

Project Documentation: LaptopRequest Catalog Item

1. Project Overview

Project Title: LAPTOP REQUEST CATALOG ITEM

Platform: ServiceNow

Objective:

To automate and simplify the process of requesting laptops for employees by creating a dynamic and user-friendly Service Catalog item in ServiceNow. This eliminates manual requests, reduces processing time, and ensures accurate data collection.

Problem Statement:

Employees in the organization require an efficient way to request laptops for official use. The existing manual process is slow, error-prone, and lacks real-time validation or interactivity. There is also no mechanism to track changes or transfer configurations between instances efficiently.

To solve these challenges, a **Service Catalog item** named "**Laptop Request**" is developed, featuring dynamic fields, form reset functionality, and update set management for easy migration between instances.

2. Goals and Objectives

- a. Create a **Service Catalog item** for employees to request laptops.
- b. Implement **dynamic field behavior** based on user input.
- c. Add a **form reset UI action** to enhance user experience.
- d. Track all configurations using **Update Sets** for deployment.
- e. Enable **easy migration** of configurations between instances.
- f. Ensure a **seamless, accurate, and efficient** request process.

3. Tools and Technologies Used

Tool/Technology	Purpose
ServiceNow Platform	Catalog creation and automation
Update Sets	Configuration tracking and migration

Catalog UI Policy	Dynamic field behavior
CatalogVariables	Capturing user input
UI Actions	Adding customform actions
JavaScript	Client-side scripting for UI Action

4. ImplementationSteps

Step 1: Create a Local UpdateSet

1. Navigate to: **All** → **System UpdateSets** → **Local Update Sets**
2. Click **New** and createa new update set with the followingdetails:
 - a. **Name:** Laptop Request
3. Click **Submit**, then **Make Current** toactivate it.
4. This ensures that all subsequent configurations are tracked under this update set.

Step 2: Create the Service Catalog Item

1. Go to **All** → **Service Catalog**→ **Maintain Items**
2. Click **New** and enter the followingdetails:
 - a. **Name:** Laptop Request
 - b. **Catalog:** Service Catalog
 - c. **Category:** Hardware
 - d. **Short Description:** Use this item to requesta new laptop
3. Click **Save** to createthe catalog item.

Step 3: Add Catalog Variables

1. Scroll down to the **Variables** related list and click **New**.
2. Add the following variables:

Variable Label	Type	Name	Order
Laptop Model	Single Line Text	laptop_model	100
Justification	Multi Line Text	justification	200
AdditionalAccessories	Checkbox	additional_accessories	300
Accessories Details	Multi Line Text	accessories_details	400

3. Save after adding all variables.

Step 4: Create CatalogUI Policy

1. Navigate to **All** → **Service Catalog** → **Maintain Items**
2. Open **Laptop Request** item
3. Scroll to the **CatalogUI Policies** related list and click **New**
4. Fill the details:
 - a. **Short Description:** Show Accessories Details
 - b. **When to Apply:**
 - i. **Field:** additional_accessories
 - ii. **Operator:** is
 - iii. **Value:** true
5. Click **Save** (not Submit).
6. Scroll down to **Catalog UI PolicyActions** and click **New**
 - a. **Variable Name:** accessories_details
 - b. **Mandatory:** True
 - c. **Visible:** True
 - d. **Order:** 100
7. Click **Save**, then save the main UI Policy form again.
8. This ensures that the “Accessories Details” field appears and becomes mandatory only when the “Additional Accessories” checkbox is checked.

Step 5: Create a UI Action (Reset Form Button)

1. Navigate to **All** → **System Definition** → **UI Actions**
2. Click **New** and enter details:
 - a. **Table:** sc_cart
 - b. **Order:** 100
 - c. **Action Name:** Reset Form
 - d. **Client:** Checked
3. Add the following script:
4.

```
function resetForm() {  
    g_form.clearForm(); // Clears all fields in the form  
    alert("The form has been reset.");  
}
```
5. Click **Save**.

Step 6: Export Changes (Update Set Migration)

1. Go to **All** → **System Update Sets** → **Local Update Sets**
2. Open the created update set **Laptop Request**
3. Set the **State** to **Complete**
4. In the **Updates** related list, verify all tracked changes
5. Click **Export to XML** – this downloads the XML file of the update set.

Step 7: Retrieve Update Set in Another Instance

- a. Open another ServiceNow instance (in incognito mode).
- b. Navigate to **All** → **System Update Sets** → **Retrieved Update Sets**
- c. Click **Import Update Set from XML**
- d. Upload the previously exported XML file.
- e. Click **Upload**, then **Preview Update Set**.
- f. Finally, click **Commit Update Set**.
- g. The catalog item and configurations are now imported successfully into the new instance.

Step 8: Test the Catalog Item

1. Search for **Service Catalog** in the navigator.
2. Open **Hardware Category** → **Laptop Request**.
3. Verify:
 - a. "AccessoriesDetails" field appears only when **Additional Accessories** is checked.
 - b. The field becomes **mandatory** when visible.
 - c. The **Reset Form** button clears all form fields.

5. Results

The **Laptop Request Catalog Item** works as intended:

- a. Users can easily request laptops using a dynamic and interactive form.
- b. The form dynamically adjusts based on user input.
- c. The reset functionality improves user experience.
- d. All changes are properly tracked and portable across instances via update sets.

6. Conclusion

The **Laptop Request Catalog Item** project effectively streamlines the laptop request process in the organization using ServiceNow's **Service Catalog** capabilities.

By automating form handling, adding dynamic visibility rules, and implementing a reset action, the system improves accuracy and efficiency.

Furthermore, the use of **update sets** ensures all configurations are safely tracked and easily migrated between environments.

This project demonstrates how ServiceNow can transform manual workflows into **automated, efficient, and user-friendly digital processes**, ultimately improving service delivery and employee satisfaction.

7 . Future Enhancements

- a . Integrate with **Approval Workflows** for managerial authorization.
- b . Auto-populate user details (department, location) from the user record.
- c . Generate an **email notification** upon submission.
- d . Implement **Asset Management Integration** to assign laptops automatically.