

Automated Network Request Management in ServiceNow

Data Handling

1. Introduction

Data handling in the Automated Network Request Management system ensures that user-submitted catalog variables are accurately captured, processed, and stored in a structured format. This phase focuses on mapping Service Catalog variables to a custom backend table using ServiceNow Process Automation and Flow Designer.

2. Variables to Custom Table Records

Objective

To store catalog request data in a structured custom table (**u_network_database**) for reporting, tracking, and auditing purposes.

3. Implementation Approach

- Process Automation is used to capture catalog variables.
- Catalog variables are mapped to corresponding fields in the custom table.
- Data is stored automatically upon request submission.

4. Variable Mapping Process

Steps Followed:

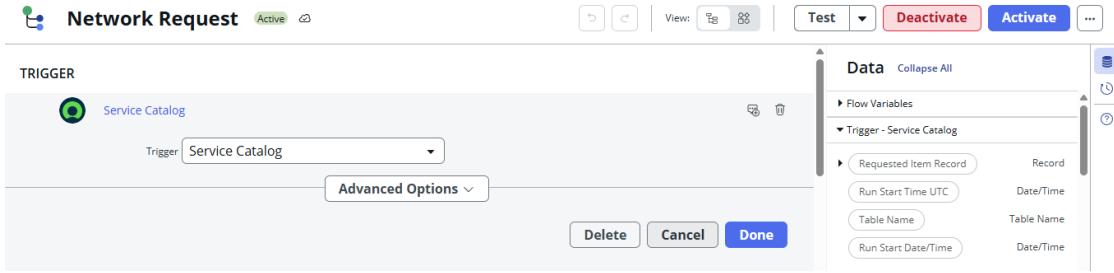


Figure: 4.1.1

This screenshot shows the 'Get Catalog Variables' action configuration. It includes sections for 'Action Properties' (set to 'Get Catalog Variables'), 'Action Inputs' (with 'Available' and 'Selected' lists), and 'Catalog variables' (with 'provide_device_details' and 'selected'). A note at the bottom states: 'Note: If removing a variable from the "Selected" list, it will be moved to "Available" list only if the variable is present in the selected Template, Item Template and Variable Sets. Otherwise, the variable is removed from both "Available" and "Selected" lists.' The right side of the screen shows the 'Data' panel with various catalog items like 'Run Start Time UTC', 'Table Name', and 'Run Start Date/Time'.

Figure: 4.1.2: Get Catalog Variables Action

This screenshot shows the 'Create Record' action configuration. It includes sections for 'Action Properties' (set to 'Create Record'), 'Action Inputs' (with 'Table' and 'Fields' sections), and a 'Fields' section where 'Request Number' is mapped to '1 - Get Catalog V... > user_id...'. The right side of the screen shows the 'Data' panel with various catalog items like 'Run Start Time UTC', 'Table Name', and 'Run Start Date/Time'.

Figure: 4.2: Create Record Variables Action

- Create a flow in **Flow Designer**.
- Configure appropriate **Trigger** (Service Catalog – Requested Item).

- Add **Get Catalog Variables** action to retrieve user inputs.
- Add **Create Record** action.
- Select **Network Database (u_network_database)** table.
- Click **Add Fields (+)** and map catalog variables to table fields.
- Save and activate the flow.

5. Benefits of Structured Data Handling

- Centralized data storage
- Improved reporting and tracking
- Enhanced audit readiness
- Easy integration with workflows and approvals

6. Conclusion

The data handling mechanism ensures reliable migration of user inputs from the Service Catalog to a structured backend table. This approach enhances data consistency, traceability, and long-term maintainability.