



# 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 .....	5
Extending Factory Track to display additional information from an M3 API .....	6
Requirement .....	6
Changes in Factory Track .....	6
Sample Output.....	9
Extending Factory Track to update additional fields in M3 .....	10
Requirement .....	10
Changes in Factory Track .....	10
Sample Output.....	13



---

## 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\_1stbalid**. Add **RESP** column to this table. Create a new user data type, as necessary.

The screenshot shows a configuration window for a SQL table. On the left, a list of columns is visible, with 'RESP' highlighted at row 37. The main area is titled 'Column' and contains the following fields:

- Column Name: \*RESP
- Schema: dbo
- Table Name: ft\_m3\_1stbalid
- Data Type: M3UsernameType
- Is Nullable:
- System Data Type: nvarchar
- Primary Key:
- Length: 10
- Decimal Position: (empty)
- Default Value: (empty)
- Key Sequence: (empty)

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

The screenshot shows the 'SQL Data Types' configuration window. On the left is a table with columns 'Schema' and 'Data Type'. The table contains the following rows:

	Schema	Data Type
1	dbo	M3UsernameType
2	dbo	M3SupplierType
3	dbo	M3StringQtyType
4	dbo	M3StringNumPac..
5	dbo	M3StringDateType
6	dbo	M3StatusType

On the right, the configuration fields are:

- Schema:
- Data Type:
- System Data Type:
- Length:
- Precision:
- Scale:

b Re-generate the triggers of the staging table.

Open the **Trigger Management** form.

Specify **ft\_m3\_Instbalid** 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**.

The screenshot shows the 'IDOs' configuration window. On the left is a table with columns 'ID Name'. The table contains the following row:

ID Name
FTM3LstBalIDBound_Ext

On the right, the configuration fields are:

- New IDO** (selected)
- IDO Name:
- Project Name:
- Description:
- Label String ID:
- Caption:
- Profile Name:
- Extends:
- Replace:
- IDO Assembly Name:
- Ext Class Name:
- Ext Class Namespace:
- Revision Num:
- Revision Date:
- Locked By:
- Access As:
- 
- 

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\_Instbalid.RESP** column. This ensures that **MMS060MI/LstBalID.RESP** outbound field gets saved in **ft\_m3\_Instbalid.RESP** column.

The screenshot shows the 'IDO Properties' dialog box with the following configuration:

- Property Name:** RESP
- IDO Name:** FTM3LstBallIDBound\_Ext
- Property Type:** Bound to Column
- Column Name:** RESP
- Sequence:** 49
- Data Type:** String
- \*Column Data Type:** M3UsernameType
- Label String ID:** sM3Responsible

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.

**gridLstBallID** 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 **FTM3LstBallIDBound\_Ext.RESP** property.

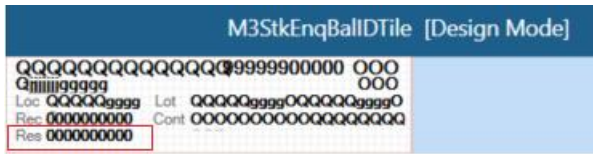
The screenshot shows the 'Edit Contained Components' dialog box with the following configuration for the 'grid2RESP' component:

- Name:** grid2RESP
- Type:** GridColumn
- Hidden:** False
- Data Source:** object2.RESP



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

**M3StkEnqBallIDTile** is the tile form used by gridLstBallID grid component in M3StockEnquiry.mobi form. Add new static components, as shown below, that will display the label and value of FTM3LstBallIDBound\_Ext.RESP IDO property.



## Sample Output

Stock Enquiry – balance identities displayed in tile format

STOCK ENQUIRY LIST BALANCE ID			
Balance ID - Select			
<input type="text"/>			
DD8020	121.676	PCE	
DD Hard-Lot0-Cont0-3Dec		APP	
Loc DD0202	Lot		
Rec	Cont		
Res: <b>FABPRODUCT</b>			
DD8020	9988.8	PCE	
DD Hard-Lot0-Cont0-3Dec		APP	
Loc DD0303	Lot		
Rec	Cont		
Res: FABPRODUCT			
DD8020	1000	PCE	
DD Hard-Lot0-Cont0-3Dec		APP	
Loc YRM0101	Lot		
Rec	Cont		
Res: FABPRODUCT			

(1 of 3)



Stock Enquiry – balance identities displayed in grid format

STOCK ENQUIRY LIST BALANCE ID									
Balance ID - Select									
<input type="text"/>									
Item	Item Name	Loc	Quantity	U/M	Lot	Cont	Responsible	Rec No	Sts
DD8020	DD Hard-Lot0-Cont0-3Dec	DD0202	121.676	PCE			FABPRODUCT		APP
DD8020	DD Hard-Lot0-Cont0-3Dec	DD0303	9988.8	PCE			FABPRODUCT		APP
DD8020	DD Hard-Lot0-Cont0-3Dec	YRM0101	1000	PCE			FABPRODUCT		APP

---

# 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		* Screen Name	* Field No	* Field Name	* Label String ID	Description
1	M3ConfirmPutaway	1	Report	1	idEmployee	sM3Employee	Employee
2	M3Count	2	Report	2	idProduct	sM3Product	Product
3	M3Inspect	3	Report	3	idManQty	sM3ManQty	Man Qty
4	M3MOIssue	4	Report	4	idScrapQty	sM3ScrapQty	Scrap Qty
5	M3MOOperation	5	Report	5	idScrapReason	sM3ScrapRsn	Scrap Rsn
6	M3Move	6	Report	6	idLabRunTime	sM3LabRunTm	Lab Run Time
7	M3PackageInPackage	7	Report	7	idLabSetup	sM3LabSetup	Lab Setup
8	M3PackageMove	8	Report	8	idMchRunTime	sM3MchRunTm	Mch Run Time
9	M3PackageRepack	9	Report	9	idMchSetup	sM3MchSetup	Mch Setup
10	M3PackageUpdate	★10	Report	10	idShift	sM3Shift	
11	M3PickAndPack	11	Search	1	idSearch	sM3Search	Search
12	M3Print	12	Search	2	idReportNo	sM3ReportNo	Report No
13	M3PrintDoc	13	Search	3	idProduct	sM3Product	Product
14	M3Putaway	14	Search	4	idOrder	sM3Order	Order
15	M3Receive	15	Search	5	idOperation	sM3Operation	Operation
16	M3ShipmentLoading	16	Search	6	idWorkCenter	sM3WorkCenter	Work Center
17	M3StockEnquiry	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	7	<input type="checkbox"/> Search field search sequence	1;2;3;4;5;6
15	REPORT OPERATION	8	<input type="checkbox"/> Sequence of filter fields	1;2;3;4;5;6;7
16	SHIPMENT LOADING	9	* <input checked="" type="checkbox"/> LIST OPERATIONS	1
17	STOCK ENQUIRY	10	<input type="checkbox"/> Auto select first record	0
18	USER SETTINGS	11	* <input checked="" type="checkbox"/> REPORT OPERATION	1
19	WAREHOUSE MOVE	12	<input type="checkbox"/> API warnings to suppress	
	Profile	13	<input type="checkbox"/> Auto confirm input field data	1=A;2=A;3=A;4=A
1	REPORT OPERATION	14	<input type="checkbox"/> Auto trigger Next when all fields confirmed	0
2	Report Operation - Demo	15	<input type="checkbox"/> Numeric keypad - Activate negative sign	1=A;2=A;3=A;4=A;5=A;6=A
		16	<input type="checkbox"/> Numeric keypad - Enable	1=B;2=B;3=B;4=B;5=B;6=B
		17	<input type="checkbox"/> Populate scan field with suggested data	
		18	<input type="checkbox"/> 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
    #endregion
}
```

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

```
protected void SetReportFilter()
{
    INSIDORow rowMOOpe = ThisForm.GetSecondaryIDOCCollection(idoReport);
    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>**.

The screenshot shows the 'Transactions Setup' form for the 'M3MOCOperation' transaction. The form includes fields for ERP Type (Infor-M3), Transaction Form Name (M3MOReportOperation.mobi), Transaction Name (M3MOCOperation), and Transaction Type (REPORT OPERATION). It also has checkboxes for 'Trans Fill wise', 'Shows Success Message', 'Applies To Order Types', and 'Supports Label Printing'. Below the form is a table of parameters:

Parameter Name	Type	Gr...	Parameter Value	Description	Parent Parameter	End User Description
moro_rpt_header	Boolean	<input type="checkbox"/>	1	sM3ReportOperUC	moro_rpt_header	REPORT OPERATION
moro_rpt_num_keypad_enable	String	<input type="checkbox"/>	1=B,2=B,3=B,4=B,5=B,6=B	sM3PMNumKeypadEnable	moro_rpt_header	Numeric keypad - Enable
moro_rpt_num_keypad_negsign	String	<input type="checkbox"/>	1=A,2=A,3=A,4=A,5=A,6=A	sM3PMNumKeypadNegSign	moro_rpt_header	Numeric keypad - Activate negative sign
moro_rpt_populate_scan_field	String	<input type="checkbox"/>		sM3PMPopulateSugData	moro_rpt_header	Populate scan field with suggested data
moro_rpt_sec_fds	String	<input type="checkbox"/>	1,2,3,4,5,6,7,8,9	sM3PMSeqInputFids	moro_rpt_header	Sequence of input fields

## Sample Output

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

The screenshot shows the 'Report Operation - Demo' screen. On the left, there is a 'Shift' input field with the value '001'. On the right, there is a list of operations with details such as 'DD8030 DD Hard-Lot3m-Cont0', 'Wc S-1 - Single strength canning', and 'Qty 100'. The screen also has navigation buttons for 'BACK' and 'NEXT'.

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:

The screenshot displays the Infor M3 BE PMS095 interface for MO Operation Display Transactions. The top navigation bar includes 'Menu', 'Start', and 'MO Operation. Display Transactions'. Below the navigation bar, there are tabs for 'ACTIONS', 'OPTIONS', 'RELATED', and 'TOOLS'. The main content area shows product details for 'DD8030 DD Hard-Lot3m-Cont0'. The MO number is '0003004738', the responsible person is '11282', the operation is '10 Single strength canning', and the work center is 'S-1 Single strength canning'. The 'Detailed Information' section is divided into two columns. The left column includes fields for 'Auto receipt', 'Order qty b/U/M', 'Used lab run tm', 'Used mch run tm', 'Run disturbance', 'Manuf quantity', 'Scrapped qty', 'Man compl flag', 'Employee no', 'Shift', 'Pin no workers', 'Resource', and 'Dev work center'. The right column includes fields for 'Auto reporting', 'Reporting no', 'Used lab setup', 'Used mch setup', 'Setup disturb', 'Mfg U/M', 'Reject reason', 'Scrap src W/C', 'Pay element', 'Trans date', 'Pin no setup', 'Rework', and 'Yield quantity'. The 'Shift' field is highlighted in yellow and shows the value '001'.

Field	Value
Product	DD8030 DD Hard-Lot3m-Cont0
MO number	0003004738
Responsible	11282
Operation no	10 Single strength canning
Work center	S-1 Single strength canning
Auto receipt	<input type="checkbox"/>
Order qty b/U/M	70
Used lab run tm	0,00 0,00
Used mch run tm	0,00 552,00
Run disturbance	
Manuf quantity	1 71
Scrapped qty	
Man compl flag	<input type="checkbox"/>
Employee no	
Shift	001
Pin no workers	0,00
Resource	
Dev work center	
Auto reporting	0-No
Reporting no	494705
Used lab setup	0,00 0,00
Used mch setup	0,00 0,00
Setup disturb	
Mfg U/M	PCE
Reject reason	
Scrap src W/C	
Pay element	
Trans date	171205 / 807
Pin no setup	0,00
Rework	<input type="checkbox"/> Cstng type 3
Yield quantity	