

Phase 5: Deployment, Documentation & Final Presentation

Troubleshooting

Overview

This section documents the troubleshooting and validation activities performed after implementing the **Automated Network Request Management in ServiceNow** solution. The goal was to ensure stable automation, accurate data handling, and reliable user experience across the Service Catalog, Flow Designer, and notification mechanisms.

Troubleshooting Areas & Actions

1. Flow Execution Monitoring

To validate automation reliability, Flow Designer execution logs were reviewed.

Actions Taken

- Used **Flow Designer** → **Execution Details** to:
 - Trace each step of the flow execution
 - Identify failures or skipped actions
 - Verify correct condition paths during approvals and record updates
- Confirmed that the flow triggers correctly on **Network Request catalog item submission**
- Ensured that approval, record creation, and update actions executed in the expected order

Result

- No blocking errors observed

- Flow executed successfully for all tested scenarios
-

2. Email Notification Validation

Email notifications were tested to confirm correct delivery and content.

Actions Taken

- Checked email activity under:
 - **System Logs** → **Email** → **Sent**
 - **System Logs** → **Email** → **Received**
- Verified:
 - Correct recipients (requester, approver)
 - Proper subject lines and message body
 - Dynamic values such as Request Number and user details
 - Successful email delivery status

Result

- Notifications triggered at request creation and approval stages
 - Emails delivered successfully with accurate content
-

3. Variable Population Issue Resolution

Issues related to missing or incorrect variable values were identified and resolved.

Actions Taken

- Debugged **Catalog Variables and Variable Sets**
- Ensured:
 - Variables were correctly selected in **Get Catalog Variables** action
 - Proper variable-to-field mappings in **Create Record** action
- Verified auto-population logic using:
 - Reference variables
 - Dot-walking for user-related fields (email, phone, name)

Result

- All required catalog variables populated correctly
 - Data stored accurately in the custom table (**u_network_database**)
-

4. Form Behaviour Testing

User interface behaviour was validated to ensure correct interaction in the Service Portal.

Actions Taken

- Tested catalog form using:
 - **Service Portal preview mode**
 - Live request submissions
- Verified:
 - UI Policies for conditional visibility
 - Mandatory and read-only field behaviour

- Dynamic display of fields based on user selections

Result

- UI behaved as expected
 - Conditional logic worked consistently without scripting issues
-

Outcome

The troubleshooting process ensured:

- Stable post-deployment automation
- Accurate variable population and data storage
- Reliable approval and email notification flow
- Smooth and error-free user experience in the Service Portal

The system is fully validated and ready for final presentation and future enhancements.