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

Summary

Symptom
As of Release 4.70 customers of SAP as well as of IS-solutions or SAP-partners can integrate the maintenance of individual customer master fields without modification into the standard-dialog of the customer master record. With the help the Business Add-In (BAdI) technology you provide interfaces with which you can maintain unknown fields in the standard.
In the customer master record dialog the fields are admitted via customer-specific subscreens into the maintenance of the customer master. The Standard is informed about these subscreens via interfaces.
In the standard batch input of the customer master there are available interfaces so you can also integrate the maintenance of the fields via batch input into batch input program RFBIDE00 which is delivered by the standard.
The ALE-outbound of the customer master provides interfaces which integrates the filling of the segments of the customer master IDoc that do not belong to the standard into the standard procedure.
The ALE-inbound of the customer master provides interfaces that allow to integrate the data contained in the customer master IDoc from the non-standard-segments into the table of the batch input data calculated by the standard-program.

More Terms
SAPMF02D
DEBMAS, IDOC_INPUT_DEBITOR, MASTERIDOC_CREATE_DEBMAS

Cause and Prerequisites
None.

Solution
You can attach own fields without modification to existing standard tables of the customer master by means of the Append technology, or you work with own transparent tables in order to store your data in the database. The basic difference is that you are responsible yourself when you use own tables for the data update the creation of change documents (if desired). If you use the Append method to attach your fields to existing standard-tables, the standard will take over the data update and the creation of change documents as well.
The following activities are required to allow the maintenance of own fields or own tables in the customer master record dialog:

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

Settings in the Customizing
In the Customizing settings of the customer master under point 'Adoption of Customer's Own Master Data Fields' you will find IMG activity 'Prepare

Modification-Free Enhancement of Customer 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 the form of a pushbutton on the main screens of the customer master. In order for the pushbutton to become visible in the dialog, you must create an active implementation of BAdI CUSTOMER_ADD_DATA and inform the standard program via method CHECK_ADD_ON_ACTIVE that you want to use your enhancement actively. Bear in mind that for the runtime of the customer master dialog of you can actively use a maximum of 10 screen groups simultaneously.
After you activated the pushbutton in the customer master dialog you branch to a screen that provides (maximally) 32 tab pages. The labeling of these tab pages is derived from the labeling which you defined in the Customizing for the tab pages of your screen group.
Every 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 CUSTOMER_ADD_DATA_CS, where you must select your screen group as filters.
Further information is available in the documentation for the IMG activity.

Which BAdIs are available?
For the modification-free inclusion of the maintenance of your fields the following BAdIs with various methods are available:

- CUSTOMER_ADD_DATA_CS

  This filter-dependent BAdI serves for the inclusion of your subscreens into the customer master record dialog. On the subscreens you implement the maintenance of your own fields. The filter is the screen group created by you in the Customizing. With 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 in which your subscreen for the runtime is included.

- CUSTOMER_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 in systems with several simultaneously active screen groups all active implementations of the individual methods will be run.

  The BAdI contains methods which occur in the interactive environment but which are independent of the subscreen container in which your subscreens are included for the runtime. For example these are methods for the defaulting of data or for the saving of the data.

- CUSTOMER_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 are run.

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

The ALE distribution provides the methods only for the segments or tables that can be edited analogously to the standard batch input or whose table enhancement is also supported in the customer 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: KNA1, KNB1, KNVV, KNVK, KNEX, KNVA, KNVP, KNVD, KNVI, KNVL.

Further information on the BAdIs is available in the Customizing settings of the customer master record 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). As object type you must specify the interface of the BAdI. The interface results from the name of the BAdI to which the ID 'IF_EX_' is prefixed (the interface of BAdI CUSTOMER_ADD_DATA is called IF_EX_CUSTOMER_ADD_DATA). A pushbutton for the documentationis available on the tab page with the individual methods. By positioning the cursor you select the method whose documentation you want to consult.

General information on the dialog
It was already described which Customizing settings are necessary so that a pushbutton about which you reach the screen with your tab pages is provided on the main screens of the customer master dialog. It is important that you inform the standard program via method CHECK_ADD_ON_ACTIVE of BAdI CUSTOMER_ADD_DATA that your enhancement is supposed to be used actively.

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

Through methods SET_DATA and GET_DATA of BAdI CUSTOMER_ADD_DATA_CS the system provides you with current data from the standard program or returns the data to the standard program. This is important if you have used the Append method to extend the standard tables by own fields and maintain these fields on one of your tab pages.

Method SET_FCODE of BAdI CUSTOMER_ADD_DATA_CS provides you with the function code that 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 change 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 use method GET_FIELDNAME_FOR_CHANGEDOC of BAdI CUSTOMER_ADD_DATA_CS to inform the standard program 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 that change documents for this change document object are written in your update routines. Use method GET_CHANGEDOC_FOR_OWN_TABLES of BAdI CUSTOMER_ADD_DATA to infrom the standard about your change document object. In this process, it is important that change documents for your change document object are stored in the system with the customer number as object ID.

With method SUPPRESS_TAXI_TABSTRIPS of BAdI CUSTOMER_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 CUSTOMER_ADD_DATA indicates that you should make an initialization of your work structures. It is called when you access the dialog and when the user changes the customer during the processing of a customer without exiting the transaction.

Method READ_ADD_ON_DATA of BAdI CUSTOMER_ADD_DATA which is called when you display or change existing customers is intended for reading your own tables and filling your work structures.

With method CHECK_DATA_CHANGED of BAdI CUSTOMER_ADD_DATA you can inform the standard program whether the user made changes to your own tables. This method is not important for you if you have used the Append method to attach your fields exclusively to existing standard-tables. Method SAVE_DATA of BAdI CUSTOMER_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 CUSTOMER_ADD_DATA will run. Here, you check the current data for consistency again.

If you work with internal number assignment during the setup of a customer, you can use method MODIFY_ACCOUNT_NUMBER of BAdI CUSTOMER_ADD_DATA to change the customer number that was determined from the system-side. For example, you could add a check digit to it. The method is not called if the creation of the customer 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 external number assignment during the setup of a customer, you can use method CHECK_ACCOUNT_NUMBER of BAdI CUSTOMER_ADD_DATA in order to carry out additional checks onto the entered customer number which are not covered by the number range. The method is not called if the creation of the customer occurs via the ALE interface.

General information on ALE and batch input

If in the following it is spoken of batch input data this means data whose structure corresponds to the background structures of the customer master. For example, these background structures are BKNA1 or BKNB1. The data can be stored in a data file as this is the case in the traditional batch input. However, the data can also be stored for the runtime in an internal table such as used in the ALE-inbound, for example. Further information on the setup of batch input data is available in the documentation of standard batch input program the customer master record RFBIDE00.

Roughly described, the following process occurs in the ALE inbound: the data from the IDoc segments is transferred to the background structures of the customer master. From these background structures the system sets up an internal table with batch input data which is transferred to standard function module DEBITOR_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 transaction for the creation or changing of a customer is always called. Here, the internal table with the batch input data transferred to module DEBITOR_BDCDATA always contains only the data which can be edited in a transaction.
The standard batch input program for customer master RFBIDE00 expects batch input data as an entry. These are processed and stored in an internal table in the batch input format from which the batch input session is generated. The common aspect between ALE-inbound and standard batch input lies in the fact that from batch input data an internal table is set up in the batch input format.

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

The ALE-outbound provides method FILL_ALE_SEGMENTS_OWN_DATA of BAdI CUSTOMER_ADD_DATA_BI which serves for the setup of the customer-specific segments. The method is called after the processing of the following standard segments: E1KNA1M, E1KNA11, E1KNVVM, E1KNVPM, E1KNVDM, E1KNVIM, E1KNVLM, E1KNB1M, E1KNBKM, E1KNVAM, E1KNVKM, E1KNEXM. Table parameter T_IDOC_DATA contains the data of the segments that are currently set up. Fill your customer-specific segment with data and attach it to table T_IDOC_DATA when it is hierarchically subordinate to the standard segment from parameter I_SEGMENT_NAME.

Method PROCESS_ALE_OWN_CHANGE_POINTER of BAdI CUSTOMER_ADD_DATA_BI is available in the ALE-outbound to trigger a customer master distribution due to the evaluation own change pointers. The method is only interesting if you store own data for the customer in own tables that process changes to these tables via 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 customer number as object ID.
The basic 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 E1KNA1M; if your customer-specific segment includes company code data, it is probably subordinate to standard segment E1KNB1M). If standard segment E1KNB1M is filled in the standard module that creates the IDocs because the company code data is to be transferred, you can - taking this as a basis - fill your customer-specific segments that are subordinate to the company code segment (by means of method FILL_ALE_SEGMENTS_OWN_DATA). Therefore you must during the evaluation of your change pointers inform the standard program about the key fields of the standard tables which are superior to your customer-specific segments so that this triggers the transfer of the standard table.

The ALE-inbound provides method PASS_NON_STANDARD_SEGMENT of BAdI

CUSTOMER_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. With method FILL_BI_TABLE_WITH_OWN_SEGMENT of BAdI CUSTOMER_ADD_DATA_BI you can then attach your data collected from the segments to the table of the batch input data calculated by the standard. In this process it is important that you append your data in the form of batch structures. These batch structures must in their structure correspond to the batch structures of the standard (for example BKNA1). 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 with 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) this split into 2 different transactions is however 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 DEBITOR_BDCDATA. Parameter I_TRANS_DATA provides you with information on with which transaction and with which organizational data the next CALL TRANSACTION is executed in order to import the customer master data.

In function module DEBITOR_BDCDATA (ALE-case) or in program RFBIDE00 (Batch input) the system calls method CHECK_DATA_ROW of BAdI CUSTOMER_ADD_DATA_BI for each record that contains batch input data for a structure that is not known to the standard. The method allows you to 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.

In function module DEBITOR_BDCDATA (ALE-case) or in program RFBIDE00 (Batch input) method FILL_FT_TABLE_USING_DATA_ROWS of BAdI CUSTOMER_ADD_DATA_BI is available. It is called once per transaction with which the data is imported in the batch input format. It allows you to integrate your batch input data into the table of the data in the batch input format created by the standard program. With parameter ET_FT this table is made available. After you call the method the standard finally attaches the function code to table ET_FTfor the saving of the data.
If you have integrated the maintenance of your own data without modification into the standard-dialog with the help of BAdIs CUSTOMER_ADD_DATA and CUSTOMER_ADD_DATA_CS, the system provides a pushbutton for branching to the maintenance of your data. In the course of this the system dynamically assigns a function code to this pushbutton which only arises during the runtime. The table of the data to be imported in the batch input format 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 known 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 prepared your modification-free enhancments of the customer master in the Customizing of the customer

master.
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. For this reason 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 to IS-solutions or SAP-partners.
- During the connection of the own fields to the tables of the standard by means of Append technology you must bear in mind that the system does not support tables whose processing does not lie within module pool SAPMF02D of the customer master. Methods SET_DATA and GET_DATA of BAdI CUSTOMER_ADD_DATA_CS do not provide any parameters for these tables so that a data exchange between the standard program and your enhancement is not supported. These are tables KNAT, KNBW, WRF4 and WRF12.
- An integration of your fields into the Customizing maintenance of the field status of the customer master fields is not supported.
- A support of the user-defined fields in the mass maintenance of the customer master record (Transaction XD99) is not available.

Header Data

Release Status: Released for Customer
Released on: 27.02.2006 09:19:56
Original Lang.: English
Priority: Recommendations/additional info
Category: Consulting

Main Component: LO-MD-BP-CM Customer Master
Additional Components:
FI-AR-AR-N Master Data

Valid Releases

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

Related Notes

Number	Short Text
580266	Enhancements without modification in vendor master
430663	Master data enhancement: Critical objects