



MAXTECH

at



MUWG

MAXIMO UTILITY WORKING GROUP

Spring 2024

POWERED BY  BPD ZENITH

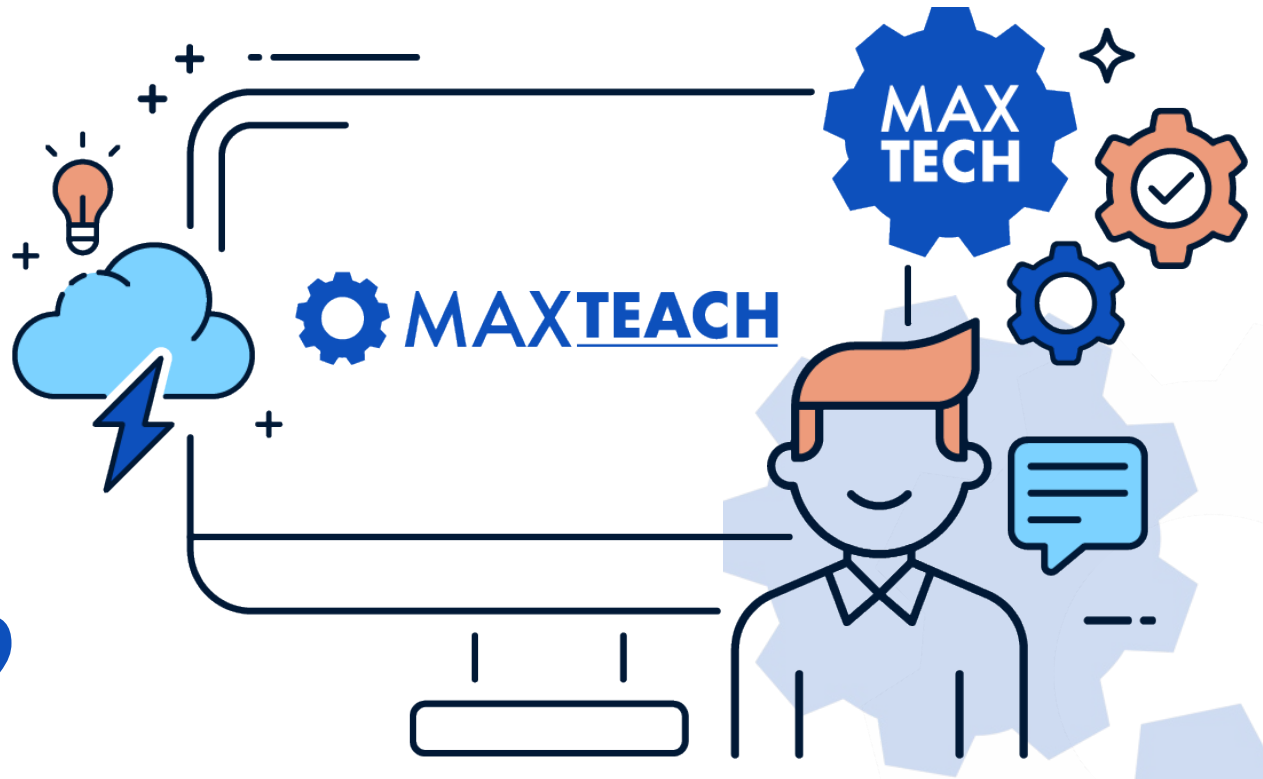
In partnership with



POWERED BY  BPD ZENITH

WHAT IS MAXTECH?

Founded in 2017 by BPD Zenith, MaxTECH is the first ever dedicated Maximo Technical User Group aimed at Maximo Administrators, Developers and Technical Support staff.



About BPD ZENITH

- Established in 1994 – Carlisle, England
- Global operations in North America, Asia Pacific and EMEA
- Leading IBM Global Partner in Maximo & IoT
- Proven provider of fully hosted and supported cloud-based IBM Maximo & AI solutions
- Dedicated Global Infrastructure & Software Support Team
- Experience across multiple asset-intensive industries
- Internal Product Development team who develop and maintain our award-winning industry accelerator templates and add-on solutions





CUSTOM APPLICATIONS USING JUST CONFIGURATION

Building an Application to Request New Items and Inventory in Maximo

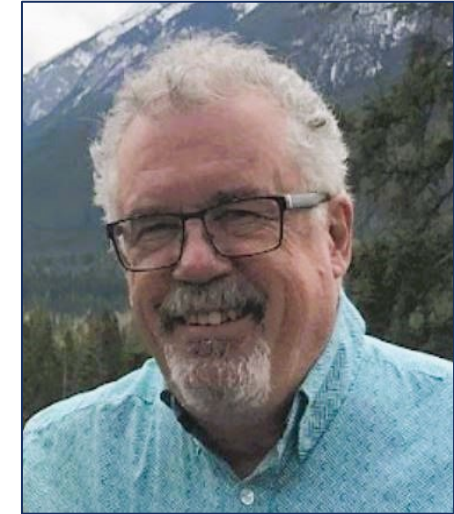
Presentation by: Stephen Hume | Maximo Director/Principal Consultant |



Introduction

Recent Projects:

- Maximo SME for Conversion from Asset Suite to Maximo
- Re-engineering of MOC process in Maximo
- Confined Space Permit Application in Maximo
- Safe Work Permit Application in Maximo
- Maximo Security and System Support – Hypercare
- Maximo Training – 2000 end users
- Upgrade to MAS 8 Planning and health check
- Maximo HSE (Incidents) Configuration and Implementation
- Maximo Supply Chain Improvements Project
- Design integration between Fastenal and Maximo
- Upgrade to 7.6.1.3
- Conversion from Asset Suite to Maximo
- Implementation of Solar Farms in Maximo



Stephen Hume
Maximo Director/Principal
Consultant

Stephen has accumulated more than two decades of experience working with Maximo and served as the former president of the Canadian Maximo User Group.

He is currently the Maximo Practice Director/Principal Consultant at Sheffield Scientific LLC.

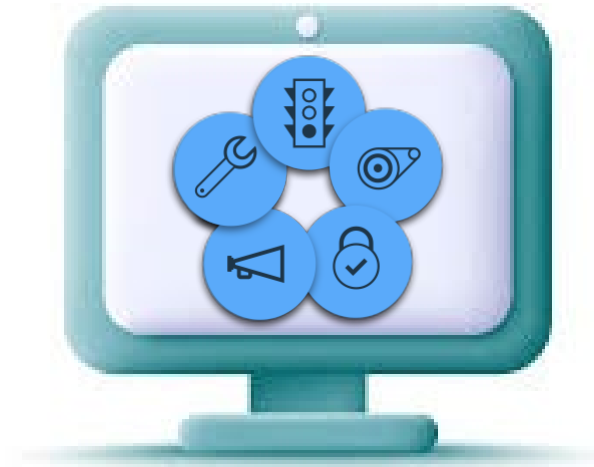
Overview

On a recent project, our customer requested the creation of an application in Maximo to allow end users to request the creation of a new Item in the Item Master and Inventory.

The business reason for this is that quite often after a plant outage, they would have spare materials left over that were a direct purchase, and rather than return them to the vendor they would like the item master created and the material placed in the storeroom for future use.

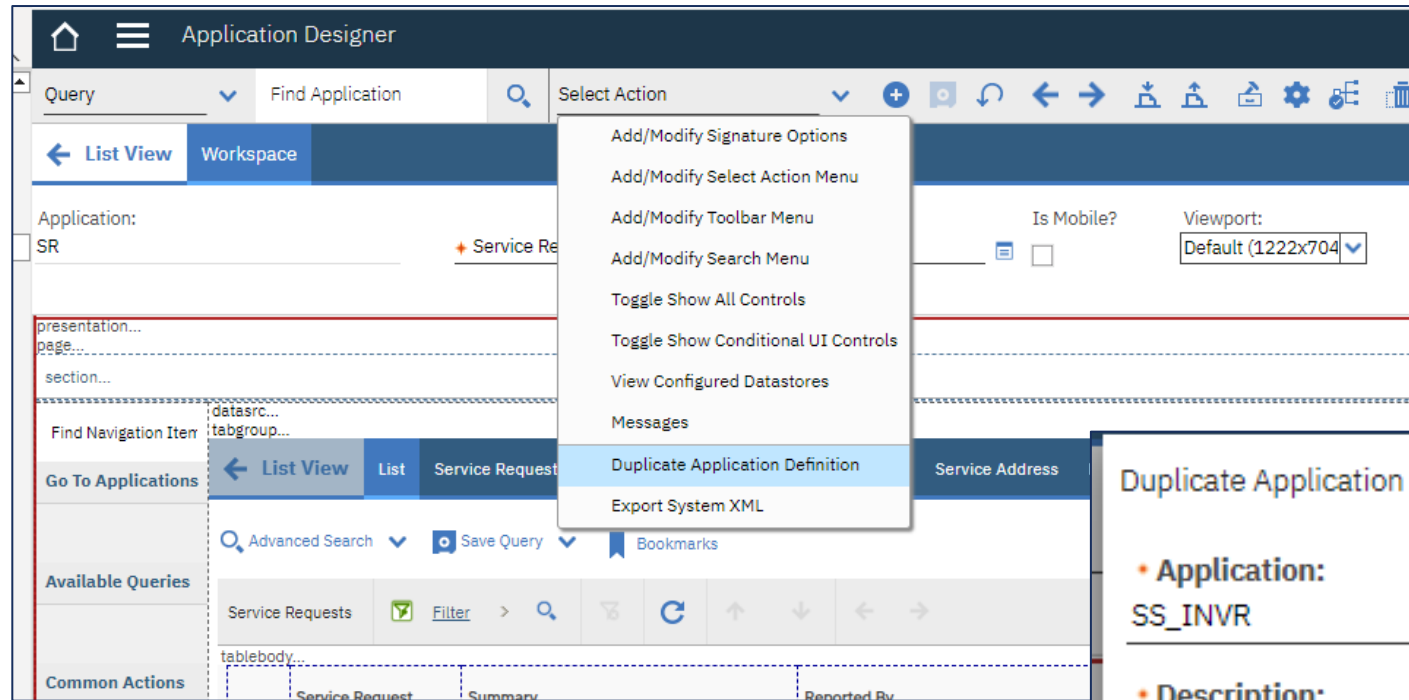
Prior to having an application to manage this process, it was done only some of the time, and became quite a burden for the warehouse personnel who had to manage pallets of miscellaneous parts left over from the outage. Those parts may have been purchased as Material or were purchased using P Cards and now need to be converted to Items and Inventory.

Demonstration

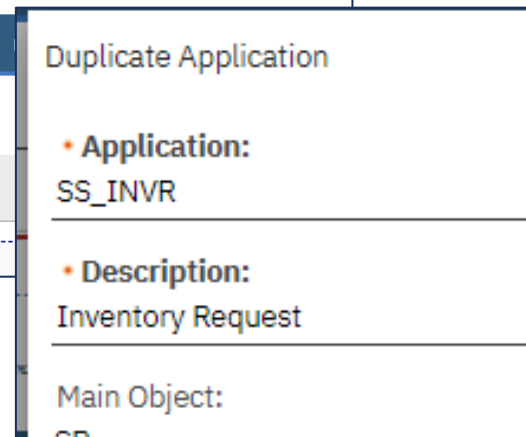


Creating the New Application

The application that will be created will be a clone of the Maximo Service Request Application.



In application designer open the SR application and then chose the select action menu item “Duplicate Application Definition”



Give the new application a Name (SS_INVR) with a description of Inventory Request

Move the application to a new module

Once the new application has been created and saved, you can move it into the new module you wish. (This is the menu item that the application will appear under for the end users.)

This can only be done by updating the back-end database using SQL.

Here is an example of the SQL needed to move the application to the Inventory Module.

```
update maxmenu  
set moduleapp = 'INVENTOR', POSITION = 95000, SUBPOSITION = 0  
where KEYVALUE = 'SS_INVR';
```

New Domains

SSSTOREROOM	Storerrooms	LOCATIONS	type = 'STOREROOM'	TABLE
SSCOMMGROUP	Commodity Group	COMMODITIES	:SSITEMSET = itemsetid and servicetype = 'PROCURE'	TABLE
SSMANUF	Get Manufacturers	COMPANIES	Type = 'M'	TABLE

Database Configuration

In order to capture the data needed to create Item and Inventory data in the new application several custom attributes need to be added to the TICKET table.

The SR is a VIEW of the Ticket Table so, changes should be made at the Ticket Table which will cascade to the view.

Attribute	Desc	Title	Type	Length	Domain	Search
SS TICKETTYPE	Ticket Type	Ticket Type	ALN	10		WILDCARD
SS ITEMNUM	Created Item	Created Item	Upper	30		WILDCARD
SS WHGROUP	Warehouse Group	Warehouse Group	Upper	30		
SS APPRGROUP	Inventory Approver Group	Approver Group				
SS ITEM AVAIL	Is there an item available to be put into the warehouse	Is there an item available to be put into the warehouse	YORN	1		EXACT
SS ITEMLOC	Where is the item currently located	Where is the item currently located	ALN	100		WILDCARD
SS ITEMORDER	Does the item need to be ordered	Does the item need to be ordered	YORN	1		EXACT
SS ITEM COST	Item Unit Cost	Unit Cost	Amount	10/2		EXACT
SS ORDERUNIT	Order Unit	Order Unit	Upper	16		WILDCARD
SS ESTANNUSAGE	Estimated Annual Usage	Estimated Annual Usage	FLOAT	8/0		EXACT
SS ESTQTYPERJOB	Estimated Quantity Per Job	Estimated Quantity Per Job	FLOAT	8/0		EXACT
SS SLTARGET	Service Level Target	Service Level Target	ALN	25		WILDCARD
SS ISREPSPARE	Is Repairable Spare	Is Repairable Spare	YORN	1		EXACT
SS REASON	Reason	Reason	ALN	50		WILDCARD
SS REASON_LONGDESCRIPTION			LONGALN	32000		TEXT
SS ORDERQTY	Order Quantity	Order Quantity	DECIMAL	15/2		EXACT
SS MINLEVEL	Reorder Point	Reorder Point	DECIMAL	15/2		EXACT
SS CCF	Cycle Count Frequency	Cycle Count Frequency	INTEGER	12		EXACT
SS ITEM COMMODITYGROUP	Commodity Group	Commodity Group	UPPER	8	SS COMMGRP	WILDCARD
SS STOREROOM	Storeroom	Storeroom	UPPER	30	SS STOREROOM	WILDCARD
SS BINNUM	Bin	Bin	ALN	40		WILDCARD
SS MODELNUM	Model Number	Model Number	ALN	50		WILDCARD
SS MANUFACTURER	Manufacturer	Manufacturer	UPPER	12		WILDCARD
SS ITEMSET	Itemset	Itemset	UPPER	8		WILDCARD
SS ISSUEUNIT	Issue Unit	Issue Unit	UPPER	14		WILDCARD

Database Configuration

New Relationships

Parent	Name	Where Clause	Child	Comments
SR	SSWORKLOG	RECORDKEY=:TICKETID AND CLASS=:CLASS AND WORKLOGID IN (SELECT WORKLOGID FROM WORKLOG WHERE CREATEDATE = (SELECT MAX(CREATEDATE) FROM WORKLOG WHERE RECORDKEY= :TICKETID AND CLASS=:CLASS))	WORKLOG	Get the latest work log entry
SR	COMPANIES	:vendor = COMPANY	COMPANIES	Get the Vendor details
SR	SSMANUF	:ssmanufacturer = COMPANY	Companies	Get the manufacturer details
SR	SSDEFINVWHGRP	persongroup in (select pg.persongroup from persongroup pg inner join persongroupteam pgt on pg.persongroup = pgt.persongroup where ath_isinvwhgroup = 1 and isnull(useforsite,(select defsite from maxuser where userid = :reportedby)) = (select defsite from maxuser where userid = :reportedby))	PERSONGROUP	Link to the default WH Group
SR	IMGLIB	refobject = 'SR' and refobjectid = :ticketuid	IMGLIB	Image linked to sr
SR	NEWIMGLIB	1=2	IMGLIB	Used to copy image to new item (inventory request)

Add new lookup for Manufacturers

Modify Lookups XML add a lookup for Manufacturers

```
<table id="manufacturers" inputmode="readonly" selectmode="single" whereclause=" type = 'M'">
  <tbody displayrowsperpage="20" filterable="true" filterexpanded="true" id="manuf_lookup_tablebody">
    <tablecol dataattribute="company" id="manuf_lookup_tablebody_col_0" mxevent="selectrecord" mxevent_desc="Go To %1" sortable="true" type="link"/>
    <tablecol dataattribute="name" id="manuf_lookup_tablebody_col_1" mxevent="selectrecord" mxevent_desc="Go To %1" sortable="true" type="link"/>
    <tablecol dataattribute="type" id="manuf_lookup_tablebody_col_2" mxevent="selectrecord" mxevent_desc="Go To %1" sortable="true" type="link"/>
    <tablecol dataattribute="orgid" id="manuf_lookup_tablebody_col_3" mxevent="selectrecord" mxevent_desc="Go To %1" sortable="true" type="link"/>
  </tbody>
</table>
```

Added Features

- Make Workflow Comments Required
- Create Log Entry from Workflow Comments
- Display Latest Log Entry on List Tab and Main Tab
- Add IMGLIB to the Application
- Create Item Master when Request is Approved
- Create Inventory Record when Request is Approved
- Create Vendor information when Request is Approved
- Create Inventory Balance information when Request is Approved
- Make fields conditionally required
- Erase data in the balance field if check box is unchecked
- Change the default numbering when a new record is created
- Create BIRT report for List of Inventory Requests and Inventory Request Details
- Conditionally Hide Sections of the Application

Make Workflow Comments Required

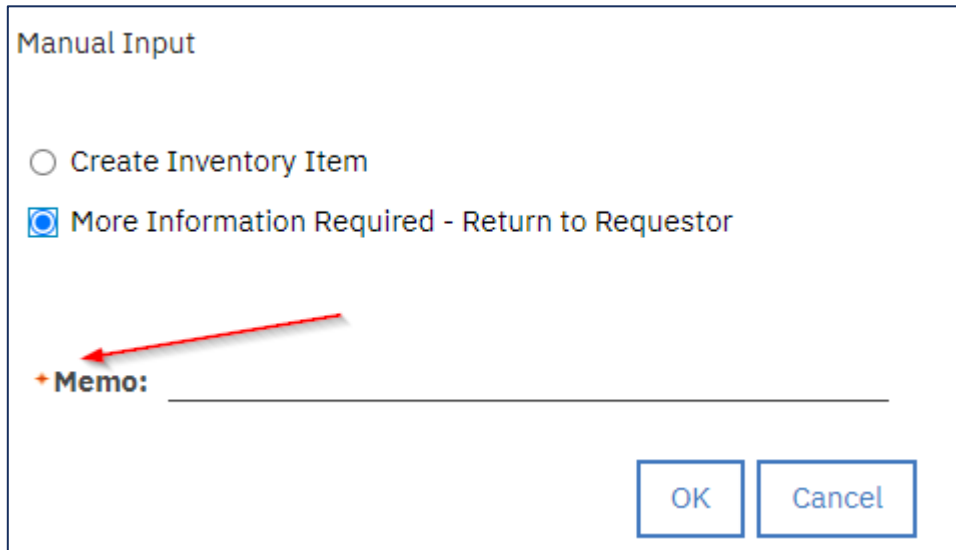
Manual Input

Create Inventory Item

More Information Required - Return to Requestor

+ Memo: _____

OK Cancel



Make Workflow Comments Required

1. Add a **YORN** field to **WFACTION** in Database Configuration
 - Title : "Memo Required"
2. Add a **YORN** field to **SHOWWFACTION** in Database Configuration
 - Title : "Memo Required"
 - Turn on Admin Mode
 - Apply Configuration Changes

Attribute: <u>SS_MEMOREQD</u>	♦ Title: <u>Memo Required</u>
Description: <u>Memo is required when this option is checked</u>	Class: _____
♦ Type: <u>YORN</u> 🔍	Domain: _____ >



In the SHOWWFACTION table this attribute is set up as non-persistent
Make sure the persistent check box is unchecked before you save your changes.

Make Workflow Comments Required

In Application Designer,
open **WFDESIGN**
and edit the following dialogs:

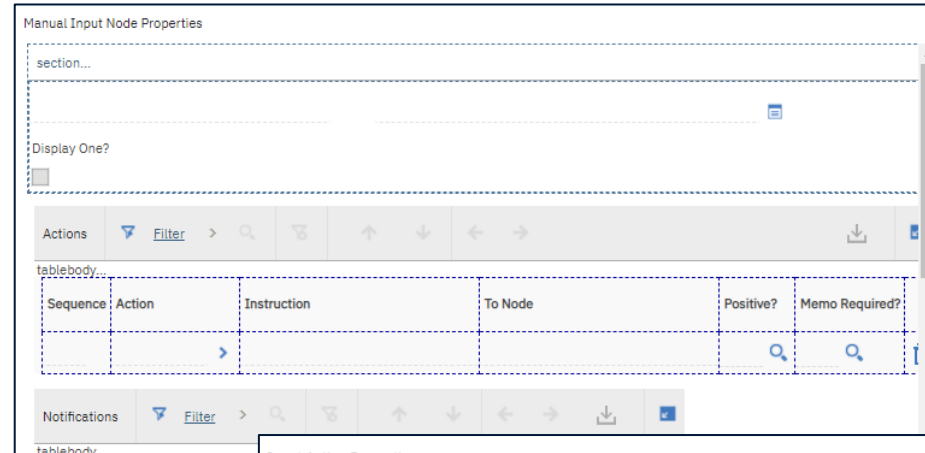
- `wfinputproperties`
- `wfinputactionproperties`

The screenshot shows the 'Application Designer' interface with the 'Edit Dialogs' dialog box open. The dialog box displays a list of dialog properties. The table below is a representation of the data shown in the dialog box.

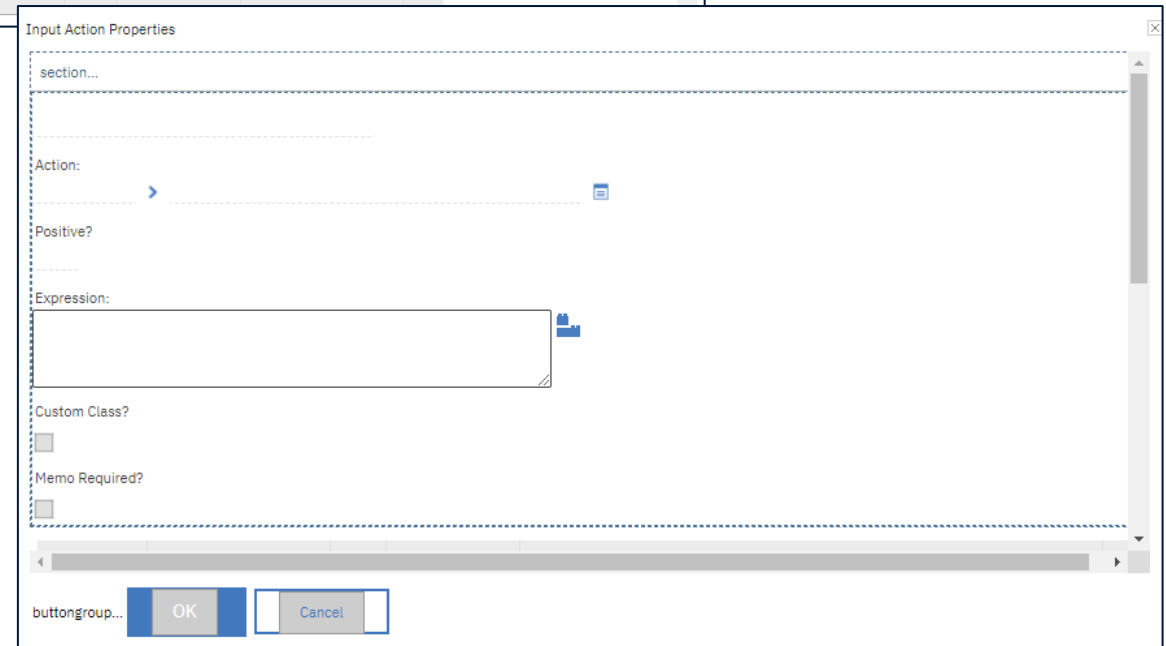
Control ID	Label
<code>wfinputproperties</code>	Manual Input Node Properties
<code>wfstoppproperties</code>	Stop Node Properties
<code>viewsync</code>	View Synchronized Processes
<code>wfinteractionproperties</code>	Interaction Node Properties
<code>wfconditionproperties</code>	Condition Node Properties
<code>wfinputactionproperties</code>	Input Action Properties
<code>wfsubprocessproperties</code>	Subprocess Node Properties
<code>wfstartproperties</code>	Start Node Properties
<code>wfactionproperties</code>	Action Properties
<code>wftaskproperties</code>	Task Node Properties
<code>appsupport</code>	Add Workflow Support to Applications
<code>wfwaitproperties</code>	Wait Node Properties
<code>apptoolbar</code>	Edit Workflow GO Buttons
<code>searchbook</code>	My Bookmarks
<code>searchmore</code>	More Search Fields

Make Workflow Comments Required

Add the **MEMOREQD** field as a checkbox on a table column in dialog wfinputproperties

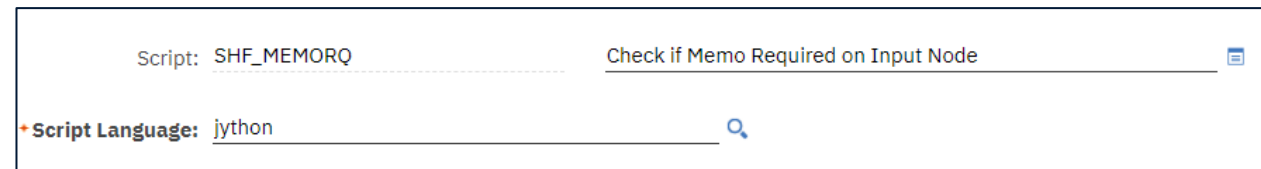
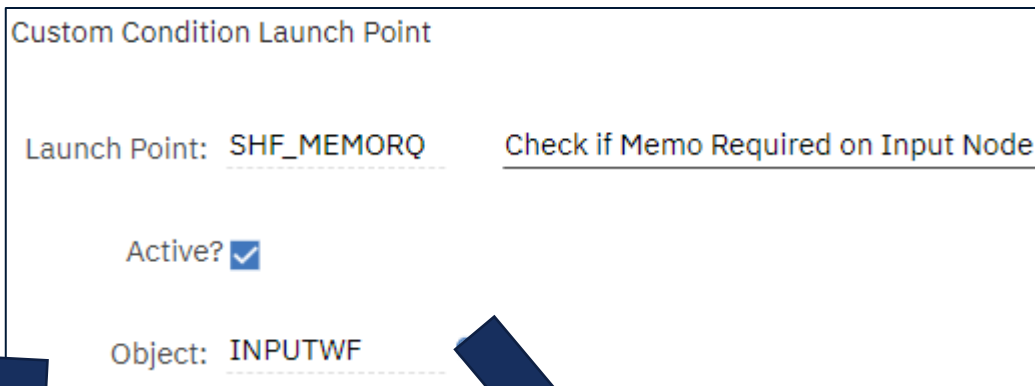
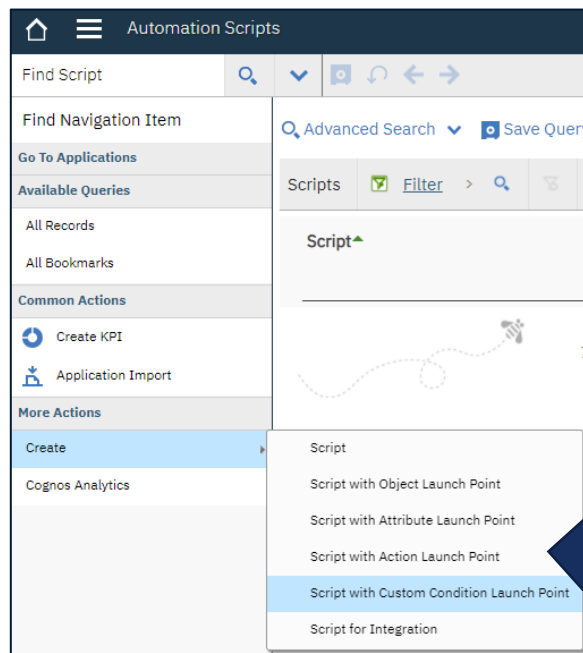


Add the **MEMOREQD** field as a checkbox in dialog winputactionproperties:



Make Workflow Comments Required

Create custom condition automation script



Make Workflow Comments Required

Create custom condition automation script

```
#####  
# autoscript: KEY_WFMEMORQ - Make memo required on Manual Input nodes when required by WF  
# launchpoints : KEY_WFMEMORQ - Condition launch point  
#  
# Determines if the memo field is required for the selected option. Set in workflow designer  
# on WFACTION object.  
#  
#####  
if (launchPoint == "SHF_WFMEMORQ"):  
    bResult = False  
    mboOwner = mbo.getOwner()  
    if mboOwner is not None:  
        mboWFProcess = mboOwner.getMboSet("WFPROCESS").moveFirst()  
        if (mboWFProcess is not None):  
            strActionId = mbo.getString("ACTIONID")  
            strProcessName = mboOwner.getString("PROCESSNAME")  
            strProcessRev = mboWFProcess.getString("processrev")  
            strWhere = "actionid = " + strActionId  
            strWhere = strWhere + " and processname = '" + strProcessName + "'"  
            strWhere = strWhere + " and PROCESSREV = " + strProcessRev + " and shf_memoreqd=1"  
            mboSetWFActionMemoRqd = mbo.getMboSet("$TEMPREL1", "WFACTION", strWhere)  
  
            if (mboSetWFActionMemoRqd is not None):  
                mboWFActionMemoRqd = mboSetWFActionMemoRqd.moveFirst()  
                if (mboWFActionMemoRqd is not None):  
                    bResult = True  
evalresult = bResult
```

```
#####  
# autoscript: KEY_WFMEMORQ - Make memo required on Manual Input nodes when required by WF  
# launchpoints : KEY_WFMEMORQ - Condition launch point  
#  
# Determines if the memo field is required for the selected option. Set in workflow designer  
# on WFACTION object.  
#  
#####  
if (launchPoint == "SHF_WFMEMORQ"):  
    bResult = False  
    mboOwner = mbo.getOwner()  
    if mboOwner is not None:  
        mboWFProcess = mboOwner.getMboSet("WFPROCESS").moveFirst()  
        if (mboWFProcess is not None):  
            strActionId = mbo.getString("ACTIONID")  
            strProcessName = mboOwner.getString("PROCESSNAME")  
            strProcessRev = mboWFProcess.getString("processrev")  
            strWhere = "actionid = " + strActionId  
            strWhere = strWhere + " and processname = '" + strProcessName + "'"  
            strWhere = strWhere + " and PROCESSREV = " + strProcessRev + " and shf_memoreqd=1"  
            mboSetWFActionMemoRqd = mbo.getMboSet("$TEMPREL1", "WFACTION", strWhere)  
  
            if (mboSetWFActionMemoRqd is not None):  
                mboWFActionMemoRqd = mboSetWFActionMemoRqd.moveFirst()  
                if (mboWFActionMemoRqd is not None):  
                    bResult = True  
evalresult = bResult
```

Make Workflow Comments Required

Create Conditional Expression



Create the Condition that will use Custom Condition Script we created:

Description: **Check if Memo is required on INPUTWF**

Type: **CLASS**

Expression: **SHF_MEMORQ:SHF_WFMEMORQ**

Class: **com.ibm.tivoli.maximo.script.ScriptCustomCondition**

Condition:	<u>SHF_MEMORQ</u>	<u>Check if Memo is required on INPUTWF</u>
+ Type:	<u>CLASS</u>	
Expression:	<input type="text" value="SHF_MEMORQ:SHF_MEMORQ"/>	
+ Class:	<u>com.ibm.tivoli.maximo.script.ScriptCustomCondition</u>	

Make Workflow Comments Required

Add Global Data Restriction

Create a required attribute restriction on object **INPUTWF** attribute **MEMO** in the Security Groups application.

Object: **INPUTWF**

Attribute: **MEMO**

Type: **REQUIRED**

Condition: **SHF_WFMEMO**

Update Library XML to make UI more responsive

If the MEMO field seems sluggish in becoming required/not required, we can make the radio button group in the "inputwf" dialog "synchronous". Setting a control as "synchronous" will ensure all UI adjustments are applied prior to letting users make additional changes.

```
<dialog beanclass="psdi.webclient.system.beans.WorkflowBean" id="inputwf" label="Manual Input">
  <section id="inputwf_sec1">
    <radiobuttongroup dataattribute="ACTIONID" descattribute="instruction" id="inputwf_t2" valueattribute="actionid" synchronous="true"/>
  </section>
  <section id="inputwf_sec2">
    <textbox dataattribute="MEMO" id="inputwf_t3"/>
  </section>
  <buttongroup id="inputwf_bg1">
    <pushbutton default="true" id="inputwf_b_1" label="OK" mxevent="dialogok"/>
    <pushbutton id="inputwf_b_2" label="Cancel" mxevent="dialogcancel"/>
  </buttongroup>
</dialog>
```


Make Workflow Comments Required

Setting Memo to be required from Interaction Node properties for the "CANCEL" option

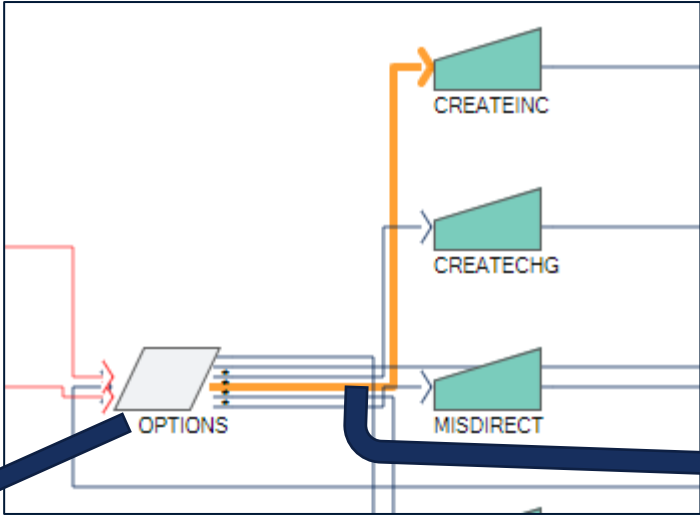
Manual Input Node Properties

Title: OPTIONS SR Options

Display One?

Actions Filter > 1 - 4 of 6

Sequence	Action	Instruction	To Node	Positive?	Memo Required?
1	SR CREATEINCI >	Create Incident	CREATEINC	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	SR CREATECHG >	Create Change	CREATECHG	<input checked="" type="checkbox"/>	<input type="checkbox"/>



Setting Memo to be required from Input Action properties for the "Create Incident" option

Input Action Properties

Instruction: Create Incident

Action: SR CREATEINCI > Create Incident Action

Positive? Y

Expression:

Memo Required?

Create Log Entry from Workflow Memo

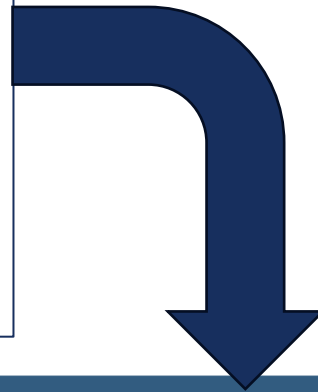
Manual Input

Create Inventory Item

More Information Required - Return to Requestor

+Memo: Please attach a vendor quote

OK Cancel



Work Log		Communication Log					
Record	Class	Created By	Date	Type	Summary	Viewable?	
> 1211	> SR	WILSON	2/26/24 2:07 PM	WORKFLOW	Please attach a vendor quote	<input type="checkbox"/>	

New Row

Create Log Entry from Workflow Memo

1. Configuration

- i. Add attributes to the WFTRANSACTION table (only needed for the automation script)
- ii. Create relationship on the WFTRANSACTION object.

2. Add the logic

- i. Create automation script

3. Extras

- i. Create a new memo type
- ii. Modify the automation script to use the new memo type

Create Log Entry from Workflow Memo

1. Add a **SITEID** field to **WFTRANSACTION** in Database Configuration
 - Title : “Siteid”
 - Description: Site needed for automation script
2. Add a **ORGID** field to **WFTRANSACTION** in Database Configuration
 - Title : “Orgid”
 - Description: orgid needed for automation script
3. Turn on Admin Mode
4. Apply Configuration Changes
5. Turn off Admin Mode

The image shows two screenshots of a database configuration interface. The top screenshot displays the 'Details' section for a field named 'SITEID'. It includes fields for 'Attribute: SITEID', 'Title: Siteid', 'Description: Site needed for automation script', 'Type: UPPER', 'Length: 8', 'Scale: 0', and 'Required?' (unchecked). The 'Advanced' section shows 'Entity: WFTRANSACTION', 'Column: SITEID', 'Same as Object: SITE', 'Same as Attribute: SITEID', 'Autonumber:', and 'Search Type: NONE'. The bottom screenshot shows the 'Details' section for a field named 'ORGID'. It includes fields for 'Attribute: ORGID', 'Title: Orgid', 'Description: orgid needed for automation script', 'Type: UPPER', 'Length: 8', 'Scale: 0', and 'Required?' (unchecked). The 'Advanced' section shows 'Entity: WFTRANSACTION', 'Column: ORGID', 'Same as Object: ORGANIZATION', 'Same as Attribute: ORGID', and 'Autonumber:'.

Create Log Entry from Workflow Memo

Create relationship on the wftransaction object

Relationship: SHF_WORKLOG

Where Clause: 1=2

Child Object: WORKLOG

Remarks: Link to the worklog object

Relationship: SHF_WORKLOG	Child Object: WORKLOG 
Where Clause: 1=2	Remarks: Link to the worklog object (no records retrieved)

Create Log Entry from Workflow Memo

Create new entry in LOGTYPE Domain

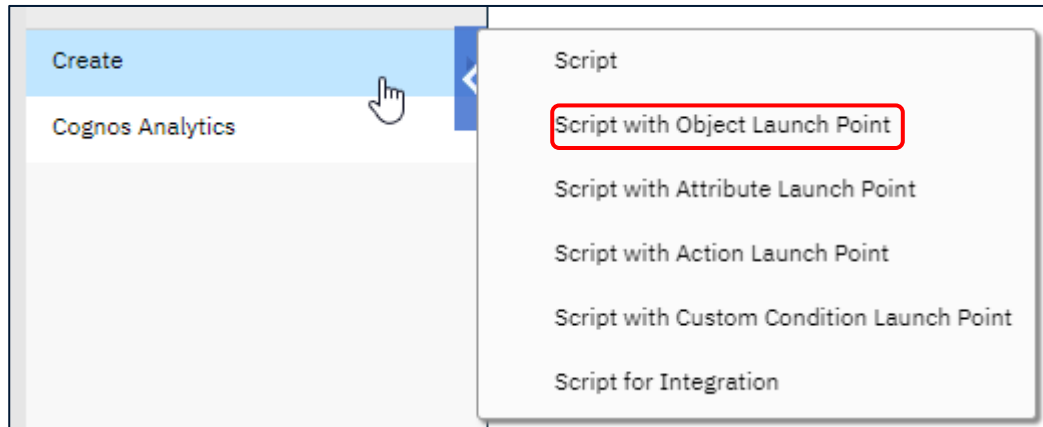
The new log type is created to allow the entries created from a workflow memo to be clearly identified when viewing the worklog entries.

♦ Internal Value: UPDATE
♦ Value: WORKFLOW
Description: Worklog created from workflow memo

Create Log Entry from Workflow Memo

Create Automation Script

Create SCRIPT with object launchpoint



Launch Point: SHFMEMOTOLOG

Object: WFTRANSACTION

Event: Save, Add, After Save

Create Log Entry from Workflow Memo

Create Automation Script

SCRIPT:

SHFMEMOTOLOG

DESCRIPTION:

copy workflow memo to worklog

SCRIPT LANGUAGE:

python



The indents are critical in the script

Note: This script is specifically designed to work with the Inventory request application and could be modified to work with any application that has associated workflow and worklog functions

```
#####  
# autoscript: SHFMEMOTOLOG - Create worklog entry from workflow memo  
# launchpoints : SHFMEMOTOLOG - Object launch point  
#####  
  
from psdi.mbo import MboConstants;  
from psdi.server import MXServer;  
  
#Check if memo is populated and WF transaction is for Inventory Request  
if mbo.getString('processname') == 'SS_INVR' and not mbo.isNull('memo') and mbo.getString('ownertable') == 'SR':  
    srSet = MXServer.getMXServer().getMboSet('sr', mbo.getUserInfo())  
    srSet.setWhere("ticketuid = '" + str(mbo.getInt('ownerid')) + "'")  
    srSet.setFlag(MboConstants.DISCARDABLE, True)  
    srSet.reset()  
  
    srMbo = srSet.moveFirst()  
  
    if srMbo:  
        workLogSet = mbo.getMboSet("SHF_WORKLOG")  
        workLogSet.setWhere("i=0")  
        workLogSet.reset()  
  
        #Add work log and populate with work order details  
  
        workLogMbo = workLogSet.add()  
  
        workLogMbo.setValue("recordkey", srMbo.getString("ticketid"), MboConstants.NOACCESSCHECK)  
        workLogMbo.setValue("class", "SR", MboConstants.NOACCESSCHECK)  
        workLogMbo.setValue("logtype", 'WORKFLOW', MboConstants.NOACCESSCHECK)  
        workLogMbo.setValue("createby", mbo.getString('personid'), MboConstants.NOACCESSCHECK)  
        workLogMbo.setValue("description", mbo.getString('memo'), MboConstants.NOACCESSCHECK)  
        workLogMbo.setValue("modifyby", mbo.getString('personid'), MboConstants.NOACCESSCHECK)  
  
        workLogSet.save()  
    else:  
        None  
else:  
    None
```

Display Latest Work Log Entry

Inventory Request	Item Description	Reported By	Status	Vendor	Company Name	Latest Log Entry
1208	super hero shirt, cotton, xxl	WILSON	PENDING			
1211	super hero shirt, cotton, xxl	WILSON	RETURNED			Please attach a vendor quote
1212	super hero shirt, cotton, med	WILSON	WAPPR			


Inventory Request

Inventory Request ID: [1211](#) Reported By: [WILSON](#) [Mike Wilson](#)

Status: [RETURNED](#) Reported Date: [1/16/24 1:14 PM](#)

+ Site: [BEDFORD](#) Ticket Type: [INVR](#)

Created Item: [1054](#) Latest Log Entry: [Please attach a vendor quote](#)

[Attachments](#) 

Display Latest Work Log Entry

Create relationship in the SR database Object.

Relationship: [SSWORKLOG](#)

Where Clause:

```
RECORDKEY=:TICKETID AND CLASS=:CLASS AND WORKLOGID IN ( SELECT WORKLOGID FROM WORKLOG WHERE CREATEDATE = ( SELECT MAX(CREATEDATE) FROM WORKLOG WHERE RECORDKEY= :TICKETID AND CLASS=:CLASS ))
```

Child Object: [worklog](#)

Remarks: [Get the latest worklog entry](#)

Relationship: [SSWORKLOG](#)

Child Object: [WORKLOG](#) 🔍

Where Clause: RECORDKEY=:TICKETID AND CLASS=:CLASS AND WORKLOGID IN (SELECT WORKLOGID FROM WORKLOG WHERE CREATEDATE = (SELECT MAX(CREATEDATE) FROM WOR

Remarks: Get the latest work log entry

Service Request: 1189	Owner: _____	Owner Group: TIER1	Status: QUEUED
Feature Class: _____	Memo: Incident created to investigate issue		

Display Latest Work Log Entry

Add table column and text box to the application using application designer.

The screenshot displays an application designer interface. On the left, a table is shown with a column header 'Latest Log Entry' highlighted in green. A context menu is open over this column, showing the following properties:

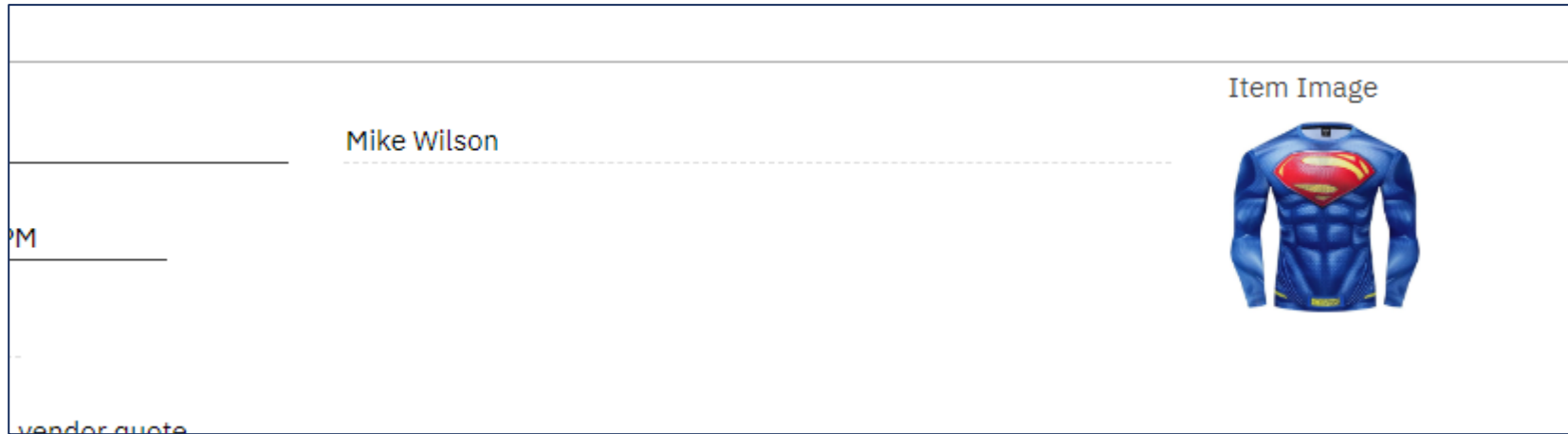
- Control ID: 1701798198025
- Default Label: Summary
- Label: Latest Log Entry
- Attribute: SSWORKLOG.DESRIPTION

On the right, the 'Textbox Properties' panel is open, showing the following settings:

- Control ID: 1701793465360
- Default Label: Summary
- Label: Latest Log Entry
- Hide Label?
- Turn Smart Fill Off?
- Attribute: SSWORKLOG.DESRIPTION

The background shows a form with various fields: 'Request ID', 'Status', 'Site' (with a search icon), 'Created Item' (with a right arrow), 'Reported By', 'Reported Date', 'Ticket Type', 'Latest Log Entry' (highlighted in green), and 'Attachments' (with a link icon).

Add IMGLIB to the Application







The steps to add Image functionality to an application are described in the IBM article linked here:
<https://www.ibm.com/support/pages/adding-picture-image-record-within-application>

One Automation Script to do the rest

The last step required to complete the process of creating new item master and inventory information for the Inventory Request.



Create an automation script with an **Action Launch point**. This action can then be used in workflow once the IR has its final approval.


Script: SSSRCRIT	Create Item from Inventory Request	
+ Script Language: python		

Action Launch Point	
Launch Point: SSSRCRIT	Create Item from Inventory Request
Active? <input checked="" type="checkbox"/>	
Object: SR	

Create Item Master When Approved

Inventory Request

Inventory Request ID: <u>INVR1216</u>	Reported By: <u>WILSON</u>	<u>Mike Wilson</u>
Status: <u>CLOSED</u>	Reported Date: <u>2/13/24 9:37 AM</u>	
Site: <u>BEDFORD</u> 	Ticket Type: <u>INVR</u>	
Created Item: <u>1071</u> 	Latest Log Entry: _____	

[Attachments](#) 

Create Item Master When Approved

The first part of the automation script does the necessary imports and sets up the necessary values.

It then using a dynamic MboSet to get the information for an Item Master record with the Item Number of 10000. This is a "dummy" item created in the item master solely for the purpose of creating new items.

The script uses a function called duplicate() to make a new copy of the dummy item, and then populates key attributes with data from the inventory request.

Once that is completed, the new item is saved (which is then available for use in the rest of the script.

```
from psdi.mbo import MboConstants
from psdi.server import MXServer
from org.apache.axiom.om.util import Base64

invbalSet = mbo.getMboSet("SS_INVBALANCES")
curBal = mbo.getDouble("SS_CURBAL")
storerm = mbo.getString("SSSTOREROOM")
org = mbo.getString("ORGID")

prefix = ""
dynclause = " itemnum = '10000' "

itemset = mbo.getMboSet("$ss_citemset","ITEM",dynclause)
print dynclause
print itemset
site = mbo.getString("SITEID")

#create the item
curItem = itemset.moveFirst()
newitem = curItem.duplicate()

newitem.setValue("ITEMSETID","SET1")
newitem.setValue("DESCRIPTION",mbo.getString("DESCRIPTION"))
newitem.setValue("ORDERUNIT",mbo.getString("SSORDERUNIT"))
newitem.setValue("ISSUEUNIT",mbo.getString("SSORDERUNIT"))
newitem.setValue("COMMODITYGROUP",mbo.getString("SSITEMCOMMODITYGROUP"))
newitem.setValue("ITEMNUM",prefix + newitem.getString("ITEMNUM"))
newitem.validate()

mbo.setValue("SSITEMNUM",newitem.getString("ITEMNUM"))

itemset.save()
```

Create Inventory Record When Approved

```
43 # now add inventory
44
45 if mbo.isNull("SSSTOREROOM") is False:
46     storedynclause = "location = '"+mbo.getString("SSSTOREROOM")+''
47     locationSet = mbo.getMboSet("$ss_locstoreset","LOCATIONS",storedynclause)
48     store = locationSet.moveFirst()
49     invset = store.getMboSet("NEW_INVENTORY")
50     newInventoryMbo = invset.addAtEnd()
51     newInventoryMbo.setPropagateKeyFlag(False);
52     newInventoryMbo.setValue("siteid", store.getString("SITEID"), 11L);
53     newInventoryMbo.setValue("orgid", "EAGLENA", 2L);
54     newInventoryMbo.setPropagateKeyFlag(True);
55     newInventoryMbo.setValue("location", mbo.getString("SSSTOREROOM"), 3L);
56     newInventoryMbo.setValue("itemsetid", newitem.getString("itemsetid"), 2L);
57     newInventoryMbo.setValue("itemnum", newitem.getString("itemnum"), 3L);
58     newInventoryMbo.setValue("orderunit", newitem.getString("orderunit"), 3L);
59     newInventoryMbo.setValue("issueunit", newitem.getString("issueunit"), 2L);
60     newInventoryMbo.setFieldFlag("conditioncode", 7L, True);
61     newInventoryMbo.setFieldFlag("condrate", 7L, True);
62
63     itemOrgInfoSet = newInventoryMbo.getMboSet("ITEMORGINFO");
64     if ((itemOrgInfoSet.isEmpty()) is False) :
65         newInventoryMbo.setValue("status", itemOrgInfoSet.getMbo(0).getString("status"), 3L);
66
67     itemOrgInfoSet.cleanup();
68     newInventoryMbo.setCostType();
69     newInventoryMbo.setValue("BINNUM",mbo.getString("SSBINNUM"))
70     newInventoryMbo.setValue("VENDOR",mbo.getString("VENDOR"))
71     newInventoryMbo.setValue("MANUFACTURER",mbo.getString("SSMANUFACTURER"))
72     newInventoryMbo.setValue("MODELNUM",mbo.getString("SSMODELNUM"))
73     newInventoryMbo.setValue("MINLEVEL",mbo.getString("SSMINLEVEL"))
74     newInventoryMbo.setValue("ORDERQTY",mbo.getString("SSORDERQTY"))
75     newInventoryMbo.validate()
76     invset.save()
```

← List View	Inventory	Reorder Details	Rotating Assets	Where Used
Item: 1071	>	test of inventory request		
Storeroom: CENTRAL	>	Central Storeroom		
Lot Type: NOLOT				

Create Vendor Data When Approved

Vendor	Manufacturer	Model	Catalog #	Order Unit	Last Price	Last Order Date	Organization	Site
> GRAINGER >	>	123456		EACH 🔍		2/13/24 📅	EAGLENA 🔍	BEDFORD

```
#update invvnedor
  invvendset = newInventoryMbo.getMboSet("INVVENDOR")
  invvnedor = invvendset.moveFirst()

  invvnedor.setValue("MODELNUM",mbo.getString("SSMODELNUM"),MboConstants.NOACCESSCHECK|MboConstants.NOVALIDATION)
  invvnedor.setValue("ISDEFAULT",True,MboConstants.NOACCESSCHECK|MboConstants.NOVALIDATION)

  invset.save()
```

Create Inventory Balance When Approved

Bin	Lot	Condition Code	Current Balance	Staged Balance	Staging Bin
> 3			5.00	0.00	<input type="checkbox"/>

```
78 #setup inventory cost
79
80   invcostset = newInventoryMbo.getMboSet("INVCOST")
81
82   invcost = invcostset.moveFirst()
83
84   invcost.setValue("STDCOST",mbo.getDouble("SSITEMCOST"),MboConstants.NOACCESSCHECK|MboConstants.NOVALIDATION )
85   invcost.setValue("LASTCOST",mbo.getDouble("SSITEMCOST"),MboConstants.NOACCESSCHECK|MboConstants.NOVALIDATION )
86
87
88   invset.save()
89
```

```
#setup inventory balance
if curBal > 0:
    invbalMbo = invbalSet.add(2L)
    invbalMbo.setValue("BINNUM", mbo.getString("SSBINNUM"),MboConstants.NOACCESSCHECK|MboConstants.NOVALIDATION)
    invbalMbo.setValue("CURBAL", curBal,MboConstants.NOACCESSCHECK|MboConstants.NOVALIDATION)
    invbalMbo.setValue("PHYSCNT", curBal,MboConstants.NOACCESSCHECK|MboConstants.NOVALIDATION)
    invbalMbo.setValue("ITEMNUM", newitem.getString("itemnum"),MboConstants.NOACCESSCHECK|MboConstants.NOVALIDATION)
    invbalMbo.setValue("SITEID", site,MboConstants.NOACCESSCHECK|MboConstants.NOVALIDATION)
    invbalMbo.setValue("ORGID", org,MboConstants.NOACCESSCHECK|MboConstants.NOVALIDATION)
    invbalMbo.setValue("RECONCILED", "0", 2L)
    invbalMbo.setValue("LOCATION", storerm,MboConstants.NOACCESSCHECK|MboConstants.NOVALIDATION)
    invbalMbo.setValue("itemsetid", 'SET1',MboConstants.NOACCESSCHECK|MboConstants.NOVALIDATION)

    invbalSet.save()
```

Copy Image from Inventory Request to Item

```
# If there is an image for the item on the inventory request, copy it to the item.

newitemid = newitem.getString("itemid")
imglibSet = mbo.getMboSet("IMGLIB")
newimgSet = mbo.getMboSet("NEWIMGLIB")

imgMbo = imglibSet.moveFirst()

while imgMbo is not None:
    itemimg = newimgSet.add()
    itemimg.setValue("REFOBJECT", "ITEM")
    itemimg.setValue("REFOBJECTID", newitemid)
    itemimg.setValue("MIMETYPE", imgMbo.getString("MIMETYPE"))
    itemimg.setValue("IMAGENAME", imgMbo.getString("IMAGENAME"))
    itemimg.setValue("IMAGE", imgMbo.getBytes("IMAGE"))
    itemimg.setValue("ENDPOINTNAME", imgMbo.getString("ENDPOINTNAME"))
    itemimg.setValue("IMGURI", imgMbo.getString("IMGURI"))
    imgMbo = imglibSet.moveToNext()

# end of image copy
```

Add Asset(s) Spare Parts

```
# if there is list of multiple assets on the inventory request add the item created to each of the assets spare parts list
assetlistSet = mbo.getMboSet("ALLMULTIASSETLOCCI")
assetlistMbo = assetlistSet.moveFirst()

while assetlistMbo is not None:
    if (not assetlistMbo.isNull("ASSETNUM")):
        assetNumList = assetlistMbo.getString("ASSETNUM")
        org = assetlistMbo.getString("ORGID")
        site = assetlistMbo.getString("SITEID")
        item = newitem.getString("ITEMNUM")
        sparepartlistSet = mbo.getMboSet("SSSPAREPART")
        sparepartlist = sparepartlistSet.add(2L)
        sparepartlist.setValue("ASSETNUM", assetNumList, MboConstants.NOACCESSCHECK | MboConstants.NOVALIDATION)
        sparepartlist.setValue("ITEMNUM", item, MboConstants.NOACCESSCHECK | MboConstants.NOVALIDATION)
        sparepartlist.setValue("ITEMSETID", "SET1", MboConstants.NOACCESSCHECK | MboConstants.NOVALIDATION)
        sparepartlist.setValue("ORGID", org, MboConstants.NOACCESSCHECK | MboConstants.NOVALIDATION)
        sparepartlist.setValue("SITEID", site, MboConstants.NOACCESSCHECK | MboConstants.NOVALIDATION)
        assetlistMbo = assetlistSet.moveToNext()

# end asset list spare parts
```

Make Fields Conditionally Required

Step 1: Create a conditional expression which checks to see if the checkbox is checked.

Step 2: Create signature option called CURBALREQ

Step 3: Add the signature option to the advanced tab for the current balance field in application designer

Step 4: Configure the conditional properties for the signature option

Step 5: Grant access to this signature option in the security groups application

Is there an item available to be put into the warehouse?

+ Balance to put into Inventory: _____

Condition: SSITEMAVAIL Item Available is checked on

+ Type: EXPRESSION

+ Expression: :ssitemavail = 1

Class:

Option: CURBALREQ

Description: Current Balance Required

Signature Option: CURBALREQ

Name	Description
EVERYONE	All Maximo Users

Name	Description
SSITEMAVAIL	Item A

Property Values for Condition SSITEMAVAIL	
Condition is true	Condition is false
Property	Value
inputmode	required

Erase Data in Field if Checkbox Unchecked

Is there an item available to be put into the warehouse?

+ Balance to put into Inventory: _____ 10

Is there an item available to be put into the warehouse?

Balance to put into Inventory: _____

If the user unchecks the item available checkbox then the balance to be put into inventory should be blanked out.

This is achieved by creating an automation script with an **attribute launch point**.

Launch Point: SS_ITEMAVAIL Item Available launchpoint Active?

Object: SR Attribute: SSITEMAVAIL

```
Created by: Stephen Hume

This script performs the following functions:
If item available is unchecked the current balance is set to null

"""
# Imports
from psdi.app.common import DateUtility
from psdi.mbo import MboConstants
from psdi.server import MXServer

# Initialise variables
attributename = mbovalue.getName()

if attributename == "SSITEMAVAIL":
    avail = mbo.getInt("SSITEMAVAIL")
    if avail == 0:
        mbo.setValue("SS_CURBAL","", MboConstants.NOACTION and MboConstants.NOACCESSCHECK)
```


Change Default Numbering for New Record

In this case, the business wanted the Ticket ID to have a specific prefix to easily differentiate it from other types of tickets that might be created in the SR application.

This was achieved by creating a special automation script called **SR.NEW**. It does not have a launch point but gets triggered every time a new SR is created.

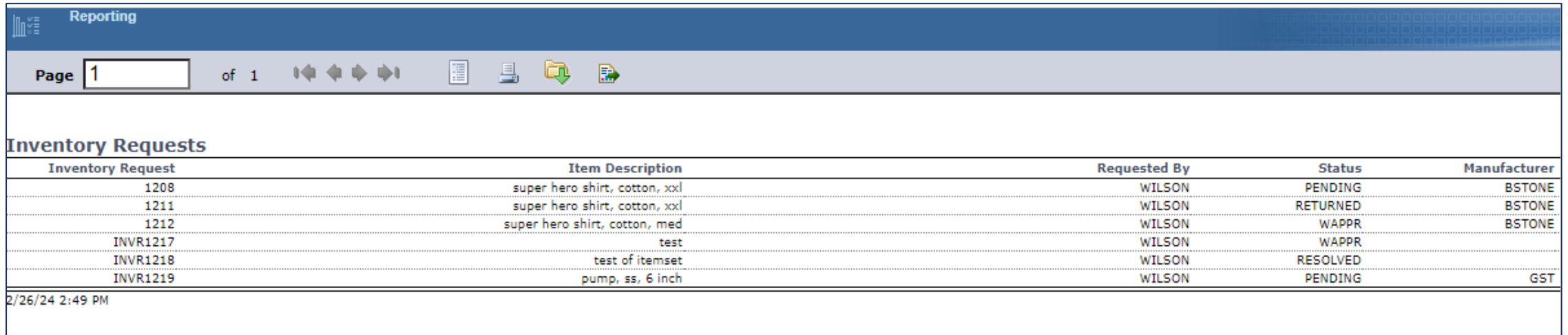
+Inventory Request ID: INVR1219

```
from java.util import Date
from psdi.server import MXServer
from psdi.mbo import SqlFormat

mxserver = MXServer.getMXServer()

if app == 'SS_INVR':
    autoKeySet = mxserver.getMboSet("AUTOKEY",mbo.getUserInfo())
    autoKeySet.setWhere("AUTOKEYNAME='SRID'")
    autoKeySet.reset()
    autoKeyMbo=autoKeySet.getMbo(0)
    prefix= 'INVR'
    seed = autoKeyMbo.getInt("SEED")
    mbo.setValue("TICKETID", prefix+str(seed))
```

Create BIRT Reports



Reporting

Page 1 of 1

Inventory Requests

Inventory Request	Item Description	Requested By	Status	Manufacturer
1208	super hero shirt, cotton, xxl	WILSON	PENDING	BSTONE
1211	super hero shirt, cotton, xxl	WILSON	RETURNED	BSTONE
1212	super hero shirt, cotton, med	WILSON	WAPPR	BSTONE
INVR1217	test	WILSON	WAPPR	
INVR1218	test of itemset	WILSON	RESOLVED	
INVR1219	pump, ss, 6 inch	WILSON	PENDING	GST

2/26/24 2:49 PM

The business requested a simple report to provide a listing of new Inventory Requests which could be scheduled to run and then emailed to specific users.

The quickest way to do this is to create an adhoc report using QBR, then export the report design file from Maximo.

Edit the rptdesign file using Eclipse BIRT to modify the where clause, remove the summary information, and then upload the new report into Maximo, granting security to the various security groups that have access to the application.

Create BIRT Reports

```
dataSet_sr = MXReportDataSetProvider.create(this.getDataSource().getName(), this.getName());
dataSet_sr.open();
var sqlText = new String();
sqlText =
    "select "
+ " sr.ticketid, sr.description, sr.reportedby, sr.status, sr.ssmanufacturer, sr.ssmodelnum, sr.sstoreroom, sr.vendor "
+ "from sr "
+ "where historyflag = '0' and sstickettype = 'INVR' "
+ " "
;
```

Once you have made SQL and layout changes in BIRT, save the report as a new name, then in Maximo go to report administration, create the new report, import the report design file, generate the request page and then grant security to the report.

Report File Name: <u>SS_INVR_List.rptdesign</u>	<u>Inventory Request List</u>		
Report Level Security	↑ ↓ ← 1 - 1 of 1 →	↓	☒
Group	Description		
> MAXADMIN	🔍 Maxadmin		🗑️

The final step is to delete the QBR report you created to start the creation of this report.

Conditionally Hide Sections of the App

The screenshot displays two overlapping windows from a Maxtech application. The top window is a request form with a text area containing "New Pump needed as part of overhaul" and a "Submit Request" button. The bottom window is an "Inventory Management" table with a yellow background, showing various fields like "Order Quantity", "Reorder Point", "Lead Time", "Commodity Group", "Storeroom", "Bin Number", "Model Num: GST-1123", and "Unit Cost: 850.00". The table also includes fields for "Created Item", "Inventory Approver Group", "Warehouse Group", "Item Set: SET1", "Site: BEDFORD", and "Ticket Type: INVR". A message "There are no rows to display." is visible in the table area.

Is Repairable Spare?

+ Reason for Request:

Font [] Size xx-small Format None

New Pump needed as part of overhaul

Submit Request

Multiple Assets,Locations Filter > 0 - 0

Asset	Location	Site
There are no rows to display.		

Inventory Management

Order Quantity: _____ Created Item: _____ Inventory Approver Group: _____

Reorder Point: _____ Warehouse Group: _____

Cycle Count Frequency: _____ Item Set: SET1

Lead Time: _____ Site: BEDFORD

Commodity Group: _____ Ticket Type: INVR

Storeroom: _____

Bin Number: _____

Model Num: GST-1123

Unit Cost: 850.00

Multiple Assets,Locations Filter > 0 - 0 of 0

Conditionally Hide Sections of the App

The business requested that the Inventory Details section of the Inventory Request be hidden from the user when they are first creating the request, and also, if the request has been sent back to them for any reason.

Here are the steps to implement this feature:

1. Create a new conditional expression which will be the condition under which the section will be hidden
2. Create a new signature option in the application
3. In application designer click on the section you want to HIDE and then right click and select properties. On the advanced tab add the signature option you just created. Then click Configure Conditional Properties

Condition Number: Description:
Type:
Expression:
Condition Class:
Reevaluate?

Option:
Description:
Option Also Grants:
Option Also Revokes:
Prerequisite:


Section Properties

General Advanced

Listeners:

Signature Option:

Sig Option Data Source ID:



Conditionally Hide Sections of the App

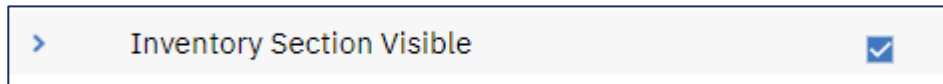
4. Chose the security group (EVERYONE or MAXEVERYONE) whichever you use at your company)

Chose the Conditional Expression you created

The Click Enter Property in the Condition is True tab

The property to put in is "display" and the value is "true"

There is no need to add a property for when the condition is false.

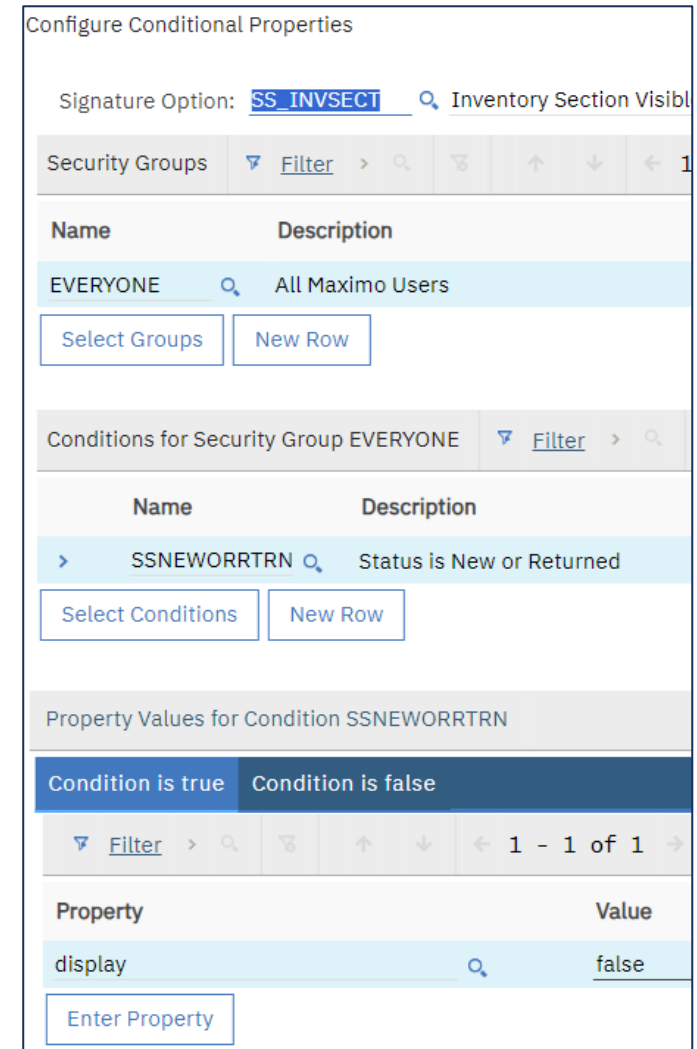


5. The next step is to grant access to this new signature option to the EVERYONE security group in the security groups application.



If you miss doing step 5 of this process, the section will be hidden for all users all of the time

6. Then log out of Maximo, log back in so that the security change will take effect.




The screenshot shows the 'Configure Conditional Properties' interface. It is divided into three main sections: Security Groups, Conditions for Security Group, and Property Values.

Security Groups: Signature Option: SS_INVSECT Inventory Section Visible. A table with columns 'Name' and 'Description' shows 'EVERYONE' with 'All Maximo Users'. Buttons: 'Select Groups', 'New Row'.

Conditions for Security Group EVERYONE: A table with columns 'Name' and 'Description' shows 'SSNEWORRTRN' with 'Status is New or Returned'. Buttons: 'Select Conditions', 'New Row'.

Property Values for Condition SSNEWORRTRN: Two tabs: 'Condition is true' (selected) and 'Condition is false'. A table with columns 'Property' and 'Value' shows 'display' with 'false'. Button: 'Enter Property'.

Credits and Acknowledgements

This presentation is a compilation of many great ideas and configurations in Maximo that I have seen and used elsewhere. Many have been topics for presentations at previous  MAXTECH Meetings on their own. The purpose of today is to show how you can bring concepts together in Maximo to make an Application Great, and for the users to LOVE what they are seeing.

If you need any of the details on how to implement any of the features described in this presentation drop me an email and I would be happy to provide them for you and your company.

Also note the features do not need to be limited to just this application but can be applied to any application in Maximo. It is the skills and concepts and being creative in your configuration which are important.

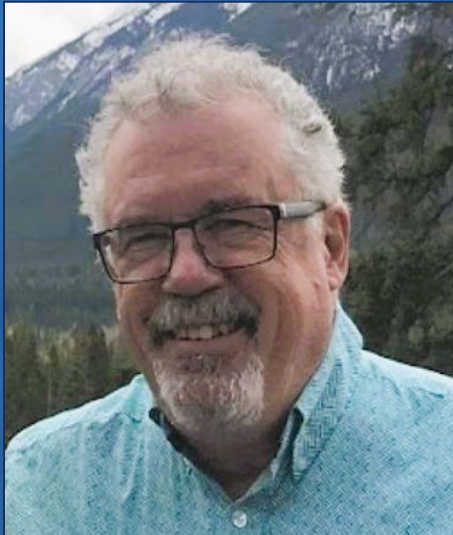
If you wish to receive copies of any of the automation scripts described in this presentation please reach out to me directly or via linkedin.

Stephen Hume

stephen.hume@sheffieldscientific.com

403 651-6117

QUESTIONS?



Stephen Hume

Maximo Principal Consultant |



<< Connect with Stephen on LinkedIn

THANK YOU



POWERED BY  BPD ZENITH

Ask us
a technical question:



maxtech@bpdzenith.com

Watch
previous webinars:



www.bpdzenith.com/na/maxtech/

Thank You to our Partners



Join our LinkedIn Group | MaxTECH – Maximo Technical User Group