Determine Setup ID in APO by Characteristic Values

Applies to:
APO 7.0 and ERP 6.0
For more information, visit the Supply Chain Management homepage.

Summary
Learn how to change the setup id of APO activities with object dependencies based on characteristic values. The setup id is used to control PP/DS sequence optimization of activities in APO planned or production orders.

Author: Dr. Frank Horlacher
Company: SAP AG
Created on: 29 May 2009

Author Bio
Dr. Frank Horlacher of ISM PI IBU Mill Solution Management WDF has 9 years experience in SAP APO development and solution management in supply chain management production planning, detailed scheduling, manufacturing execution and sequencing system integration. Area of Expertise:SAP APO Modules: DP, SNP, PP/DS, ATP, CIF; SAP ERP Modules: SD, MM, PP, FI-CO
Table of Contents

Introduction .........................................................................................................................................................3
Prerequisites .......................................................................................................................................................3
Create Reference Characteristic in ERP ............................................................................................................4
Add Reference Characteristic to Variant Class of Material in ERP .................................................................6
Map Object Characteristics in APO ....................................................................................................................7
Transfer Characteristic and Class to APO ...........................................................................................................8
Check Reference Characteristic in APO ............................................................................................................9
Create Function Module in ERP .......................................................................................................................10
Create Variant Function in ERP .......................................................................................................................13
Create Function Module in APO .......................................................................................................................15
Create Variant Function in APO .......................................................................................................................18
Create Object Dependency in ERP ..................................................................................................................22
Assign to Operation of Routing in ERP ............................................................................................................24
Resend PDS to APO from ERP ........................................................................................................................25
Order Creation ..................................................................................................................................................27
Setup Optimization ...........................................................................................................................................28
Related Content ................................................................................................................................................33
Copyright ...........................................................................................................................................................34
Introduction

I will guide through a step by step procedure how to configure the characteristic dependent determination of the setup id in the SAP ERP and APO system. As a result the APO PP/DS module will create planned orders with operations (activities) with setup ids according to the rules you defined in object dependencies based on characteristic values of the output node of the planned order.

Prerequisites

As a prerequisite you need have created setup groups in the ERP system and transferred them to APO. You can maintain the setup groups in APO with transaction /SAPAPO/CDPSC6 - Production Planning -> Setup Group/Setup Matrix -> Maintain Setup Groups.

<table>
<thead>
<tr>
<th>Location</th>
<th>SetupGroup</th>
<th>Description</th>
<th>SetupGrp Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>M113</td>
<td>MM_HR3</td>
<td>Hot rolling grade 3</td>
<td></td>
</tr>
<tr>
<td>M113</td>
<td>MM_HR4</td>
<td>Hot rolling grade 4</td>
<td></td>
</tr>
<tr>
<td>M113</td>
<td>MM_HR5</td>
<td>Hot rolling grade 5</td>
<td></td>
</tr>
<tr>
<td>M113</td>
<td>MM_HR6</td>
<td>Hot rolling grade 6</td>
<td></td>
</tr>
<tr>
<td>M113</td>
<td>MM_ROLL</td>
<td>Setup group rolling</td>
<td></td>
</tr>
<tr>
<td>M113</td>
<td>MP_SHEET</td>
<td>Setup group sheeting</td>
<td></td>
</tr>
<tr>
<td>M210</td>
<td>MM_ROLL</td>
<td>Setup Groups for rolling</td>
<td></td>
</tr>
<tr>
<td>M210</td>
<td>MP_SHEET</td>
<td>Setup Groups for sheet</td>
<td></td>
</tr>
<tr>
<td>M210</td>
<td>M_SETUP_ROLLING</td>
<td>Setup Groups for rolling</td>
<td></td>
</tr>
<tr>
<td>M210</td>
<td>P_SETUP_SHEETING</td>
<td>Setup Groups for sheet</td>
<td></td>
</tr>
<tr>
<td>M900</td>
<td>1250</td>
<td>MM9_SHEET_RL</td>
<td></td>
</tr>
<tr>
<td>M900</td>
<td>1550</td>
<td>MM9_SHEET_RL</td>
<td></td>
</tr>
</tbody>
</table>
Create Reference Characteristic in ERP

The first step is to create a reference characteristic in the ERP system.

Transaction

CT04 - Classification System -> Master Data -> Characteristics

With the reference characteristic you overrule the master data settings. In our case we overrule the setup group in the PDS.
The most important setting is done in the Additional data tab of the characteristic maintenance. There you enter the reference to the master data table and field. Now you might ask: how do I know which field to enter here. A good idea is to consult the SAP help for object dependencies where you can find the referenceable fields or you go to the master data object itself e.g. the routing and put the cursor in the field you want to overrule and press the F1 button there you can see the table and the field the particular screen field is representing.

Above that you also need to check the referenceable fields in the APO PDS because not all ERP master data fields exist in APO. Please read note 610873. You can check the fields in APO by using the ABAP dictionary with transaction SE11 and look to structures beginning with /SAPAPO/CULL_CFG…
Add Reference Characteristic to Variant Class of Material in ERP

Add the newly created characteristic to the variant class of the material.

Transaction: CL02 - Classification System -> Master Data -> Classes

Do not forget to enter the organizational area for APO integration in the new line.
Map Object Characteristics in APO

Prior to activating the integration model to APO for characteristics and classes you need to map the object characteristics from ERP structures to APO structures because the data model is different.

You do that in the customizing of APO.

Advanced Planning and Optimization → Master Data → Classification and Configuration → Map Object Characteristics

![Change View "Map Object Characteristics": D...](image-url)
Transfer Characteristic and Class to APO

You transfer the reference characteristic and the variant class to APO by changing and activating your APO integration model for characteristics and classes in ERP.

Transactions:

CFM1 - Integration Model -> Create
CFM2 - Integration Model -> Activate
Check Reference Characteristic in APO

You can now see the new reference characteristic in APO. The characteristic should have the correct APO reference table and field we defined before in the mapping.

Transaction: CT04 - Master Data -> Classification System -> Characteristics
Create Function Module in ERP

We now create a function module with which we want to change the setup group from the master data PDS to the new value. We need a function module to be used in a variant function we use later in the object dependency.

Transaction: SE80 - Tools -> ABAP Workbench -> Overview -> Object Navigator

The function module needs to be created in ERP and APO because of the correct syntax check in both systems although in this example we only need it in APO for execution.

The function module needs the default parameters for variant function modules.
This is the interface and the empty code of the function module:

```
FUNCTION Z_MMF_SETUP_GROUP.
* **----------------------------------------------------------------------
* ** Local Interface:
* ** IMPORTING
* ** VALUE(GLOBALS) LIKE CUOV_00 STRUCTURE CUOV_00
* ** TABLES
* ** MATCH STRUCTURE CUOV_01
* ** QUERY STRUCTURE CUOV_01
* ** EXCEPTIONS
* ** FAIL
* ** INTERNAL_ERROR
* **----------------------------------------------------------------------
ENDFUNCTION.
```

You do not need to add any source code to the function module in the ERP system as the function module is used in APO PDS explosion.
Create Variant Function in ERP

We now create the variant function which wraps the function module we created in the last step. The variant function can be called in the object dependency.

Transaction: CU65 - Variant Configuration -> Tools -> Function -> Create

**Change Function: Basic Data**

- **Function**: Z_MMF_SETUP_GROUP
- **Status**: released
- **Group**: 
- **Authorization**: 

**Function Module**

- **Status**: 
- **Source Code**: LZMILL_NETU07
- **Last Changed By**: HORLACHER
- **Changed On**: 10.05.2009

**Function Group**

- **Function Group**: ZMILL_NET
- **Program**: SAPLZMILL_NET
- **Person Responsible**: 600107
Determine Setup ID in APO by Characteristic Values

In this example I created a variant function which changes the setup group based on the steel grade of the planned order output.
Create Function Module in APO

To execute the object dependency in the PDS explosion in APO PP/DS we also need to create the function module in the APO, this time with implemented source code.

Transaction: SE80 - Tools -> ABAP Workbench -> Overview -> Object Navigator

The function module needs to have exactly the same name as in ERP.

Repeat the parameter definition of the function module interface as described in step Create Function Module in ERP.

This is the example ABAP code I used:

```abap
FUNCTION z_mmf_setup_group.

** Local Interface:
** IMPORTING
** VALUEGLOBALS LIKE CUOV_00 STRUCTURE CUOV_00
** TABLES
** MATCH STRUCTURE CUOV_01
** QUERY STRUCTURE CUOV_01
** EXCEPTIONS
** FAIL
** INTERNAL_ERROR

DATA:
lv_apo_group_id TYPE /sapapo/cdps_setup_id,
lv_apo_item_id TYPE /sapapo/cdps_setup_id,
lv_setup_group TYPE /sapapo/cdps_setup_group,
lv_setup_item TYPE /sapapo/cdps_setup_item,
lv_setup_id TYPE /sapapo/cdps_setup_id,
ls_cuov_01 TYPE cuov_01,

* Location
  lv_location TYPE /sapapo/locno,
* Location structure
  ls_location TYPE /sapapo/loc_locno_rstr,
* Location table
  lt_location TYPE /sapapo/loc_locno_rtab,
* Location ID table
  lt_locid TYPE /sapapo/locid_tab,
* Location ID
  ls_locid TYPE /sapapo/locid.

READ TABLE query
  WITH KEY varnam = 'MM_IGRADE'
```
Determine Setup ID in APO by Characteristic Values

```plaintext
INTO ls_cuov_01.
CHECK sy-subrc = 0.
IF ls_cuov_01-atwrt = '3'.
  lv_setup_group = 'MM_HR3'.
ENDIF.
IF ls_cuov_01-atwrt = '4'.
  lv_setup_group = 'MM_HR4'.
ENDIF.
IF ls_cuov_01-atwrt = '5'.
  lv_setup_group = 'MM_HR5'.
ENDIF.
IF ls_cuov_01-atwrt = '6'.
  lv_setup_group = 'MM_HR6'.
ENDIF.

lv_location = 'M113'.

* fill location structure
ls_location-sign = 'I'.
ls_location-option = 'EQ'.
lv_location-low = lv_location.
APPEND ls_location TO lt_location.

* get location id
CALL FUNCTION '/SAPAPO/DM_LOCID_GET'
EXPORTING
  i_locno_rtab  = lt_location
IMPORTING
  e_locid_tab   = lt_locid
EXCEPTIONS
  no_location   = 1
  not_qualified = 2
  OTHERS        = 3.
IF sy-subrc <> 0.
  * MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
  * WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.

READ TABLE lt_locid INTO ls_locid INDEX 1.
IF sy-subrc = 0.
  CALL FUNCTION '/SAPAPO/DM_SETUP_ID_READ'
  EXPORTING
    i_locid       = ls_locid
    i_setup_group = lv_setup_group
    i_setup_item  = lv_setup_item
IMPORTING
    e_setup_id    = lv_setup_id
EXCEPTIONS
  not_found     = 1
  OTHERS        = 2.
  CALL FUNCTION '/SAPAPO/DM_SETUP_IDS_GET'
  EXPORTING
    i_setup_id       = lv_setup_id
```

© 2009 SAP AG
Determine Setup ID in APO by Characteristic Values

```
IMPORTING
  e_setup_group_id  = lv_apo_group_id
  e_setup_item_id   = lv_apo_item_id
EXCEPTIONS
  not_found        = 1
  OTHERS           = 2.

CLEAR ls_cuov_01.
ls_cuov_01-varnam = 'MM_PLPO_RFGRP'.
* also fill atwrt due to error in code in standard
  ls_cuov_01-atwrt = lv_setup_group.
  ls_cuov_01-atflv = lv_apo_group_id.
  ls_cuov_01-atfor = 'NUM'.
APPEND ls_cuov_01 TO match.
ENDIF.
ENDFUNCTION.
```

What do I do in the code? First I wrote down the if-clause what should happen with the setup group for what steel grade. The setup groups are defined in a setup matrix in the master data. The steel grades are predefined values from the configuration. You can also use a z-table to store the information of steel grade and setup group.

In APO the setup group is only for display the real value for scheduling is represented by the setup id which is only known internally.

The function module to read the setup ids which are used in scheduling in the end needs to know the location id. As you probably do not know the GUID of the location I included a function module to get the LOCID from the location name.

In the end you set the reference characteristic value in the match table.

But there is a trick you absolutely need to follow because of the different formats of setup group and setup id in APO and ERP. The code which interprets the object dependencies for setup in APO also wants to know the character value even if the setup id is a numeric value. So the line

```
  ls_cuov_01-atwrt = lv_setup_group.
```

is very important to make the change work.
Create Variant Function in APO

After we finished the function module we can now create the variant function in APO. The transaction for that might not be found in the tree of the SAP Menu. But it is the same transaction code as in the ERP.

CU65 - Create Function

When you create the variant function the system wants to have a status. In APO there are no default statuses for variant functions. So you need to create them in customizing.

If you press F4 value help on the status field when creating the function the system asks you to create a status. Just press yes and maintain the entry in the customizing table.
At least maintain the “released” status.
Determine Setup ID in APO by Characteristic Values

Then you can go on creating the variant function with the characteristics.
**Change Function: Characteristics**

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Input Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM_PLO_RF_GRP</td>
<td>☐</td>
<td>Setup group category</td>
</tr>
<tr>
<td>MM_JGRADE</td>
<td>☑</td>
<td>Internal steel grade</td>
</tr>
</tbody>
</table>
Create Object Dependency in ERP

The next step is to create the object dependency in ERP and assign it to the operation of the routing.

I use the transaction PMEVC - Modeling Environment for Variant Configuration for easier maintenance of object dependencies.

![Maintain Dependency: Basic Data](image)

- **Dependency**: MMF_SETUP_HR
- **Description**: Determine Setup in HR
- **Status**: 2 - In preparation
- **Dependency Group**: SAPAPOACT - Dependency for activity level in APO
- **Created By**: HORLACHER
- **Created On**: 19.05.2009
The dependency group must be SAPAPOACT to integrate the object dependency to the APO activity.

This is the sample code for the dependency:

```plaintext
FUNCTION Z_MMF_SETUP_GROUP
  (MM_IGRADE =$PARENT.MM_IGRADE,
   MM_PLPO_RFGRP =$SELF.MM_PLPO_RFGRP)

We define here that the steel grade of the parent node influences the reference characteristic of the activity in APO.
Assign to Operation of Routing in ERP

The object dependency needs to be assigned to the operation of the routing with the resource for setup optimization later on in APO.

You can navigate to the object maintenance for the routing out of the PMEVC - Modeling Environment for Variant Configuration transaction via the menu.
Resend PDS to APO from ERP

Now that changed the routing in ERP we want to update the PDS in APO with transaction CURTO_CREATE - Transfer Production Data Structure (PDS).

Check PDS in APO

Let’s check if the transfer has worked and the dependency can be seen in the APO PDS.

Transaction: /SAPAPO/CURTO_SIMU - Production Data Structure (PDS) -> Display Production Data Structures
The procedure is indeed on the right activity and you can also see the default setup group of the PDS as defined in the master data. This is the value we want to change based on different characteristic values of the output node in the order.
Order Creation

Now that we created all the master data let’s look at the order creation if our settings take effect.

I simply create an order in the /SAPAPO/RRP3 - Production Planning -> Interactive Production Planning -> Product View. In characteristic value assignment I use a steel grade which should result in a different setup group from the default setting in the PDS.

And indeed the setup group was changed to the expected value for the operation.
**Setup Optimization**

After order creation we use the PP/DS Optimizer to schedule the operations with setup groups in a sequence according to the setup matrix we defined and entered in the resource master.

Transactions:
/SAPAPO/CDPSC7 - Production Planning -> Setup Group/Setup Matrix -> Maintain Setup Matrix
/SAPAPO/RES01 - Master Data -> Resource -> Resource
Assign setup matrix to the resource.
After the planning run the operations are scheduled infinitely to the resource with no particular sequence.
The next step is the finite scheduling with the PP/DS Optimizer.

Setup costs need to have weights in the basic settings to make the optimizer create a sequence with low setup costs according to the rules defined in the setup matrix.
After the optimization we can see the finite schedule on the resource.

The operations with the same setup group are scheduled next to each other. The sequence is also from a low number to the higher number e.g. 3, 4, 5, and 6. The only exception is the first green operation. Here other constraints exist which prevent the scheduling next to the other green operation.

The setup optimization works in parallel with block scheduling.

In the screenshot you can see that the operations have different colors according to the setup group. The coloring can be done in the customizing of the graphical planning board.
Related Content
The Importance of Batches for Metals Companies

Product Definition in the Primary Metals Industry

SAP Best Practices for Primary Steel