ServiceNow Application Developer

Server-side Scripting > Exercise: Create the NeedItUtils Script Include

In this exercise, you will move the date verification logic you wrote in the *NeedIt When needed field date* Business Rule lab into a Script Include. You will call the Script Include method from the Business Rule.

Create the Script Include

1. Create a Script Include.

   a. In Studio, click the **Create Application File** link.

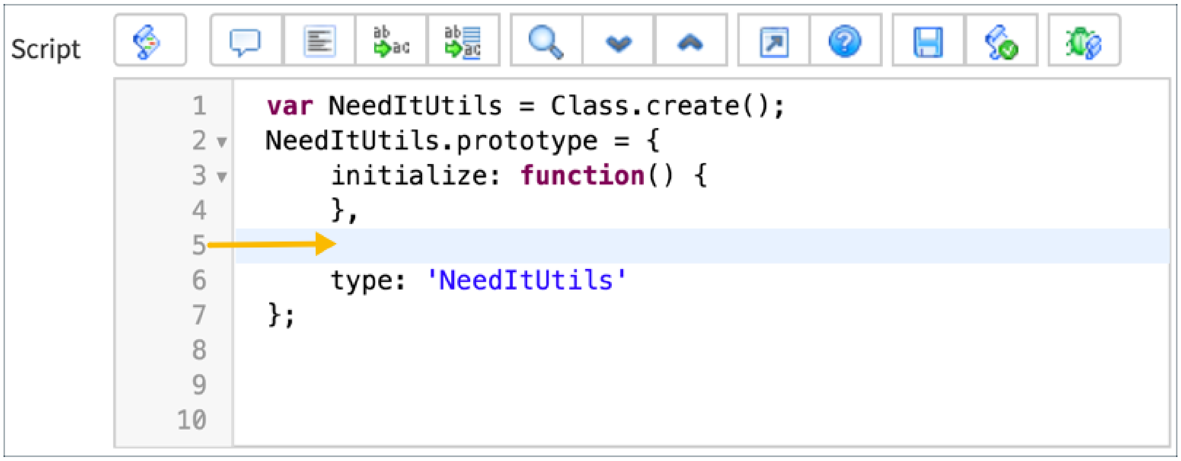
   b. In the *Filter...* field enter the text **Script** OR select **Server Development** from the categories in the left hand pane.

   c. Select **Script Include** in the middle pane as the file type then click the **Create** button.

2. Configure the Script Include:

   Name: **NeedItUtils**
   API Name: **(this field value is automatically populated)**
   Application: **(this field value is automatically populated)**
   Accessible from: **This application scope only**
   Active: **Selected (checked)**
   Description: **Utils Script Include for the NeedIt application.**

3. Copy the *isDatePast* function. Paste it into the Script Include after the *initialize* function and before the *type* property.



```
var NeedItUtils = Class.create();
NeedItUtils.prototype = {
    initialize: function() {
    },

    type: 'NeedItUtils'
};
```

```
isDatePast: function(strDate){
        // Create GlideDateTime objects for the current date and the passed
in date
        var rightnow = new GlideDateTime();
        var testdate = new GlideDateTime(strDate);

        // If the testdate is before rightnow, return true, else return false
        if (testdate.before(rightnow)) {
            return true;
        }
        else {
            return false;
        }
    },
```

4. Click the **Submit** button.

Modify the NeedIt When needed field date Business Rule

1. In Studio, use the Application Explorer to open **Server Development > Business Rules > NeedIt When needed field date**.

2. Delete the script logic for testing if the *When needed* date is in the past:

```
// rightnow stores the current time
    var rightnow = new GlideDateTime();
    // Create a GlideDateTime object for the When needed date
    var whenNeeded = new GlideDateTime(current.u_when_needed);

    // If the When needed date is before rightnow, do not write the record to the database
    // Output an error message to the screen
    if(whenNeeded.before(rightnow)){
    gs.addErrorMessage("When needed date cannot be in the past.  Your request has not been saved to the database.");
    current.setAbortAction(true);
}
```

3. Copy and paste the new script logic into the Business Rule:

```
// Instantiate the NeedItUtils class.  Call the isDatePast method and pass
    // the u_when_needed value.
    var niutils = new NeedItUtils();
    var isPast = niutils.isDatePast(current.u_when_needed);

    // If the isDatePast method returns true, the date is in the past.
    if(isPast == true){
        gs.addErrorMessage("When needed date cannot be in the past.  Your request has not been saved to the database.");
        current.setAbortAction(true);
    }
```

4. Click the **Update** button.

▶ **QUESTION**: Why is it a good idea to move the date validation logic out of the Business Rule and into the Utils Script Include?

Test the Script Include and Business Rule

1. Return to the main ServiceNow browser window.

2. Use the Application Navigator to open **NeedIt > Create New**.

3. Put values in all mandatory fields and set the *When needed* field date in the past.

4. Click the **Additional actions** menu (☰) and select the **Save** menu item. Do not click the *Submit* button because that will take you away from the form.

5. You should see an error message stating that the *When needed* date cannot be in the past. If you do not see the error message, debug and re-test. NOTE: If you worked the Challenge part of *Exercise: Create a Business Rule*, that logic might now be broken.

Challenge

If you worked the Challenge part of *Exercise: Create a Business Rule*:

1. Add a function, *isDateToday*, to the NeedItUtils Script Include. Move the logic to determine if the *When needed* date to the *isDateToday* function.

2. Make any modifications necessary to the *NeedIt When needed field date* Business Rule to use the *isDateToday* method.

If you did not work the Challenge part of Exercise: Create a Business Rule you can still do this Challenge:

1. Add a function called *isDateToday* to the *NeedItUtils* Script Include. Write logic to determine if the *When needed* date is after today.

2. Add whatever logic is necessary to the *NeedIt When needed* field date Business Rule to use the *isDateToday* method to test whether the When needed date is today. Do not allow users to save new records if the *When needed* value is today.

▶ **CHALLENGE SOLUTION**: