SDN Community Contribution

(This is not an official SAP document.)

Disclaimer & Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.

Applies To:

Business Add-Ins

Summary

Definition: User Exits allow customers to attach additional code to standard SAP source code without having to modify the original object. Business Add-Ins are a new SAP enhancement technique based on ABAP Objects.

By: Aveek Ghose

Company: Consultant, IBM India

Date: 08 April 2005

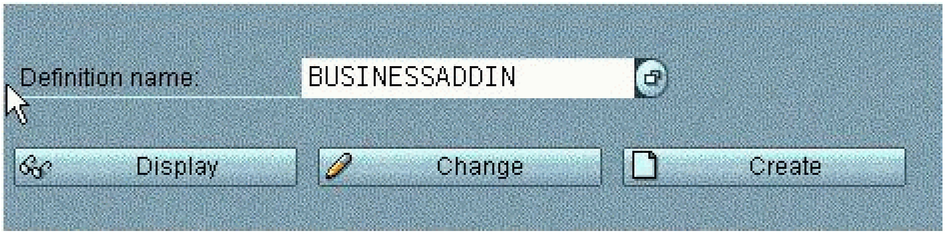
Table of Contents

Definition

User Exits allow customers to attach additional code to standard SAP source code without having to modify the original object. Business Add-Ins are a new SAP enhancement technique based on ABAP Objects.

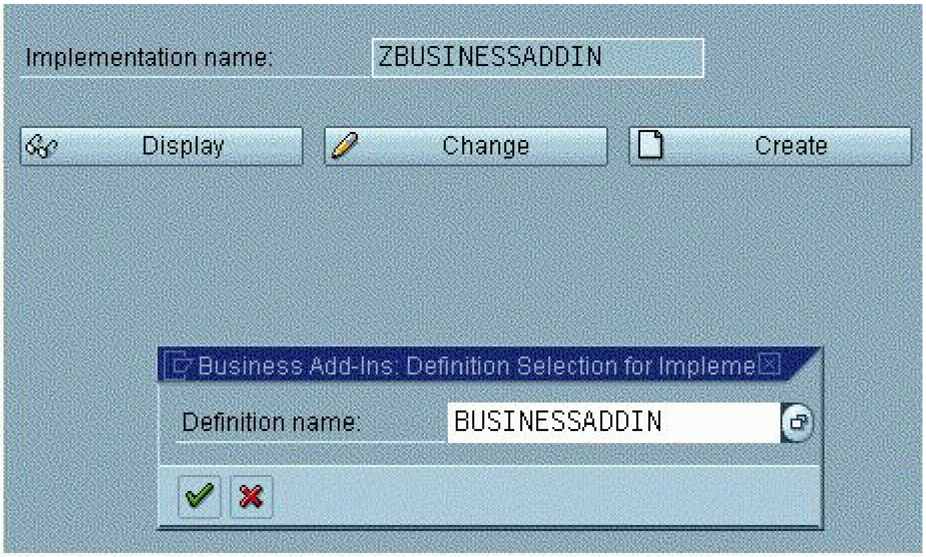
- Business Add-In has an interface and methods for that interface.
- Choose *Tools ->ABAP Workbench -> Utilities -> Business Add-Ins* (transaction SE18) from the SAP menu.

- Enter the name of the Business Add-In.



Definition name: BUSINESSADDIN

- Business Add-Ins have two parts: One is the **Definition** and the other is its **Implementation**.
  - In the implementation view, the users of Business Add-Ins can customize the logic they need or use a standard logic if one is available.
  - To create an implementation, choose *ABAP Workbench ->Utilities -> Business Add-Ins* (transaction SE19) from the SAP menu or double-click on the corresponding activity in the Implementation Guide.
  - Enter a name for the implementation and choose *Create.*
  - Enter the name of the add-in you want to create an implementation for in the dialog box that appears.



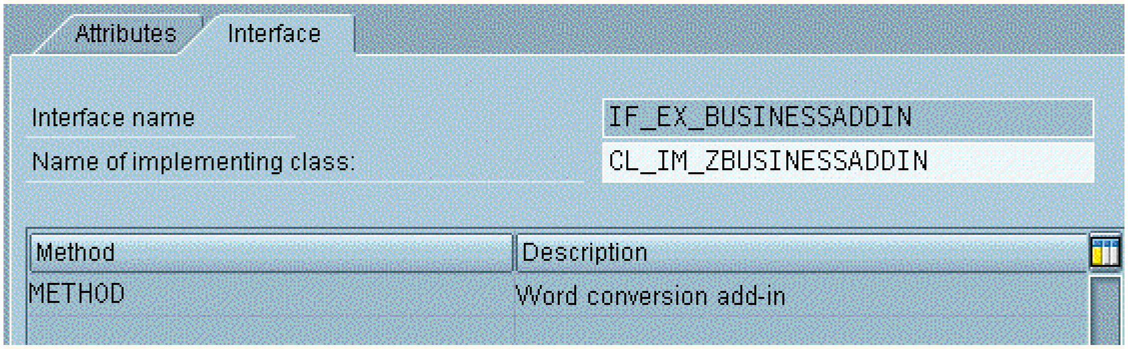
Implementation name: ZBUSINESSADDIN

Business Add-Ins: Definition Selection for Impleme

Definition name: BUSINESSADDIN

- Enter a short text describing your implementation on the following screen.

- Choose the *Interface* tab.



Interface name: IF_EX_BUSINESSADDIN

Name of implementing class: CL_IM_ZBUSINESSADDIN

Method	Description
METHOD	Word conversion add-in

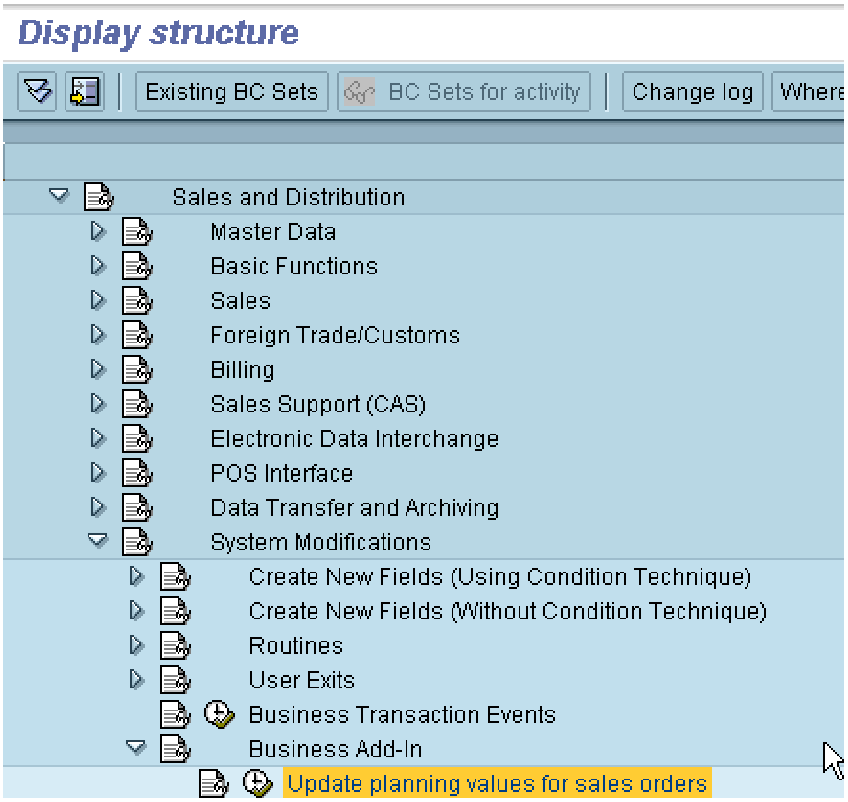
- In order to implement a method for your add-in, double-click on the method to branch to the Class Builder.

- Insert the desired source code for the implementation between the **method if_ex_businessaddin~method.** and **endmethod.**
- Save your entries and return to the *Change Implementation* screen.

- Choose *Activate.* You may now use this implementation when the application program is executed.

- The Business Add-In enhancement technique differentiates between enhancements that can only be implemented once and enhancements that can be used actively by any number of customers at the same time.

- Customers can find the enhancements present in their system in the IMG and in the component hierarchy. Whenever they want to use a Business Add In, they must create their own implementation of the add-in. The enhancements' names and corresponding documentation should help in deciding which add-in they want to create an implementation for. As discussed above, during implementation creation, a class for implementing the enhancement's interface is also created. Customers must first implement methods and user interface enhancements, and then activate their implementations of the enhancement. The enhancement's active components are then called at runtime.



Display structure

- Sales and Distribution
  - Master Data
  - Basic Functions
  - Sales
  - Foreign Trade/Customs
  - Billing
  - Sales Support (CAS)
  - Electronic Data Interchange
  - POS Interface
  - Data Transfer and Archiving
  - System Modifications
    - Create New Fields (Using Condition Technique)
    - Create New Fields (Without Condition Technique)
    - Routines
    - User Exits
    - Business Transaction Events
    - Business Add-In
      - Update planning values for sales orders

Filter-Dependent Business Add-Ins

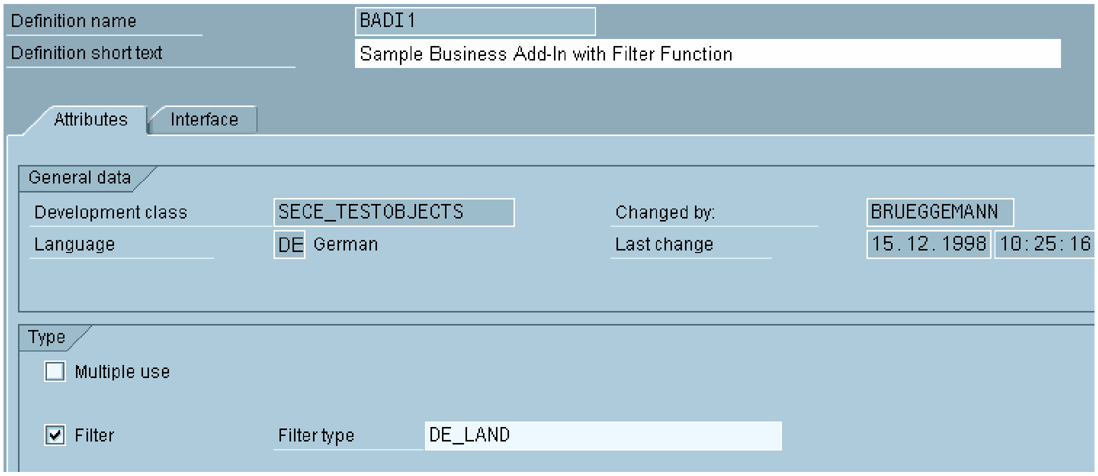
Business Add-Ins may be implemented depending on a specific filter value. If the standard allows for an enhancement for, for example, country-specific versions, it is likely that various partners will want to implement this enhancement. Distinct implementations can then be created and activated according to country.

Enter a filter type when defining your enhancement (a country or industry sector, for example). All methods created in the enhancement's interface have filter value 'FLT_VAL' as their import parameter. The application program provides the filter value to the enhancement method. The method then selects the active implementation for that value.

A description follows of how a filter-dependent Business Add-In works in the context of the string conversion example. In the following example, different implementations will be called using different filter values.

Defining a Filter-Dependent Business Add-In

To define a filter-dependent Business Add-In, first select a filter dependant Business Add-In and select the *Filter* checkbox.



Definition name	BADI1		
Definition short text	Sample Business Add-In with Filter Function		

General data

Development class	SECE_TESTOBJECTS	Changed by:	BRUEGGEMANN
Language	DE German	Last change	15.12.1998 10:25:16

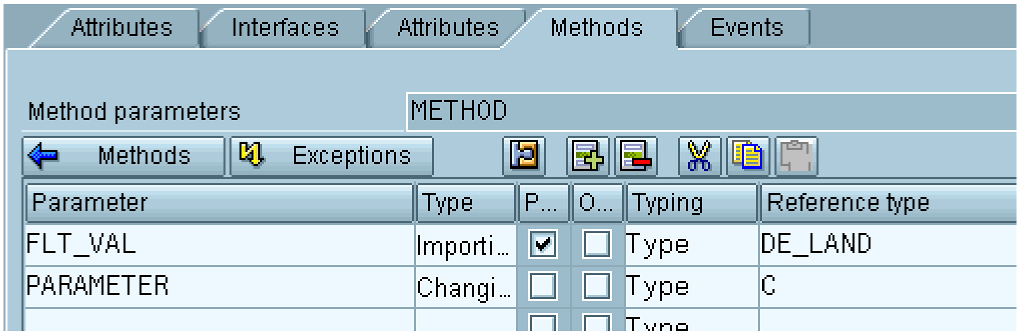
Type

☐ Multiple use

☑ Filter Filter type DE_LAND

Enter the data element you want as a filter type or select a filter type using the possible entries help.

The filter value is declared using parameter *flt_val* and is preset in the list of parameters.



Method parameters METHOD

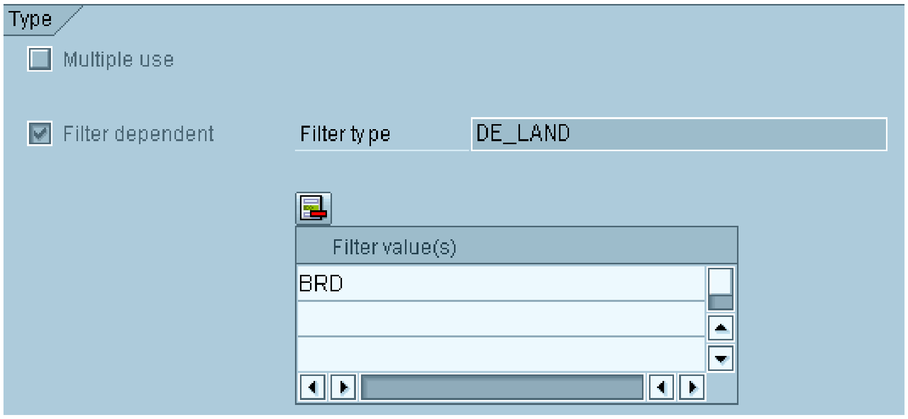
Parameter	Type	P...	O...	Typing	Reference type
FLT_VAL	Importi...	☑	☐	Type	DE_LAND
PARAMETER	Changi...	☐	☐	Type	C
		☐	☐	Type	

Calling a Filter-Dependent Business Add-In from an Application Program

If you want to use a filter-dependent Business Add-In, you will need an implementation for each relevant filter value. Multiple filter values may use the same implementation, however.

When implementing a filter-dependent Business Add-In, proceed as follows:

1. Create an implementation by referring to the corresponding Business Add-In definition.
2. Enter a characteristic filter value for the implementation, or choose F4 and select a value from the list of possible entries displayed. In principle, it is possible to define multiple characteristic filter values for each implementation.



Type

□ Multiple use

☑ Filter dependent

Filter type: DE_LAND

Filter value(s)
BRD

1. Use the *Class Editor* to fill the interface method.
2. Repeat steps 1-3 for each implementation that you create.
3. Activate your implementations.

Now, whenever the standard SAP program is executed, different filter value-specific implementations are executed.

Author Bio

Aveek Ghose is a consultant with IBM, India

M.S. Information Systems 1993, George Mason University, Fairfax, VA, USA.

M.S. in Economics, 1991, Virginia Tech, Blacksburg, VA, USA.