ServiceNow Developer

ServiceNow Implementation Development Guide.

Client Script & Catalog Client Script

Content

01 What is Client Script & Catalog Client Script

02 Client Script Form

03 Catalog Client Script Form

04 Types of Client Script & Catalog Client Script

05 Difference between Client Script & Catalog Client Script

What is Client Script & Catalog Client Script ?

Client scripts allow the system to run JavaScript on the client (web browser) when client-based events occur, such as when a form loads, after form submission, or when a field changes value.

Use client scripts to configure forms, form fields, and field values while the user is using the form. Client scripts can:

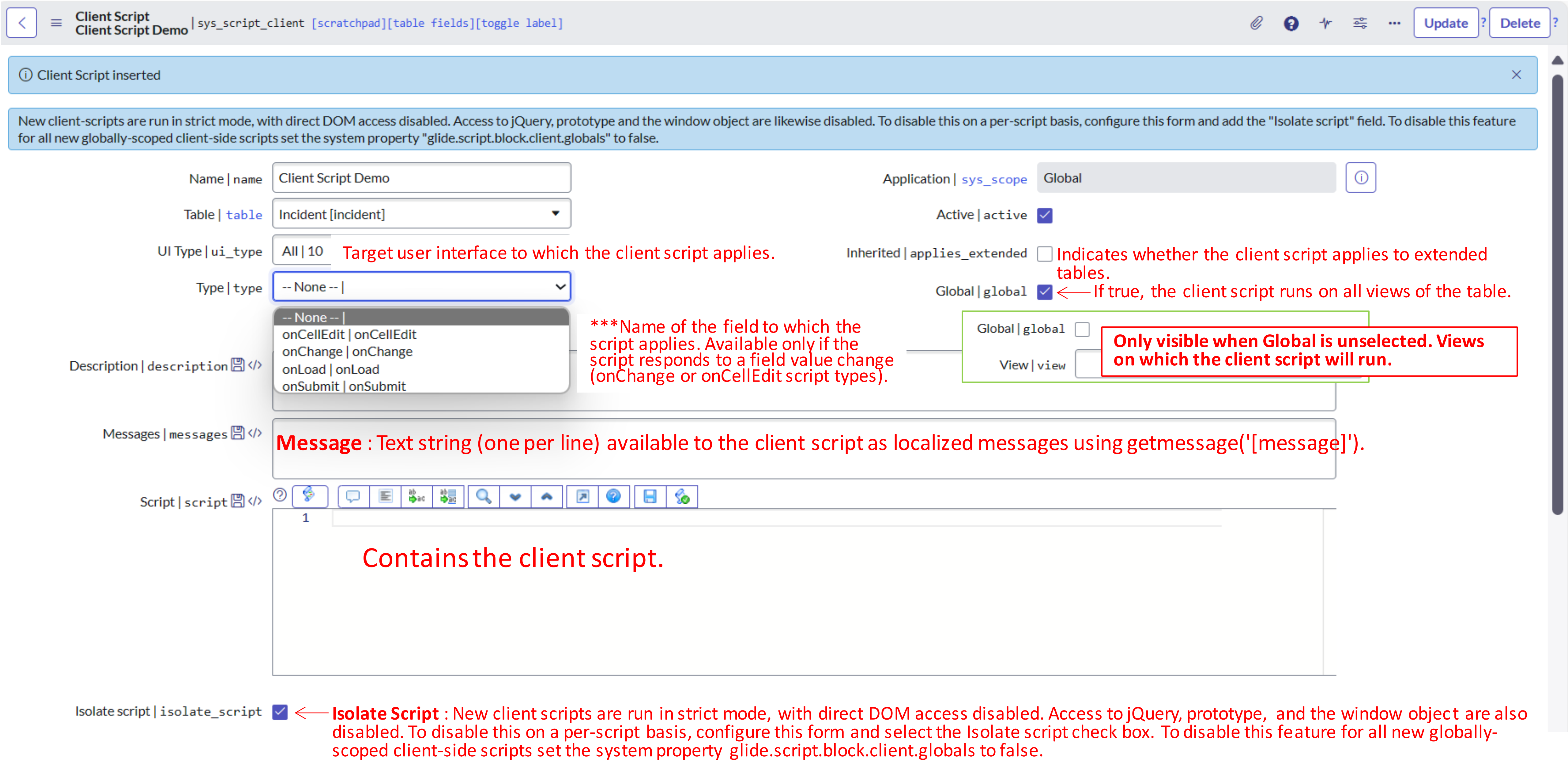
- make fields hidden or visible
- make fields read only or writable
- make fields optional or mandatory based on the user's role
- set the value in one field based on the value in other fields
- modify the options in a choice list based on a user's role
- display messages based on a value in a field

Catalog Client Script : Scripts can apply to **service catalog items or variable sets**, allowing administrators to use the same functionality that is available on other forms.

You can use client side scripts to:

- Get or set variable values.
- Hide or display variables.
- Make variables mandatory or not.
- Validate form submission.
- Add something to the cart.
- Order something immediately.

Client Script Form



Client Script
Client Script Demo
| sys_script_client [scratchpad][table fields][toggle label]
Update
Delete

ⓘ Client Script inserted

New client-scripts are run in strict mode, with direct DOM access disabled. Access to jQuery, prototype and the window object are likewise disabled. To disable this on a per-script basis, configure this form and add the "Isolate script" field. To disable this feature for all new globally-scoped client-side scripts set the system property "glide.script.block.client.globals" to false.

Name | name: Client Script Demo

Application | sys_scope: Global

Table | table: Incident [incident]

Active | active ☑

UI Type | ui_type: All | 10 — Target user interface to which the client script applies.

Inherited | applies_extended ☐ — Indicates whether the client script applies to extended tables.

Type | type: -- None -- |

- -- None -- |
- onCellEdit | onCellEdit
- onChange | onChange
- onLoad | onLoad
- onSubmit | onSubmit

Global | global ☑ ← If true, the client script runs on all views of the table.

***Name of the field to which the script applies. Available only if the script responds to a field value change (onChange or onCellEdit script types).

Global | global ☐

View | view

Only visible when Global is unselected. Views on which the client script will run.

Description | description

Messages | messages

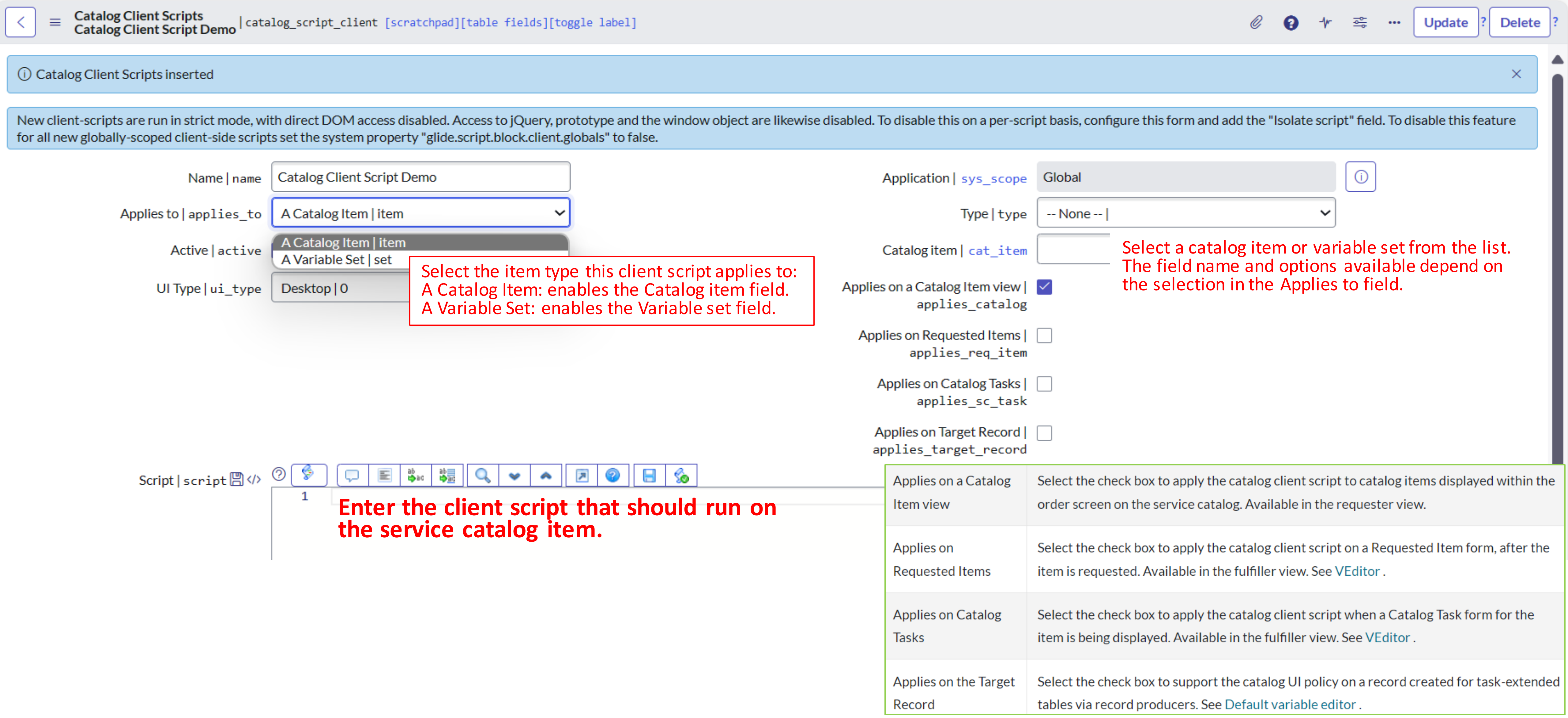
Message : Text string (one per line) available to the client script as localized messages using getmessage('[message]').

Script | script

Contains the client script.

Isolate script | isolate_script ☑ ← **Isolate Script** : New client scripts are run in strict mode, with direct DOM access disabled. Access to jQuery, prototype, and the window object are also disabled. To disable this on a per-script basis, configure this form and select the Isolate script check box. To disable this feature for all new globally-scoped client-side scripts set the system property glide.script.block.client.globals to false.

Catalog Client Script Form



Catalog Client Scripts
Catalog Client Script Demo
| catalog_script_client [scratchpad][table fields][toggle label]
Update
Delete

ⓘ Catalog Client Scripts inserted

New client-scripts are run in strict mode, with direct DOM access disabled. Access to jQuery, prototype and the window object are likewise disabled. To disable this on a per-script basis, configure this form and add the "Isolate script" field. To disable this feature for all new globally-scoped client-side scripts set the system property "glide.script.block.client.globals" to false.

Name | name	Catalog Client Script Demo
Applies to | applies_to	A Catalog Item | item
Active | active	
UI Type | ui_type	Desktop | 0
Application | sys_scope	Global
Type | type	-- None -- |
Catalog item | cat_item	
Applies on a Catalog Item view | applies_catalog	☑
Applies on Requested Items | applies_req_item	□
Applies on Catalog Tasks | applies_sc_task	□
Applies on Target Record | applies_target_record	□
Script | script	1

Select the item type this client script applies to:
A Catalog Item: enables the Catalog item field.
A Variable Set: enables the Variable set field.

Select a catalog item or variable set from the list. The field name and options available depend on the selection in the Applies to field.

Enter the client script that should run on the service catalog item.

Applies on a Catalog Item view	Select the check box to apply the catalog client script to catalog items displayed within the order screen on the service catalog. Available in the requester view.
Applies on Requested Items	Select the check box to apply the catalog client script on a Requested Item form, after the item is requested. Available in the fulfiller view. See VEditor .
Applies on Catalog Tasks	Select the check box to apply the catalog client script when a Catalog Task form for the item is being displayed. Available in the fulfiller view. See VEditor .
Applies on the Target Record	Select the check box to support the catalog UI policy on a record created for task-extended tables via record producers. See Default variable editor .

Types of Client Script & Catalog Client Script

Client Script	Catalog Client Script

onLoad
function onLoad() {
//Type appropriate comment here, and begin script below
}
	onLoad
	function onLoad() {
	//Type appropriate comment here, and begin script below
	}

onSubmit
function onSubmit() {
//Type appropriate comment here, and begin script below
}
	onSubmit
	function onSubmit() {
	//Type appropriate comment here, and begin script below
	}

onChange
function onChange(control, oldValue, newValue, isLoading, isTemplate) {
if (isLoading || newValue === '') {
return;
}
//Type appropriate comment here, and begin script below
}
	onChange
	function onChange(control, oldValue, newValue, isLoading) {
	if (isLoading || newValue == '') {
	return;
	}
	//Type appropriate comment here, and begin script below
	}

onCellEdit
function onCellEdit(sysIDs, table, oldValues, newValue, callback) {
var saveAndClose = true;
//Type appropriate comment here, and begin script below
callback(saveAndClose);
}	

Difference btw Client Script & Catalog Client Script

Client Script	Catalog Client Script
Types of Client Script - onLoad, onChange, onSubmit, onCellEdit.	Types of Catalog Client Script - onLoad, onChange, onSubmit.
Client Script applies to Table.	Catalog Client Script applies on Catalog Items & Variable Set.
Client Script can be View Specific or can apply to all Views using Global checkbox.	Catalog Client Script can be applies to Catalog Items View, Requested Item, Catalog Tasks, Target Record
Client Script can be applies to extended tables.	
onCellEdit Client Script executes, when User change value of any field from List view.	N/A (***no onCellEdit Script type in Catalog Client Script)

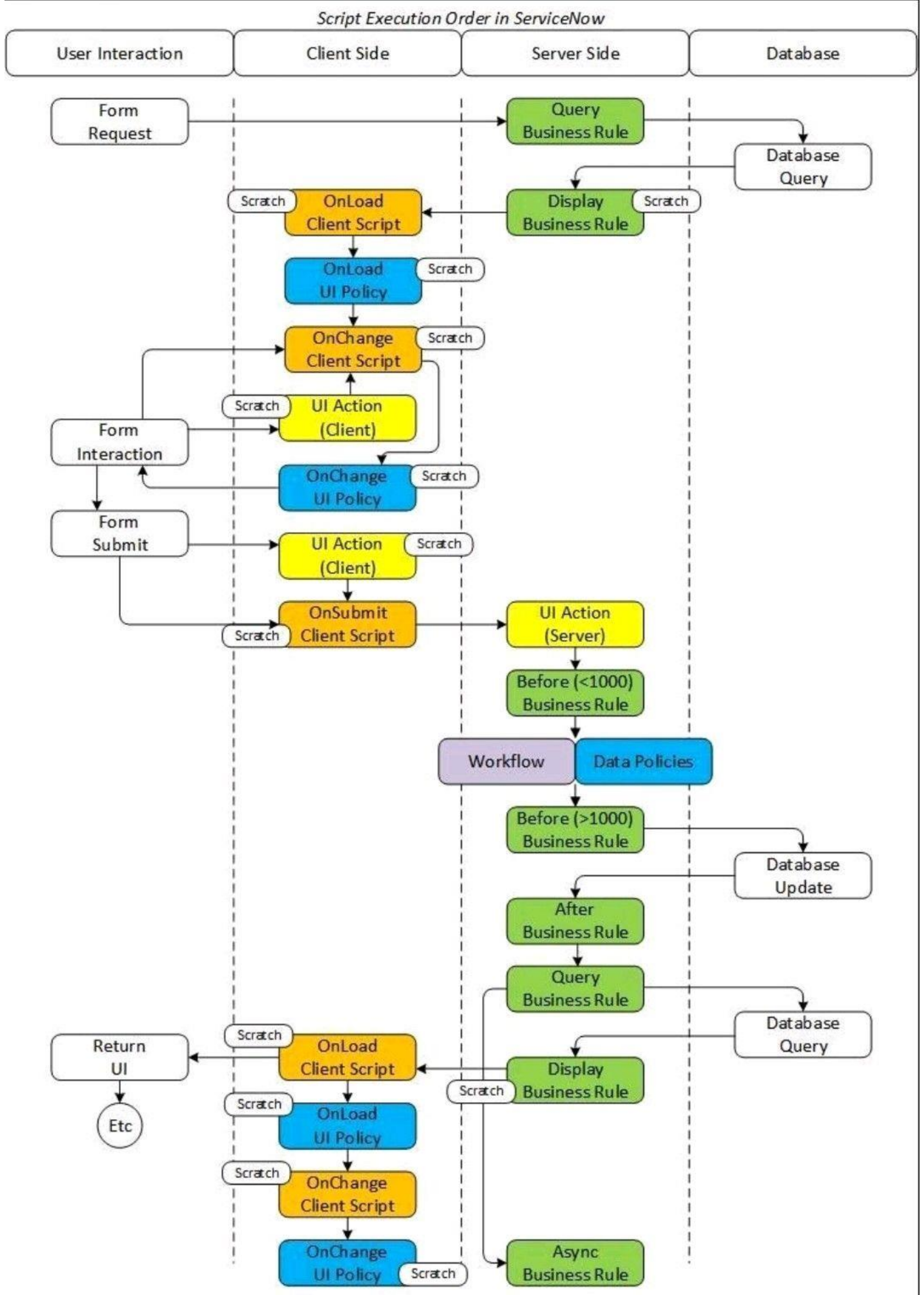
Script Execution Order

Different colors are used to represent different types of ServiceNow artifacts. "Scratch" indicates access to g_scratchpad, which is reset to an empty state when the form is submitted, so the g_scratchpad from the second Display Business Rule starts new.

While attempting to determine the precise sequence of all the steps involved in this process, several interesting observations were made:

1. Setting values in g_scratchpad through a Display Business Rule makes them accessible in ACLs.
2. It's worth noting that g_scratchpad remains available throughout the client's work until submission. This means that it can also be used to track client-side activity like a client state parameter.
3. Data Policies are executed at the "Before Business Rule" order number 1000. This means that they are executed after all Business Rules with order values less than 1000 and before all Business Rules with order values greater than 1000. It's essential to be aware that Business Rules executed before may interact with the checks performed by Data Policies.
4. Flows are executed after the data is committed to the database, unlike Business Rules and Workflows, which are executed before the data is saved. They allow you to automate tasks and integrations between ServiceNow and other systems.
5. The Script includes in ServiceNow can be invoked by Business Rules or any other server-side scripts. Therefore, their execution order is not fixed and depends on the server-side entity that calls them.

Script Article Execution shared by : Vyshnavi Raja



THANKS

Sangeeta Yadav
SERVICENOW