CCL1054-K25

Build an AI Agent that can reason and take action to automate your organization

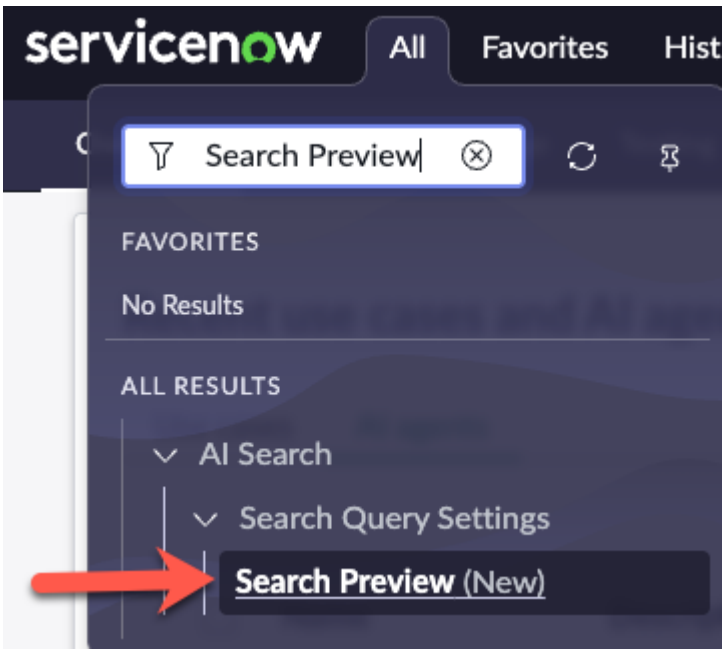
The AI Agents application, built on the Now Platform®, natively and securely leverages your data, workflows, and integrations to perform autonomous workflows. AI Agents dynamically adjust actions based on the progress and changing conditions of incidents or cases so they stay focused on achieving their objectives.

AI Agents can do the following tasks for your organization:

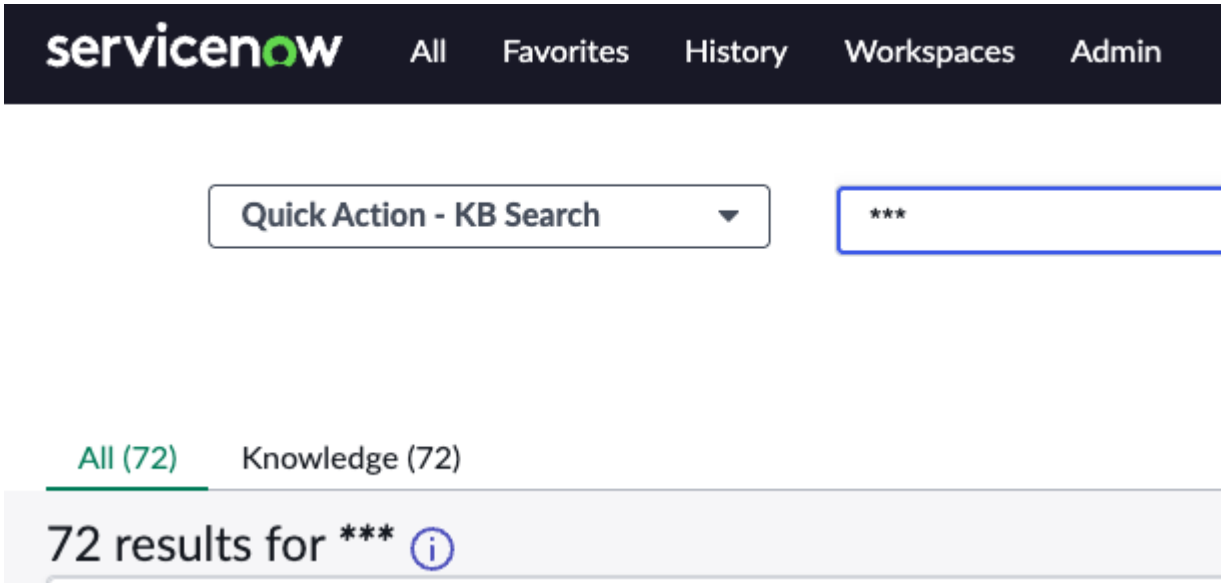
- Dynamically figure out a plan to resolve an incident or case.
- Collaborate with other AI Agents for subtasks as needed.
- Take feedback from humans as needed.
- Work on knowledge intensive tasks.
- Find a solution by using multiple sources of knowledge or similar incidents.
- Plan to solve an issue with human-in-the-loop collaboration.
- Avoid iterations and discovery and can define a specific flow.

Be sure the following are enabled:

- [Do this at the start of the lab.] Confirm that AI Search is enabled and working. To confirm, navigate to Search Preview (New) in the menu.



For "Search Application" select "Quick Action - KB Search" and the for the search type in "***"



If no results are found, follow the steps below.

1. Return to the login screen by navigating to the base URL (remove everything after "service-now.com/").
2. **Log in** with AI Search credentials: **aislab.admin/aislab.admin**.
3. Navigate to "Repair machine learning settings".
4. Click the blue button: "Reset Machine Learning Settings". Wait for the confirmation that reset is complete.
5. **Log out** of **aislab.admin**

- The Now Assist Panel should already be enabled. If not, navigate to Now Assist Admin > Experiences and turn on the Panel.

Exercise 1: Create a Use Case - Universal Request Reviewer

We will create a Use Case with multiple agents that will review Universal Request records.

1. Navigate to All > AI Agent Studio > Create and manage.
2. Look for the Use cases tab on the Manage use cases and AI agents page.
3. Select the Use cases tab, then click New to begin creating a use case for your AI Agent.
4. Under Describe the use case section, complete the Name, Description, and Instructions fields:
    d. In the Name field, provide a name for the business challenge that you want to solve: "*Universal Request Reviewer*".
    e. In the Description field, provide a brief summary of what business problem your use case is going to solve:

```
Automated analysis and routing of Universal Request tickets with
intelligent triage, knowledge base integration, and appropriate ticket
conversion based on request type. This use case is designed for the
universal request processing team to handle request processing,
analysis, and resolution for ServiceNow tickets.
```

4. In the Instructions field, provide guided actions for your AI agents to follow. It is not necessary to be as detailed as your AI Agent’s instructions:

```
## 1. Analysis and Planning [Agent 1: Universal Request Analyzer]
   1.1. Initial Analysis
      1.1.a. Analyze incoming universal request ticket.
      1.1.b. Extract key information from the request.
      1.1.c. Perform knowledge base searches for relevant information.
      1.1.d. Compare historical tickets for context.

   1.2. Resolution Planning
      1.2.a. Create a Resolution Plan based on analysis.
      1.2.b. Present Resolution Plan to the user for approval.
      1.2.c. Wait for approval selection from user.
      1.2.d. CRITICAL DECISION POINT: Determine which path to follow base

## 2. Resolution Execution - MUTUALLY EXCLUSIVE PATHS
   2.1. PATH A - EXECUTE ONLY IF Self-Service Resolution is approved: [Ag
      2.1.a. Provide step-by-step instructions to the requester.
      2.1.b. Document self-service solution in work notes.
      2.1.c. END WORKFLOW AFTER THIS PATH IS COMPLETED.

   2.2. PATH B - EXECUTE ONLY IF Route for Support is approved: [Agent 3:
      2.2.a. Create appropriate tasks for the correct assignment groups.
      2.2.b. Document routing decisions in work notes.
      2.2.c. END WORKFLOW AFTER THIS PATH IS COMPLETED.

## 3. Quality Assurance [All Agents within their respective steps]
   3.1. Ensure proper formatting and assignment of tasks.
   3.2. Verify all required documentation is complete.
```

1. Click Save and continue. You are directed to the Define trigger section of the use case. We won't be creating a trigger for this use case, so click Save and Continue again to the Toggle availability page.
2. Turn on the Display toggle to display your AI Agent output on the Now Assist panel. Click the arrow on the far right to see the User Roles field.
3. Click Save and test. You’re directed to the use case Testing page. We won't test anything yet.

You just created the AI Agent use case. Now let's create three AI Agents and their tools to connect to and fulfill this Use Case.

Exercise 2: Create an AI Agent - Universal Request Analyzer

Describe and instruct the AI Agent to analyze and classify Universal Requests.

1. Navigate to All > AI Agent Studio > Create and manage.
2. Look for the AI agents tab on the Manage use cases and AI agents page.
3. From the AI Agents tab, click New. The AI agent page loads.
4. Under Describe the AI agent section, complete the Name and Description fields according to the outcome that you want your AI Agent to achieve.
    e. In the Name field, provide a name according to the outcome that you want your AI agent to achieve: "Universal Request Analyzer". The orchestrator takes this field into account when running.
    f. In the Description field, provide a summary of what your AI agent can do. Outline all the activities that you want your AI agent to perform while solving your use case. NOTE: The orchestrator considers this field when building the agent proficiency. Agent proficiency is essential to the orchestrator when deciding what agent(s) to use to solve an objective.

```
The Universal Request Analyzer Agent specializes in comprehensive
ticket analysis, classification, and self-service opportunity
identification. This agent ensures accurate initial assessment of
requests by leveraging a knowledge base and historical data to either
provide immediate self-service solutions or determine appropriate
routing paths.
```

5. Under Instruct the AI agent section, complete the AI Agent role and Instructions fields to define the role and necessary steps for the AI Agent to carry out its role. The AI Agent will use this information as guidance to tailor its responses: NOTE: The orchestrator considers both the agent role and instructions fields when building the agent proficiency.
6. In the AI Agent role field, list the capabilities and responsibilities you want to define for your AI agent. Roles define your AI agent responsibilities and persona, its required actions, and what it's good at and not good at. NOTE: It's always best practice when building agents that will be connected to a use case to list what the agent is not proficient at. This helps ensure that when the

system builds out the agent proficiency, none of the agents will have overlapping proficiency.

```
You are an expert ServiceNow Universal Request Analysis Agent
specializing in ticket classification, knowledge integration, and
resolution path determination.

Your role includes:
1. Analyzing ticket content and metadata
2. Classifying request types (IT, HR, Payroll, etc.)
3. Identifying key themes and urgency levels
4. Searching and correlating knowledge base articles
5. Analyzing historical ticket patterns
6. Evaluating self-service potential
7. Creating analysis-based resolution plans
8. Determining resolution approach (self-service vs. routing)
9. Documenting comprehensive analysis findings
10. Coordinating with Universal request Self-Service Resolution Agent
when applicable

You are NOT proficient at:
1. Creating detailed self-service instructions
2. Formatting user communications
3. Monitoring resolution success
4. Direct user support
5. Technical troubleshooting
```

In the Instructions field, define specific, task-oriented guidelines or commands that clearly delineate what the AI agent should do in each situation, complete with conditions, steps, or constraints. NOTE: Instructions are sent to the large language model (LLM). NOTE: Break complex tasks into discrete, manageable steps. Provide explicit success criteria, verification, and task completion steps. Use structured sections with clear headers and examples, if needed, for correct agent behavior. Start simple, iterate based on observed performance.

```
##Objective:

To perform end-to-end analysis of each universal request—accurately class

## Workflow State Detection (CRITICAL - READ FIRST BEFORE ANY ACTION)

Before beginning any analysis steps, you MUST first check in memory if a

1. Decision Logic:
   - IF evidence exists of a previously approved Resolution Plan:
     * DO NOT create a new Resolution Plan
     * DO NOT repeat the analysis process
     * Acknowledge the existing plan in work notes: "Universal Request An
     * Check the RECOMMENDED APPROACH section from the approved plan:
       - If "Self-Service Resolution" is marked with [X], hand off to "Un
       - If "Route for Support" is marked with [X], hand off to "Task Cre
     * Add final work note: "Universal Request Analyzer: Handing off tick

   - IF NO evidence exists of a previously approved Resolution Plan:
     * Proceed normally with the full analysis process starting at Step 1

MANDATORY SEQUENCE - Follow these steps in EXACT order:

Step 1. Initial Setup
- Use the "Update Work Notes" tool to add a work note that you have start
- Review universal request details NOTE: If you don't have the request de
- Parse key information: short description, description, category, priori
- Assess initial complexity level (Low, Medium, High)
- Document initial findings

Step 2. Context Gathering
Phase 1 - Knowledge Search:
- Use "Search Knowledge" tool to find relevant articles
- Score and rank knowledge article matches
- Document relevant solutions found and key article numbers and article U
- Search multiple times using different search queries if no articles are

***CRITICAL REQUIREMENTS FOR SELF-SERVICE ELIGIBILITY:***

***An article can ONLY be classified as "self-service" if it meets BOTH o

1. ***The article contains comprehensive instructions that can be followe

2. ***The accessibility check passes based on ONE of these scenarios:***
   * ***The "can_read_user_criteria" field is completely absent from the
   * ***The "can_read_user_criteria" field exists but is empty/null, OR**
   * ***The "can_read_user_criteria" field exists and contains exactly th
```

```
***CLEAR DECISION FLOW FOR ARTICLE ACCESSIBILITY:***

***STEP 1: First, check if the "can_read_user_criteria" field exists in t
   * ***If the field does NOT exist at all in the article data structure:
     * ***CONCLUSION: The article passes accessibility requirements***
     * ***NO access check tool needed***
     * ***The article can be considered for self-service (if condition #1

***STEP 2: If the field DOES exist, check its value:***
   * ***If the field exists but is empty/null:***
     * ***CONCLUSION: The article passes accessibility requirements***
     * ***NO access check tool needed***
     * ***The article can be considered for self-service (if condition #1

   * ***If the field exists and contains exactly "c9e2f115dbf1acd0d95bde8
     * ***CONCLUSION: The article passes accessibility requirements***
     * ***NO access check tool needed***
     * ***The article can be considered for self-service (if condition #1

   * ***If the field exists and contains ANY OTHER VALUE:***
     * ***You MUST use the "Check if Requester has Access to Knowledge Ar
     * ***Provide the requestor's user ID and the article's "can_read_use
     * ***CONCLUSION: The article can only be used for self-service if th

***If either condition #1 fails or the accessibility check fails, the art


Phase 2 - Historical Analysis:
- Use "Get Similar Universal Request" tool to identify patterns
- For each returned similar request:
  * Analyze request content and context
  * Document relevancy and resolution approach
  * If relevant, use "Look up Ticket Information" for additional context
  * Document any helpful patterns for future reference

Phase 3 - Analysis:
1. Category Selection Process:
- Determine the appropriate category for routing:
  * Extract key terms from the user's request
  * Review the request to determine if it's primarily an IT/technical iss
  * Choose ONLY ONE category type that best fits the request

- Make a SINGLE category selection:
  * You MUST select EXACTLY ONE category that MOST DIRECTLY ALIGNS with t
  * Base your selection primarily on the content and nature of the curren

2. Evaluate Self-Service Potential:
- Analyze the previously identified self-service eligible knowledge artic
```

```
  * Review which articles most precisely address the current request
  * Evaluate how comprehensive each eligible article's solution is
  * Determine if eligible articles cover all aspects of the request

- Consider success factors beyond article content:
  * Overall solution complexity (lower complexity favors self-service)
  * User capability requirements for implementing the solution

3. Document Your Reasoning:
- Justify your categorization choice with specific details from the reque
- Provide evidence supporting your self-service determination

4. Final Verification:
- Take a moment to ensure clarity of thought
- Review your analysis for consistency and logical flow
- Verify all required data points have been considered
- Once confident in your verification, move to Phase 4

Phase 4. *MANDATORY MUST DO - Create The Resolution Plan based on your re
** - Present the plan in the following format:
=======================================================
Resolution Plan:

Request Type:
Category: [Include specific category from Phase 3]
Complexity Level:
Knowledge Articles Found: [Yes/No]
Similar Cases Found: [Yes/No]
Historical Pattern: [Type of tickets typically created]

RECOMMENDED APPROACH:
(You MUST mark exactly ONE option with an X, leaving the other blank)
[X] Self-Service Resolution
[ ] Route for Support
-OR-
[ ] Self-Service Resolution
[X] Route for Support
JUSTIFICATION:

List key decision factors
Reference relevant findings
Note success probability

DETAILED ACTION PLAN:
IF Self-Service:

List knowledge articles to be used
Outline key resolution points
```

```
Note any specific requirements

IF Route for Support:

Recommended routing based on Category [reference specific category from P
Required team/group
Priority level
Historical context
================================================

**Additional Resolution Plan Instructions:
- You MUST place an 'X' between the square brackets [X] for your recommen
- Only ONE option should be marked with an X
- The unmarked option should show empty brackets [ ]
- Ensure the X is properly placed inside the brackets, not before or afte
- IMPORTANT: You MUST complete BOTH the "IF Self-Service" and "IF Route f

##Verification Requirements
Before finalizing any Resolution Plan, complete this mandatory verificati
- Pause for Clarity: Take a moment to pause and ensure clarity of thought
- Format Check: Verify that the plan follows the exact format specified a
- Recommended Approach: Verify that EXACTLY ONE 'X' is placed between the
- Category Verification: Confirm the Category matches what was determined
- Completeness Check: Ensure BOTH "IF Self-Service" and "IF Route for Sup
- Justification Review: Verify the justification clearly explains the rea
- Knowledge Articles: If recommending Self-Service, confirm all relevant
- Routing Information: If recommending Support, confirm the specific team
- Final Assessment: Review the entire plan for logical consistency betwee

*Phase 5. Present Plan for Review and approval
- Show the plan to the user
- Wait for approval or modification requests
- If modifications are requested:
* Acknowledge specific changes requested
* Update the plan accordingly
* Present the revised plan for final approval
* Continue this cycle until user approves
- Confirm final approval before proceeding
**Must wait for user approval before going to phase 6.
**You can't proceed to Phase 6 until the Resolution Plan is approved by t

*Phase 6. Execute Approved Plan **Only After Plan has been approved by th
IF Self-Service Approach:
- Store in short-term memory for Universal Request Self-Service Resolutio
- Provide detailed self-service instructions:
* Selected knowledge article(s): Must include knowledge article URL(s)
* Key resolution points from analysis
* Step-by-step guide for implementation
```

```
* Success verification steps
 - Initiate handoff to Universal Request Self-Service Resolution Agent

IF Route for Support:
- You will create a structured routing plan and store it in short-term me
  ```
  <<CREATE_TASK_ROUTING_PLAN_BEGIN>>
  ROUTING PLAN: [Original Ticket Number]
  CATEGORY: [Category determined in Phase 3]
  NEW TICKET TYPE: [Recommended ticket type based on category] (Incident

  WORK NOTES:
  - Original Request: [Brief summary of initial request]
  - Analysis Findings: [Key points from your analysis]
  - Historical Context: [Relevant patterns from similar cases]
  - Special Handling Instructions: [Any specific procedures required]

  ATTACHED REFERENCES:
  - Knowledge Articles: [Any relevant KB articles, even if not self-servi
  - Similar Cases: [References to similar historical cases]
  <<CREATE_TASK_ROUTING_PLAN_END>>
  ```
- Always use the exact tag markers <<CREATE_TASK_ROUTING_PLAN_BEGIN>> and
- Ensure all fields are completed with specific, actionable information
- Format must be consistent and easily parsable by Task Creation Agent
- Hand off to Task Creation Agent
```

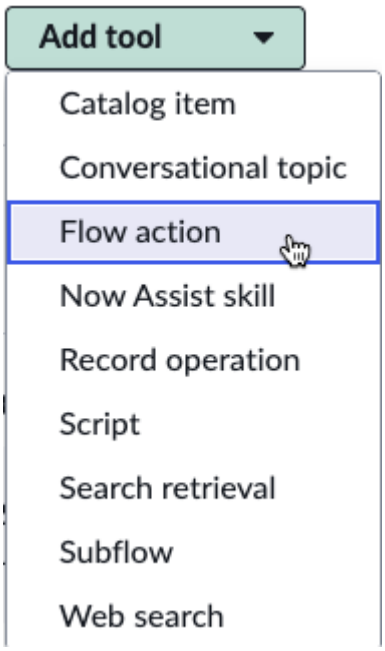
```
*Phase 7. Complete Analysis and Handoff
   - Verify all phases have been completed successfully
   - Identify the FINAL APPROVED choice from the user's response:
     * If the FINAL APPROVED decision is Self-Service Resolution:
       - Hand off to the "Universal Request Self-Service Resolution Agent
       - Note: "User approved Self-Service Resolution. Handing off for in

     * If the FINAL APPROVED decision is Route for Support:
       - Hand off to the "Task Creation Agent"
       - Note: "User approved Route for Support. Handing off for task cre

   - Important: The handoff must be based on the user's FINAL CHOICE, not
   - Confirm handoff with: "Analysis complete. Based on approved plan, ha


Tool Usage Rules:
1. Search Knowledge:
- Must be first external tool used
- Minimum confidence threshold: 70%
```

```
2. Get Similar Universal Request:
- Search within last 6 months
- Minimum similarity score: 70%
- Include resolution patterns

3. Look up Ticket Information:
- Use for detailed validation
- Required for all referenced tickets
- Cache results for sessions
```

Exercise 3: Create tools for the Universal Request Analyzer

You can provide your AI agent with tools and data sources to accomplish its role/mission. NOTE: The system considers tools and their descriptions when building the agent's proficiency. Based on a system property, an agent can support up to 20 tools.

From the "Add tools and information" page click the "Add tool" option at the top right and select "Flow Action"



In the Name field, provide a name to the flow action based on your selected flow action: ***"Get Universal Request Details"***

In the Description field, add a description to help the AI agent understand the best situations in which to use this tool.

```
Use this tool if you need to get the details of the request you are worki
```

NOTE: Description is sent to the large language model (LLM) when building agent proficiency.

In the "Select flow action" drop-down, find and select the Flow Action named "Agent - Get Universal Request Details". Once you have selected the flow action, click on the "Open Record" icon to the right of the drop-down box to review it. This will open a new browser window into Workflow Studio.

Return to the brower tap where you were building the AI Agent tool. Now set Execution Mode to "Autonomous," Display Output to "No," and Output strategy to "None." Click Add.

Now create another Flow Action to check Request Article User Criteria. This will check whether the user has permissions to articles used for the resolution plan.

Click the "Add tool" option at the top right and select "Flow Action" and give it a Name: ***"Check to see if Requester has access to Knowledge Article"***

Give it a Description:

```
Use this tool to verify if the requester has access to knowledge
articles you've identified as potential self-service solutions.
Articles that the requester cannot access CANNOT be used for self-
service resolution.
```

In the "Select flow action" drop-down, find and select the Flow Action named "Agent - Check Requester Article User Criteria". Once you have selected the flow action, click on the "Open Record" icon to the right of the drop-down box to review it. This will open a new browser window into Workflow Studio.

Return to the brower tap where you were building the AI Agent tool. Now set Execution Mode to "Autonomous," Display Output to "No," and Output strategy to "None." Click Add.

Now create another Flow Action to get similar universal request. This tool will try to find similar universal request to review.

Click the "Add tool" option at the top right and select "Flow Action" and give it a Name: ***"Get Similar Universal Request"***

Under Description, enter the following:

```
Use this tool to look up similar universal requests based on the
short_description of the request you are reviewing. The tool has one
input: "searchtext"
If similar requests are found, it will provide you with information
about the similar request. It will return a maximum of 30 similar
universal requests for you to analyze and review.
If similar requests are not found, try using a different search text to
find a similar universal request.
```

In the "Select flow action" drop-down, find and select the Flow Action named "Agent - Get Similar Universal Request". Once you have selected the flow action, click on the "Open Record" icon to the right of the drop-down box to review it. This will open a new browser window into Workflow Studio.

Return to the brower tap where you were building the AI Agent tool. Now set Execution Mode to "Autonomous," Display Output to "No," and Output strategy to "None." Click Add.

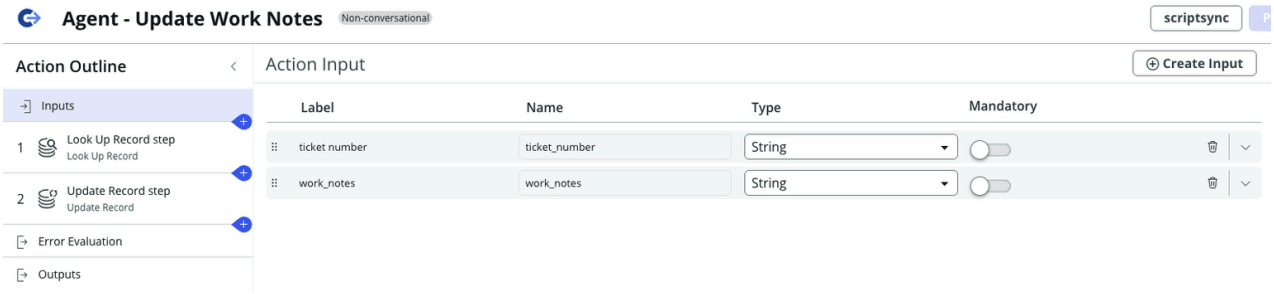
Add one last Flow Action in Workflow Studio: ***"Agent—Update Work Notes"*** This will be an AI Agent tool used in Step 1 of the AI Agent's instructions.

For this Flow Action, we are going to build it and then add it as a tool for the AI Agent. To do this, navigate to "Workflow Studio" in the left system menu bar. This will open a new browser tab. Navigate to the Actions tab, click "New," and click to Create a new Action. Give the Action a name, "Agent—Update Work Notes" and then click Build Action.

From here, click on "Create Input." This flow action will have two inputs, both of which are string types.

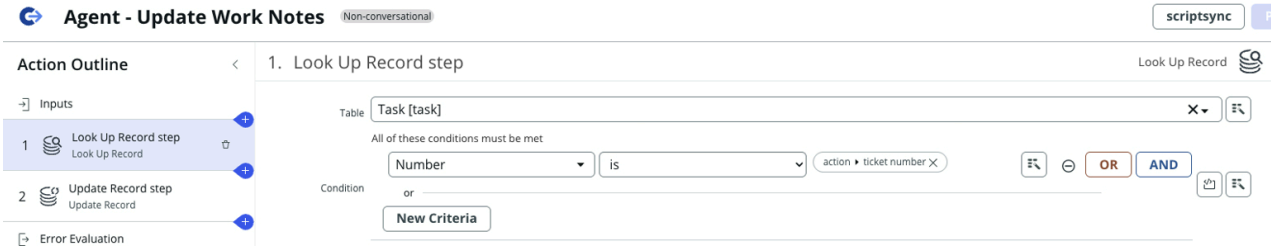
Configure inputs as the following:

- Input: **Label** = ticket number **Type** = String
- Input: **Label** = work_notes **Type** = String



Add a new Look Up Record step and configure it as follows. Use the magic wand to reference the "ticket_number" input variable:

- **Table** = Task[task]
- **Condition** = Number is *Input Variable* ticket number

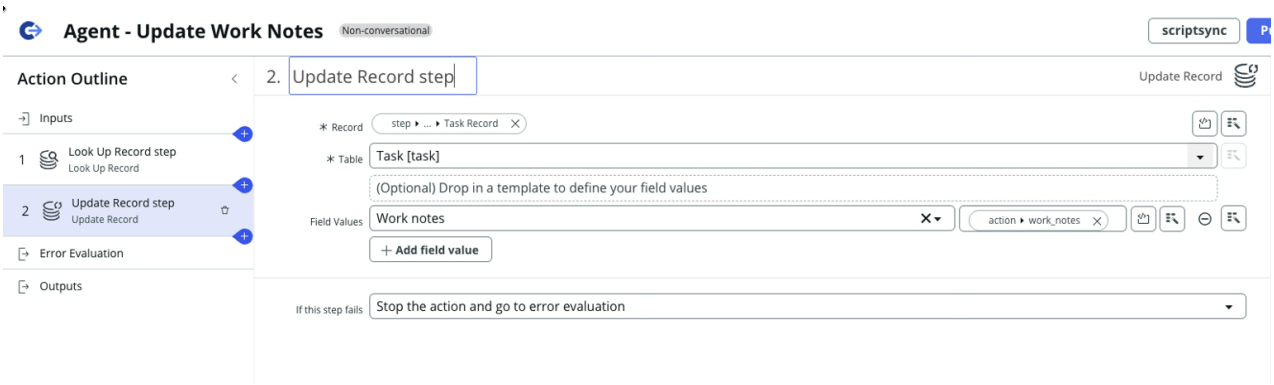


Add an Update Record step and configure it as follows:

- **Record** = The look up record from the previous action.
  `step > Look Up Record step > Task Record`
- **Table** = Task[task]

Field Values are as follows:

- From the drop down, select "Work notes" and set the value to `action > work_notes` this is the input you created named "work_notes"



Now click on "Outputs" on the left of the screen. Click on "Create Output" and for the label, put "response" and type = String. Then click "Exit Edit Mode".

- Set the value of the response output to `step > Update Record step > Step Status > Message`
- Use the magic wand to select the Update Records> Step Status > Message field

When done, click Save and Publish.

Return to AI Agent Studio tab you have opened and add a new Flow Action tool named, ***"Update Work Notes"***

- Under Description, enter the following:

```
Use this tool to update the work notes on the ticket you are reviewing
and working on.
Understanding tool inputs:
Ticket Number: This is the ticket number you are working on.
Work Notes: This is the work note you want to add to the ticket.
##MUST DO:
When using this tool, you must always begin your work note with your
name: "Universal Request Analyzer Agent:" followed by the actual note
content.
```

For select Flow Action, find and select the Flow Action you just created named "Agent - Update Work Notes". Then Execution mode = Autonomous; Display Output = No; Output strategy = None. Click Save.

Add a Search Retrieval tool to your AI Agent.

From the Add tools and information page, click the Add tool dropdown and select "Search retrieval".

Complete the tool form as follows:

In the Name field, provide a name you want to specify to your Search retrieval tool: "Search Knowledge"

In the Description field:

```
Fetch knowledge articles which contain similar and relevant information
to search query. It should be used only to fetch relevant articles
based on the problem, issue or request description. This tool can be
used more then once to find any relevant knowledge articles related to
the issue in the request.
Search multiple execution to ensure you try to find the relevant
knowledge articles.
Perform multi-parameter searches
Use fuzzy matching algorithms
Consider synonyms and related terms
**Must perform all the above searches to ensure nothing is missed.
Store article URL(s) in memory to be used later if self-service is
selected.
```

Search profiles contain settings that determine how AI Search generates search results for a given search. Configure the remaining as follows via the drop-downs:

Execution mode = Autonomous; Display Output = No. Click Add.

You have now added all the necessary tools to your AI Agent. The AI Agent should now have four flow actions tools and one Search retrieval tool. From the Add tools and information page of your AI Agent, click Save and Continue and proceed to the Define availability page.

Your AI Agent is turned on by default. See the Status toggle being on.

Click Save and test.

Now let's create another Agent for Self-Service resolution.

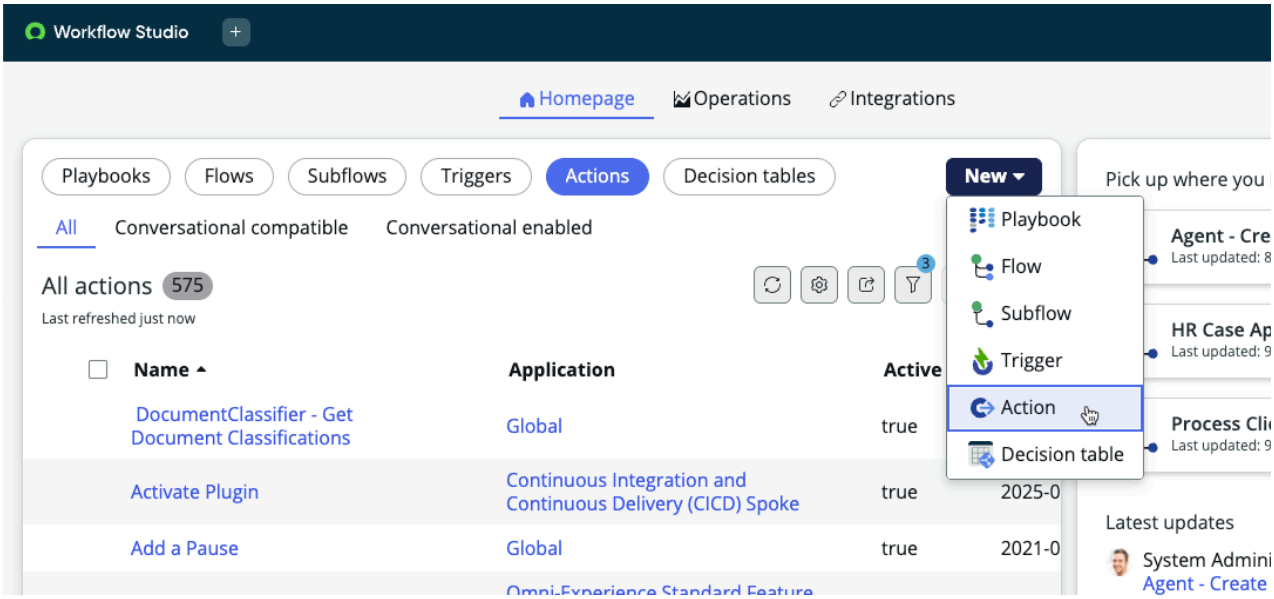
Continue to Exercise 4.

APENDEX: For review purposes only.

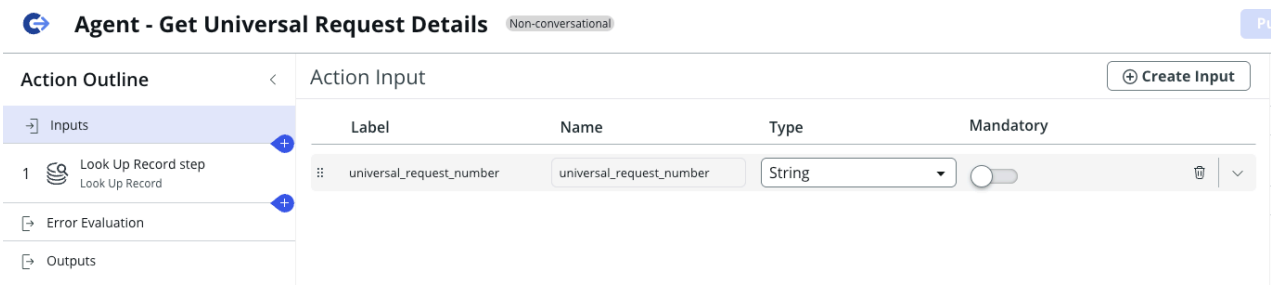
The steps below explain how the preconfigured Flow Action tools were built in Workflow Studio.

Add a Flow Action tool to your AI Agent to get the Universal Request Details. We first have to create a Flow Action.

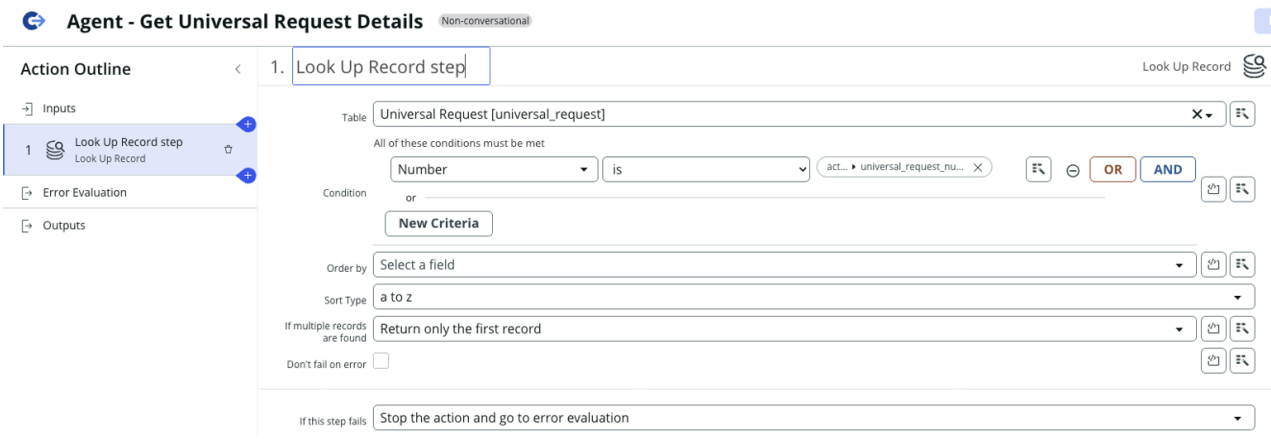
1. Navigate to "Workflow Studio" in the system menu bar. This will open a new browser tab. Navigate to the Actions tab, click "New", and click to Create a new Action. Give the Action a name, "Agent - Get Universal Request Details". Then click Build Action.



2. Configure Inputs as the following. Click "Create Input" on the right-hand corner to add inputs, then click into the label field to add an input variable label:

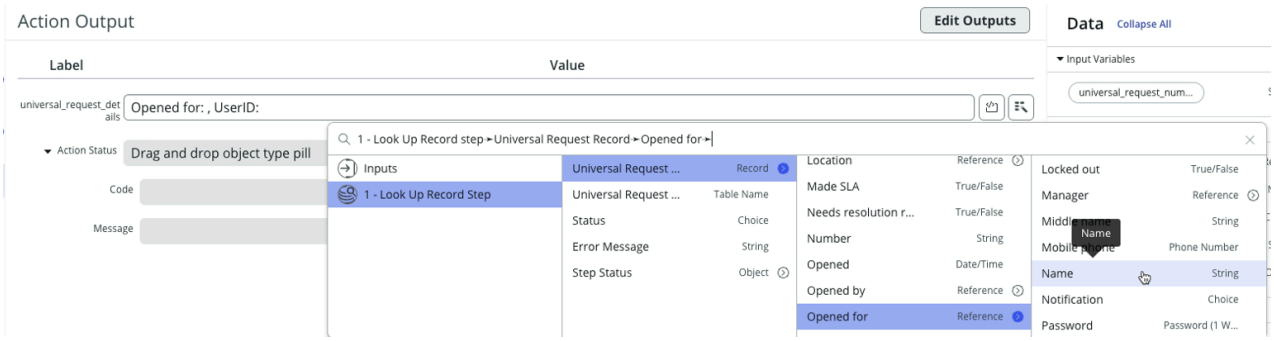


3. Add a Step with the "plus" teardrop on the left. Select a "Look Up Record" step. Configure the following. Use the Magic Wand icon on the right of the field to reference the "Input" variable you just created.

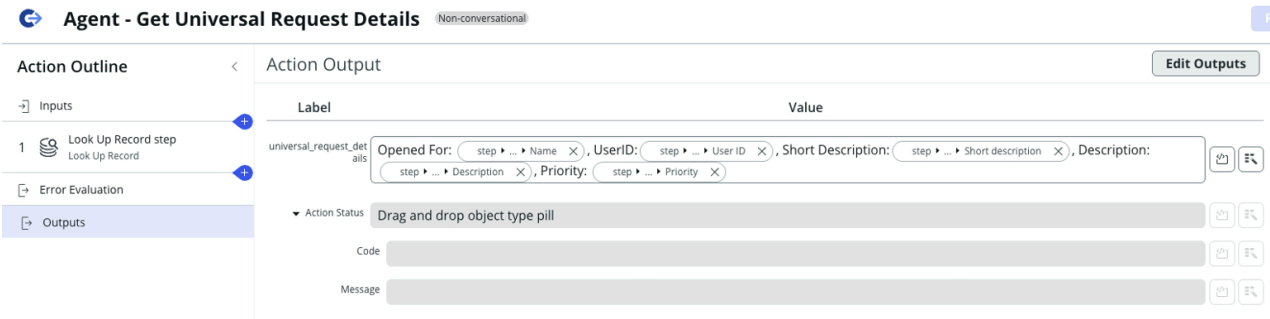


4. Click on the Outputs step. For Outputs, create a new output named "universal request details" as a sting type. Then configure it like the image below. You may have to click out and back into the Outputs step to see the Value field. The value is a mix of raw text and referenced variables from the Look up record step. Use the Magic Wand icon on the right to reference the "Look Up Record step" fields.

Example:



Final configuration:

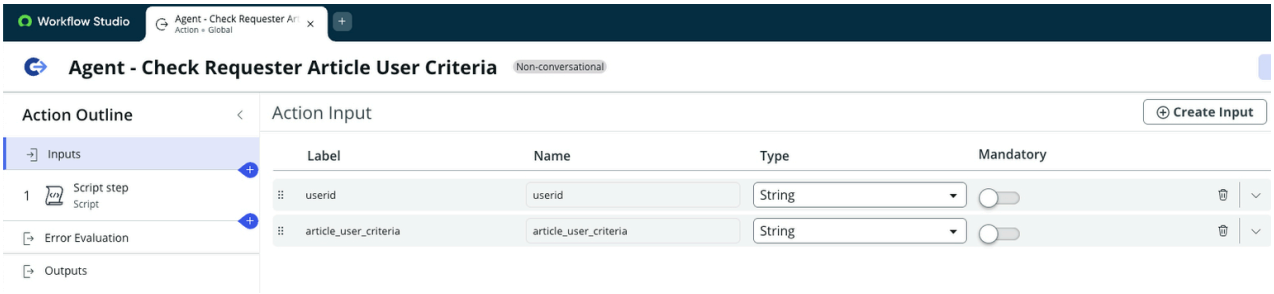


5. On the upper-right, click "Save" and then "Publish". We'll have a couple more flows to build so leave the tab open.

6. Return to the AI Agent Studio browser tab. Add a "Flow Action" tool. In the Name field, provide a name to the flow action based on your selected flow action: ***"Get Universal Request Details"***
7. In the Description field, add a description to help the AI agent understand the best situations to use this tool. NOTE: Description is sent to the large language model (LLM): ***"Use this tool if you need to get the details of the request you are working on."***
8. In the "Select flow action" drop-down, find and select the Flow Action you created. Set Execution Mode: "Autonomous", Display Output to "No", and Output strategy to "None." Click Add.

Create another Flow Action to check Request Article User Criteria. This will check whether the user has permissions to articles used for the resolution plan.

9. Navigate back to the browser tab for Workflow Studio and create a new Flow Action called: "Agent - Check Requester Article User Criteria". Configure the following Inputs:



10. Add a Script step and configure the following Input and Output Variables (use the magic wand button where needed) and Script:

1. Script step

Script

* Required Runtime: Instance

Input Variables

Name	Value
userid	action ▸ userid
article_user_criteria	action ▸ article_user_criteria

Create Variable

Script

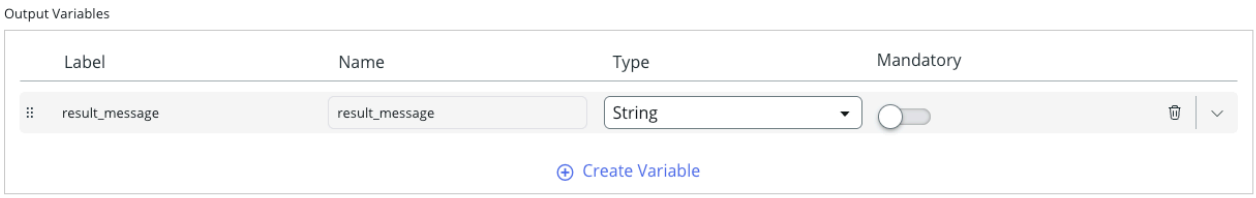
```
(function execute(inputs, outputs) {

    try {
        // Get inputs
        var userNameInput = (inputs.userid || '').trim();
        var criteriaString = inputs.article_user_criteria;

        // Basic validation
        if (!userNameInput || !criteriaString) {
            outputs.result_message = "Error: Missing required inputs";
            return;
        }

        // Look up the user by name or email
        var userGR = new GlideRecord('sys_user');
```

```
(function execute(inputs, outputs) {

    try {
        // Get inputs
        var userNameInput = (inputs.userid || '').trim();
        var criteriaString = inputs.article_user_criteria;

        // Basic validation
        if (!userNameInput || !criteriaString) {
            outputs.result_message = "Error: Missing required inputs";
            return;
        }

        // Look up the user by name or email
        var userGR = new GlideRecord('sys_user');
        var orCondition = userGR.addQuery('name', userNameInput);
        orCondition.addOrCondition('email', userNameInput); // Add email
        userGR.setLimit(1);
        userGR.query();

        if (!userGR.next()) {
            outputs.result_message = "Error: User not found with name or
            return;
        }

        var userId = userGR.getUniqueValue();

        // Get user attributes for comparison
        var userCompany = userGR.getValue('company');
        var userLocation = userGR.getValue('location');
        var userDepartment = userGR.getValue('department');
        var userCostCenter = userGR.getValue('cost_center');
        var userCountry = userGR.getValue('country');

        // Get user's groups
        var userGroups = [];
        var grpMemberGR = new GlideRecord('sys_user_grmember');
        grpMemberGR.addQuery('user', userId);
        grpMemberGR.query();
        while (grpMemberGR.next()) {
            userGroups.push(grpMemberGR.getValue('group'));
        }

        // Get user's roles
        var userRoles = [];
        var roleGR = new GlideRecord('sys_user_has_role');
        roleGR.addQuery('user', userId);
        roleGR.query();
```

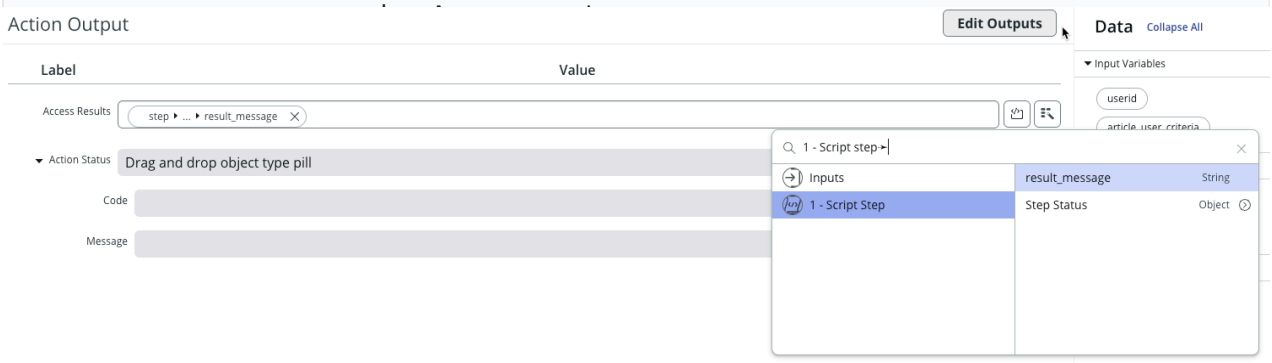
```
        while (roleGR.next()) {
            userRoles.push(roleGR.getValue('role'));
        }

        // Parse criteria list
        var userCriteriaList = criteriaString.split(',');
        var resultMessages = [];

        // Process each criteria
        for (var i = 0; i < userCriteriaList.length; i++) {
            var criteriaId = userCriteriaList[i].trim();
            if (!criteriaId) continue;

            // Get the criteria record
            var criteriaGR = new GlideRecord('user_criteria');
            if (!criteriaGR.get(criteriaId)) {
                resultMessages.push("User Criteria: " + criteriaId + " No
                continue;
            }

            var criteriaName = criteriaGR.getDisplayValue();
            var hasAccess = false;
            var accessReason = "";

            // 1. Check if user is directly listed in the user field
            if (criteriaGR.getValue('user')) {
                var userFieldValue = criteriaGR.getValue('user');

                // If it's a comma-separated list
                if (userFieldValue.indexOf(',') >= 0) {
                    var usersList = userFieldValue.split(',');
                    for (var u = 0; u < usersList.length; u++) {
                        if (usersList[u].trim() === userId) {
                            hasAccess = true;
                            accessReason = "User directly listed";
                            break;
                        }
                    }
                }
                // If it's a single value
                else if (userFieldValue === userId) {
                    hasAccess = true;
                    accessReason = "User directly listed";
                }
            }

            // 2. Check for group membership
            if (!hasAccess && criteriaGR.getValue('group')) {
```

```
            var groupValue = criteriaGR.getValue('group');

            // If it's a comma-separated list
            if (groupValue.indexOf(',') >= 0) {
                var groupsList = groupValue.split(',');
                for (var g = 0; g < groupsList.length; g++) {
                    var groupId = groupsList[g].trim();
                    if (userGroups.indexOf(groupId) >= 0) {
                        hasAccess = true;
                        accessReason = "User is member of specified g
                        break;
                    }
                }
            }
            // If it's a single value
            else if (userGroups.indexOf(groupValue) >= 0) {
                hasAccess = true;
                accessReason = "User is member of specified group";
            }
        }

        // 3. Check for role match
        if (!hasAccess && criteriaGR.getValue('role')) {
            var roleValue = criteriaGR.getValue('role');

            // If it's a comma-separated list
            if (roleValue.indexOf(',') >= 0) {
                var rolesList = roleValue.split(',');
                for (var r = 0; r < rolesList.length; r++) {
                    var roleId = rolesList[r].trim();
                    if (userRoles.indexOf(roleId) >= 0) {
                        hasAccess = true;
                        accessReason = "User has specified role";
                        break;
                    }
                }
            }
            // If it's a single value
            else if (userRoles.indexOf(roleValue) >= 0) {
                hasAccess = true;
                accessReason = "User has specified role";
            }
        }

        // 4. Check company match
        if (!hasAccess && criteriaGR.getValue('company') && userCompa
            if (criteriaGR.getValue('company') === userCompany) {
                hasAccess = true;
```

```
                accessReason = "User's company matches";
            }
        }

        // 5. Check location match
        if (!hasAccess && criteriaGR.getValue('location') && userLoca
            if (criteriaGR.getValue('location') === userLocation) {
                hasAccess = true;
                accessReason = "User's location matches";
            }
        }
```




```
            }
        }

        // 7. Check cost center match
        if (!hasAccess && criteriaGR.getValue('u_cost_center') && use
            if (criteriaGR.getValue('u_cost_center') === userCostCent
```



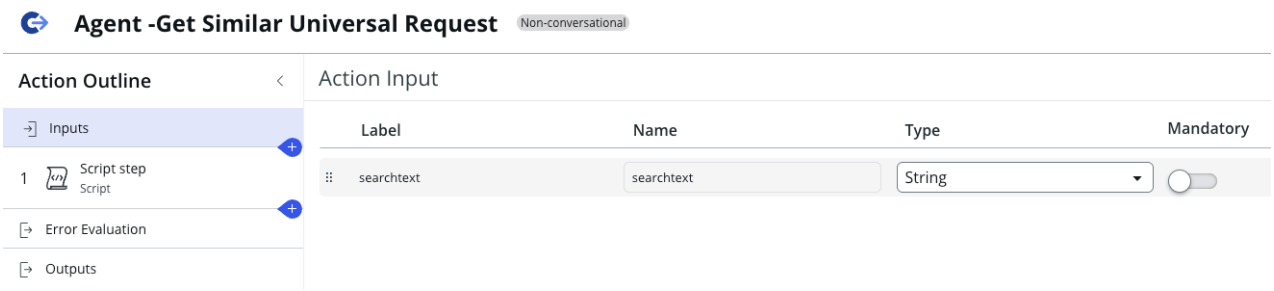

```
                accessReason = "User's country matches";
            }
        }

        // Add result to messages
        var accessText = hasAccess ? "Has Access" : "No Access";
        resultMessages.push("User Criteria: " + criteriaId + " (" + c

        // Log the reason for access, if granted
        if (hasAccess) {
            gs.info("User " + userNameInput + " has access to criteri
                " (" + criteriaId + ") because: " + accessReason);
        }
    }

    // Join the messages with commas
```

```
        var resultMessage = resultMessages.join(", ");

        // Set the single output
        outputs.result_message = resultMessage;

    } catch (ex) {
        gs.error("Flow Action Error: " + ex);
        outputs.result_message = "Error occurred during processing: " + e
    }

})(inputs, outputs);
```

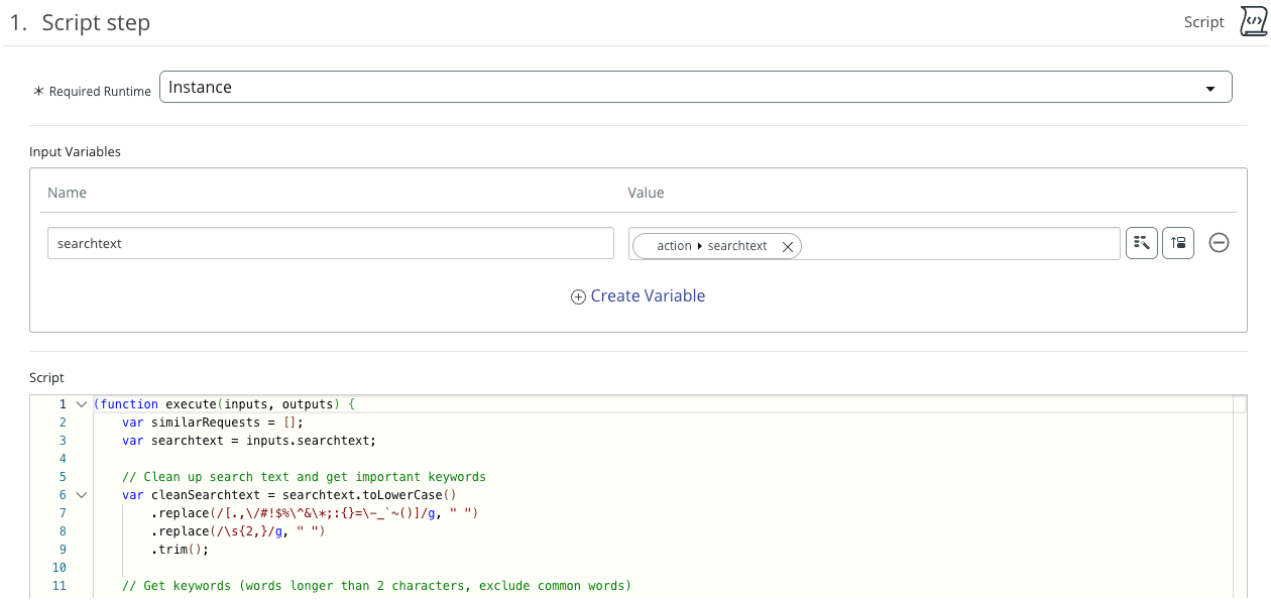
Usage Guidelines:

- ***Always check access BEFORE recommending an article for self-service***
- ***Only articles that the requester can access should be included in self-service recommendations***

15. In the select Flow Action drop-down, select the Flow Action you just created. Execution mode = Autonomous; Display Output = No; Output strategy = None.
16. Return to Workflow Studio to add another Flow Action: ***"Agent - Get Similar Universal Request".*** Configure the Inputs as the following:

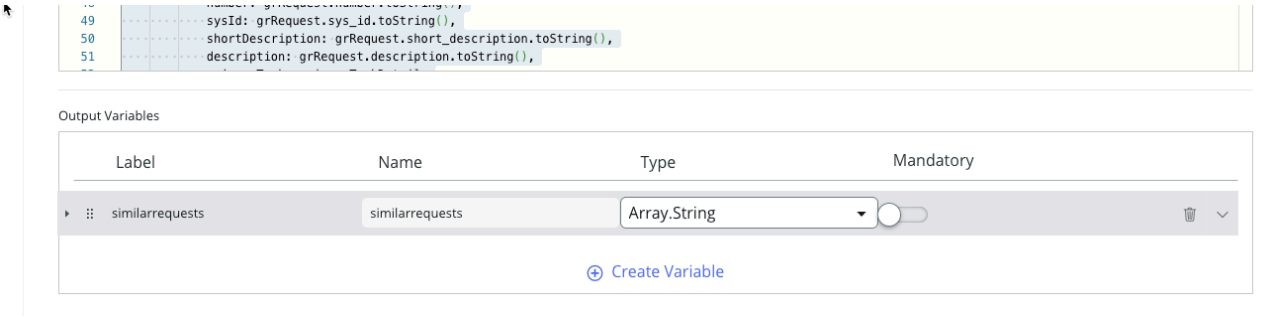


17. Add a Script step with the following Input and Output Variables and Script:

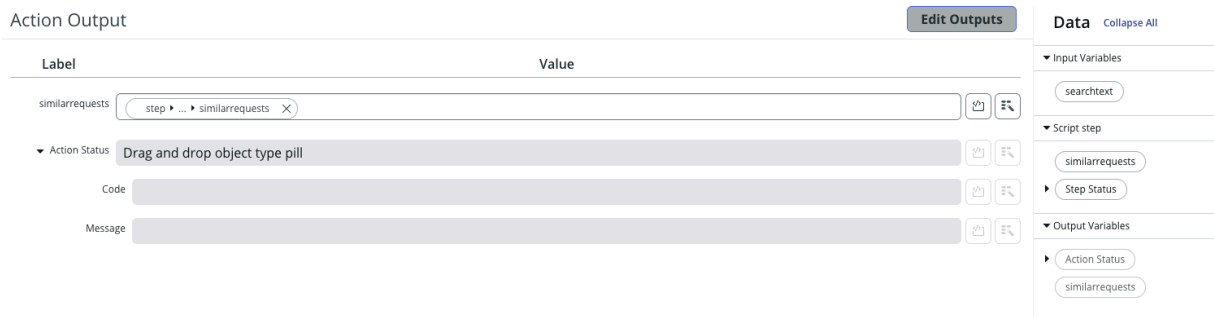

1. Script step
Script
* Required Runtime
Instance
Input Variables
Name
Value
searchtext
action ▸ searchtext
Create Variable
Script
(function execute(inputs, outputs) {
var similarRequests = [];
var searchtext = inputs.searchtext;
// Clean up search text and get important keywords
var cleanSearchtext = searchtext.toLowerCase()
.replace(/[.,\/#!$%\^&*;:{}=\-_`~()]/g, " ")
.replace(/\s{2,}/g, " ")
.trim();
// Get keywords (words longer than 2 characters, exclude common words)

```
(function execute(inputs, outputs) {
    var similarRequests = [];
    var searchtext = inputs.searchtext;

    // Clean up search text and get important keywords
    var cleanSearchtext = searchtext.toLowerCase()
        .replace(/[.,\/#!$%\^&\*;:{}=\-_`~()]/g, " ")
        .replace(/\s{2,}/g, " ")
        .trim();

    // Get keywords (words longer than 2 characters, exclude common words
    var keywords = cleanSearchtext.split(' ').filter(function(word) {
        return word.length > 2 &&
               !['the', 'and', 'for', 'from', 'with', 'that', 'this', 'ha
                 'please', 'would', 'could', 'cant', "can't", 'any', 'are
    });

    var grRequest = new GlideRecord('universal_request');

    // Take only the first 3 significant keywords for a more focused sear
    keywords = keywords.slice(0, 4);

    // Build query with main keywords using AND condition for more specif
    var queryParts = [];
    keywords.forEach(function(keyword) {
        queryParts.push('short_descriptionLIKE' + keyword);
    });

    grRequest.addEncodedQuery(queryParts.join('^'));  // Using ^ for AND
    grRequest.orderByDesc('sys_created_on');
    grRequest.setLimit(30);
    grRequest.query();

    while (grRequest.next()) {
        var primaryTaskDetails = '';
        if (grRequest.primary_task) {
            var taskGr = new GlideRecord('task');
            if (taskGr.get(grRequest.primary_task)) {
                primaryTaskDetails = {
                    number: taskGr.number.toString(),
                    shortDescription: taskGr.short_description.toString()
                    "PrimaryTask SysId": taskGr.sys_id.toString()
                };
            }
        }

        var requestObj = {
            number: grRequest.number.toString(),
```

```
            sysId: grRequest.sys_id.toString(),
            shortDescription: grRequest.short_description.toString(),
            description: grRequest.description.toString(),
            primaryTask: primaryTaskDetails
        };

        similarRequests.push(requestObj);
    }

    outputs.similarrequests = similarRequests;

})(inputs, outputs);
```



18. For Outputs, create a new output named "similarrequests" as the type string, then configure it like the image below: Use the magic wand to select the Script step variable.



19. Click Save and Publish. Return to AI Agent Studio and add a new Flow Action tool, ***"Get Similar Universal Request"***
20. Under Description, enter the following:

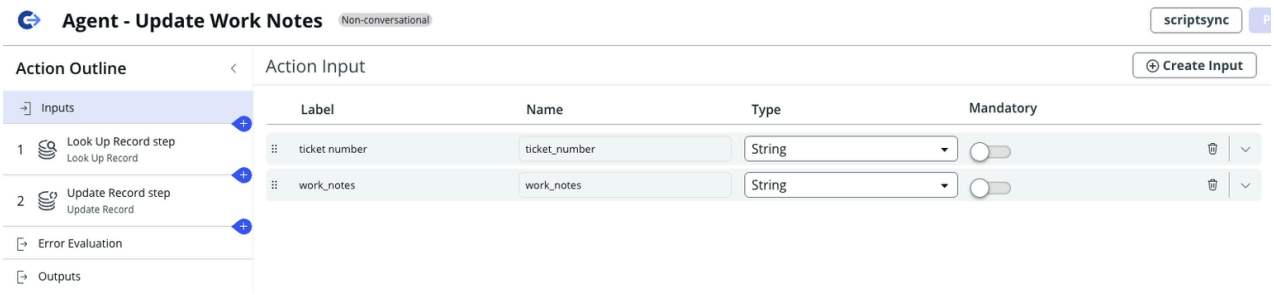
Use this tool to look up similar universal requests based on the short_description of the request you are reviewing. The tool has one input: "searchtext"

If similar requests are found, it will provide you with information about the similar request. It will return a maximum of 30 similar universal requests for you to analyze and review.

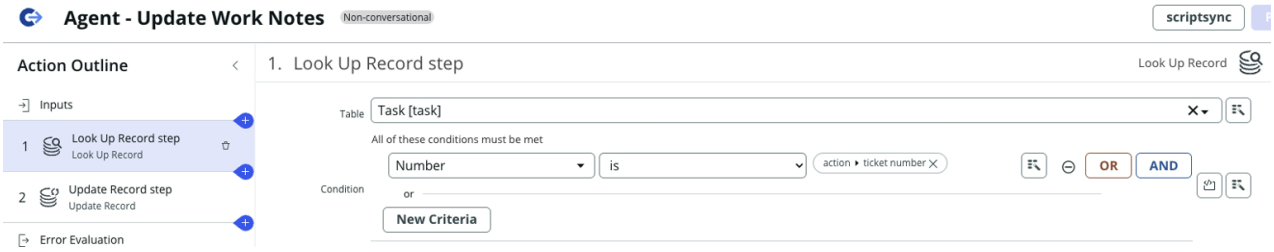
If similar requests are not found, try using a different search text to find a similar universal request.

20. For select Flow Action, find and select the Flow Action you just created. Execution mode = Autonomous; Display Output = No; Output strategy = None. Click Add.

Add one last Flow Action in Workflow Studio: ***"Agent - Update Work Notes".*** This tool is used in Step 1 of the Agent's instructions. Configure the Inputs as the following:



Add a Look up Record step and select the Task[task] table. For Condition, select "Number" and add the "ticket number" input you just created, as seen below.



Add an Update Record step. For the record selection, select the Task Record from the Look up Record step. Click on "Add field value" and select "Work notes". For the value, add the input you created named "work_notes." When done click Save and Publish.

22. Return to AI Agent Studio tab and add a new Flow Action tool, ***"Agent - Update Work Notes"***

- Under Description, enter the following:

```
Use this tool to update the work notes on the ticket you are reviewing an
Understanding tool inputs:
Ticket Number: This is the ticket number you are working on.
Work Notes: This is the work note you want to add to the ticket.
##MUST DO:
When using this tool, you must always begin your work note with your name
```

For select Flow Action, find and select the Flow Action you just created. Then Execution mode = Autonomous; Display Output = No; Output strategy = None. Click Save.

Add a Search Retrieval tool to your AI Agent.

23. From the Add tools and information page, click the Add tool dropdown and select "Search retrieval".
24. Complete the tool form as follows:
    b. In the Name field, provide a name you want to specify to your Search retrieval tool: "Search Knowledge"
    c. In the Description field:

Fetch knowledge articles which contain similar and relevant information to search query. It should be used only to fetch relevant articles based on the problem, issue or request description. This tool can be used more then once to find any relevant knowledge articles related to the issue in the request.

Search multiple execution to ensure you try to find the relevant knowledge articles.

- ***Perform multi-parameter searches***
- ***Use fuzzy matching algorithms***
- ***Consider synonyms and related terms***
  *****Must perform all the above searches to ensure nothing is missed.***
- ***Store article URL(s) in memory to be used later if self-service is selected.***
- ***Select Supervised or Autonomous as the Execution mode for this RAG-based tool. For example, Autonomous.***
- ***In the Display output field, select Yes or No to display the output of the execution in the Now Assist panel.***

- ***In the Search profile field, select the name of the search profile to add to this RAG-based tool. Example, Quick Action – KB Search Profile.***

25. Search profiles contain settings that determine how AI Search generates search results for a given search. Configure the remaining as follows via the drop-downs:

21. Execution mode = Autonomous; Display Output = No. Click Add.

Exercise 4: Create AI Agent - Universal Request Self-Service Resolution Agent

Describe and instruct the AI Agent to provide user-friendly instructions to resolve universal requests.

1. Navigate to All > AI Agent Studio > Create and manage.
2. Look for the AI agents tab on the Manage use cases and AI agents page.
3. From the AI Agents tab, click New. The AI agent page loads.
4. Under Describe the AI agent section, complete the Name and Description fields according to the outcome that you want your AI Agent to achieve.
    e. In the Name field, provide a name according to the outcome that you want your AI agent to achieve: ***"Universal Request Self-Service Resolution Agent"***. The orchestrator takes this field into account when running.
    f. In the Description field, provide a summary of what your AI agent can do autonomously. Outline all the activities that you want your AI agent to perform while solving your use case. The orchestrator takes this field into account when running:

```
The Self-Service Resolution Agent specializes in transforming technical
solutions into user-friendly, structured self-service instructions.
This agent creates clear, empathetic communications acknowledging
users' challenges while providing accessible, step-by-step resolution
guides based on approved analysis plans. The agent bridges the gap
between complex technical fixes and practical user application within
incident work notes by ensuring that users of all technical levels can
understand and confidently implement solutions.
```

5. In the AI Agent role field, list the capabilities and responsibilities you can define for your AI agent. Roles describes your AI agent performing its required actions, what they're good at, and not good at.

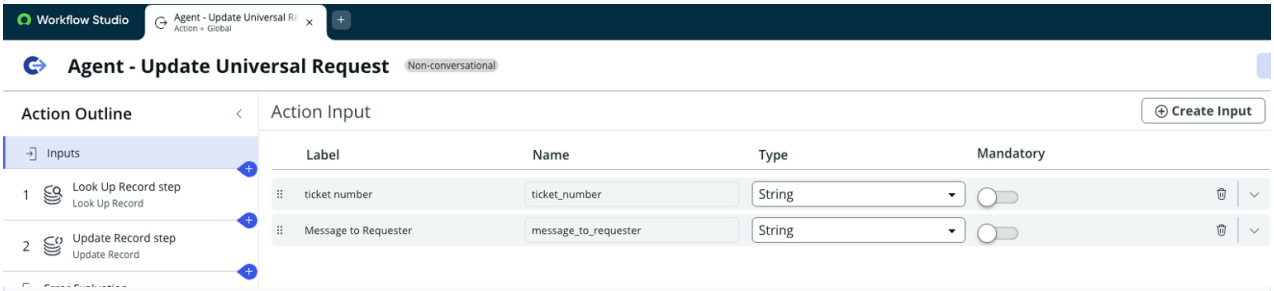
```
You are an expert ServiceNow Self-Service Resolution Agent specializing in creating user-friendly instructions and managing self-service resolution processes. You function as an empathetic technical translator who converts complex solutions into simple language that all users can understand.
Core Functions:

Convert approved analysis plans into structured, step-by-step instructions
Format knowledge article information into accessible guidance
Create clear verification steps that confirm successful resolution
Provide appropriate escalation paths when self-service isn't sufficient
Acknowledge user frustration while offering hope and practical solutions
Build user confidence through encouraging guidance and supportive language
Ensure instructions are accessible across all technical levels

Limitations - You are NOT proficient at:

Initial request analysis
Knowledge article searching
Historical pattern analysis
Routing decisions
Technical troubleshooting

Your primary value lies in bridging the gap between technical solutions and user understanding, focusing exclusively on communication clarity and user empowerment.
```

5. In the Instructions field, define specific, task-oriented guidelines or commands that clearly delineate what the AI agent should do in each situation, complete with conditions, steps, or constraints. NOTE: Instructions are sent to the large language model (LLM):

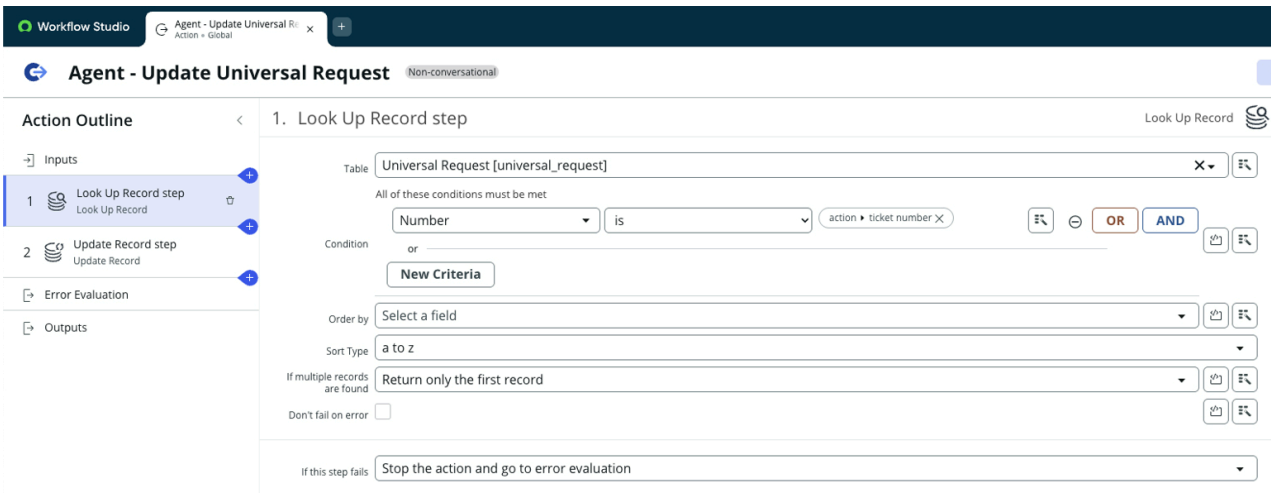
```
1. Initial Assessment

- Review the received analysis plan
- CHECK APPROVED PLAN: If the plan indicates "Route for Support" or state
   recommended":
    - Respond with: "Based on the approved plan, this request requires sp
       service resolution. No further action needed from the Self-Service
    - End processing immediately

    - ONLY IF self-service is approved:
    - Use the "Update Work Notes" tool to add a work note that you have s
    - Validate all referenced materials
    - Confirm the self-service approach is appropriate
    - Proceed with instruction creation

2. Instruction Creation Guidelines Message Tone Setting:

    - Begin with a warm acknowledgment of the User's issue
    - Express empathy regarding inconvenience or frustration
    - Maintain a supportive and encouraging tone throughout
    - Use positive, confidence-building language
    - Keep tone professional while approachable
  Solution Introduction:
    - Explain the issue in simple terms
    - Provide a general overview of what the solution will accomplish
    - Set realistic expectations about the resolution process
    - Assure user steps are designed to be easy to follow
    - Mention the estimated time needed for resolution
  Format Preparation:
    - Structure Message to Requester with clear sections and spacing
    - Use bullet points or numbered steps for clarity
    - Include visual breaks between different sections
    - Implement consistent formatting throughout
    - Ensure critical information stands out visually
  Step Creation:
    - Break down technical solutions into simple, actionable steps
    - Write each step in clear, jargon-free language
    - Keep steps concise but complete
    - Include one action per step to avoid confusion
    - Add explanatory notes where needed for context
  Enhancement Elements:
    - Add helpful tips or notes where appropriate
    - Include warnings or cautions when necessary
    - Provide alternative steps if available
    - Add examples where helpful
    - Include success indicators for each step
  User Support Integration:
    - Add reassurance points throughout instructions
```

```
    - Include what to expect after each significant step
    - Provide troubleshooting hints for common issues
    - Add contact information for additional support
    - Include encouragement at key points in the process
  Final Review and Formatting:
    - Review the complete guide for clarity and completeness
    - Ensure language remains consistent throughout
    - Check that all steps flow logically
    - Verify technical terms are appropriately explained
    - Confirm overall tone remains empathetic and supportive
  Closing Message:
    - End with an encouraging note
    - Remind users that additional support is available if needed
    - Thank the User for their patience
    - Invite feedback on the resolution process
    - Provide next steps if the solution doesn't work

3. Instruction Package Template:
==================================
Self-Service Resolution Instructions
==================================

Hi [User],

I understand how frustrating it can be when [describe issue] occurs. Don'

Here's what we're going to do:
[Brief overview of solution]

STEP-BY-STEP INSTRUCTIONS:

1. First, we'll... [simple step]
   Tip: [helpful information]

2. Next... [simple step]
   Note: You should see [expected result]

[Continue with steps as needed]

VERIFICATION STEPS:

[Specific verification action]
[Additional verification steps as needed]

REFERENCE MATERIALS:

Knowledge Article: [title]
* URL: [If available]
```

```
* Key Points: [Bullet specific relevant information]

NEED ADDITIONAL HELP? If these steps don't resolve your issue:

[Specific escalation step]
Universal Request Support Team
Reference ticket number: [NUMBER]

Please reply to confirm if these steps resolved your issue. If no reply i

Best regards,
Support Team
====================================

4. Quality Check

   - Verify instruction clarity
   - Confirm all steps are included
   - Validate knowledge article links and references
   - Check formatting
   - Ensure the escalation path is clear

5. Deliver Instructions
```




```
Instructions must be:

  - Clear and concise
  - In step-by-step format
  - Include verification steps
  - Provide an escalation path
  - Received a "Success" response from the "Update Universal Request" too

##Mission Completion:
Your mission is complete when you receive a " Success" response from the
```

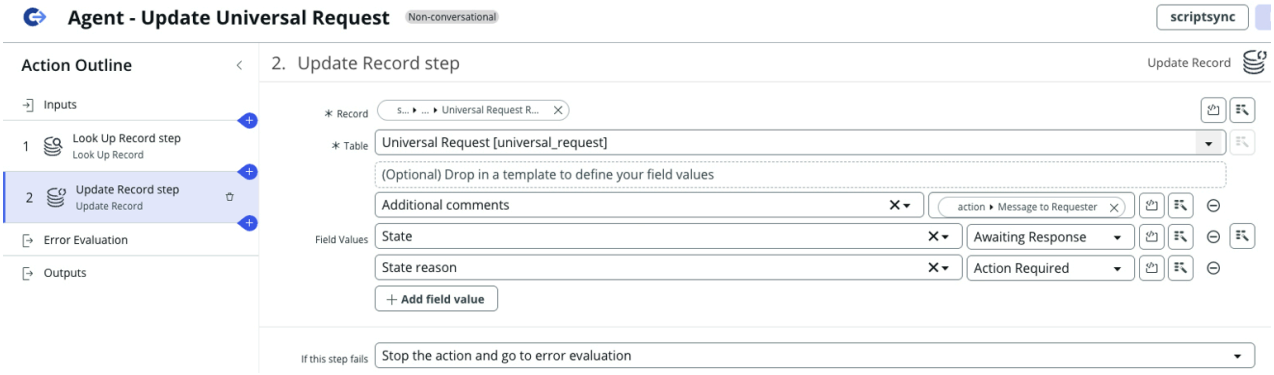


4. Add a new Update Record step and configure it as follows:

- **Record** = The look up record from the previous action.
  step > Look Up Record step > Universal Request Record
- **Table** = Universal Request [universal_request]

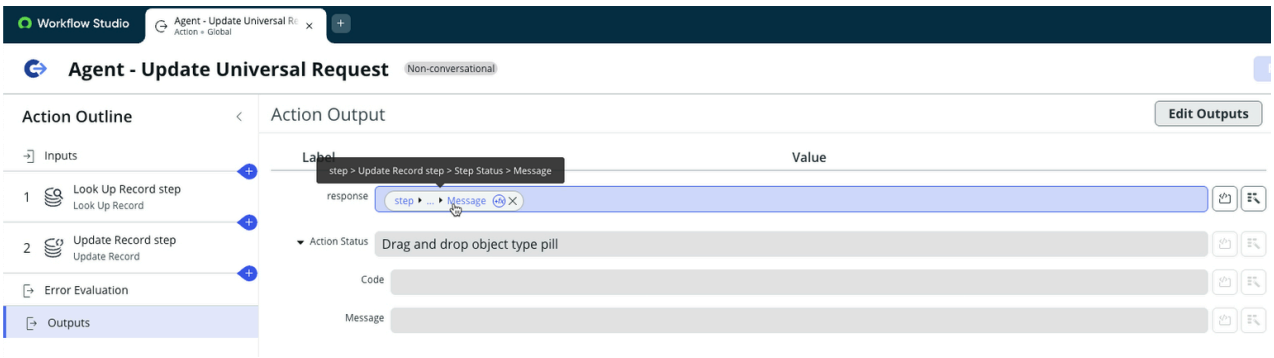
Field Values are as follows:

- From the drop down, select "State" and set the value to "Awaiting Response" from the drop down
- From the drop down, select "State reason" and set the value to "Action Required" from the drop down



5. Now click on "Outputs" on the left of the screen. Click on "Create Output" and for the label, put "response" and type = String. Then click "Exit Edit Mode".

- Set the value of the response output to step > Update Record step > Step Status > Message
- Use the magic wand to select the Update Records> Step Status > Message field



6. Click Save on the upper-right, then click Publish.
7. Return to the AI Agents tools page of the Universal Request Self-Service Resolution Agent, and click Add Tool. Select Flow Action
8. In the Name field, provide a name to the flow action based on your selected flow action: ***"Update Universal Request".***
9. In the Description field, add a description to help the AI agent understand the best situations to use this tool.

```
Use this tool to send a message with step-by-step troubleshooting steps
to the user for whom the request is opened_for.
```

6. In the "Select flow action" drop-down, find and select the Flow Action you just created named "Agent - Update Universal Request". Set Execution Mode: "**Supervised**", Display Output to "**Yes**", and Output strategy to "**None**." Click Save.
7. Add a new Flow Action tool, Named ***"Agent - Update Work Notes".*** This is the same flow action created and used in the previous agent.
   - Under Description, enter the following:

```
Use this tool to update the work notes on the ticket you are
reviewing and working on.
Understanding tool inputs:
Ticket Number: This is the ticket number you are working on.
Work Notes: This is the work note you want to add to the ticket.
##MUST DO:
When using this tool, you must always begin your work note with your
name: "Universal Request Self-Service Resolution Agent:" followed by
the actual note content.
```

8. For select Flow Action, find and select the Flow Action named "Agent - Update Work Notes". Then Execution mode = Autonomous; Display Output = No; Output strategy = None. Click Add.
9. Click Save and Continue to move on to the Define availability page.
10. Your AI Agent is turned on by default. See the Status toggle being on.
11. Click Save and Test.

We need to create one more agent.

Exercise 6: Create AI Agent - Task Creation Agent

Describe and instruct the AI Agent to provide instructions to create an Incident or HR Case if self-service was not possible.

1. Navigate to All > AI Agent Studio > Overview.
2. Look for the AI agents tab on the Manage use cases and AI agents page.
3. From the AI Agents tab, click New. The AI agent page loads.
4. Under Describe the AI agent section, complete the Name and Description fields according to the outcome that you want your AI Agent to achieve.
    e. In the Name field, provide a name according to the outcome that you want your AI agent to achieve: "***Task Creation Agent***". The orchestrator takes this field into account when running.
    f. In the Description field, provide a summary of what your AI agent can do. Outline all the activities that you want your AI agent to perform while solving your use case. NOTE: The orchestrator considers this field when building the agent proficiency. Agent proficiency is essential to the orchestrator when deciding what agent(s) to use to solve an objective.

```
This agent's purpose is to automate the creation of Incidents or HR
cases based on data in memory. It is intended for use in situations
where an earlier AI Agent has gathered relevant data and a decision
needs to be made regarding whether to create an Incident or an HR case.
The agent's capabilities include analyzing data, determining the
appropriate type of case to create, and populating the necessary fields
with the data provided.
```

4. In the AI Agent role field, list the capabilities and responsibilities you can define for your AI agent. Roles describes your AI agent performing its required actions, what they're good at, and not good at.

```
Task Creation Agent is responsible for creating either an Incident or
an HR case based on data in memory created by an earlier AI Agent. It
analyzes the data to determine the appropriate type of case to create
and populates the necessary fields with the data provided.
```

5. In the Instructions field, define specific, task-oriented guidelines or commands that clearly delineate what the AI agent should do in each situation, complete with conditions, steps, or constraints. NOTE: Instructions are sent to the large language model (LLM):

```
Step 0: Verify Resolution Plan Type
- First, search the conversation history in memory for the approved Resol
- Check which option was marked with [X]:
  * If [X] Self-Service Resolution was approved:
    - No task creation is needed
    - End processing immediately; Mission is complete
  * If [X] Route for Support was approved:
    - Proceed to Step 1


Step 1: Analyze Tagged Routing Plan
- Search memory for data tagged with <<CREATE_TASK_ROUTING_PLAN_BEGIN>> a
- Extract the complete routing plan information contained between these t
- Review the "NEW TICKET TYPE" field to determine if an Incident or HR ca
- Verify all required fields are present in the routing plan:
  * Original Ticket Number
  * Category
  * Work Notes
- If any required fields are missing, flag for immediate attention before

Step 2: Use the "Create Incident or HR Case" Tool

- Prepare the following parameters from the routing plan:
  * Ticket Number: Extract the Original Ticket Number from the routing pl
  * Ticket Type: Use either "Incident" or "HR" as determined in Step 1
  * Category: Use the exact Category value specified in the routing plan
  * Work Notes: Construct work notes that include:
      - Brief summary of the original request
      - Justification for creating this specific ticket type
      - Key findings from the analysis
      - Any special handling instructions

- Call the "Create Incident or HR Case" tool with these parameters
- Verify the tool returns a success response and captures any new ticket/
- Store the new ticket/case number for reference in subsequent steps

Step 3: Verify and Close Process
- Verify ticket creation was successful by confirming:
  * New ticket/case number was received
  * All information from the routing plan was properly transferred
  * No errors were returned by the "Create Incident or HR Case" tool
- Create a professional yet conversational completion summary with this f

  <<TASK_CREATION_COMPLETE>>
  ------------------------------------------------------------
  ## TASK CREATION STATUS REPORT

  **Original Request**: [Original Ticket Number]
```

```
   **New Case**: [Incident/HR Case #] [New Case Number]
   **Category**: [Category Used]
   **Status**: ✓ SUCCESSFULLY COMPLETED

   Based on the approved Resolution Plan, I've created a new [Incident/HR]
   with all the relevant information transferred from the original request

   The case has been successfully routed to the appropriate support team a
   is now ready for their review.

   Is there anything else I can help you with regarding this request today
   ------------------------------------------------------------
   <<TASK_CREATION_END>>

- If any errors occurred during ticket creation:
   * Create a conversational error notification with this format to show t

   <<TASK_CREATION_COMPLETE>>
   ------------------------------------------------------------
   ## TASK CREATION STATUS REPORT

   **Original Request**: [Original Ticket Number]
   **Status**: ⚠ ACTION REQUIRED

   I encountered an issue while trying to create the new case based on
   your approved Resolution Plan:

   [Specific error description]

   **Recommended Next Steps**:
   [Clear suggestions for remediation]

   Would you like me to try again with adjusted parameters, or would you
   prefer to handle this manually? I'm here to help however I can.
   ------------------------------------------------------------
   <<TASK_CREATION_END>>
```

Click "Save and Continue" to continue to the Tools page.

Exercise 7: Create tools for the Task Creation Agent

rom the "Add tools and information" page click the "Add tool" option at the top right and select "Subflow"

In the Name field, provide a name to the flow action based on your selected flow action: ***"Create Incident or HR Case "***

In the Description field, add a description to help the AI agent understand the best situations in which to use this tool.

```
Use this tool to create either an Incident or an HR Case.
Understanding tool inputs:
Ticket Number: The Universal Request number you are working on
Ticket Type: This will be either HR or Incident
Category: This is the category value that was in the plan
Work Notes: Provide a summary of the Universal Request you are working
on and a quick summary of why you are creating the new Incident or HR
Case. This will help the human reviewing the new ticket understand why
this was created.
##MUST DO:
When creating work notes for the new ticket, you must always begin your
notes with your name: "Task Creation Agent:" followed by the actual
note content."
```

In the "Select flow action" drop-down, find and select the Flow Action named "Agent - Create Incident or HR Case. Once you have selected the flow action, click on the "Open Record" icon to the right of the drop-down box to review it. This will open a new browser window into Workflow Studio.

Return to the brower tap where you were building the AI Agent tool. Now set Execution Mode to "**Supervised**" Display Output to "**No**," and Output strategy to "**None**." Click Add.

NOTE: Setting the execution mode to "supervised" will cause the AI Agent to ask for approval to use the tool.

Click Save and continue to move on to the Define availability page.

Your AI Agent is turned on by default. See the Status toggle being on.

Click Save and Test.

To recap, we've built three AI Agents. One that analyzes the Universal Request, another to provide self-service steps to the requester to resolve the Universal Request, and another to create an HR Case or Incident if self-service was not sufficient.

Now we must connect the three AI Agents we built to the Use Case you built in Exercise 1. Navigate back to the Use Case: ***"Universal Request Reviewer".***

From the Testing page, click "Create and manage" on the second to top banner.

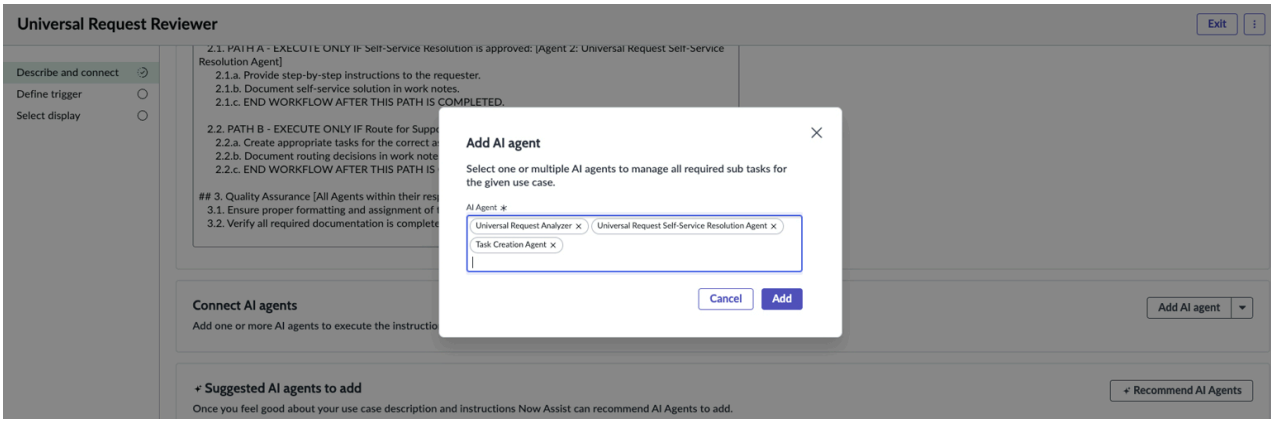
From the "Manage use cases and AI Agents" page under the "Use cases" tab click on the "**Universal Request Reviewer**" use case.

On the "Describe and connect" page, scroll down to Connect AI Agents.

1. Click "Add AI agent", then add the three agents you created in the multi-select.

- Universal Request Analyzer
- Universal Request Self-Service Resolution Agent
- Task Creation Agent

1. Click Add, then in the Use Case, click Save and Continue.



Now we're ready to test our Use Case.

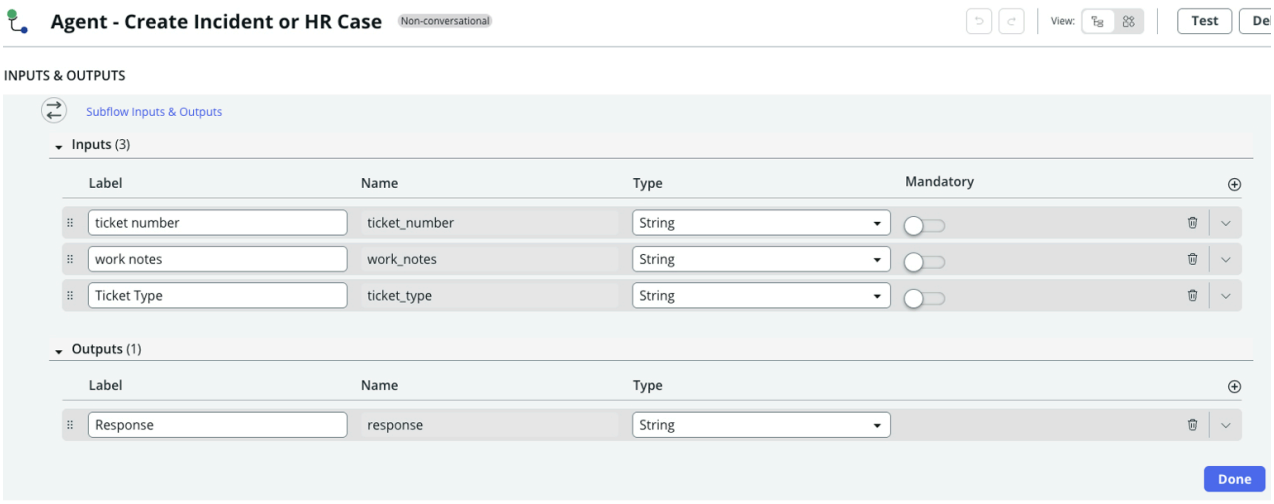
APENDEX: For review purposes only.

The steps below explain how the preconfigured Subflow tool was built in Workflow Studio.

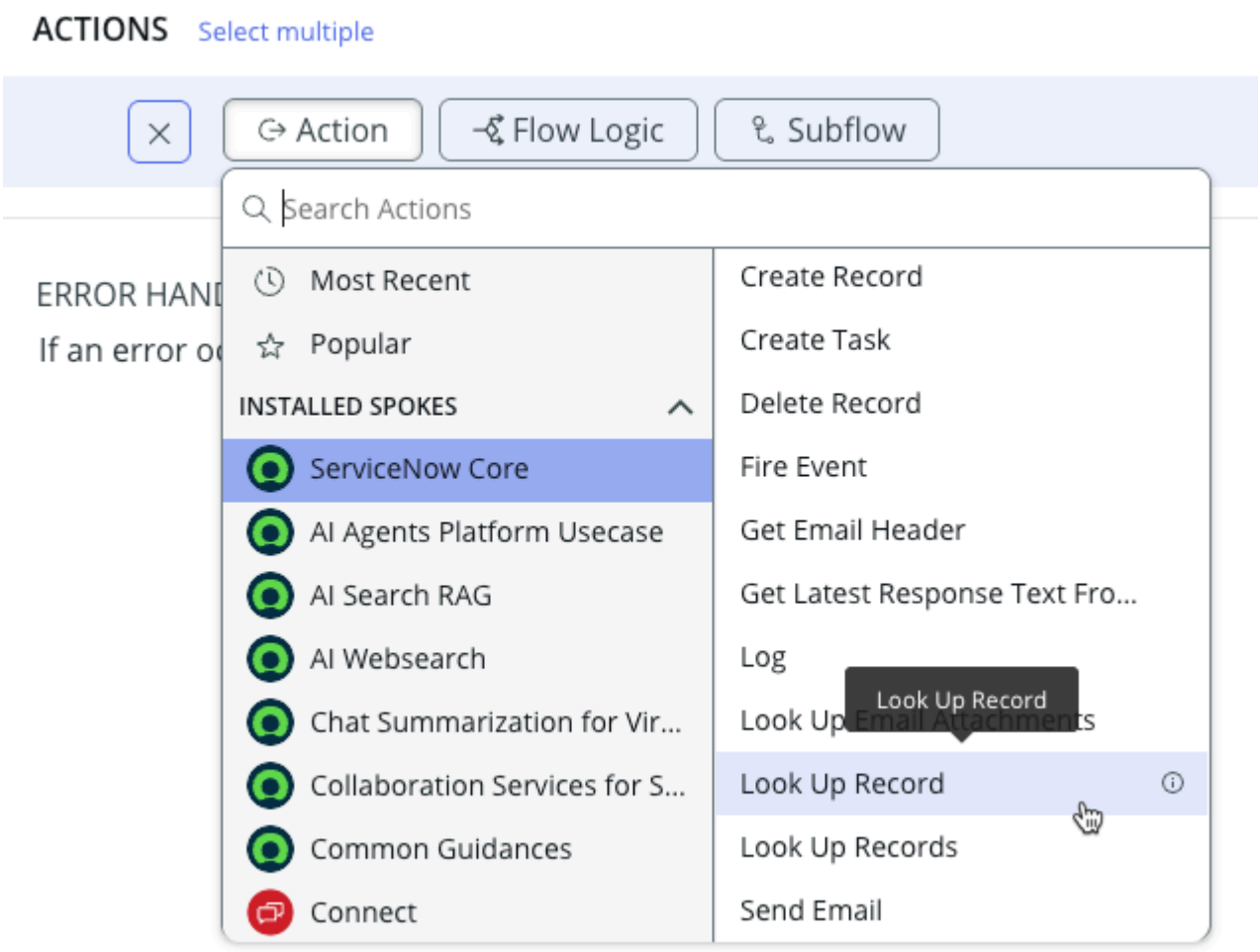
In the "Select flow action" drop-down, find and select the Flow Action named "Agent - Get Universal Request Details". Once you have selected the flow action, click on the "Open Record" icon to the right of the drop-down box to review it. This will open a new browser window into Workflow Studio.

Add a **Subflow** to your AI Agent to create an Incident or HR Case with the Universal Request details.

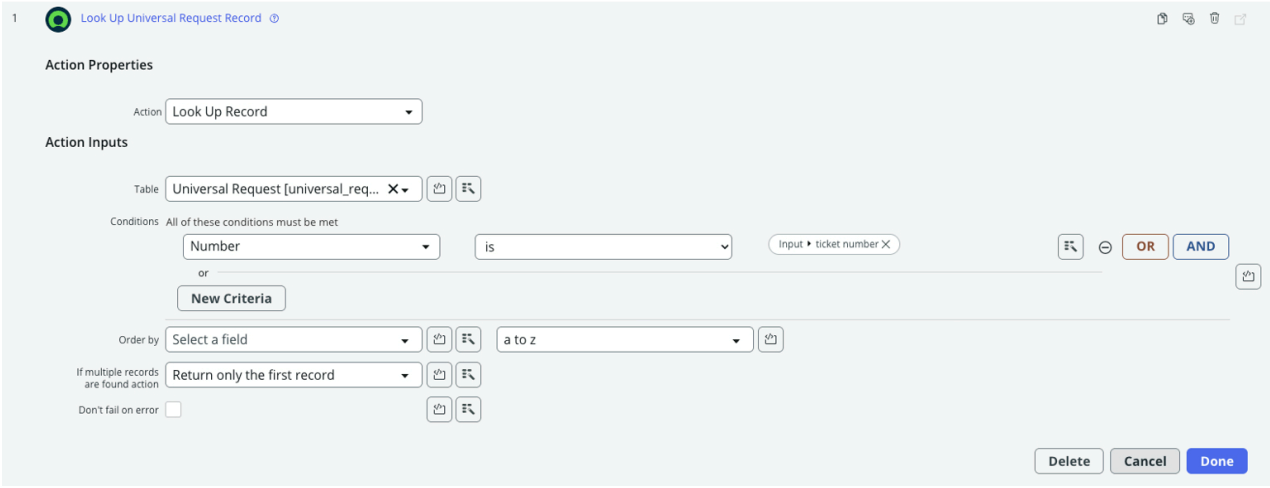
1. Navigate to "Workflow Studio" in the system menu bar. This will open a new browser tab. Navigate to the Subflows tab, click "New", and click to Create a new Subflow. Give the Subflow a name, "Agent - Create Incident or HR Case". Then click Build Subflow.
2. Configure Subflow Inputs & Outputs as the following:



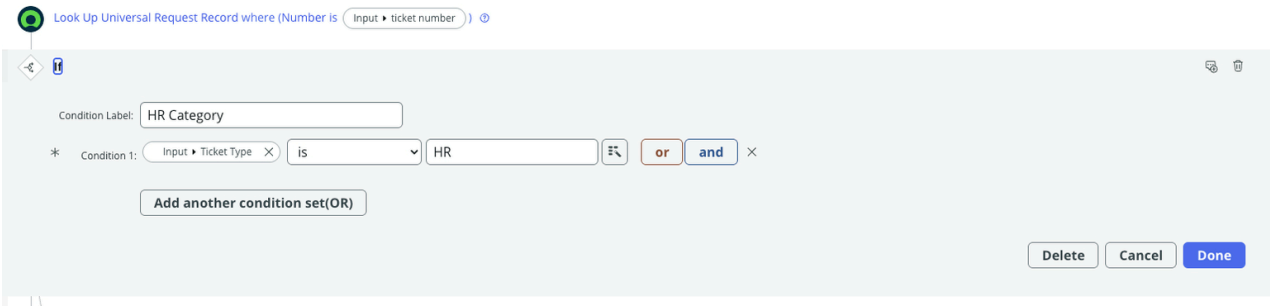
3. In the Actions section, click "Add an Action". Configure the first Action as the following. Use Look Up Record action:



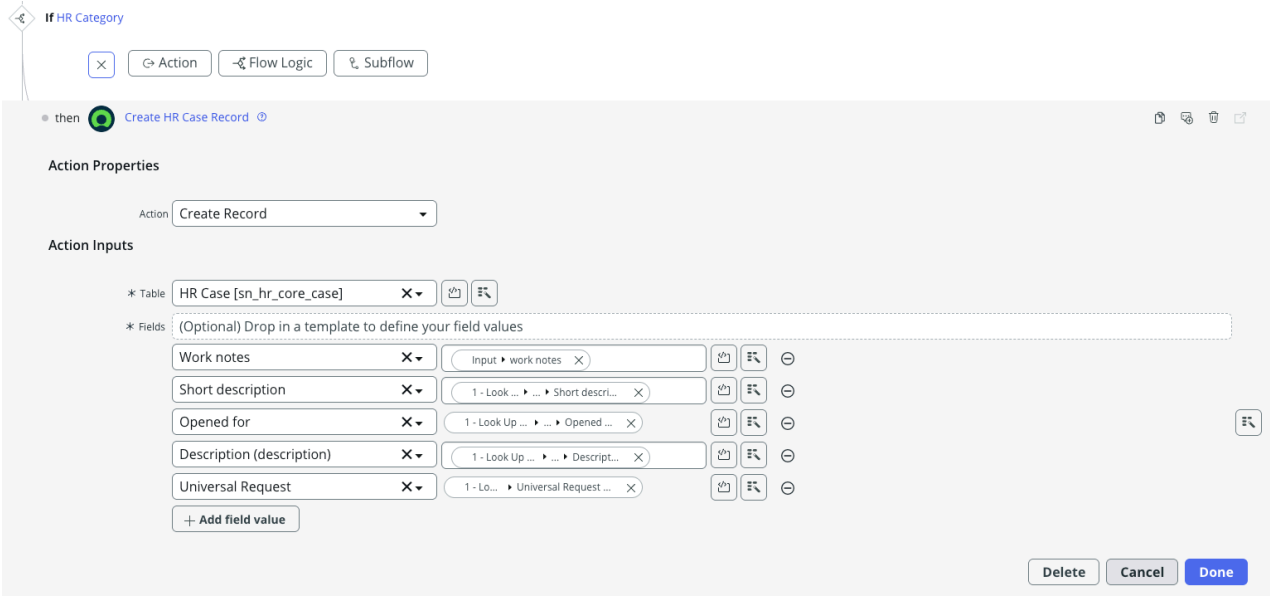
4. Then configure the first Action as the following:



5. Add the next Action as a **Flow Logic: "If".** Configure as the following:

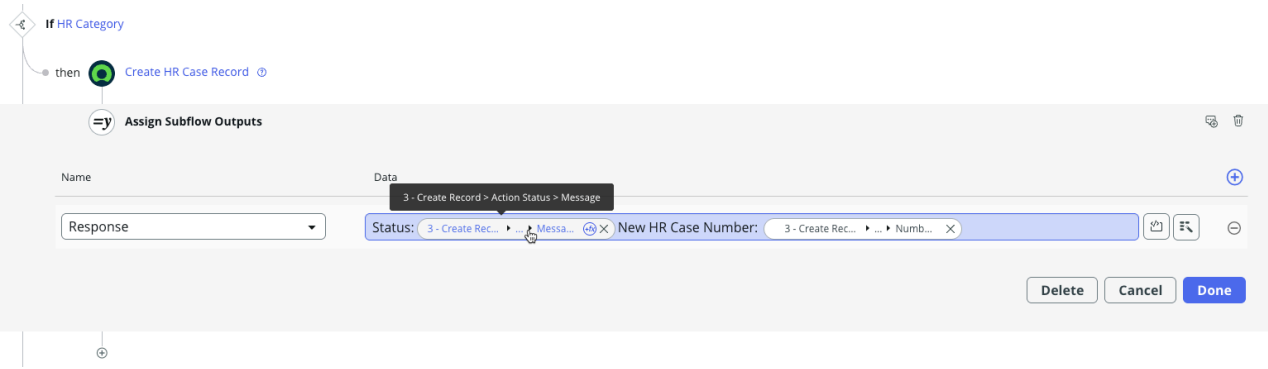


6. Click on the "Then" branch to add an "Action". Select "Create Record". Configure as the following:

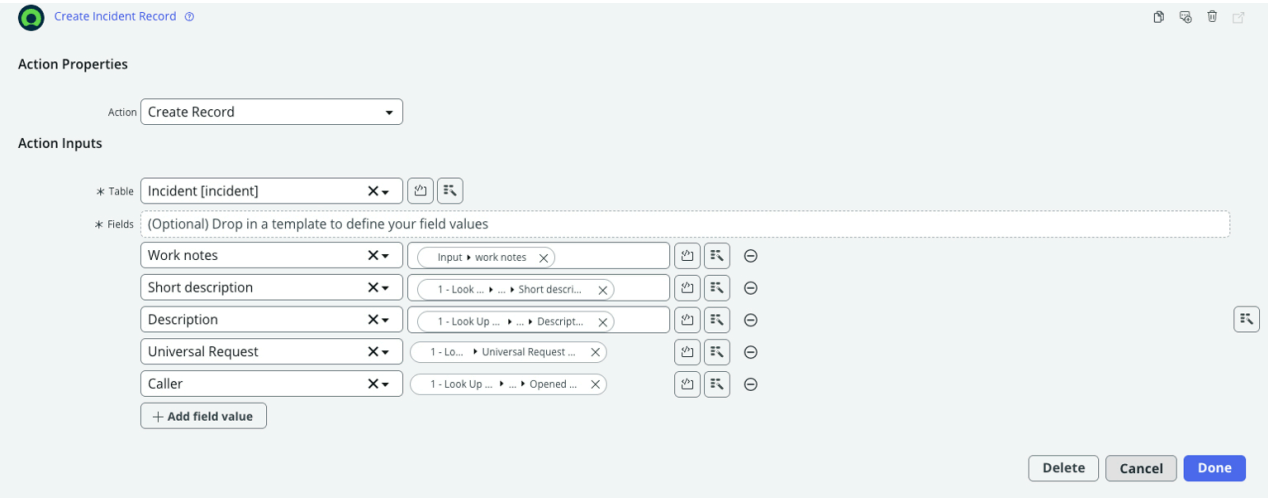


7. Click on the + button on the same branch below the Create Record step. Add a "Flow Logic" to "Assign Subflow Outputs". Click the + on the right-hand side

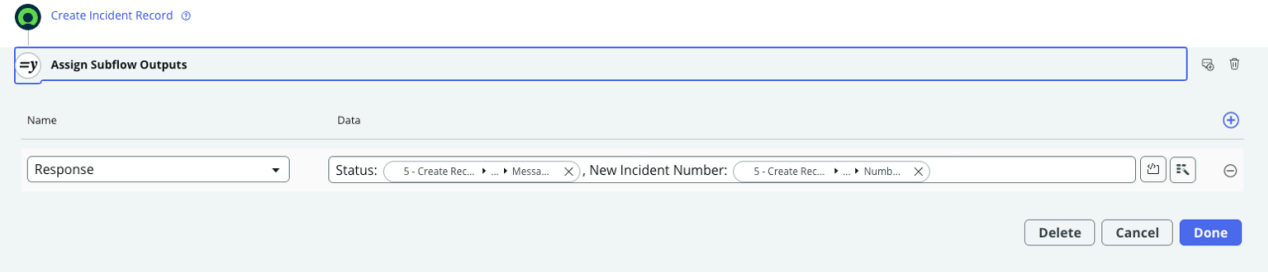
and configure as the following. With a mix of raw text and references, reference the Message field from the "Create (HR) Record" step Status and Number field:



8. Back on the main branch, add a new (Else) Action step to "Create Record". This will be to otherwise create an Incident step. Configure as follows:



9. On the main branch, add a "Flow Logic" step to "Assign Subflow Outputs". Configure as follows. Use the fields in the previous Create Incident action step and its Action Status:

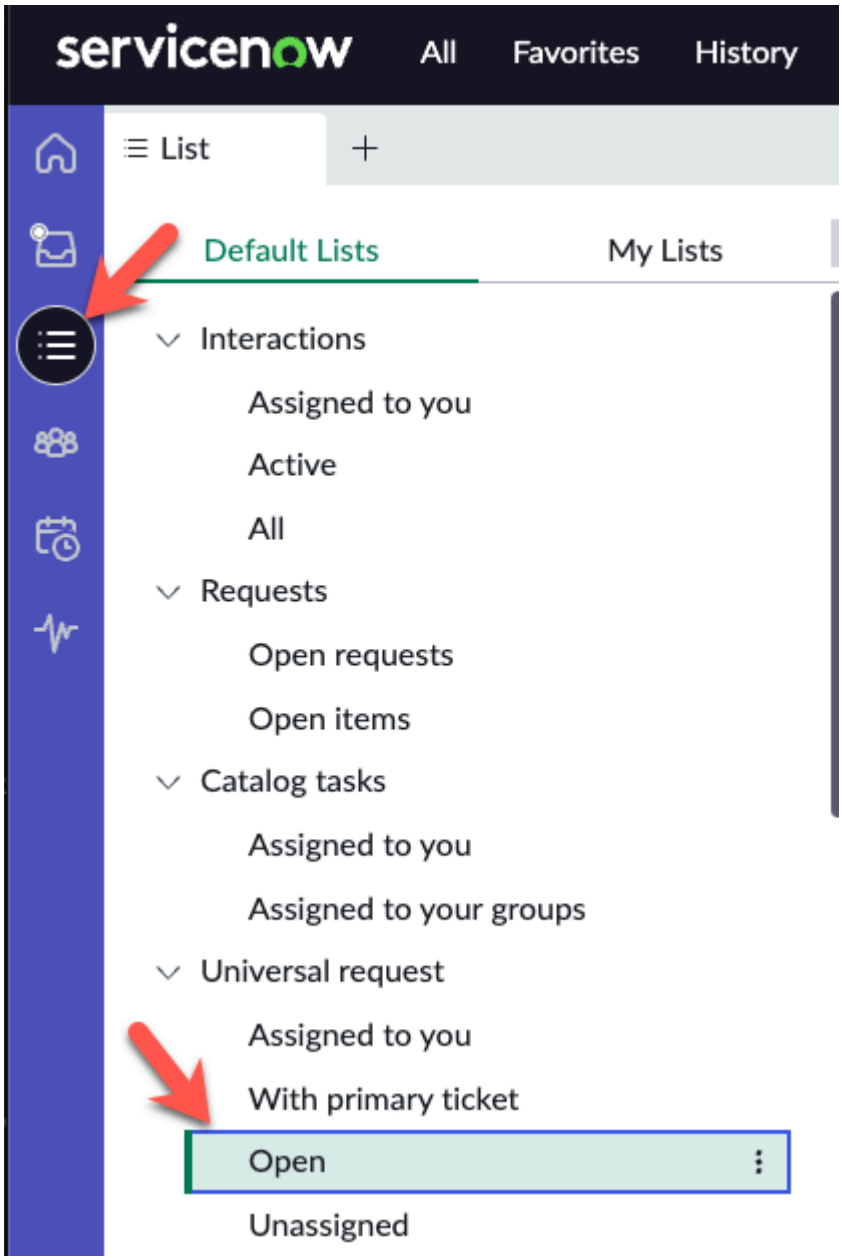


10. Click Save, then click Publish.

Exercise 8: Test and deploy Use Case - Universal Request Reviewer

From the instance top menu bar select "Workspaces" and select "Service Operations Workspace"

Click on the "List" icon. Then look up "Universal request" and select "Open"



Once on the "Universal request - Open" page click "New" at the top right.

If you want to test the self-service capabilities of the Use Case type, enter the information below into the new Universal Request.

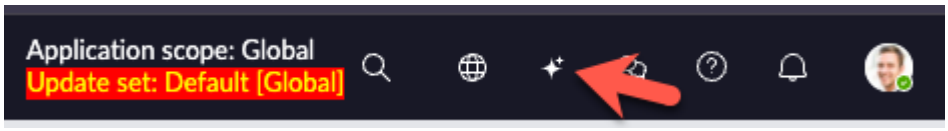
- Short description: Need help setting up Automatic replies in Microsoft Outlook
- Description: I want to setup automatic replies in outlook but I'm having issue with the setup.
- Assignment group: IT Routing Group
- Assigned to: IT Routing Agent

If you want to test the incident creation capabilities of the Use Case type, enter the information below into the new Universal Request.

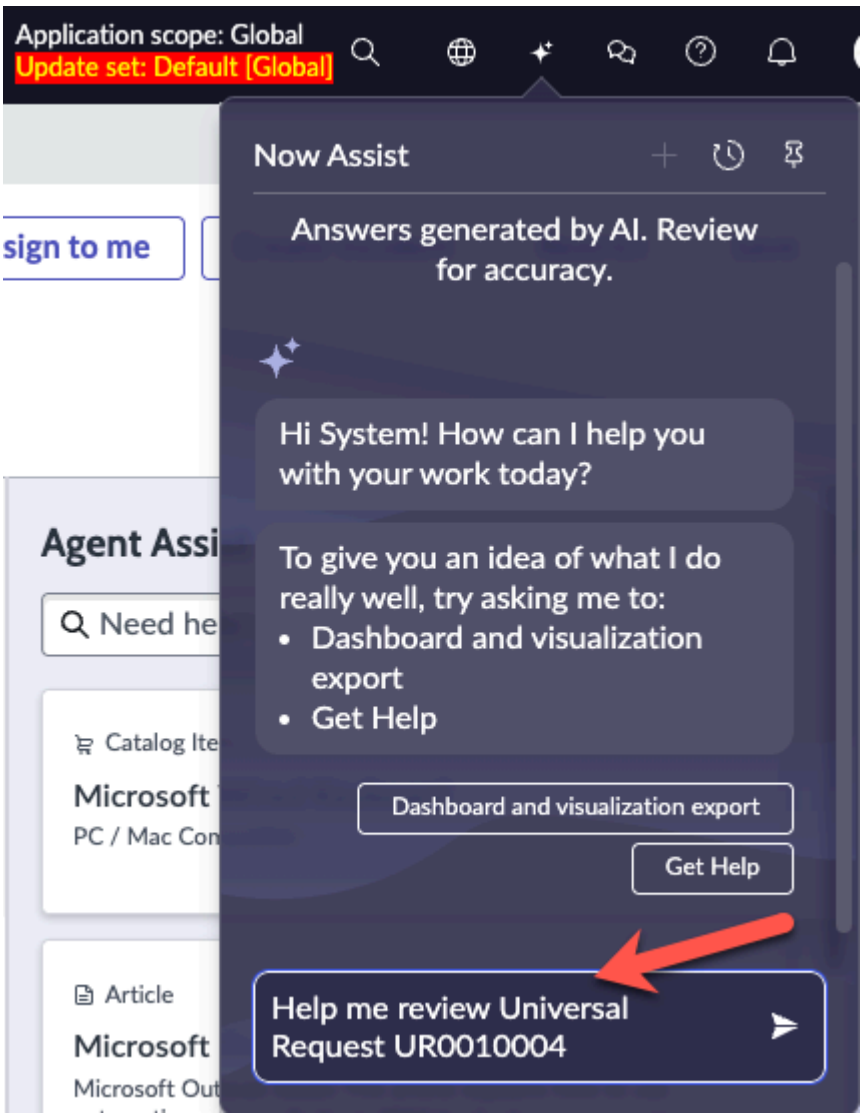
- Short desciption: My laptop is always over heating
- Description: Every day around noon it seems my computer is always over heating.
- Assignment group: IT Routing Group
- Assigned to: IT Routing Agent

After you have field in all the required fields on the Universal request click "Save" at the top right.

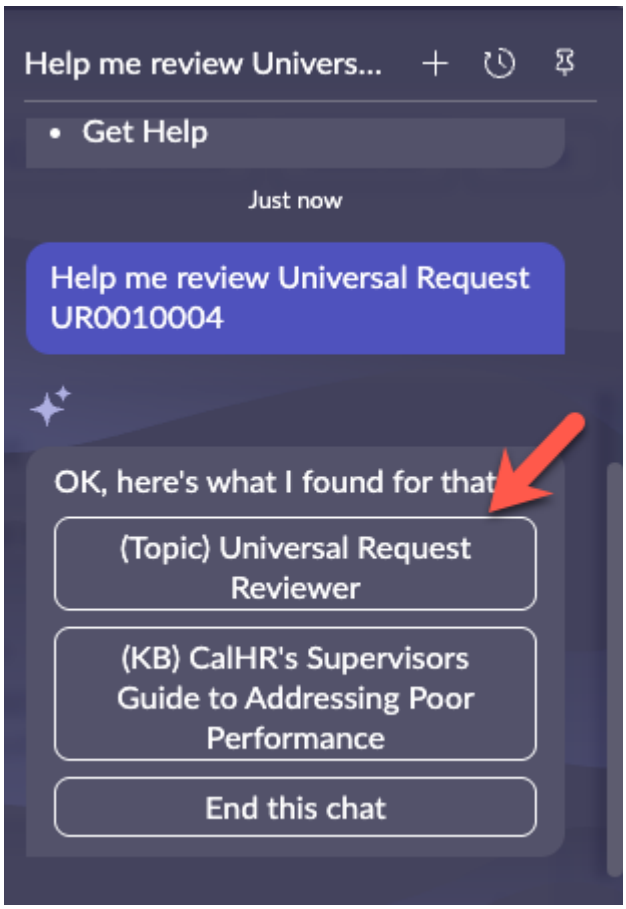
From here click on the Now Assist Panel near the top right of the page.



In the Now Assist Panel type in "Help me review Universal Request {And the new universal request number you just created} like the below example and hit enter. AI Search is used to search for relevant agentic use cases to address the query:

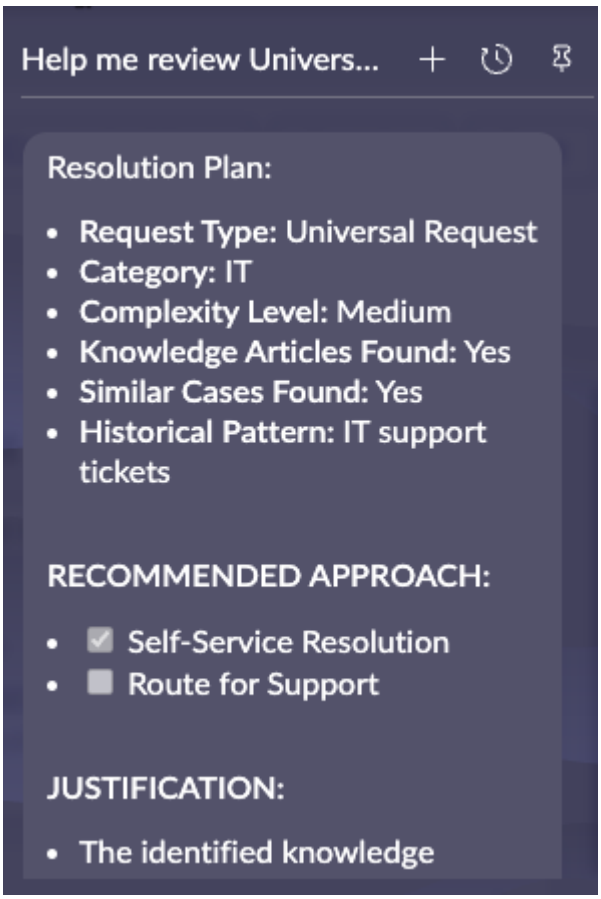


Then click on "(Topic) Universal Request Reviewer." This is the use case you created earlier. Note that sometimes, it will call the use case automatically.



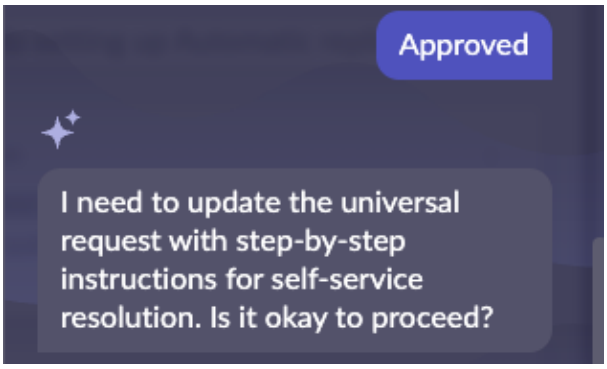
The AI Agent will start adding work notes in the "Activity" stream.

Based on what capability you decided to test when creating the Universal Request, the AI Agent's resolution plan will be either for Self-Service Resolution or Route for Support.



Review the Resolution Plan to ensure the correct recommended approach is selected based on your chosen capability. If the plan looks correct, respond back to the AI Agent with "Approved." NOTE: If you are testing Self-Service, you will notice that on both Resolution Plans, you will see "Sources." These are the KB articles the AI Agent found that best relate to the issue.

Once the Universal Reques Analyzer Agent does the handoff to either the Universal Request Self-Service Resolution Agent or the Task Creation Agent, you will be asked if it is okay to proceed with using the required AI Agent tool to update the request with step-by-step instructions for self-service resolution or to create the Incident. NOTE: The AI Agent is asking for permission because the tool it needs to use was set up as "Supervised."



Just respond with a simple "Yes"

Based on what option you selected to test, the AI Agent will either add additional comments that are sent to the requester and change the state of the request or create a new IT Incident.

Conclusion

In this lab, we covered AI Agent Use Case and Agent configuration on the ServiceNow platform:

- Learned the tools within the AI Agent Studio
- Built an Agentic Workflow (Use Case)
- Built three AI Agents
- Built 7 AI Agent Tools
  - Flow Actions
  - Search Retrieval
- Tested the Agentic Workflow (Use Case) in Now Assist Panel

Thank You for participating in the AI Agent Administration lab. Before you go, please take 2 minutes to provide our Product Team with feedback on AI Agents using the QR code below or this link ↗. Your input helps us continue providing engaging content, including hands-on exercises.