2070351 - Example for adjustment of the MFS putaway strategy

Version	2	**Type**	SAP Note
Language	English	**Master Language**	German
Priority	Correction with medium priority	**Category**	Consulting
Release Status	Released for Customer	**Released On**	31.10.2014
Component	SCM-EWM-MFS (Material Flow System)		

Please find the original document at **https://launchpad.support.sap.com/#/notes/ 2070351**

Symptom

This SAP Note shows how the MFS standard putaway strategy can be adjusted to implement project-specific logic.

Solution

Example scenario 1:

The EWM 9.2 standard for a multi-depth automatic storage does not contain an exception logic if a handling unit (HU) cannot be put away because the product-batch combination of the HU does not yet exist in any bin, and there is no free space left. In this case, the HU circles through the conveyor system until a space becomes free due to a stock removal.

For this sample implementation, the HU is to be added to existing stock in a bin that is already partially occupied. During determination, the system determines the first bin that still has sufficient capacity to receive the HU in accordance with cross-line stock putaway. This improves bin usage, however, stock transfers are required if a covered HU is to be removed. Moreover, the system takes now into account all product-batch combinations that exist in the storage bin for the regular addition to existing stock.

This simple strategy is to show how you have to carry out a corresponding implementation. The strategy you finally choose depends on the priorities of the warehouse and is project-specific.

Implementation:

1. Create a new subclass of the class /SCWM/CL_CORE_PUTSTRA.
2. Create a new method, such as DET_BIN_ANY, in which you determine a partially occupied storage bin that already contains a different product-batch combination. For this, you can use the method DET_EMPTY_BIN as a guideline, which calls the method SELECT_EMPTY_BINS. In SELECT_EMPTY_BINS, storage bins from the table SCWM/LAGPN are already selected. During this, the field KZLER in the WHERE condition is used to include only storage bins that are completely empty. Without this condition, the system selects, as required for the sample implementation, partially occupied bins that have sufficient free capacity. Make sure that you really create a new method for this instead of overwriting an existing method because the standard behavior is still required for the determination of empty spaces.
3. Overwrite the method DET_BIN and copy the original program code into it. Call your new method DET_BIN_ANY if the methods DET_BIN_FOR_PRODHU or DET_BIN_FOR_PPHU could not determine a storage bin. This adjustment is required for the rough target bin determination in the aisle decision point.
4. Overwrite the method DET_EMPTY_BIN_AT_WT_CREA and copy the original program code into it.

Call your new method DET_BIN_ANY if the method DET_EMPTY_BIN could not determine a storage bin. This adjustment is required for the creation of the putaway warehouse task in the resource reception point.

5. Create a new implementation for the BAdI /SCWM/EX_MFS_ACT_CASE_RBD for the rough target bin determination in the aisle decision point. For the implementation, copy the sample class /SCWM/CL_EI_MFS_ACT_CASE_RBD and replace the instance of the class /SCWM/CL_CORE_PUTSTRA with an instance of your subclass in the source code to call the new logic.
6. Create a new implementation of the BAdI /SCWM/EX_CORE_PTS_EMPTY_BIN for the determination of an empty storage bin during the warehouse task creation in the resource reception point. For the implementation, copy the sample class /SCWM/CL_EX_CORE_PTS_EMPTY_BIN and replace the instance of the class /SCWM/CL_CORE_PUTSTRA with an instance of your subclass in the source code to call the new logic.

Example scenario 2:

In the EWM 9.2 standard system, in a multi-depth automatic storage, only HUs are stored together in one bin that contain the same product-batch combination. To further optimize bin occupation for an FIFO stock removal strategy, the goods receipt date is also to be considered.

Implementation:

1. Create a new subclass of the class /SCWM/CL_CORE_PUTSTRA. As an alternative, you can reuse the class from example scenario 1, to combine both implementations.
2. Overwrite the method GET_STOCK_FOR_PROD of the class /SCWM/CL_CORE_PUTSTRA. First, call the implementation of the method of the superior class to read the stock for the current product. Subsequently, determine the goods receipt date of the stock to be put away and of the stock that has been read previously. Delete the stock lines whose goods receipt date does not match that of the stock to be put away.
3. Create a new implementation for the BAdI /SCWM/EX_MFS_ACT_CASE_RBD for the rough target bin determination in the aisle decision point. For the implementation, copy the sample class /SCWM/CL_EI_MFS_ACT_CASE_RBD and replace the instance of the class /SCWM/CL_CORE_PUTSTRA with an instance of your subclass in the source code to call the new logic.
4. Create a new implementation of the BAdI /SCWM/EX_CORE_PTS_MD_ADDBIN for the determination of a bin for addition to existing stock during the warehouse task creation in the resource reception point. For the implementation, copy the sample class /SCWM/CL_EX_CORE_PTS_MD_ADDBIN and replace the instance of the class /SCWM/CL_CORE_PUTSTRA with an instance of your subclass in the source code to call the new logic.

Software Components

Software Component	Release
SCMEWM	920 - 920