Note Language: English **Version: 2** **Validity:** Valid Since 27.02.2006

Summary

Symptom
As of Release 4.70 customers of SAP and IS-solutions or SAP-partners do have the possibility to integrate the maintenance of own vendor master fields without modification into the standard-dialog of the vendor master.
With the help the Business Add-In (BAdI) technology you can provide interfaces withi which you can maintain unknown fields in the standard.
In the vendor master dialog the fields are admitted via customer-specific subscreens into the maintenance of the vendor master. These subscreens are announced to the standard via interfaces.
The interfaces are available in the standard batch input of the vendor master so that the maintenance of the fields can also be integrated via batch input into the batch input program RFBIKR00 delivered by the standard.
The ALE-outbound of the vendor master provides interfaces which integrate the filling of the segments of the vendor master IDoc that do not belong to the standard into the standard procedure.
The ALE inbound of the vendor master provides interfaces that allow to integrate the data the non-standard-segments contained in the vendor master IDoc into the table of the batch input data calculated by the standard-program.

More Terms
SAPMF02K
CREMAS, IDOC_INPUT_CREDITOR, MASTERIDOC_CREATE_CREMAS

Cause and Prerequisites
no

Solution
User-specific fields can be attached without modification to existing standard tables of the vendor master by means of the Append technology, or you work with user-specific transparent tables in order to store your data in the database. The basic difference between these is that you are responsible yourself when you use your own user-specific tables for the data update and the creation of change documents (if desired at all). If you attach your fields to existing standard-tables by means of the Append-method, the data update and the creation of change documents are taken over by the standard.
The following activities are necessary in order to allow the maintenance of user-specific fields or tables in the vendor master dialog:

- Settings in the Customizing
- Providing subscreens about about which a maintenance of your fields is allowed from out of the dialog.
- Creating active BAdI implementations so that the data exchange or the communication between the standard and your enhancements can occur via the provided BAdI methods.

Settings in the Customizing
In the Customizing settings of the vendor master under point 'Adoption of

Customer's Own Master Data Fields' you find IMG activity 'Prepare Modification-Free Enhancement of Vendor Master Record'. Here you can create a screen group with your name. In addition, you can assign up to 32 labelings of tab pages to your screen group.
The name of the screen group appears in form of a pushbutton on the main screens of the vendor master. In order for the pushbutton becoming visible online, you must create an active implementation of BAdI VENDOR_ADD_DATA and inform the standard program via method CHECK_ADD_ON_ACTIVE that you want to use your enhancement actively. Please bear in mind that during the runtime of the vendor master dialog a maximum of 10 screen groups can be open actively at the same time.
After pressing the pushbutton in the vendor master dialog you branch to a screen that provides (a maximum of) 32 tab pages. The labeling of these tab pages results from the labeling which you defined in the Customizing for the tab pages of your screen group.
Each tab page provides an own subscreen on which you can edit your own data. The integration of your subscreen into the dialog occurs by means of methods of the filter-dependent BAdI VENDOR_ADD_DATA_CS, where you must select your screen group as filter.
Further information is available in the documentation for the IMG activity.

Which BAdIs are available?
The following BAdIs with diverse methods are available for the modification-free inclusion of the maintenance of your fields:

- VENDOR_ADD_DATA_CS

  This filter-dependent BAdI serves for the inclusion of your subscreens into the vendor master dialog. On the subscreens you implement the maintenance of your own fields. The filter is the screen group that you created in the Customizing. Due to the filter dependency it is achieved that only the active implementation of the respective screen group will run in the flow control of the subscreen container from the standard which includes your subscreen at runtime.

- VENDOR_ADD_DATA

  This BAdI does not have any filter dependency. As the BAdI is delivered as a repeatedly usable BAdI, you should be aware that all active implementations of the individual methods will run in systems in which several screen groups are simultaneously active.

  The BAdI includes methods which indeed occur in the interactive environment, but which are independent of the subscreen container in which your subscreens are included at runtime. For example, these might be methods for the preassignment of data or for the saving of data.

- VENDOR_ADD_DATA_BI

  This BAdI does not have any filter dependency. As the BAdI is delivered as a repeatedly usable BAdI, you should be aware that in systems in which several screen groups are simultaneously active all active implementations of the individual methods will run individually.

The BAdI is used in the field of ALE distribution and standard batch input. There are available methods with which you can fill your own ALE segments in the ALE outbound or make an evaluation of your own change pointers. The ALE-inbound and the Standard-Batch-Input of the vendor master provide methods with which you can integrate your own data into the batch input data calculated by the standard program.

Within the ALE distribution the methods are only provided for the segments or tables that can be edited analog to the standard batch input or whose table enhancement is also supported in the vendor master dialog. Only when these conditions are met, a user can connect himself without modification in order to import own data. The following tables and their ALE segments are supported: LFA1, LFB1, LFM1, LFM2, LFBK, WYT1, WYT1T, WYT3.

Further documentation on the BAdIs is available in the Customizing settings of the vendor master under the IMG activities for the individual BAdIs. An online documentation of the individual BAdI-methods is available in the system. The easiest way to access these is via the Class Builder (Transaction SE24). Specify the interface of the BAdI as an object type. The interface results from the name of the BAdI to which ID 'IF_EX_' is prefixed (the interface of the BAdI VENDOR_ADD_DATA is called IF_EX_VENDOR_ADD_DATA). The tab page with the individual methods provides a pushbutton for the documentation. By means of cursor positioning you select the method whose documentation you want to look at.

General information on the dialog

It was already described which Customizing settings are required for providing a pushbutton on the main screens of the vendor master dialog with which you can reach the screen with your tab pages. It is important that you use method CHECK_ADD_ON_ACTIVE of BAdI VENDOR_ADD_DATA to inform the standard program that your enhancement is supposed to be used actively.

With method GET_TAXI_SCREEN of BAdI VENDOR_ADD_DATA_CS you inform the standard program about the subscreen which shall be displayed on a tab page.

Methods SET_DATA and GET_DATA of BAdI VENDOR_ADD_DATA_CS provide current data from the standard program or return the data to the standard program. This is important if you have used the Append method to extend the standard tables by your own fields and you now maintain these fields on one of your tab pages.

Method SET_FCODE of BAdI VENDOR_ADD_DATA_CS informs you about the function code which the user triggered after the screen with your tab page was displayed to him. However, the method is not called if the user wants to branch to one of your other tab pages. This change is implemented through the standard program in connection with method GET_TAXI_SCREEN.

If a user wants to look at the change documents for a field on your subscreen, you inform the standard program via method GET_FIELDNAME_FOR_CHANGEDOC of BAdI VENDOR_ADD_DATA_CS about the name of the field for which the change documents are to be determined. If this is a field from one of your own tables you must then define an own change

document object for this table and ensure in your update routines that the system writes change documents for this change document object. With method GET_CHANGEDOC_FOR_OWN_TABLES of BAdI VENDOR_ADD_DATA you inform the standard about your change document object. It is important that you store change documents for your change document object with the vendor number as Object-Id in the system.

With method SUPPRESS_TAXI_TABSTRIPS of BAdI VENDOR_ADD_DATA_CS you can hide tab pages depending on the activity and the organizational data. For example, you can hide a tab page which contains company code data if no company code was entered.

Method INITIALIZE_ADD_ON_DATA of BAdI VENDOR_ADD_DATA signals that you should make an initialization of your work structures. It is called when accessing the dialog and when the user without exiting the transaction changes the vendor during the processing of a vendor.

Method READ_ADD_ON_DATA of BAdI VENDOR_ADD_DATA which is called during the display or change of existing vendors is intended for reading your own tables and filling your work structures.

With method CHECK_DATA_CHANGED of BAdI VENDOR_ADD_DATA you can inform the standard program about whether the user made changes to your own tables. This method is not important for you if you have exclusively attached your fields to existing standard tables by means of the Append method. Method SAVE_DATA of BAdI VENDOR_ADD_DATA is then available in order to save your own tables. It is important that you do not execute any COMMIT WORK in your implementation of method SAVE_DATA; the triggering of the update tasks occurs in the standard program.

After the user pressed the 'Save' button, method CHECK_ALL_DATA of BAdI VENDOR_ADD_DATA will run. Here, you can check the current data regarding consistency again.

If you work with the internal number assignment during the setup of a vendor, method MODIFY_ACCOUNT_NUMBER of BAdI VENDOR_ADD_DATA provides the option to change the vendor number determined by the system. For example, it could be supplemented by a check digit. The method is not called if the creation of the vendor occurs via the ALE interface.
It is the BadI responsibility to ensure the checks and locking mechanism for the number generated by it.

If you work with the external number assignment during the setup of a vendor, the system provides method CHECK_ACCOUNT_NUMBER of BAdI VENDOR_ADD_DATA for the execution of additional checks on the entered vendor number which are not covered by the number range. The method is not called if the creation of the vendor occurs via the ALE interface.

General information on ALE and batch input

If in the following it is referred to batch input data, this will mean data whose structure corresponds to the batch structures of the vendor master. For example, these batch structures are BLFA1 or BLFB1. The data can be stored in a data file as this is the case in the traditional batch input. They can, however, also be stored also at runtime in an internal table as for example it is used in the ALE inbound. Further information on the setup of batch input data is available in the documentation of the standard batch input program for vendor master RFBIKR00.

Roughly described, the following process occurs in the ALE inbound: the data from the IDOC segments is transferred to the batch structures of the vendor master. From these batch structures an internal table with batch input data is set up which is transferred to standard function module VENDOR_BDCDATA. This standard module processes the batch input data and stores its contents in an internal table in the batch input format which is finally imported by means of CALL TRANSACTION USING .... In this process, the system always calls the transaction for the creation or changing of a vendor. The internal table transferred to module VENDOR_BDCDATA with the batch input data always contains only the data which can be edited in a transaction.
The standard batch input program for vendor master RFBIKR00 expects batch input data as an entry. These are processed and stored in an internal table from which the batch input session is generated in the batch input format. The common aspect between ALE inbound and standard batch input is that from batch input data the system sets up an internal table in the batch input format.

Within the ALE technology for the distribution of vendor master data only the end user has the possibility to admit own fields into customer-specific segments and to distribute these together with the standard segments. This option is not available for IS-solutions or SAP-partners as the extensibility of ALE provides only a single-level concept.

In the ALE outbound provides method FILL_ALE_SEGMENTS_OWN_DATA of BAdI VENDOR_ADD_DATA_BI which serves for the setup of the customer-specific segments. Ineach case the method is called after the processing of the following standard-segments: E1LFA1M, E1LFA1A, E1LFB1M, E1LFM1M, E1LFM2M, E1WYT3M, E1LFBKM, E1WYT1M, E1WYTTM. Table parameter T_IDOC_DATA contains the data from the segments that were set up so far. You must fill your customer-specific segment with data and attach it to table T_IDOC_DATA if it is hierarchically subordinate to the standard segment from parameter I_SEGMENT_NAME.

Method PROCESS_ALE_OWN_CHANGE_POINTER of BAdI VENDOR_ADD_DATA_BI is available in the ALE outbound in order to trigger a vendor master distribution due to the evaluation of own change pointers. The method is only interesting if you store your own data for the vendor in your user-defined tables that treat changes to these tables through an own change document object and if change pointers are generated from the change documents. The change documents for this change document object must be stored in the system with the vendor number as object ID.
The idea behind this is that you distribute your own data via customer-specific segments and that these segments are subordinate to the standard segments (if your customer-specific segment includes general data it is subordinate to standard segment E1LFA1M; if your customer-specific segment includes company code data it is probably subordinate to standard segment E1LFB1M). For example, when you fill standard segment E1LFB1M in the standard module that creates the IDocs because the company code data is to be transferred, you can accordingly fill your customer-specific segments, that are subordinate to the company code segment (by means of method FILL_ALE_SEGMENTS_OWN_DATA). During the evaluation of your change pointers you must therefore inform the standard-program about the key fields of the standard tables which are superior to your customer-specific segments so that the transfer of the standard table is triggered by that.

The ALE inbound provides method PASS_NON_STANDARD_SEGMENT of BAdI VENDOR_ADD_DATA_BI which is called during the processing of every

customer-specific segment. It serves for the transfer of the segment-data to your application so that you can collect your data. Using method FILL_BI_TABLE_WITH_OWN_SEGMENT of BAdI VENDOR_ADD_DATA_BI you can attach your data collected from the segments to the table of the batch input data calculated by the standard. It is important that you append your data in the form of batch structures. These batch structures must correspond in their structure to the batch structures of the standard (for example BLFA1). The first byte must include the record type ('2), the next 30 bytes must include the name of the batch structure. The remaining bytes are freely available. The overall length of the structure must not exceed 2000 bytes.
The separation between the two methods PASS_NON_STANDARD_SEGMENT and FILL_BI_TABLE_WITH_OWN_SEGMENT is necessary as the segments of an IDoc are not necessarily processed by a transaction.
Example: the IDoc contains both general data and company code data. While the general data is already existing in the system, the company code data does not exist yet. As a consequence, the system calls 2 transactions about which the data is imported. During the processing of the customer-specific segments in the standard (that is, the call of method PASS_NON_STANDARD_SEGMENT), however, this split into 2 different transactions not carried out yet. Method FILL_BI_TABLE_WITH_OWN_SEGMENT is called before the table of the batch input data is transferred to function module VENDOR_BDCDATA. Parameter I_TRANS_DATA provides information on with which transaction and with which organizational data the next CALL TRANSACTION is executed in order to import the vendor master data.

In function module VENDOR_BDCDATA (ALE case) or in program RFBIKR00 (Batch input) the system calls method CHECK_DATA_ROW of BAdI VENDOR_ADD_DATA_BI for every record that contains batch input data on a structure that is not known to the standard. With this method you can execute checks on the batch input data. If you are the owner of the batch input data (the BAdI is repeatedly usable), you must inform the standard program in parameter E_STRUCTURE_CHECKED that you checked the data.

Method FILL_FT_TABLE_USING_DATA_ROWS of BAdI VENDOR_ADD_DATA_BI is available in function module VENDOR_BDCDATA (ALE case) or in program RFBIKR00 (Batch input). It is called once per transaction with which the data is imported in the batch input format. With that you can integrate your own batch input data in the batch input format into the data table created by the standard program. This table is available via parameter ET_FT. After the call of the method the function code for the saving of the data is finally attached to table ET_FT by the standard.
If you have integrated the maintenance of your own data without modification into the standard-dialog with the help of BAdIs VENDOR_ADD_DATA and VENDOR_ADD_DATA_CS, a pushbutton for branching to the maintenance of your data is generated. In this case, the system dynamically assigns a function code to this pushbutton which only results during the runtime of the dialog. The table of the data in the batch format to be imported that you must extend by means of method FILL_FT_TABLE_USING_DATA_ROWS by the processing of your data must include the function code for branching to the maintenance of your data if you want to maintain your data on your screens. As the function code is not know at the time of the calculation of the data to be imported, you must select character string 'BAO' for the function code of your pushbutton followed by the screen group, under which you in the Customizing of the vendor master prepared your enhancements of the vendor master without modification.
Example: if your enhancement was defined under screen group Z1 in the Customizing, you must select 'BAOZ1' as a function code for branching to

the maintenance of your data.

Restrictions

- With regard to extensibility the ALE technology provides only a single-level concept. Therefore, only the end user can admit own fields into customer-specific segments and distribute these together with the standard segments. This option is not available for IS-solutions or SAP-partners.
- During the connection of your own fields to the tables of the standard by means of the Append technology you must bear in mind that the system does not support tables whose processing does not appear within module pool SAPMF02K of the vendor master. In methods SET_DATA and GET_DATA of BAdI VENDOR_ADD_DATA_CS there are parameters available for these tables so that a data exchange between the standard program and your enhancement is not supported. These are tables LFAT, LFEI and LFBW.
- An integration of your fields into the Customizing maintenance of the field status of the vendor master fields is not supported.
- A support of the user-defined fields in the mass maintenance of the vendor master (Transaction XD99) is not available.
- Contact persons for a vendor (Table KNVK) are only available in a Retail system.

Header Data

Release Status: Released for Customer
Released on: 27.02.2006 09:20:29
Original Lang.: English
Priority: Recommendations/additional info
Category: Consulting

Main Component: LO-MD-BP-VM Vendor Master
Additional Components:
FI-AP-AP-N Master Data

Valid Releases

Software Component	Release	From Release	To Release	and Subsequent
SAP_APPL	470	470	470	

Related Notes

Number	Short Text
577502	Enhancements without modification in customer master record
430663	Master data enhancement: Critical objects