prompt templates centrally, you can maintain prompt logic in one place and reuse it across workflows. This improves consistency, simplifies updates, and provides better control over how AI-powered workflow steps and orchestration logic are executed.

Prompt templates also support prompt variables, allowing workflow context to be embedded dynamically at runtime. Use these variables to pass workflow inputs, custom values, and outputs from previously executed tasks into the prompt. This makes it possible to build prompts that adapt to workflow state and runtime data without hardcoding values.

For example, in a loan processing workflow, you can use a prompt template to evaluate customer eligibility using applicant details supplied as workflow input. The template can also

include outputs from earlier workflow steps, such as identity verification or income validation results.

In an insurance claims workflow, you can use a prompt template to review claim details submitted by a customer, such as accident description, claim amount, and supporting documents. The prompt can instruct the model to classify the claim as **Valid**, **Needs More Information**, or **Potentially Fraudulent**, and provide a short justification. This result can then be used to determine the next workflow step, such as requesting additional documents, invoking another task, or escalating the claim for manual review.

Topics

- [Create a Prompt Template Definition](#)
Prompt template definitions are reusable and structured natural language system instructions—such as those used for summarizers or planners—that guide the output of tasks within a workflow.
- [Update a Prompt Definition](#)
You can update description and details in the prompt template.
- [View a Prompt Definition](#)
You can retrieve and view the details of a prompt definition.
- [Delete a Prompt Definition](#)
You can delete a prompt definition when it is no longer required.

5.2.1 Create a Prompt Template Definition

Prompt template definitions are reusable and structured natural language system instructions—such as those used for summarizers or planners—that guide the output of tasks within a workflow.

Create a prompt template definition, which represents a reusable, structured prompt that incorporates placeholders and variables, which allows for dynamic application within workflows. This ensures consistency, flexibility, and efficiency. The same prompt template definition can be reused and tailored to individual applications within various workflows. By allowing you to introduce placeholders and variables, prompt template definitions enable dynamic interaction and make it easy to adapt prompts to different scenarios and use cases.

To create a prompt template definition:

1. Open the navigation menu and click **Agentic AI**.
2. Click the **Prompt Template** tab.

The **Prompt Template Definitions** list page opens. All the prompt templates that you have defined are displayed in a table.

3. Click .

The **New Prompt Definition** dialog box appears.

4. Enter the following information.
 - **Name:** Enter a unique and descriptive name to identify this prompt template definition in workflows. The name can be up to 128-characters long. Use only letters, numbers, underscores (_), and hyphens (-). Spaces and other special characters are not supported. You cannot change this name later.
 - **Description:** Optional. Enter a description for the prompt template definition.

- **Prompt Text:** Create a reusable prompt template for AI-powered workflow execution. Use prompt templates in GenAI Tasks for single-prompt use cases, such as summarizing text or classifying input, and in the Agentic Planner to guide dynamic decision-making during workflow execution. Here's an example prompt template which retrieves the credit score and takes an appropriate action based on the credit score.

You are an AI planner for a loan approval workflow.
 Your goal is to decide the next tool to call based on the results of previous steps.
 Follow the conditions below exactly.
 First step: Connect to sqlcl 'oracledb_mcp' using connect tool and change status of
 Loan application with APPLICATION_ID=\${workflowId} to UNDER_REVIEW.
 UPDATE LOAN_APPLICATIONS SET APPLICATION_STATUS = 'UNDER_REVIEW' WHERE
 APPLICATION_ID = workflowId;
 Then, if no tasks have been run, call the `document\_verification\_task`.

Second step: After document verification: If
 `document\_verification\_task` failed, the process stops.
 Respond with a final status of 'FAILED'. If it succeeded, call the
 `compliance\_agent` and `loan\_processing\_agent`
 in parallel. After compliance and processing:
 * If `compliance\_agent` failed due to an 'AML_CHECK', call
 `notify\_aml\_check\_failure\_to\_admin`
 and `human\_aml\_verification` in parallel.
 * For any other failure, the process stops. Respond with a final
 status of 'FAILED'.
 * If all tasks succeed, the process is complete. Respond with a
 final status of 'SUCCESS'.

Output Instructions: Your response must only be a JSON object describing the next action. It should specify the `status` and a list of `next\_tools\_to\_call`. If the process is finished, the list should be empty.

You can also add variables, in the format \${workflowId}, where workflowId is the name of the variable. At runtime the variables are replaced with actual values.

5. Click **Submit**.

Your new definition appears in the list of available prompt template definitions. Use a prompt template definition to guide the behavior of AI agents within a workflow.

5.2.2 Update a Prompt Definition

You can update description and details in the prompt template.

1. Open the navigation menu and click **Agentic AI**.
2. Click the **Prompt Template** tab.

The **Prompt Template Definitions** list page opens. All the prompt templates that you have defined are displayed in a table.

3. Identify the prompt definition that you want to update.
4. Click  **(Edit)**.

5. You can edit the description and the details you have entered in the prompt template.
6. Click **Submit**.

5.2.3 View a Prompt Definition

You can retrieve and view the details of a prompt definition.

To view details of a prompt definition:

1. Open the navigation menu and click **Agentic AI**.
2. Click the **Prompt Template** tab.

The **Prompt Template Definitions** list page opens. All the prompt templates that you have defined are displayed in a table.

3. Identify the prompt definition that you want to view.
4. Click  (**Show Details**).
5. Click **Close** when you finish viewing the details.

5.2.4 Delete a Prompt Definition

You can delete a prompt definition when it is no longer required.

Before you delete a prompt definition, ensure that it is not being used by any workflow. The workflow execution fails if it can't find a specified prompt definition.

To delete a prompt definition:

1. Open the navigation menu and click **Agentic AI**.
2. Click the **Prompt Template** tab.

The **Prompt Template Definitions** list page opens. All the prompt templates that you have defined are displayed in a table.

3. Identify the prompt definition that you want to delete.
4. Click  (**Delete**).
5. When prompted, confirm the deletion.

5.3 Manage Agent Profiles

An agent profile provides the centralized configuration for reusable AI agents in MicroTx Workflows.

It defines key agent settings in one place, including the agent's role, instructions, prompt variables, internal tools, MCP servers, LLM profile, and model parameters. This makes agents easier to configure, reuse, and maintain across different use cases.

Agent profiles support two primary execution modes. They can run within workflows through Agentic Tasks, and they can also run as conversational agents through the chat API. When an Agent Profile is executed within a workflow, it benefits from the workflow engine's durability guarantees, making it suitable for reliable, long-running, and multi-step execution.

You can also use agent profiles to bring externally defined agent templates into MicroTx Workflows. You can port templates defined for external agents into an agent profile, and then execute it inside a workflow using the MicroTx runtime model.

For example, a document verification agent profile can define instructions for validating submitted documents, prompt variables for workflow data, internal tools or MCP-based integrations for OCR and income verification, and model settings tailored to the use case. You can use the same agent profile either inside a workflow for durable execution or through the conversational API for interactive use.

This approach helps developers build flexible, reusable, and consistently governed agent definitions across workflow and conversational scenarios.

Topics

- [Create an Agent Profile](#)
Create an agent profile to define a reusable AI agent in MicroTx Workflows. An agent profile provides a centralized place to configure an agent's role, instructions, prompt variables, tools, MCP servers, LLM profile, model parameters, memory behavior, execution limits, capabilities, and guardrails.
- [Update an Agent Profile](#)
- [View an Agent Profile](#)
You can retrieve and view the details of an agent profile definition.
- [Delete an Agent Profile](#)
You can delete an agent profile when it is no longer required.

5.3.1 Create an Agent Profile

Create an agent profile to define a reusable AI agent in MicroTx Workflows. An agent profile provides a centralized place to configure an agent's role, instructions, prompt variables, tools, MCP servers, LLM profile, model parameters, memory behavior, execution limits, capabilities, and guardrails.

Use agent profiles to run agents within workflows through Agentic Tasks or as conversational agents through the chat API. You can port templates defined for external agents and execute it within MicroTx Workflows.

Before creating an agent profile, ensure that the required LLM profile, tool configurations, and MCP server definitions are already available.

1. Open the navigation menu and click **Agentic AI**.
2. Click the **Agent Profile** tab.

The **Agent Profile Definitions** list page opens. All the agent profiles that you have defined are displayed in a table.

3. Click .

The **New Agent Profile Definition** dialog box appears.

4. Enter the following information.
 - **Name:** Enter a unique and descriptive name. This name is used to reference the agent profile definition in an Agentic Task in a workflow or through the chat conversation API using the agentName parameter. The name can be up to 128-characters long. Use only letters, numbers, underscores (_), and hyphens (-). Spaces and other special characters are not supported.
 - **Description:** Optional. Provide a brief description of this agent profile.
 - **Role:** Specify the role of the agent to be included in the execution prompt. This field helps establish the agent's high-level behavior, responsibilities, boundaries, and tone

as shown in the following example. Keep the content concise and stable so it consistently guides the agent during execution. The maximum supported length is 200000 characters.

You are an expert Oracle Database Assistant. Your primary function is to help users interact with the database by translating their natural language requests into valid SQL queries. You are precise, knowledgeable, and cautious, especially with data modification commands.

- **Instruction:** Enter detailed instructions to define the agent's behavior during execution. Describe what the agent should do when invoked, how it should respond, how it should use available tools, and any behavioral rules or constraints it must follow while interacting with workflow inputs, tools, or users. Keep the instructions clear, specific, and action-oriented so the agent behaves consistently as shown in the following example. The maximum supported length is 400000 characters.

Your primary task is to parse the user's request and convert it into a valid Oracle SQL query.

1. Analyze the user's request to understand their goal (e.g., retrieve data, describe a table, count rows).
2. Construct the appropriate SQL query using Oracle syntax.
3. Present the results from the database to the user in a clear and easy-to-understand format. For example, a markdown table for data.

- **Capabilities:** Select one or more of the following options to define the agent's operation modes.
 - **Workflow:** Allows agent to run within MicroTx Workflows, such as through an Agentic Task, while a workflow manages the execution.
 - **Conversational:** Allows agent to run as a conversational agent in free-form chat interactions through the conversation API.
- **LLM Profile:** Select an LLM profile that powers the agent's reasoning and language tasks.
Note: Agentic task does not support Cohere models.
- **Use Memory:** Optional. This setting applies only to conversational capability and it does not change behavior for workflow or agentic task execution. It controls whether a conversational agent uses prior chat context, including recent messages and summary, to answer new prompts in the same chat session. When enabled, the agent keeps contextual continuity across turns, so responses can reference earlier parts of the conversation. When disabled, the agent responds using only the current input and profile instructions; prior session context is not used to generate response. It is enabled by default.

 Caution

Disabling memory may reduce response relevance in multi-turn conversations since earlier context is not considered.

- **Max Messages:** Optional. Sets the maximum number of chat messages kept in the agent's in-memory context window during execution. When the limit is exceeded, older messages are evicted and only the most recent messages are sent to the model. This helps control token usage and cost. The default value is 20.

- **Prompt Variables:** If the prompt contains any variables, add the variables and specify the values that replace these variables at runtime.
- **Guardrails - Stop Words:** Define runtime stop-word rules that filter both requests and responses. For each stop-word rule, specify:
 - words: a case-insensitive list of terms or phrases to detect or match.
 - scope: specify where the system applies the rule
 - * INPUT: scan text before sending it to the model.
 - * OUTPUT: scan the model's response before returning it.
 - * BOTH: scan both input and output.
 - action: specify what the system does when it finds a match:
 - * MASK: replace the matched text with ***.
 - * FAIL: stop execution and raise a guardrail violation error.

At runtime, the system evaluates all rules. It runs input guardrails before the LLM call and output guardrails after the LLM generates a response. It matches content using case-insensitive string matching. If you omit or provide an invalid scope or action, the system does not enforce that rule. If you omit words or set it to an empty list, the rule has nothing to match.

- **Temperature:** Optional. Controls randomness of the generated response. Lower values generate more deterministic responses. Higher values generate more varied responses. The default value is 0.2.
- **Max Tokens:** Optional. Sets the maximum number of tokens that the model can generate. The default value is 1024.
- **Top K:** Optional. Number of highest-probability tokens considered during generation. The default value is 40.
- **Top P:** Optional. Controls nucleus sampling value. The default value is 0.9.
- **Max Tool Calls:** Optional. Limits the number of tool invocations per session. This prevents runaway invocation loops and constrains execution cost and side effects. The default value is 10.
- **MCP Servers:** Optional. Select one or more Model Context Protocol (MCP) servers that the agent can access and use for executing tasks or accessing resources.
- **Tools:** Optional. Select one or more tool configurations that you have created previously in MicroTx Workflows. The agent accesses and invokes the specified tools during workflow execution.

5. Click **Submit**.

Your new definition appears in the list of available agent profile definitions. You can invoke the agent profile by name in workflows or tasks.

Example

The following JSON code provides sample values for parameters of an agent profile.

```
{
  "name": "orderProcessorAgent",
  "description": "Handles order-related workflow and chat queries",
  "role": "Process and respond to all order fulfillment workflow tasks",
  "instruction": "Always validate input, check inventory before approving"
```

```

order, escalate exceptions",
  "llmProfile": {
    "model": "gpt-4",
    "vendor": "OpenAI"
  },
  "tools": [
    "OrderDBLookupTool",
    "EmailSender"
  ],
  "mcpServers": [
    "inventoryMCP",
    "shippingMCP"
  ],
  "memory": true,
  "maxMessages": 30,
  "maxToolCalls": 5,
  "capabilities": [
    "WORKFLOW",
    "CONVERSATIONAL"
  ],
  "promptVariables": {
    "companyBrand": "MicroTx",
    "region": "APAC"
  },
  "temperature": 0.5,
  "maxTokens": 2048,
  "topK": 50,
  "topP": 0.9,
  "maxCompletionTokens": 512,
  "guardrails": {
    "stopWords": [
      {
        "words": [
          "password"
        ],
        "scope": "INPUT",
        "action": "MASK"
      }
    ]
  }
}

```

5.3.2 Update an Agent Profile

1. Open the navigation menu and click **Agentic AI**.
2. Click the **Agent Profile** tab.
The **Agent Profile Definitions** list page opens. All the agent profiles that you have defined are displayed in a table.
3. Identify the agent profile definition that you want to update.
4. Click  (**Edit**).

5. You can update all the field values you specified when creating an agent profile, except for the name.
6. Click **Submit**.

5.3.3 View an Agent Profile

You can retrieve and view the details of an agent profile definition.

1. Open the navigation menu and click **Agentic AI**.
2. Click the **Agent Profile** tab.
The **Agent Profile Definitions** list page opens. All the agent profiles that you have defined are displayed in a table.
3. Identify the agent profile definition that you want to view.
4. Click  (**Show Details**).
5. Click **Close** after viewing the details.

5.3.4 Delete an Agent Profile

You can delete an agent profile when it is no longer required.

Before deleting an agent profile, ensure that it is not referenced by any workflow or being used in any active chat conversations. If a workflow or conversation attempts to invoke a deleted agent profile, the execution fails because the specified agent profile can no longer be resolved.

1. Open the navigation menu and click **Agentic AI**.
2. Click the **Agent Profile** tab.
The **Agent Profile Definitions** list page opens. All the agent profiles that you have defined are displayed in a table.
3. Identify the agent profile definition that you want to delete.
4. Click  (**Delete**).
5. When prompted, confirm the deletion.

6

Known Issues in MicroTx Workflows

Learn about the issues you may encounter and how to work around them.

Topics:

- [Cannot return to the Workbench page from the workflow execution details page](#)
- [Failed to generate response for Agentic Task](#)

Cannot return to the Workbench page from the workflow execution details page

When you run a workflow in the Workbench, a new workflow execution ID is displayed under **Execution History**. If you click the workflow execution ID, the status of the workflow execution is displayed on a new page. From this new page, you cannot return to the Workbench page by clicking **Workbench** in the navigation menu.

Workaround:

1. Open the navigation menu, and click any option other than Workbench, such as **Agentic AI** or **Workflow Builder**.
2. Open the navigation menu, and then click **Workbench**. The Workbench page is displayed.

Failed to generate response for Agentic Task

When you use Cohere models in the LLM profile of an Agentic Task, it throws the following error:

```
Task <task-id> failed with status: FAILED and reason: 'Failed to generate response for Agent <agent-name>'
```

Workaround:

Do not use Cohere models in Agentic tasks. You can use other models in the LLM profile of an Agentic task.