# Infor Factory Track

## Extensibility Guide for Factory Track for M3

**Copyright © 2018 Infor**

**Important Notices**

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement, the terms of which separate agreement shall govern your use of this material and all supplemental related materials ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above. Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Without limitation, U.S. export control laws and other applicable export and import laws govern your use of this material and you will neither export or re-export, directly or indirectly, this material nor any related materials or supplemental information in violation of such laws, or use such materials for any purpose prohibited by such laws.

**Trademark Acknowledgements**

The word and design marks set forth herein are trademarks and/or registered trademarks of Infor and/or related affiliates and subsidiaries. All rights reserved. All other company, product, trade or service names referenced may be registered trademarks or trademarks of their respective owners.

**Publication Information**

Release: Infor Factory Track 6.01.10

Publication date: January 4, 2018

# Contents

# Use of this document

This document intends to provide information on how Factory Track for M3 can be extended to meet some customization needs. This document includes common customization scenarios and examples.

# Extending Factory Track to display additional information from an M3 API

This section explains how an existing transaction in Factory Track for M3 can be extended to display additional fields from an M3 API.

## Requirement

As a sample requirement, **Responsible** field of a balance identity must be displayed on the List Balance ID Screen of Stock Enquiry transaction. Responsible corresponds to **MMS060MI/LstBalID.RESP** outbound field.

## Changes in Factory Track

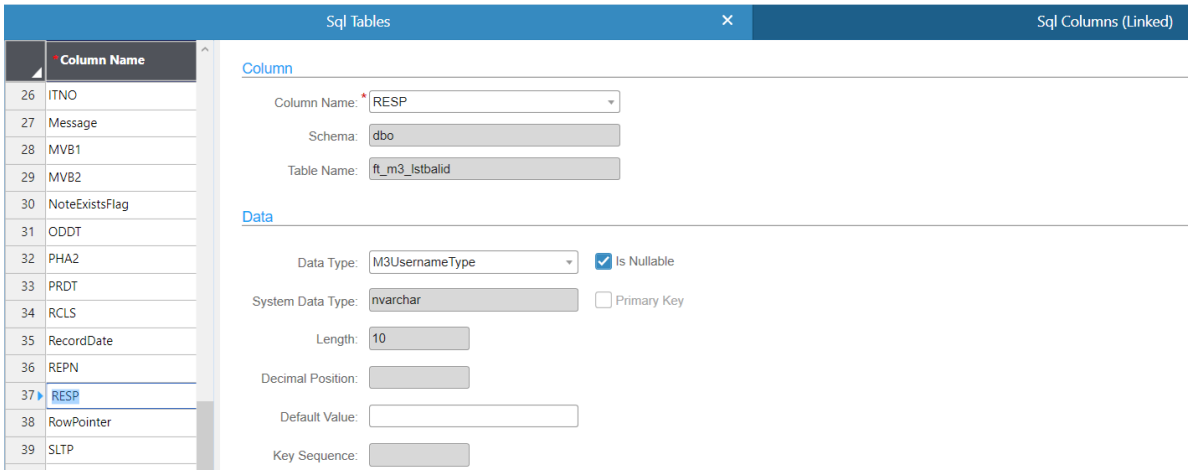
Above requirement entails changes in Factory Track database, IDO and Forms.

Determine if the IDO used in the form uses a staging table or not. This can be checked in the Table references of the IDO. FT staging tables are prefixed with “ft_m3”.

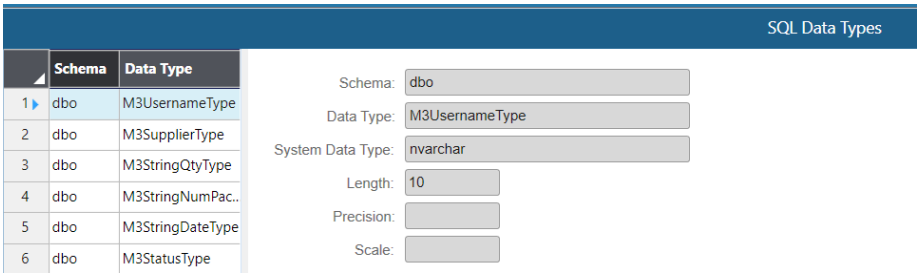
1 If the IDO uses a staging table, perform the steps below. Otherwise, proceed to step 2.

a Add a new column to the staging table.

**FTM3LstBalIDBound** IDO uses a staging table called **ft_m3_lstbalid**. Add **RESP** column to this table. Create a new user data type, as necessary.



MMS060MI/LstBalID.RESP is a 10-character, alphanumeric output field.



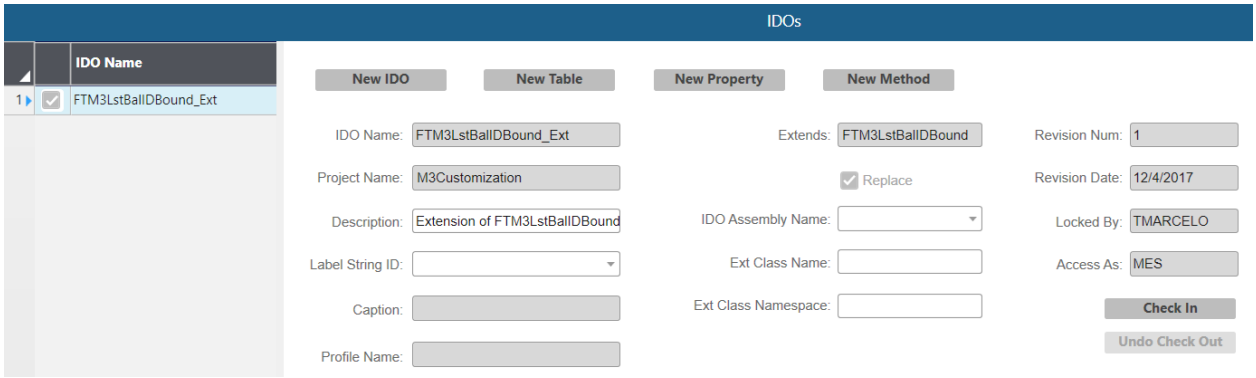

b Re-generate the triggers of the staging table.

Open the **Trigger Management** form.

Specify **ft_m3_lstbalid** in the Starting and Ending Fields, then select **Generate**.

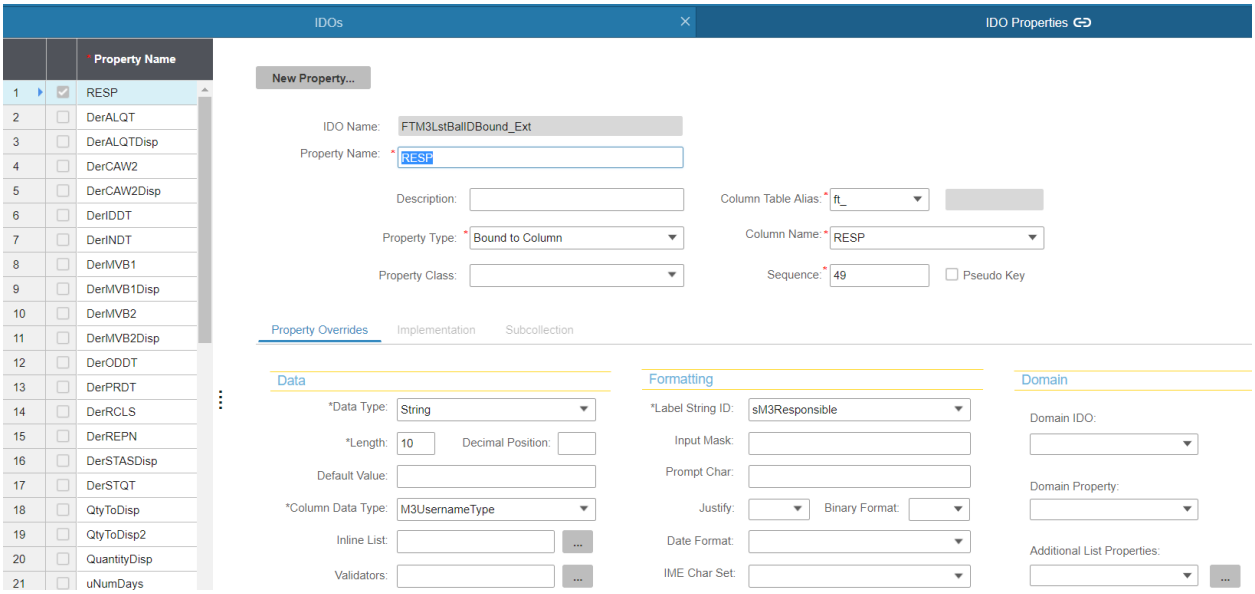
2 Create an extension of the FT IDO, and add a new property to the extension IDO.

a Create a new IDO **FTM3LstBalIDBound_Ext** as an extension of FTM3LstBalIDBound IDO. When creating the IDO, select the option **Extend and Replace**.




b Add new property **RESP** to **FTM3LstBalIDBound_Ext** IDO**.**

Since **FTM3LstBalIDBound_Ext** uses a staging table, set the property type to **Bound to Column**, and bind RESP property to **ft_m3_lstbalid.RESP** column. This ensures that MMS060MI/LstBalID.RESP outbound field gets saved in ft_m3_lstbalid.RESP column.

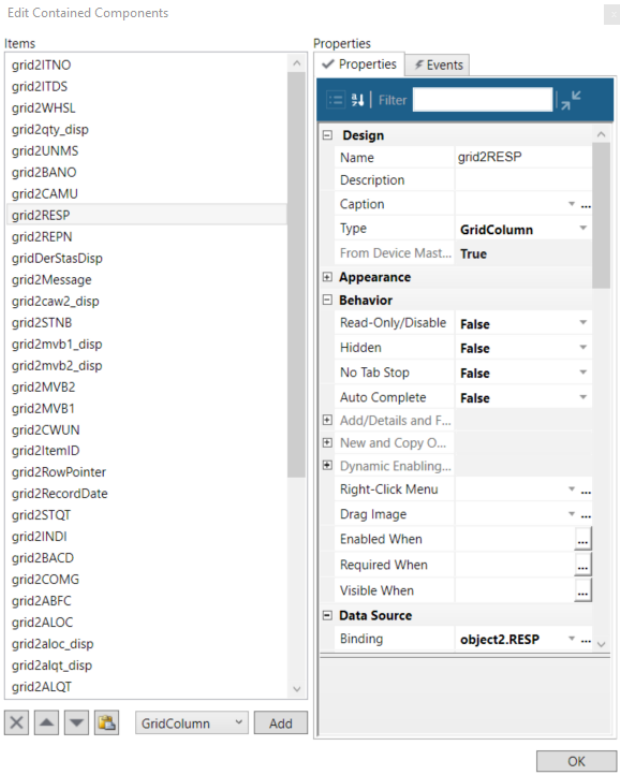



c **Save** the IDO changes

3 Discard the IDO cache.

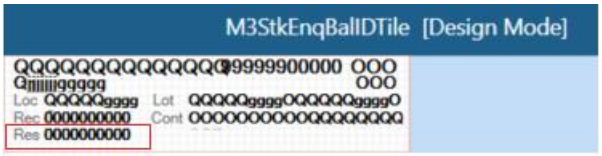
4 Update the grid in form. Add a new column bound to the new IDO property.

**gridLstBalID** is the grid component in **M3StockEnquiry.mobi** form that displays the balance identities. Add new grid column **grid2RESP**. Make sure Hidden is set to False, and column is bound to **FTM3LstBalIDBound_Ext.RESP** property.



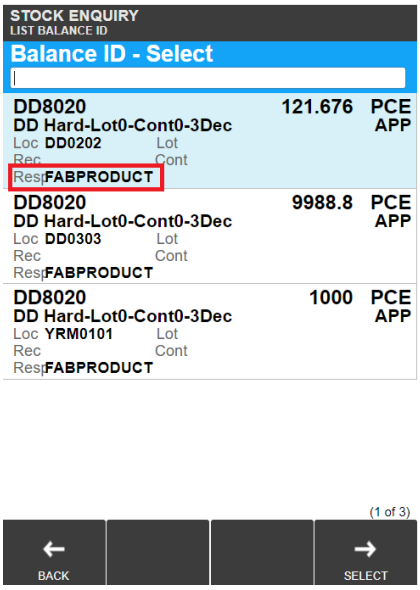
5 Update the tile form. Add new component/s to display the new IDO property.

**M3StkEnqBalIDTile** is the tile form used by gridLstBalID grid component in M3StockEnquiry.mobi form. Add new static components, as shown below, that will display the label and value of FTM3LstBalIDBound_Ext.RESP IDO property.

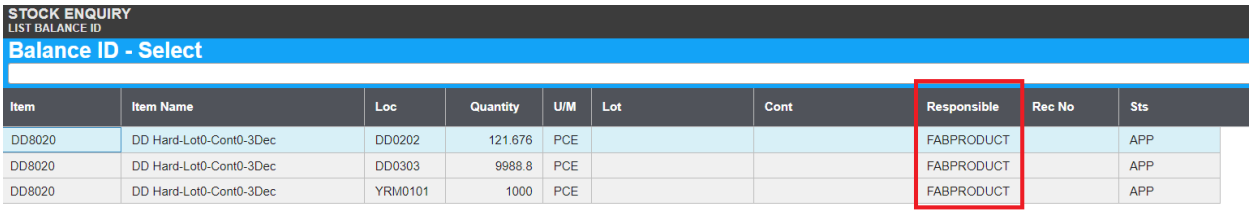



# Sample Output

Stock Enquiry – balance identities displayed in tile format




Stock Enquiry – balance identities displayed in grid format



# Extending Factory Track to update additional fields in M3

This section explains how to add a new input field in an existing Factory Track transaction form, and use the entered value to update a corresponding field in M3.

## Requirement

As a sample requirement, **Shift** must be displayed as an input field on the Report Operation Screen of MO Report Operation transaction. Shift corresponds to **PMS070MI/RptOperation.SHFC** input field.

## Changes in Factory Track

Above requirement entails changes in Factory Track database and form.

1 Add and display a new input field on the report screen of the transaction.

a Add Shift field to MO Report Operation – Report Operation screen in the M3 Transaction Screen Fields form.

Open **M3TranScreenFields** form.

Select **M3MOOperation** transaction from the left pane.

Select **Create a new object…** for the collection on the right pane, then specify the following values:

- Screen name = **Report**

  Screen name identifies the form page within the transaction form. Since Shift needs to be added to the Report screen, specify Report in this column.

- Field no = **10**

  This value identifies the sequence number of the field.

- Field name = **idShift**

  Specify a unique value to identify the field. The naming standard is **id<fieldname>**.

- Label String ID = **sM3Shift**

Specify the string ID that will be used as a field label. Create a new string, if necessary.

**Save** the new record.

M3 Transaction Screen Fields

| | Transaction Name |
|---|---|
| 1 | M3ConfirmPutaway |
| 2 | M3Count |
| 3 | M3Inspect |
| 4 | M3MOIssue |
| 5 | M3MOOperation |
| 6 | M3Move |
| 7 | M3PackageInPackage |
| 8 | M3PackageMove |
| 9 | M3PackageRepack |
| 10 | M3PackageUpdate |
| 11 | M3PickAndPack |
| 12 | M3Print |
| 13 | M3PrintDoc |
| 14 | M3Putaway |
| 15 | M3Receive |
| 16 | M3ShipmentLoading |
| 17 | M3StockEnquiry |

| | * Screen Name | * Field No | * Field Name | * Label String ID | Description |
|---|---|---|---|---|---|
| 1 | Report | 1 | idEmployee | sM3Employee | Employee |
| 2 | Report | 2 | idProduct | sM3Product | Product |
| 3 | Report | 3 | idManQty | sM3ManQty | Man Qty |
| 4 | Report | 4 | idScrapQty | sM3ScrapQty | Scrap Qty |
| 5 | Report | 5 | idScrapReason | sM3ScrapRsn | Scrap Rsn |
| 6 | Report | 6 | idLabRunTime | sM3LabRunTm | Lab Run Time |
| 7 | Report | 7 | idLabSetup | sM3LabSetup | Lab Setup |
| 8 | Report | 8 | idMchRunTime | sM3MchRunTm | Mch Run Time |
| 9 | Report | 9 | idMchSetup | sM3MchSetup | Mch Setup |
| 10 | Report | 10 | idShift | sM3Shift | |
| 11 | Search | 1 | idSearch | sM3Search | Search |
| 12 | Search | 2 | idReportNo | sM3ReportNo | Report No |
| 13 | Search | 3 | idProduct | sM3Product | Product |
| 14 | Search | 4 | idOrder | sM3Order | Order |
| 15 | Search | 5 | idOperation | sM3Operation | Operation |
| 16 | Search | 6 | idWorkCenter | sM3WorkCenter | Work Center |
| 17 | Search | 7 | idLot | sM3LotMO | Lot (MO) |

b Change the value of Report Operation – Sequence of input fields parameter of your selected transaction profile.

Open the **Profile Parameter Maintenance** form.

Select **REPORT OPERATION** and your **Profile Name** from the left pane.

Select **REPORT OPERATION – Sequence of input fields** parameter, and update the parameter value. Value must include “10” to display the Shift field.

Profile Parameter Maintenance

| | |
|---|---|
| 14 | RECEIVE |
| 15 | REPORT OPERATION |
| 16 | SHIPMENT LOADING |
| 17 | STOCK ENQUIRY |
| 18 | USER SETTINGS |
| 19 | WAREHOUSE MOVE |

| | Profile |
|---|---|
| 1 | REPORT OPERATION |
| 2 | Report Operation - Demo |

| | | | | |
|---|---|---|---|---|
| 7 | | Search field search sequence | ☐ | 1;2;3;4;5;6 |
| 8 | | Sequence of filter fields | ☐ | 1;2;3;4;5;6;7 |
| 9 | * | LIST OPERATIONS | ☑ | 1 |
| 10 | | Auto select first record | ☐ | 0 |
| 11 | * | REPORT OPERATION | ☑ | 1 |
| 12 | | API warnings to suppress | ☐ | |
| 13 | | Auto confirm input field data | ☐ | 1=A;2=A;3=A;4=A |
| 14 | | Auto trigger Next when all fields confirmed | ☐ | 0 |
| 15 | | Numeric keypad - Activate negative sign | ☐ | 1=A;2=A;3=A;4=A;5=A;6=A |
| 16 | | Numeric keypad - Enable | ☐ | 1=B;2=B;3=B;4=B;5=B;6=B |
| 17 | | Populate scan field with suggested data | ☐ | |
| 18 | | Sequence of input fields | ☐ | 1;2;3;4;5;6;7;8;9;10 |

c Update logic in the form script to handle the new field.

Open the **form script** of **M3MOReportOperation.mobi**.

In **SetParameters()**, update the valid values of parameter **PM_moro_rpt_seq_flds**. Set the fourth value in the tuple to “10”. This ensures that 10 (for Shift field) will be taken as a valid parameter value.

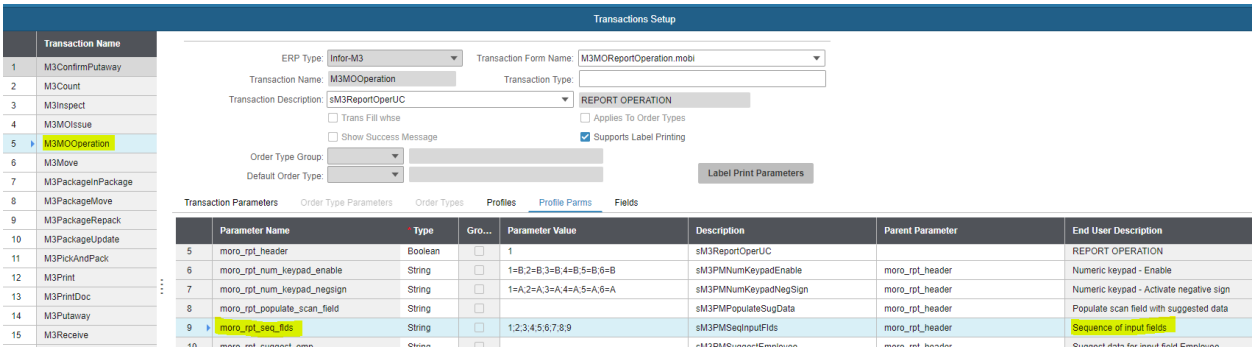
```
protected void SetParameters()
{
    COMMON CODE: Single value param from list of valid range- PA1/PB1/PC3

    #region COMMON CODE: Semicolon-delimited value param- PA6/PA7/PC1/PC4/PC9
    ThisForm.Variables("bScrapRsnInPC4").SetValue(false);
    List<Tuple<string, string, int, int, string>> tupDelimitedParms = new List<Tuple<string, string, int, int, string>>(); //parm name, parm form var, min val, max val,
    tupDelimitedParms.Add(new Tuple<string, string, int, int, string>("PM_moro_search_search_seq", "pmPA6", 1, 6, "1;2;3;4;5;6")); //PA6: Search field search sequence
    tupDelimitedParms.Add(new Tuple<string, string, int, int, string>("PM_moro_search_seq_flds", "pmPA7", 1, 7, "1;2;3;4;5;6;7")); //PA7: Sequence of filter fields
    tupDelimitedParms.Add(new Tuple<string, string, int, int, string>("PM_moro_rpt_suppress_warn", "pmPC1", 1, 4, "")); //PC1: API warnings to suppress
    tupDelimitedParms.Add(new Tuple<string, string, int, int, string>("PM_moro_rpt_seq_flds", "pmPC4", 1, 10, "1;2;3;4;5;6;7;8;9")); //PC4: Sequence of input fields
    tupDelimitedParms.Add(new Tuple<string, string, int, int, string>("PM_moro_rpt_populate_scan_field", "pmPC9", 1, 4, "")); //PC9: Populate scan field
```

In **SetReportFilter()**, update the logic so it passes the scanned value in idShift field to PMS070MI/RptOperation.SHFC.

```
protected void SetReportFilter()
{
    IWSIDOCollection gridRpt = ThisForm.GetSecondaryIDOCollection(idoReport);
    IWSIDORow rowMOOpe = ThisForm.GetSecondaryIDOCollection(idoLstMOOpe).CurrentItem;
    string inpEmp = "", inpManQty = "", inpScrapQty = "", inpScrapRsn = "", inpLabRunTm = "", inpLabSetup = "", inpMchRunTm = "", inpMchSetup = "", inpShift = "";

    for (int i = 0; i < gridRpt.GetNumEntries(); i++)
    {
        if (gridRpt.GetObjectProperty(propIdentifier, i).Equals(rowEmployee))
            inpEmp = gridRpt.GetObjectProperty(propValue, i);
        else if (gridRpt.GetObjectProperty(propIdentifier, i).Equals(rowManQty))
            inpManQty = gridRpt.GetObjectProperty(propExtra1, i);
        else if (gridRpt.GetObjectProperty(propIdentifier, i).Equals(rowScrapQty))
            inpScrapQty = gridRpt.GetObjectProperty(propValue, i);
        else if (gridRpt.GetObjectProperty(propIdentifier, i).Equals(rowScrapReason))
            inpScrapRsn = gridRpt.GetObjectProperty(propValue, i);
        else if (gridRpt.GetObjectProperty(propIdentifier, i).Equals(rowLabRunTime))
            inpLabRunTm = gridRpt.GetObjectProperty(propValue, i);
        else if (gridRpt.GetObjectProperty(propIdentifier, i).Equals(rowLabSetup))
            inpLabSetup = gridRpt.GetObjectProperty(propValue, i);
        else if (gridRpt.GetObjectProperty(propIdentifier, i).Equals(rowMchRunTime))
            inpMchRunTm = gridRpt.GetObjectProperty(propValue, i);
        else if (gridRpt.GetObjectProperty(propIdentifier, i).Equals(rowMchSetup))
            inpMchSetup = gridRpt.GetObjectProperty(propValue, i);
        else if (gridRpt.GetObjectProperty(propIdentifier, i).Equals("idShift"))
            inpShift = gridRpt.GetObjectProperty(propValue, i);
    }

    List<Tuple<string, string>> lstRpt = new List<Tuple<string, string>>();
    lstRpt.Add(new Tuple<string, string>("FACI", Application.Variables("M3_facility").Value));
    lstRpt.Add(new Tuple<string, string>("PRNO", rowMOOpe.Properties[propPRNO].Value));
    lstRpt.Add(new Tuple<string, string>("MFNO", rowMOOpe.Properties[propMFNO].Value));
    lstRpt.Add(new Tuple<string, string>("OPNO", rowMOOpe.Properties[propOPNO].Value));
    lstRpt.Add(new Tuple<string, string>("MAUN", rowMOOpe.Properties[propMAUN].Value));
    lstRpt.Add(new Tuple<string, string>("UMAT", inpLabRunTm));
    lstRpt.Add(new Tuple<string, string>("UMAS", inpLabSetup));
    lstRpt.Add(new Tuple<string, string>("UPIT", inpMchRunTm));
    lstRpt.Add(new Tuple<string, string>("USET", inpMchSetup));
    lstRpt.Add(new Tuple<string, string>("MAQA", inpManQty));
    lstRpt.Add(new Tuple<string, string>("SCQA", inpScrapQty));
    lstRpt.Add(new Tuple<string, string>("SCRE", ThisForm.Variables(vRejReason).Value));
    lstRpt.Add(new Tuple<string, string>("EMNO", inpEmp));
    lstRpt.Add(new Tuple<string, string>("SHFC", inpShift));
    lstRpt.Add(new Tuple<string, string>(propDSP1, ThisForm.Variables(propDSP1).Value));
    lstRpt.Add(new Tuple<string, string>(propDSP2, ThisForm.Variables(propDSP2).Value));
    lstRpt.Add(new Tuple<string, string>(propDSP3, ThisForm.Variables(propDSP3).Value));
    lstRpt.Add(new Tuple<string, string>(propDSP4, ThisForm.Variables(propDSP4).Value));

    buildFilter(lstRpt, idoRptOperation);
}
```

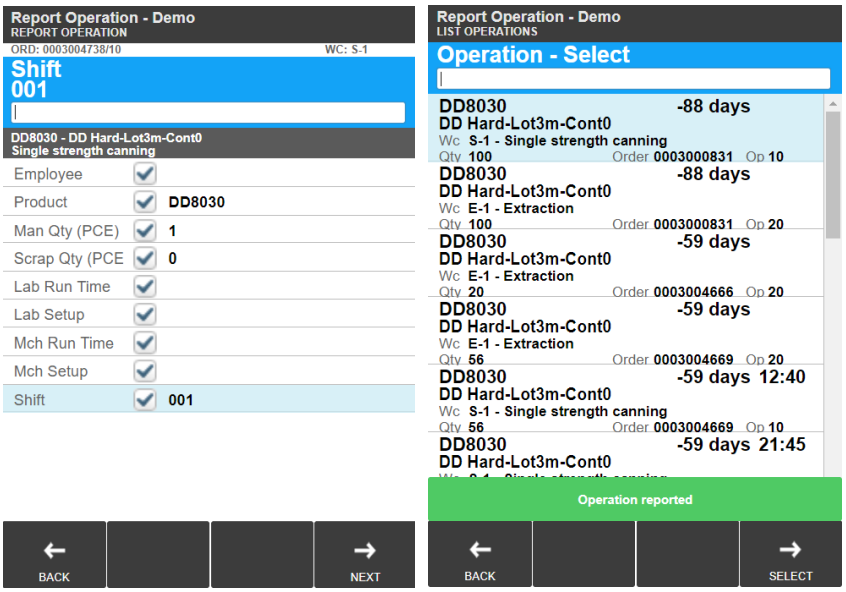
**Save** the form script changes.

**Note:** Parameter name may be identified using the **Transactions Setup** form, as shown below. Parameters in the form script are always in the format **PM_<parameter name>**.




# Sample Output

Shift is displayed as an input field in the Report Operation screen of MO Report Operation transaction.




REST API call to PMS070MI/RptOperation:

https://nlbavwm3e154.infor.com:54008/m3api-rest/execute/PMS070MI/RptOperation;m3user=TMARCELO;maxrecs=0?FACI=A01&PRNO=DD8030&MFNO=0003004738&OPNO=10&MAUN=PCE&UMAT=&UMAS=&UPIT=&USET=&MAQA=1&SCQA=0&SCRE=&EMNO=&SHFC=001&DSP1=&DSP2=&DSP3=&DSP4=

In M3 BE PMS095 (MO Operation Display Transactions), the reported Shift is reflected accordingly: