Processor		
Processing Status	new	
Implementation Status	Cannot be implemented	
Language	E	
Short Text	MM17: BAdI implementation for append fields	
Component	CA-GTF-MS	Cross-Application Mass Maintenance

Description

Symptom

Append fields of the material master tables are not mass-maintainable.

Other terms

MASS; MM17; BUS1001

Reason and Prerequisites

A function is missing.
Up to now, append fields of the material master tables could not be provided without modification in the mass maintenance (refer to SA
P Note 414020).
The advance correction of this SAP Note implements a BAdI whose methods are called in the save module of the material master mass main
tenance prior to the call of the ALE inbound processing. In the BAdI methods, the system can transfer the field values of the append f
ields without modification to enhanced IDOC segments and - if required - you can change the message type and basic type for the ALE in
bound processing.
Prerequisite for the implementation of customer-specific append fields in mass maintenance MM17:
The system must be able to process the customer-specific append fields in the ALE inbound processing of the material master maintenanc
e (refer to SAP Note 44410 and 116311).

Solution

For Releases > 4.70, only the points 14-19 are relevant
Comment about the Note Assistant:
Using the Note Assistant (transaction SNOTE), you can implement only program changes. If you want to implement this SAP Note using the
 Note Assistant, you have to first manually create all new objects (table types, type group, BadI definition, and BAdI methods for the

sample implementation).
1. Create the new table type MARA_TAB as follows:

a) Call transaction SE11.

b) Select "Data type", enter the new data type MARA_TAB, and choose "Create".

c) In the "Create type MARA_TAB" dialog box, select "Table type", and press Enter.

d) Enter the short description "MARA Table".

e) Switch to the "Line Type" tab, select "Line Type", and enter the line type MARA. Switch to the "Initialization and Access" tab and
select the access type "Standard Table".

f) Switch to the "Key" tab page, select the key definition "Standard Key" and the key category "Non-Unique".

g) Save the new table type MARA_TAB in the development class MGA. Then check and activate it.
2. Create the new table type MAKT_TAB as follows:

a) Call transaction SE11.

b) Select "Data type", enter the new data type MAKT_TAB, and choose "Create".

c) In the "Create type MAKT_TAB" dialog box, select "Table type", and press Enter.

d) Enter the short description "MAKT Table".

e) Switch to the "Line Type" tab, select "Line Type", and enter the line type MAKT. Switch to the "Initialization and Access" tab and
select the access type "Standard Table".

f) Switch to the "Key" tab page, select the key definition "Standard Key" and the key category "Non-Unique".

g) Save the new table type MAKT_TAB in the development class MGA. Then check and activate it.
3. Create the new table type MARC_TAB as follows:

a) Call transaction SE11.

b) Select "Data type", enter the new data type MARC_TAB, and choose "Create".

c) In the "Create type MARC_TAB" dialog box, select "Table type", and press Enter.

d) Enter the short description "MARC Table".

e) Switch to the "Line Type" tab, select "Line Type", and enter the line type MARC. Switch to the "Initialization and Access" tab and
select the access type "Standard Table".

f) Switch to the "Key" tab page, select the key definition "Standard Key" and the key category "Non-Unique".

g) Save the new table type MARC_TAB in the development class MGA. Then check and activate it.
4. Create the new table type MBEW_TAB as follows:

a) Call transaction SE11.

b) Select "Data type", enter the new data type MBEW_TAB, and choose "Create".

c) In the "Create type MBEW_TAB" dialog box, select "Table type", and press Enter.

d) Enter the short description "MBEW Table".

e) Switch to the "Line Type" tab, select "Line Type", and enter the line type MBEW. Switch to the "Initialization and Access" tab and
select the access type "Standard Table".

f) Switch to the "Key" tab page, select the key definition "Standard Key" and the key category "Non-Unique".

g) Save the new table type MBEW_TAB in the development class MGA. Then check and activate it.
5. Create the new table type MARD_TAB as follows:

a) Call transaction SE11.

b) Select "Data type", enter the new data type MARD_TAB, and choose "Create".

c) In the "Create type MARD_TAB" dialog box, select "Table type", and press Enter.

d) Enter the short description "MARD Table".

e) Switch to the "Line Type" tab, select "Line Type", and enter the line type MARD. Switch to the "Initialization and Access" tab and
select the access type "Standard Table".

f) Switch to the "Key" tab page, select the key definition "Standard Key" and the key category "Non-Unique".

g) Save the new table type MARD_TAB in the development class MGA. Then check and activate it.
6. Create the new table type MARM_TAB as follows:

a) Call transaction SE11.

b) Select "Data type", enter the new data type MARM_TAB, and choose "Create".

c) In the "Create type MARM_TAB" dialog box, select "Table type", and press Enter.

d) Enter the short description "MARM Table".

e) Switch to the "Line Type" tab, select "Line Type", and enter the line type MARM. Switch to the "Initialization and Access" tab and
select the access type "Standard Table".

f) Switch to the "Key" tab page, select the key definition "Standard Key" and the key category "Non-Unique".

g) Save the new table type MARM_TAB in the development class MGA. Then check and activate it.
7. Create the new table type MEAN_TAB as follows:

a) Call transaction SE11.

b) Select "Data type", enter the new data type MEAN_TAB, and choose "Create".

c) In the "Create type MEAN_TAB" dialog box, select "Table type", and press Enter.

d) Enter the short description "MEAN Table".

e) Switch to the "Line Type" tab, select "Line Type", and enter the line type MEAN. Switch to the "Initialization and Access" tab and
select the access type "Standard Table".

f) Switch to the "Key" tab page, select the key definition "Standard Key" and the key category "Non-Unique".

g) Save the new table type MEAN_TAB in the development class MGA. Then check and activate it.
8. Create the new table type MLAN_TAB as follows:

a) Call transaction SE11.

b) Select "Data type", enter the new data type MLAN_TAB, and choose "Create".

c) In the "Create type MLAN_TAB" dialog box, select "Table type", and press Enter.

d) Enter the short description "MLAN Table".

e) Switch to the "Line Type" tab, select "Line Type", and enter the line type MLAN. Switch to the "Initialization and Access" tab and
select the access type "Standard Table".

f) Switch to the "Key" tab page, select the key definition "Standard Key" and the key category "Non-Unique".

g) Save the new table type MLAN_TAB in the development class MGA. Then check and activate it.
9. Create the new table type MLGT_TAB as follows:

a) Call transaction SE11.

b) Select "Data type", enter the new data type MLGT_TAB, and choose "Create".

c) In the "Create type MLGT_TAB" dialog box, select "Table type", and press Enter.

d) Enter the short description "MLGT Table".

e) Switch to the "Line Type" tab, select "Line Type", and enter the line type MLGT. Switch to the "Initialization and Access" tab and
select the access type "Standard Table".

f) Switch to the "Key" tab page, select the key definition "Standard Key" and the key category "Non-Unique".

g) Save the new table type MLGT_TAB in the development class MGA. Then check and activate it.
10. Create the new table type MVKE_TAB as follows:

a) Call transaction SE11.

b) Select "Data type", enter the new data type MVKE_TAB, and choose "Create".

c) In the "Create type MVKE_TAB" dialog box, select "Table type", and press Enter.

d) Enter the short description "MVKE Table".

e) Switch to the "Line Type" tab, select "Line Type", and enter the line type MVKE. Switch to the "Initialization and Access" tab and
select the access type "Standard Table".

f) Switch to the "Key" tab page, select the key definition "Standard Key" and the key category "Non-Unique".

g) Save the new table type MVKE_TAB in the development class MGA. Then check and activate it.
11. Create the type group MGMC

a) Call transaction SE11.

Call the "Other Dictionary Objects" dialog box by choosing "Utilities -> Other Dictionary Objects..." in the menu. Select "Type group"
, enter MGMC as the name of the type group, and choose "Create" (F5).

As of Release 4.70 (SAP R/3 Enterprise), you select the "Type group" dictionary object directly on the initial screen of transaction S
E11, enter MGMC as the name of the type group, and choose "Create".

In the "Type group MGMC: Create text" dialog box, enter the short description "Type Group Mass Maintenance Material Master", and choos
e "Save" (Enter).

In the "Create Object Directory Entry" dialog box, enter the development class or (as of Release 4.70) the package name MGA, and choos
e "Save" (Enter).

The creation of the type group automatically generates the first line of the type group MGMC:

```
          TYPE-POOL mgmc .
```

b) Implement the advance correction (refer to the corrections) in the type group MGMC. The already automatically generated context blo
ck "TYPE-POOL mgmc" forms the basis for the advance correction and adds the required type declarations.

Save, check, and activate the type group MGMC.
12. Create the BAdI definition MG_MASS_NEWSEG as follows:

a) Call transaction SE18 ("BAdI Builder").

b) Enter the following attributes:

Definition name: MG_MASS_NEWSEG

Type: Do not set the "Multiple Use" indicator.

Do not set the "Filter-Depend." indicator. Short text: "User-Specific Fields & Segments in Mass Maintenance"; Development class: MGA

c) Switch to the "Interface" tab and call the methods by double-clicking the interface name IF_EX_MG_MASS_NEWSEG.

d) Enter the method RETURN_IDOC_TYPE with the type "Instance Method" and the description "Change Message Type and IDoc Type".

e) Call the parameters of the method RETURN_IDOC_TYPE and enter the two parameters MESSAGE_TYPE and IDOC_TYPE with the type "Changing" , typing method "Type", and associated type EDI_MESTYP or EDI_IDOCTP. Do not set the "Pass Value" and "Optional" indicators, and leave the "Default Value" column initial.

Parameter	Type	Typing Method	Reference type	Description
MESSAGE_TYPE	CHANGING	TYPE	EDI_MESTYP	Message Type
IDOC_TYPE	CHANGING	TYPE	EDI_IDOCTP	Basic Type

f) Enter the method ADD_NEW_SEGMENT with the type "Instance Method" and the description "Get New Segments".

g) Call the parameters of the method ADD_NEW_SEGMENT and add the following parameters:

Parameter	Type	Typing Method	Reference type	Description
SMARA	Importing	Type	MARA_TAB	MARA Table
SMAKT	Importing	Type	MAKT_TAB	MAKT Table
SMARC	Importing	Type	MARC_TAB	MARC Table
SMBEW	Importing	Type	MBEW_TAB	MBEW Table
SMARD	Importing	Type	MARD_TAB	MARD Table
SMARM	Importing	Type	MARM_TAB	MARM Table

SMEAN	Importing Type	MEAN_TAB	MEAN Table
SMLAN	Importing Type	MLAN_TAB	MLAN Table
SMLGT	Importing Type	MLGT_TAB	MLGT Table
SMVKE	Importing Type	MVKE_TAB	MVKE Table
SELDATA	Importing Type	MASS_TABDATA	Table with Change Infos (Mass Maintenance)
T_E1MARAM	Importing Type	MGMC_E1MARAM_TAB	E1MARAM Table
T_E1MAKTM	Importing Type	MGMC_E1MAKTM_TAB	E1MAKTM Table
T_E1MARCM	Importing Type	MGMC_E1MARCM_TAB	E1MARCM Table
T_E1MBEWM	Importing Type	MGMC_E1MBEWM_TAB	E1MBEWM Table
T_E1MARDM	Importing Type	MGMC_E1MARDM_TAB	E1MARDM Table
T_E1MARMM	Importing Type	MGMC_E1MARMM_TAB	E1MARMM Table
T_E1MEANM	Importing Type	MGMC_E1MEANM_TAB	E1MEANM Table
T_E1MLANM	Importing Type	MGMC_E1MLANM_TAB	E1MLANM Table
T_E1MLGTM	Importing Type	MGMC_E1MLGTM_TAB	E1MLGTM Table
T_E1MVKEM	Importing Type	MGMC_E1MVKEM_TAB	E1MVKEM Table
T_IDOC_DATA	Changing Type	EDIDD_TT	Table Type for

			EDIDD	
MSG	Changing	Type	MASS_MSGS	Message for Mass Changes

For all these parameters, do not set the "Pass Value" and "Optional" indicators, and leave the "Default Value" column initial.

h) Return to the attributes of the BadI definition, and in the menu, choose "Goto -> Sample code -> Create" to generate the BadI methods for the sample implementation.

i) Maintain the sample source code from the advance correction (see corrections:

METH CL_EXM_IM_MG_MASS_NEWSEG IF_EX_MG_MASS_NEWSEG~ADD_NEW_SEGMENT

METH CL_EXM_IM_MG_MASS_NEWSEG IF_EX_MG_MASS_NEWSEG~RETURN_IDOC_TYPE)

as sample implementations.

j) Save, check, and activate the BAdI.
13. Implement the advance correction (see corrections) in the function module MASS_MATMAS03_CHANGE. This advance correction calls the BAdI methods in the save module of the mass maintenance before the save module MASS_MATMAS03_CHANGE calls the ALE inbound processing of the material master maintenance for the update of the mass changes.

14. In the save module MASS_MATMAS03_CHANGE, the system first calls the BAdI method ADD_NEW_SEGMENT. Using a suitable implementation of this method, the system can supply the required material master append fields with new field values.
The BAdI is delivered with a sample implementation that describes how to supply customer-specific append fields or IS-OIL-specific append fields with new field values.

a) Prerequisite:
Customer-specific append fields:
Add the required customer-specific IDoc segments (for example, ZE1MARAM for the MARA-data, ZE1MARCM for the MARC data, and so on) to the standard IDocs as described in SAP Notes 44410 and 116311.
IS-OIL-specific append fields:
In the IS-OIL SAP R/3 system, the required IDoc segments E1OILMA, E1OILMC, and E1OILMW already exist.

a) Use transaction SE19 to create an implementation name, for example ZZ_MG_MASS_NEWSEG for the BAdI

definition name MG_MASS_NEWSEG. Call the method ADD_NEW_SEGMENT in transaction SE19 by switching to the "Interface" tab and double-clicking the method ADD_NEW_SEGMENT.
Copy the sample implementation of the method ADD_NEW_SEGMENT delivered with the BAdI (Call transaction SE18 -> enter the definition name MG_MASS_NEWSEG -> choose "Display" -> in the menu, choose "Goto -> Sample code -> Display".) or sample implementation CL_EXM_IM_MG_MASS_NEWSEG of the interface method IF_EX_MG_MASS_NEWSEG~ADD_NEW_SEGMENT described in the correction instructions (refer to the corrections) into the implementation you have created.

b) Adjust the sample implementation to your requirements. In SAP R/3 systems that are not IS-OIL-specific, all source code parts that are flagged as IS-OIL-specific must be deleted. In IS-OIL SAP R/3 Systems, all source code parts that are flagged as customer-specific must be deleted if no other customer-specific append fields are to be supplied in addition to the IS-OIL-specific append fields.

c) In the local TYPES declaration of your method implementation, you must delete the lines flagged with "<<< DELETE LINE" because these lines are required in the sample implementation for syntactic reasons. You must delete the comment "*" before the lines flagged with "<<< INSERT LINE" in order to include the customer-specific or IS-OIL-specific IDoc segments. In the sample implementation, this is not possible for syntactic reasons because the required customer-specific or IS-OIL-specific IDoc segments do not exist in the SAP R/3 standard system. Check, save, and activate the created implementation of the method ADD_NEW_SEGMENT.

15. In the save module of the mass maintenance MASS_MATMAS03_CHANGE, the message type MESSAGE_TYPE = 'MATMAS' and the basic type IDOC_TYPE = 'MATMAS03' are set for the ALE inbound processing.
Prior to the call of the ALE inbound processing, the system calls the second BAdI method RETURN_IDOC_TYPE. In the method RETURN_IDOC_TYPE, you can change the message type and/or the basic type.

SAP R/3 standard system

In SAP R/3 standard systems, neither the message type nor the basic type can be changed, otherwise the ALE inbound processing does not work correctly.

IS-OIL SAP R/3

In IS-OIL systems, you must replace the message type MESSAGE_TYPE = 'MATMAS' with 'OILMAT' and the basic type IDOC_TYPE = 'MATMAS03' with 'OILMAT03' if you want to change IS-OIL-specific fields via MM17. To do so, similar to the above described implementation of the method ADD_NEW_SEGMENT, use transaction SE19 to call the implementation name (for example ZZ_MG_MASS_NEWSEG) that you have created. Then copy the sample implementation for the method RETURN_IDOC_TYPE to the implementation of the method RETURN_IDOC_TYPE that you have created. Save, check, and activate the created implementation for the method RETURN_IDOC_TYPE.
16. Note that the system processes your implementations of the BAdI methods ADD_NEW_SEGMENT and RETURN_IDOC_TYPE only if you choose "Activate" (Ctrl+Shift+F4) on the initial screen of transaction SE19 and if you enter the implementation name you have chosen (for example ZZ_MG_MASS_NEWSEG).
17. To include the required append fields, which should be generally available in the mass maintenance for

changes, in the field list
(table MASSFLDLST) of the mass maintenance, proceed as follows:

a) Call transaction MASSOBJ.

b) Select the object type BUS1001 ("Materials (industry)" and double-click "Application tables".

c) Select the desired table, for example, MARA, and call the field list.

d) Add the required new entries and save your changes.

In IS-OIL SAP R/3 systems, you must add the following new entries to the field list of the mass maintenance so that you can select the IS-OIL-specific material master fields in the mass maintenance for changes:

Object type	Table name	Field name
BUS1001	MARA	OIGROUPNAM
BUS1001	MARA	OITRIND
BUS1001	MARA	OIHMTXGR
BUS1001	MARC	OITAXGRP
BUS1001	MARC	OIOILCON
BUS1001	MARC	OIEDCODE
BUS1001	MARC	FDICH
BUS1001	MARC	COEFF
BUS1001	MARC	UOMGR
BUS1001	MARC	UMRSL
BUS1001	MARC	ABFAC
BUS1001	MBEW	OITAXKEY
BUS1001	MBEW	OIPPINV
BUS1001	MBEW	OIHANTYP

18. In IS-OIL SAP R/3 systems, the mass maintenance functions work correctly only if the advance corrections of the IS-OIL-specific SAP Notes 567779 and 593215 are implemented.

19. The mass maintenance of customer-specific append fields works only if the prerequisites for the distribution of these fields are fulfilled via ALE (refer to SAP Notes 44410 and 116311):

As already described above, customer-specific segments must be added to the standard IDocs for the mass maintenance and for the ALE distribution of customer-specific append fields. In the ALE inbound processing of the material master maintenance, the field values of customer-specific append fields must be transferred from the customer-specific IDoc segments to the internal tables of the ALE inbound processing. For the processing of enhanced IDocs, the enhancement MGV00001 with the customer exit EXIT_SAPLMV02_002 is available for the posting of the IDocs.

If the check of foreign key dependencies or of domain fixed values or another check is required for a customer field in the ALE inbound processing, you can implement the required checks in the function exit EXIT_SAPLMGMU_001 (SAP enhancement MGA00001).

Valid Releases

Software Component	Release

SAP Application	
	46C
	470
	500
	600
	602
	603
	604
	605
	606
	616
	617
	618
	617
	101
	102
	103
	104
	105

References to Relevant Support Packages

Software Component	Release	Package Name
SAP Application		
	46C	SAPKH46C43
	470	SAPKH47008