Capgemini

ABAP Part III

Lesson 06: Business Add-Ins

Lesson Objectives

After completing this lesson, participants will be able to -

- Explain why SAP introduced classic BAdI's as an enhancement in Release 4.6.
- Search for available Business Add-Ins in SAP programs
- Use Business Add-Ins to implement program enhancements
- Explain what an extensible filter type is
- Explain the default and sample code of a BAdI

Introduction

BAdI – Business Add-ins

New SAP enhancements using SAP Objects.

Users of BAdI can customize the logic according to the specific requirements (User-defined) or use the standard logic available.

Each Business Add-In has

- At least one BAdI definition
- A BAdI interface
- A BAdI class that implements the interface

For User-defined BAdI,

- developer creates an interface for the add-in.
- Enhancement management creates an adapter class that implements the interface
- Developer creates an instance of the class in the application program and calls the corresponding methods.

Introduction (Contd.).

For standard BAdI, interface and class will be predefined by SAP.

Adapter class performs these tasks

- Control (the class calls all active implementations)
- Filtering (If the Add-in has to be executed under certain conditions, the class ensures that only certain implementations are executed)

In BAdI, all the enhancement components are grouped together.

- Program Enhancements (interface methods)
- Menu Enhancements (function codes in interface definition)
- Screen Enhancements

Advantages of BAdI’s

Business Add-Ins no longer assume a two-level infrastructure (SAP and customer solutions), but instead allow for a multi-level system landscape (SAP, country-specific versions, industry solutions, partner, customer, and so on). You can create definitions and implementations of Business Add-Ins at any level of the system landscape.

SAP guarantees the upward compatibility of all Business Add-In interfaces. Release upgrades do not affect enhancement calls from within the standard software nor do they affect the validity of call interfaces. You do not have to register Business Add-Ins in SSCR.

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

Business Add-Ins can be defined according to filter values. This allows you to differentiate between Add-In implementations using the filter criteria as per customer’s requirements.

Demo: Find Standard BAdI using CL_EXITHANDLER=>GET_INSTANCE

Demo: Find Standard BAdI using SE24

Demo: Find Standard BAdI using SQL Trace

Demo: Find Standard BAdI using Repository Information System

BAdI Definition: Interface Methods

1.The system proposes a name for the interface and the generated class.

2.The name of the generated class is composed as follows:

- Namespace prefix
- CL_ (to signify a class in general)
- EX_ (stands for "exit")

Name of Business Add-In

3.A BAdI interface can have several interface methods.

BAdI Definition: Interface Methods Parameters

You can use all of the normal functions of the Class Builder. For example, you can:

- Define interface methods
- Define interface parameters for the methods
- Declare the attributes of the interface

If the business add-in is filter-dependent, you must define an import parameter flt_val for each method. Otherwise, you define the interface parameters you need for the enhancement.

BAdI Definition: Activating the Interface

Once you have finished working on your interface, you must activate it. This generates the adapter class for the Business Add-In.

If you change the interface, the adapter class is automatically regenerated.

Demo: Create a customized BAdI

Enhancement Spot

- BADI definitions were directly defined in SE18, but from ECC 6.0, SAP has introduced the concept of Enhancement Spots
- An enhancement spot is an object, which can contain one or more BAdI definitions

Demo: Create an Enhancement spot

Menu Enhancements

Menu Enhancements are renamed as Function Code Enhancements

The Function Code should start with a ‘+’

Function Code Enhancements can only be used for single-use add-ins

Should not be filter-dependent

Have to be created in conjunction with Program Enhancements (Interfaces)

Function Code Enhancements

Right-click on the BADI definition to Create a Menu Enhancement

Select ‘Add Menu Enhancement’

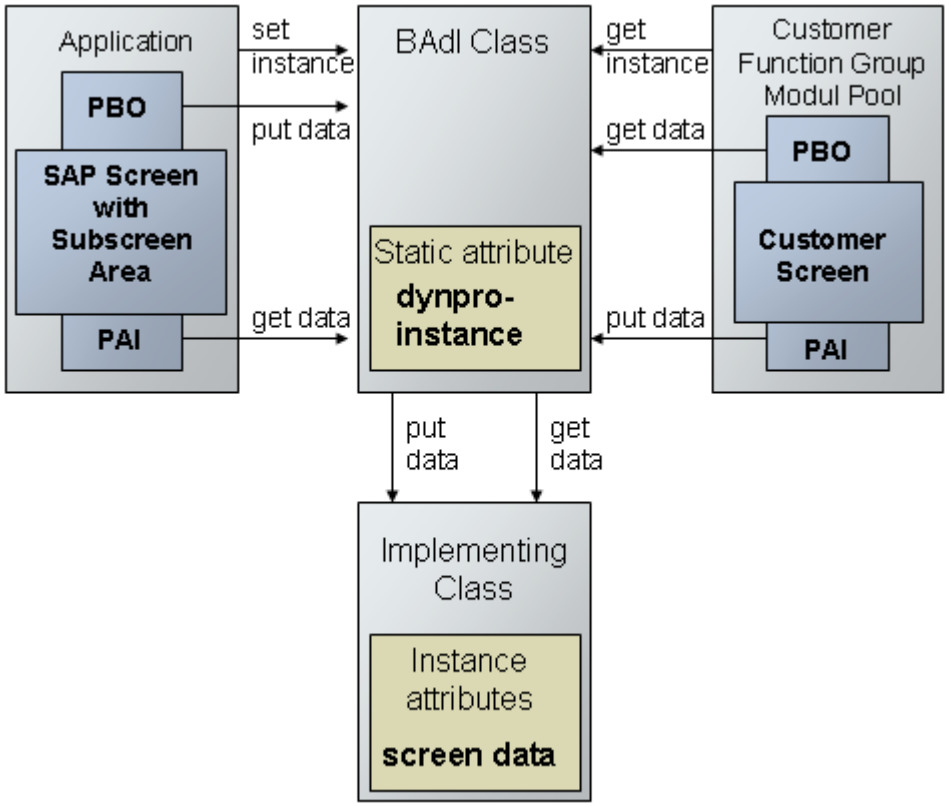
Provide the Program Name, Function Code and the description

Create the Implementation for the Function Code Enhancement in SE19

Provide the Function text, Icon text, etc.

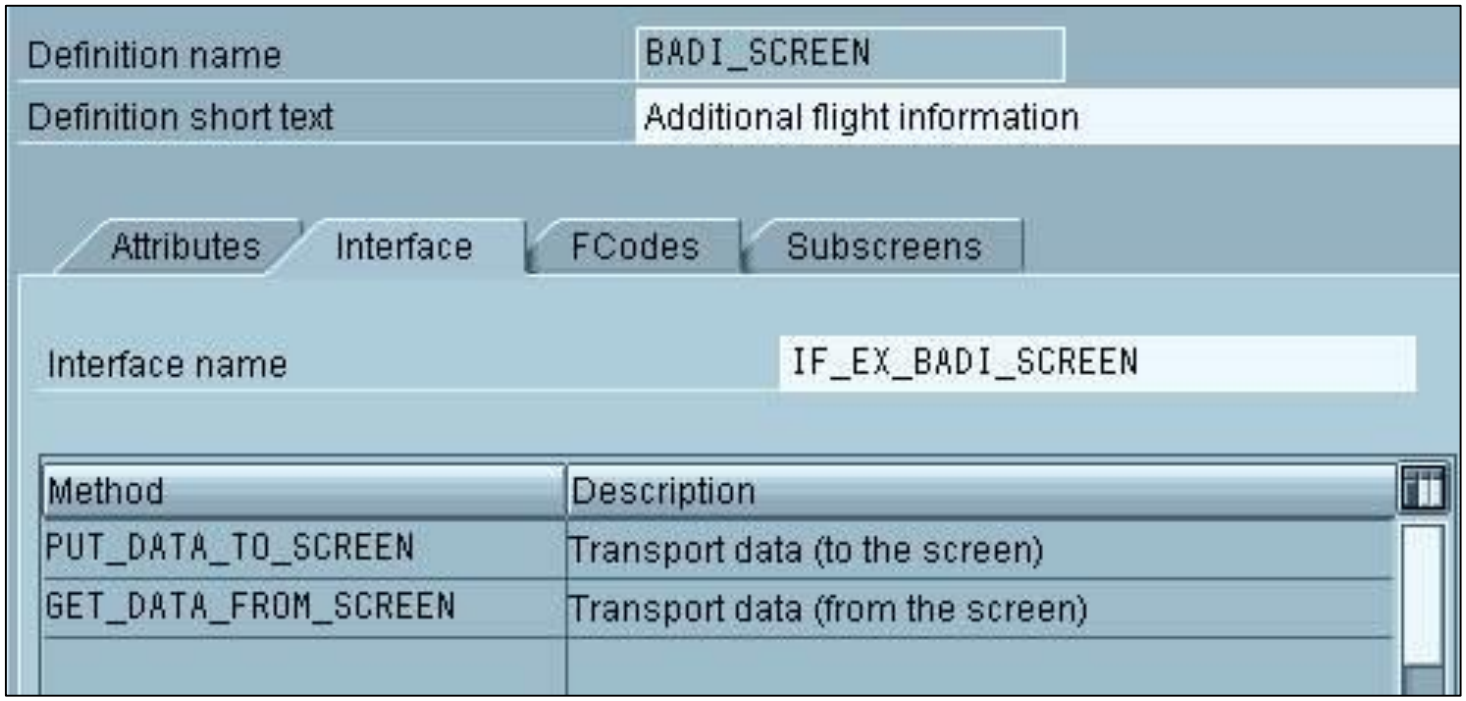
Develop the processing logic in the interface exit of the BADI

Screen Enhancement



Defining Screen Enhancements

Create the following methods



Multiple Use BAdI’s

Multiple implementations are possible for the same BAdI

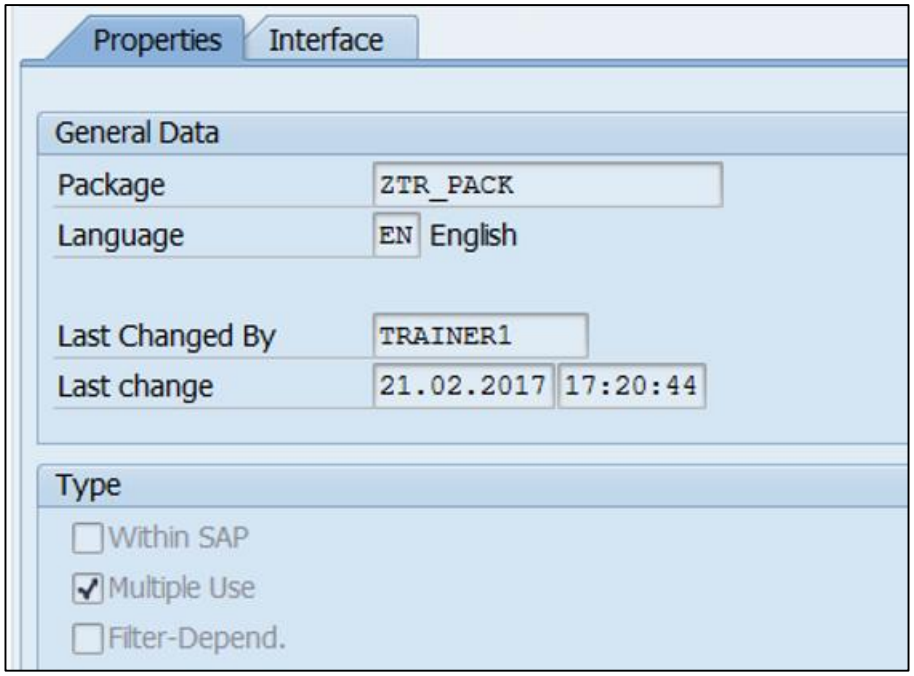
There is no sequence control for multiple implementation since at the time of definition, it doesn’t know which implementation will be active.

All active implementations will be triggered by the application program.

To display the list of all implementations of a BAdI definition, go to Implementation -> Display in SE18.

When defining a Multiple use BAdI, the interface methods of the BAdI should not contain export or returning parameters

Multiple Use BAdI’s



Demo: Mutiple Use BAdI

Filters

- BAdI’s are implemented based on some filter values
- Filter type must be specified while defining the enhancement.
- It can be a single filter value or a set of values.
- All methods in the interface will have the filter value FLT_VAL as their importing parameter.
- The method then selects the active implementation based on the data provided in the filter value.

Filters

- A filter type can be a data element or a structure. A data element must fulfill the following criteria:
- The data element's domain may contain a maximum of 30 characters and must be of type Character.
- The data element must
- Either have a search help with a search help parameter of the same type as the data element and this parameter must serve as both the Import and export parameter
- OR
- Element's domain must have fixed domain values or a value table containing a column with the same type as the data element.

Filters

- Select the option Create Filter

Filters

- Specify the required parameters – BAdI Filter Field, Filter Type and Description
- When creating the BAdI implementation, the filter values for which the implementation should be processed should be defined

Summary

In this lesson, you have learnt:

- Why SAP introduced classic Badi's as an enhancement in Release 4.6.
- How to Search for available Business Add-Ins in SAP programs
- How to Use Business Add-Ins to implement program enhancements
- What an extensible filter type is
- The default and sample code of a BAdI

Review Question

Question 1: ____ Tcode is used to define a BAdI.

Question 2: : ____ Tcode is used to implement a BAdI.