

Cognizant
Digital Nurture 3.0
Deep Skilling

Module 4 – ServiceNow Scripting Fundamentals and Functions

Overview:

This is a comprehensive course aimed at teaching learners how to script on the ServiceNow platform effectively. The session covers scripting fundamentals and provides a solid foundation for ServiceNow developers to enhance their scripting skills. By the end of the course, participants will have the knowledge and tools required to script on the ServiceNow platform confidently.

Scripting on the ServiceNow Platform

Scripting is a fundamental part of customizing and enhancing functionality on the ServiceNow platform. The platform primarily uses JavaScript for server-side and client-side scripting. In this course, you would typically cover the following scripting areas:

1. Client-Side Scripting

Client-side scripting runs on the user's browser and primarily controls the user interface (UI) and user interactions. Some key components include:

- **Client Scripts:** These scripts run on form load, form change, or form submit and are used to manipulate field values, hide/show elements, or perform form validation.

Example:

```
function onLoad() {  
    var user = g_user.getFullName();  
    alert('Welcome, ' + user);  
}
```

- **UI Policies:** These are declarative scripts that control form behavior such as hiding fields, making fields mandatory, or read-only based on conditions.

- **UI Actions:** Scripts attached to buttons or form actions, allowing custom actions like saving data or performing complex operations when triggered.

2. Server-Side Scripting

Server-side scripting runs on the ServiceNow servers and is used to interact with the database, perform business logic, and automate backend processes.

- **Business Rules:** These are triggered by database actions like record insert, update, delete, or query. They can enforce rules, automatically assign tasks, or trigger workflows.

Example:

```
(function executeRule(current, previous) {  
    if (current.state == 'Closed') {  
        current.close_notes = 'Automatically closed by rule';  
    }  
})(current, previous);
```

- **Script Includes:** These are reusable server-side functions or classes that can be called from other scripts, such as Business Rules or Workflow scripts.

- **GlideRecord API:** This API is used to query, insert, update, or delete records from ServiceNow's database.

Example:

```
var incident = new GlideRecord('incident');

incident.addQuery('priority', 1);

incident.query();

while (incident.next()) {

    gs.info('High priority incident: ' + incident.number);

}
```

3. Scripting in Workflows

Workflows automate processes and may require scripting for more complex logic. Key scripting elements in workflows include:

- **Workflow Script Actions:** JavaScript used to customize the logic in a workflow step, allowing you to interact with records and manage transitions between steps.
- **Flow Designer:** Low-code/no-code automation, but for advanced needs, you can add Script Actions to define custom logic within flows.

4. GlideAjax (Client-Server Communication)

GlideAjax allows client scripts to communicate with server-side code, making it possible to retrieve data from the server asynchronously without reloading the page.

- Client-Side (GlideAjax Call):

```
var ga = new GlideAjax('MyScriptInclude');

ga.addParam('sys_id', g_form.getValue('sys_id'));

ga.getXMLAnswer(function(response) {

    var answer = response.responseXML.documentElement.getAttribute('answer');

    g_form.setValue('short_description', answer);

});
```

- Server-Side (Script Include):

```
var MyScriptInclude = Class.create();

MyScriptInclude.prototype = {

    initialize: function() {},

    myFunction: function() {

        return 'This is the server response!';

    }

}
```

```
}  
};
```

5. Scripted REST APIs

ServiceNow allows you to create custom REST APIs using scripts. This is especially useful for integrations with other systems.

- **Scripted REST API:** Define a REST endpoint and write server-side scripts to process incoming requests and send responses.

Example:

```
(function process(request, response) {  
    var result = {};  
    result.message = "Hello from ServiceNow";  
    response.setBody(result);  
})(request, response);
```

6. Event Management Scripting

ServiceNow's Event Management module uses scripting to handle alerts and events.

- **Event Rules:** These can trigger Business Rules or Scripts to handle specific event conditions, such as escalating incidents or notifying users when an alert is raised.

7. Security and Best Practices

- **Use Script Includes for Reusability:** Avoid repetitive code by writing reusable functions.

- **Validate User Inputs:** Always validate and sanitize user inputs in both client-side and server-side scripts to prevent security vulnerabilities.

- **Avoid Hard-Coding Values:** Use system properties or data lookups instead of hard-coding values in scripts.

Understanding How ServiceNow Functions

ServiceNow is a cloud-based platform designed to manage digital workflows and business processes across various departments such as IT, HR, security, and customer service. The platform functions as a comprehensive service management tool by automating routine tasks, facilitating communication between different business units, and providing real-time data and reporting through dashboards.

Key functional elements include:

- 1. Workflows:** ServiceNow automates workflows by allowing users to define processes for tasks such as ticketing, service requests, and approvals.
- 2. Forms and Lists:** Data in ServiceNow is displayed in forms (individual records) and lists (a collection of records) that users interact with to manage and update information.
- 3. Self-Service Portals:** ServiceNow enables the creation of user-friendly portals where employees, customers, or users can submit requests, track incidents, and access knowledge articles.
- 4. Integration Capabilities:** The platform supports integration with other third-party tools and applications via APIs, providing seamless data exchange and process automation across systems.

How to Properly Configure and Personalize the ServiceNow Platform

To get the most out of ServiceNow, it's essential to configure and personalize the platform based on your organization's needs. Here's how you can do it:

1. Personalizing the User Interface:

- **Branding and Themes:** Administrators can configure the platform's look and feel by modifying the logo, colors, and header styles to align with the organization's branding.
- **Homepage Configuration:** Users and admins can create personalized dashboards and homepages with widgets that display real-time data and reports.

2. Setting Up User Roles and Permissions:

- **Roles:** ServiceNow allows you to assign roles such as Administrator, ITIL User, or Requester to users based on their job function.
- **Access Control Rules:** Define who can view or modify specific records, forms, or fields by configuring access control lists (ACLs).

3. Form and List Configuration:

- **Form Layout:** Admins can modify the layout of forms by adding, removing, or rearranging fields to improve usability.
- **List Views:** Customize list views by defining which fields (columns) are displayed, and applying filters to focus on relevant data.

4. Notification Settings:

- Set up email or push notifications to notify users of important events, such as when incidents are created, or tasks are assigned.

5. Workflows and Business Rules:

- **Business Rules:** Automate processes by configuring business rules that trigger actions when certain conditions are met (e.g., auto-assigning incidents based on priority).

- **Flow Designer:** Create or modify automated workflows for processes like approvals, task assignments, and request fulfillment without needing complex coding.

Incident Module

- **Purpose:** The Incident Management module is designed to manage unplanned disruptions or issues affecting IT services. It ensures that incidents are tracked, prioritized, and resolved efficiently to minimize business impact.

- **Key Features:**

- **Incident Creation:** Users can create incidents manually or automatically (via email, self-service portal, or monitoring tools).

- **Categorization and Prioritization:** Incidents can be categorized (e.g., hardware, software) and prioritized based on business impact and urgency.

- **Assignment:** Incidents are assigned to the appropriate support teams or individuals based on predefined workflows.

- **Resolution and Closure:** Once resolved, incidents are closed with proper documentation of the resolution method.

Problem Module

- **Purpose:** The Problem Management module helps organizations investigate the root cause of incidents, preventing them from recurring in the future. It aims to identify and eliminate underlying causes of issues affecting IT services.

- **Key Features:**

- **Problem Detection:** Problems can be identified through recurring incidents or proactive analysis of potential risks.

- **Root Cause Analysis:** Problem Management facilitates investigations to determine the root cause of issues, often using techniques such as the "5 Whys" or fishbone diagrams.

- **Known Errors:** When a workaround is discovered but the problem is not fully resolved, a "Known Error" can be documented for reference.

- **Problem Resolution:** Once the root cause is identified, corrective actions are taken, and the problem is closed.

Change Module

- **Purpose:** The Change Management module ensures that changes to IT systems, infrastructure, or services are managed in a controlled way to reduce risk and avoid service disruptions.

- Key Features:

- **Change Request Creation:** Change requests can be created for updates, patches, or new service rollouts.

- **Risk and Impact Assessment:** Each change is assessed for risk and impact, and appropriate actions are planned to minimize disruptions.

- **Approval Workflows:** Change requests go through an approval process involving key stakeholders before implementation.

- **Implementation and Review:** Once approved, changes are implemented, monitored, and reviewed to ensure they meet expectations.

- **Post-Implementation Reviews (PIRs):** Evaluate whether the change was successful and identify any improvements for future changes.

Forms in ServiceNow

Forms in ServiceNow are used to display individual records, such as an incident, request, or change. Forms are highly customizable and allow users to interact with fields, such as:

- **Field Types:** Forms can include different field types (text, date, number, choice, etc.).

- **Client Scripts:** You can create custom scripts to control the behavior of forms (e.g., making fields mandatory or read-only based on certain conditions).

- **Form Sections:** Forms can be broken down into sections, improving clarity and organization.

Form Features:

- **Activity Stream:** Shows a history of changes, comments, and work notes for the record.

- **Attachments:** Users can attach relevant files, screenshots, or documents to records via forms.

- **Related Lists:** Displays related records, such as tasks or incidents associated with the current record.

Lists in ServiceNow

Lists in ServiceNow provide a way to view multiple records simultaneously in a tabular format. Users can sort, filter, and group list data to focus on the most relevant records.

- **Personalizing List Views:** Users can add, remove, or reorder columns to display the most relevant fields. Filters can also be applied to refine data.

- **List Actions:** Lists allow mass actions such as updating multiple records, exporting data to Excel, or running reports on selected records.

Key List Features:

- **Column Sorting:** Allows sorting of records by clicking on column headers.
- **Filter Conditions:** Users can apply filter conditions to display only records that match specific criteria (e.g., all incidents with a priority of "Critical").
- **List Editing:** Enables inline editing, allowing users to make quick updates directly in the list view.