**Analysis notes:**

|  |  |
| --- | --- |
| **Script Overview** | |
| Script Name | 360CCC UE Create Work Order Button |
| Script ID | customscript\_360\_ue\_create\_wo\_button |
| Script File | 360CCC\_SO\_UE\_CreateWorkOrderButton.js |
| Script Link | [360CCC\_SO\_UE\_CreateWorkOrderButton.js](https://4449625.app.netsuite.com/app/common/media/mediaitem.nl?id=734609) |
| Type | User Event |
| Version | 2.0 |
| Deployed On | Sales Order |
| Triggered On | Before Load |
| Environment | Production |
| Status | Released |

**Script Link:** [**360CCC UE Create Work Order Button**](https://4449625.app.netsuite.com/app/common/scripting/script.nl?id=385&whence=)

**Logic Overview:**

**Function Name:** BeforeLoad

**Before Load Function:**

* If the user event type is not view then do not return
* If function **generation in progress** returns true, checks the session to see if this sales order has already been submitted.
* There is a **displayQueuedMessage** function displays a green banner across the top of the sales order if we have just been redirected back from the Suitelet. It also clears the session variable.
* Else If it calls function **buttonShouldDsiplay,** if status not equal ‘pending fulfillment’ and ‘Pending Billing/Partially Fulfilled’ this function won’t execute. If location does not exist, this function won’t execute.
* If the above condition satisfies, call the function **displayFulfillmentDateMessage**, displays a red banner across the top of the sales order if we have just been redirected back from the Suitelet and the fulfillment date is not today's date.
* Then call the function **add Button,** Generate Work Orders.