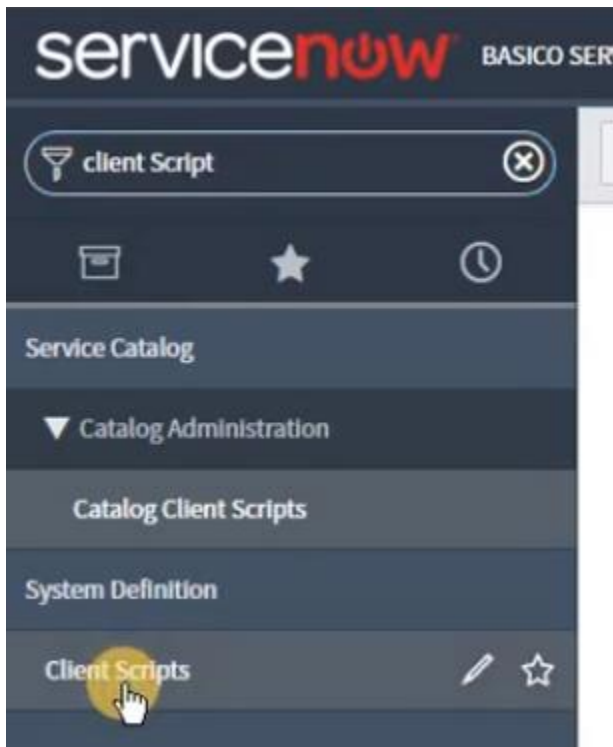


Step One: Go to Application Navigator > Client Scripts :



...Create New.

Step Two: Fill out the form for the new Client Script:

The name will be GlideFormDemo.

The Table will be Incident for this example.

The UI Type will be Desktop for this example.

The type will be onChange... it can NEVER BE for onCellEdit for g_form.

The Field Name will be "Caller" so that the onChange is reflected accordingly on the form.

Client Script
New record

Name:

Table:

UI Type:

Type:

Field name:

Description:

Messages:

Script:

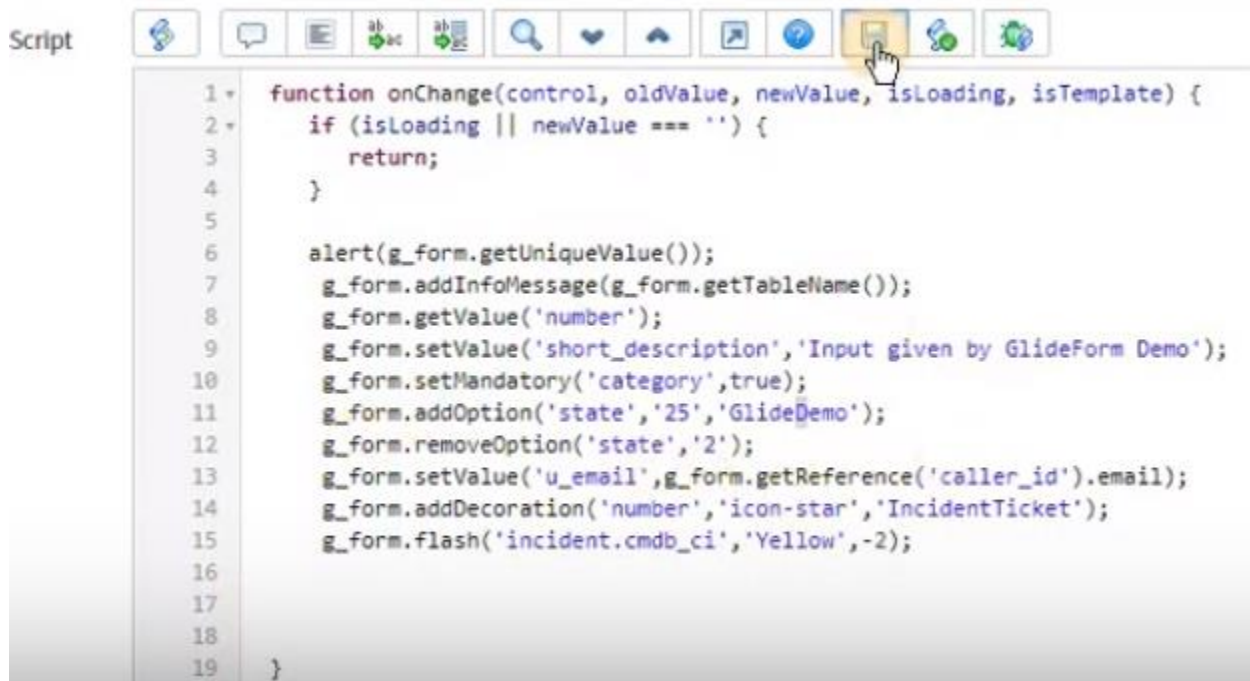
Application:

Active: ☒

Inherited: ☐

Global: ☒

Step Three: Write the Script:



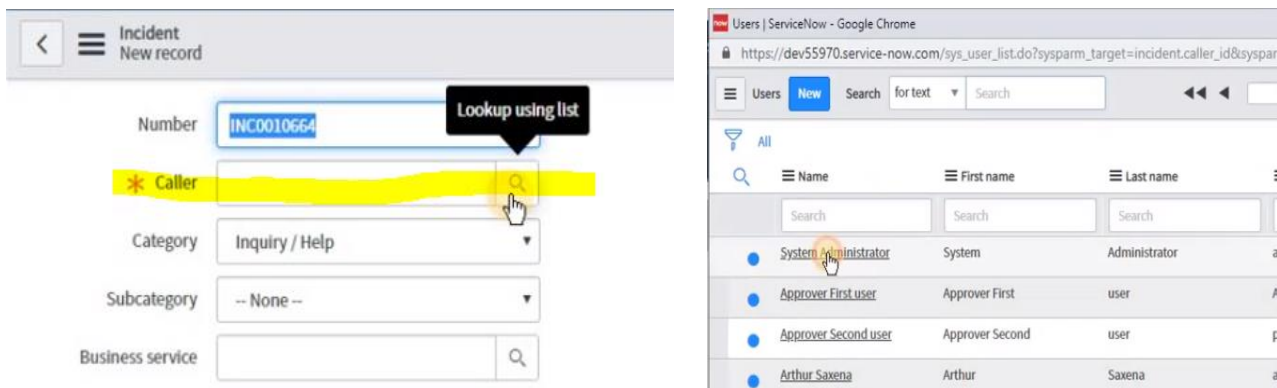
The screenshot shows a script editor window with a toolbar at the top. The toolbar includes icons for undo, redo, search, and other editing functions. The script editor displays a JavaScript function named `onChange` that is triggered when a value in a GlideForm changes. The function checks if the form is loading or if the new value is empty. If not, it performs several actions: it alerts the user with the unique value of the form, adds an info message with the table name, gets the value of the 'number' field, sets the 'short_description' field to 'Input given by GlideForm Demo', sets the 'category' field to be mandatory, adds an option to the 'state' field with value '25' and label 'GlideDemo', removes an option with value '2', sets the 'u_email' field to the email of the caller, adds a star icon to the 'number' field with the label 'IncidentTicket', and flashes a yellow message on the incident command bar.

```
1+ function onChange(control, oldValue, newValue, isLoading, isTemplate) {  
2+   if (isLoading || newValue === '') {  
3     return;  
4   }  
5  
6   alert(g_form.getUniqueValue());  
7   g_form.addInfoMessage(g_form.getTableName());  
8   g_form.getValue('number');  
9   g_form.setValue('short_description', 'Input given by GlideForm Demo');  
10  g_form.setMandatory('category', true);  
11  g_form.addOption('state', '25', 'GlideDemo');  
12  g_form.removeOption('state', '2');  
13  g_form.setValue('u_email', g_form.getReference('caller_id').email);  
14  g_form.addDecoration('number', 'icon-star', 'IncidentTicket');  
15  g_form.flash('incident.cmdb_ci', 'Yellow', -2);  
16  
17  
18  
19 }
```

Save or Update.

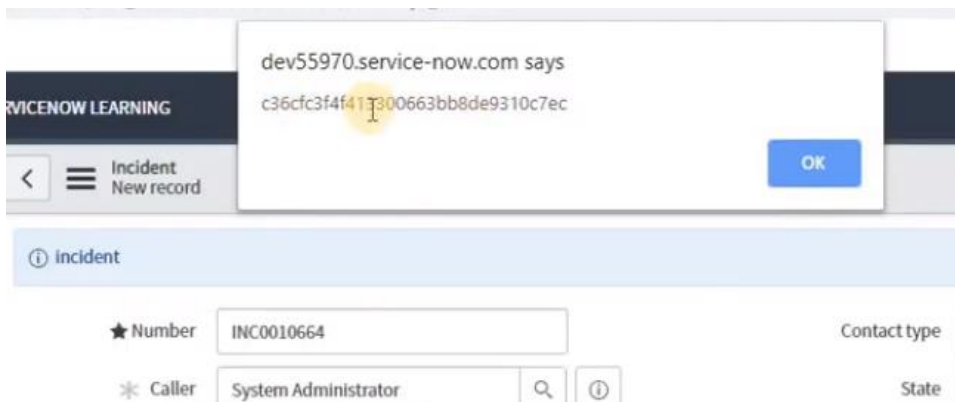
Next, Refresh the **New Incident** Form and try out the `g_form` object and its methods.

...Now let's test the OnChange type on the Caller:



The left screenshot shows the 'Incident New record' form. The 'Number' field contains 'INC0010664'. A callout bubble says 'Lookup using list'. The 'Caller' field is highlighted in yellow and has a search icon. The 'Category' is 'Inquiry / Help' and 'Subcategory' is '-- None --'. The right screenshot shows the 'Users' list in ServiceNow. The 'System Administrator' user is selected, highlighted in yellow, and circled with a red dot.

...when we choose the caller, the alert of the unique value (32-character alphanumeric string) of the new record:



The top screenshot shows an alert dialog box with the text: 'dev55970.service-now.com says c36cfc3f4f411300663bb8de9310c7ec'. The 'OK' button is visible. The bottom screenshot shows the 'Incident' form with the 'Number' field set to 'INC0010664' and the 'Caller' field set to 'System Administrator'.

...this is thanks to the alert statement:

```

1  function onChange(control, oldValue, newValue, isLoading, isTemplate) {
2      if (isLoading || newValue === '') {
3          return;
4      }
5
6      alert(g_form.getUniqueValue());
7      g_form.addInfoMessage(g_form.getTableName());
8      g_form.getValue('number');
9      g_form.setValue('short_description', 'Input given by GlideForm Demo');
10     g_form.setMandatory('category', true);

```

Also, we see the display of what type of Table it is (Incident table).

The screenshot shows a ServiceNow interface. At the top, a dark blue header contains the text "VICENOW LEARNING". Below this, a navigation bar shows a back arrow, a hamburger menu, and the text "Incident New record". A modal dialog box is open in the center, displaying the text "dev55970.service-now.com says" followed by a long alphanumeric string "c36cfc3f4f411300663bb8de9310c7ec". A yellow circle highlights the character "1" in the string. An "OK" button is in the bottom right of the dialog. Below the dialog, a light blue bar contains a yellow icon and the word "Incident". The main form area has fields for "Number" (INC0010664), "Caller" (System Administrator), "Contact type", and "State".

...this is thanks to the statement containing the addInfoMessage method:

The screenshot shows a ServiceNow script editor. The left sidebar is labeled "Script". The editor has a toolbar with various icons. The main area contains a JavaScript function:

```
1 function onChange(control, oldValue, newValue, isLoading, isTemplate) {  
2     if (isLoading || newValue === '') {  
3         return;  
4     }  
5  
6     alert(g_form.getUniqueValue());  
7     g_form.addInfoMessage(g_form.getTableName());  
8     g_form.getValue('number');  
9     g_form.setValue('short_description', 'Input given by GlideForm Demo');  
10    g_form.setMandatory('category', true);  
11    g_form.addOption('state', '25', 'GlideDemo');  
12    g_form.removeOption('state', '2');  
13    g_form.setValue('u_email', g_form.getReference('caller_id').email);
```

...and so on...etc, etc (you get the idea!)