# SAP ENHANCEMENT FRAMEWORK

## **ADAPTING SAP**

The enhancement is a concept of adding your own functionality to SAP's standard business applications without having to modify the original applications. To modify the standard SAP behavior as per customer requirements.

### Ways of adaptations:

- Customizing Configurations done in SAP (T Code SPRO)
- Enhancement Additional Customer Requirements typical for a client and which is not available in Standard SAP
- Modifications Changing the Standard SAP Code to enhance the code to add new functionalities.

# TYPES OF ENHANCEMENTS IN SAP

### User exit

### Customer exit

- Menu Exits
- Screen Exits
- Function Module Exits
- Field Exits

### BADI (Business Add-Ins)

- Classic BADIs
- New BADIs.

### **Enhancement Frame-Work**

- Enhancement Points
- Enhancement Sections
- Implicit & Explicit Enhancements

## **USER EXIT**

**User Exits** are "form-based" custom enhancements. These are FORMS and are called by SAP standard programs using PERFORM.

- I. Implemented as Subroutines.
- 2. Created in SAP namespace.
- 3. You require access key to make changes

These subroutines will begin with USEREXIT\_.

## **CUSTOMER EXIT**

SAP creates customer exits for specific programs, screens, and menus within standard applications. These exits do not contain any functionality. Instead, the customer exits act as hooks. You can hang your own add-on functionality onto these hooks.

- 1.Implemented as FM with an Include.
- 2. Created in customer namespace.
- 3. Do not require access key to make changes.

### **Types of Customer Exits**

- I. Function Module exits
- 2. Screen exits
- 3. Menu exits

## **CUSTOMER EXIT**

#### I. Function Module exits

Function module exits are exits developed by SAP. The exit is implemented as a call to a function module. The code for the function module is written by the developer. You are not writing the code directly in the function module, but in the include that is implemented in the function module.

Format: CALL CUSTOMER-FUNCTION '910'

The naming standard of function modules for function module exits is:

EXIT\_<program name>\_<3 digit suffix>

### 2. Screen Exits:

Allow customer to add fields to a screen via a sub screen in an SAP program. The sub screen is called within the standard screen's flow logic.

Format: CALL CUSTOMER-SUBSCREEN CUSTSCRI

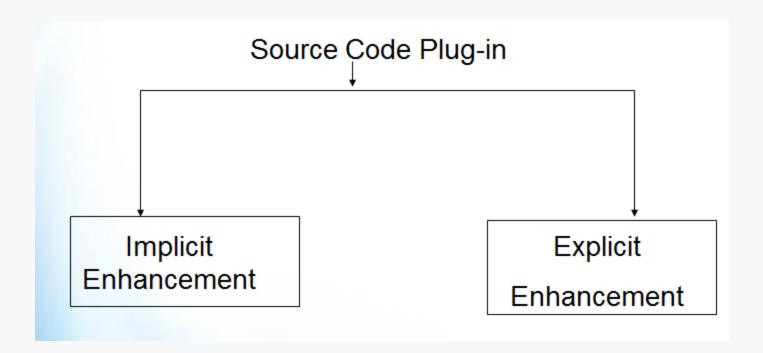
### 3. Menu exits:

Menu exits allow you to add your own functionality to menus. Menu exits are implemented by SAP and are reserved menu entries in the GUI interface. The developer can add his/her own text and logic for the menu.

Function codes for menu exits all start with "+".

Format: +CUS (additional item in GUI status)

## **SOURCE CODE PLUG-IN**



# IMPLICIT ENHANCEMENT VS. EXPLICIT ENHANCEMENT

### **Implicit Enhancement Option**

 At common enhancement places, implicit Enhancement options are available.

### Examples:

- End of Executable Program, Include, Function group, Dialog module
- Begin/End of Form routine / Function module / Method
- End of a structure
- End of Private/Protected/Public Section of a local class

### **Explicit Enhancement Option**

- Predefined enhancement options can be defined in source code.
- They are additionally stored inside Enhancement Spots.



BADI stands for Business Add Ins Just like Customer Exits
BADI help hook custom enhancements to SAP functionality.
BADI's are based on ABAP Objects

### Types of BADI's

Multi Use BADI Filter Dependent BADI

### **Transaction Codes**

**SE18** – Badi Definition. It is like Interface

**SEI9** – To implement BADI (Create implementation for BADI)

SE24 – To find out implementations for BADI

## HOW TO FIND BADI — METHOD 1?

Go to se24 (Display class cl\_exithandler)

Double click on the method GET\_INSTANCE.

Put a break point at Line no.25 (CASE sy-subrc).

**Execute SAP standard transaction** 

Press the required button for which you need to write an exit logic, the execution will stop at the break point.

Check the values of variable 'exit\_name', it will give you the BADI name called at that time.

This way you will find all the BADIs called on click of any button in any transaction

### HOW TO FIND BADI — METHOD 22

Finding BADI Using Repository Information System (TCODE- SE84).

Go to "Maintain Transaction" (TCODE- SE93).

Enter the Transaction 'MM02' for which you want to find BADI.

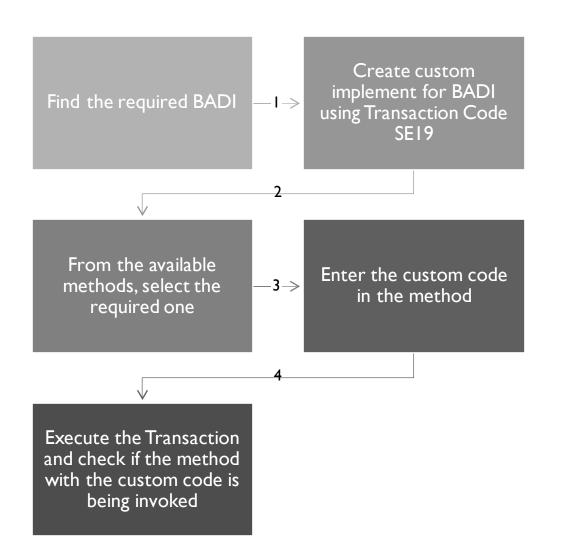
Get the Package Name. (Package ME in this case)

Go to TCode: SE84->Enhancements->Business Add-inns->Definition

Enter the Package Name and Execute.

This way you will find all the BADIs called on for this transaction

## IMPLEMENT STANDARD BADI'S



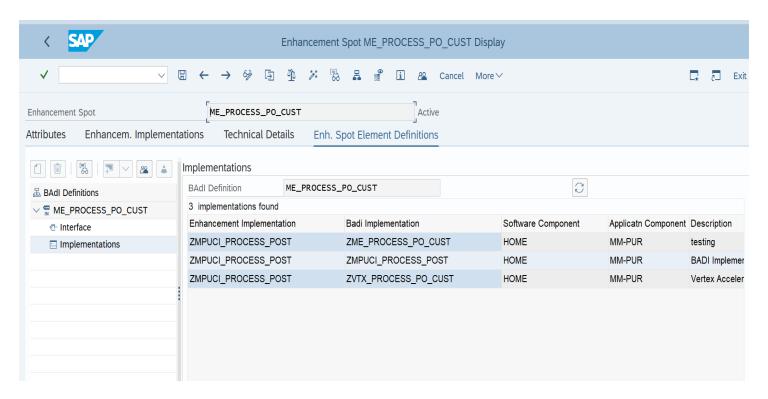
## TYPES OF BADI

### Multiple use BADI

- Multiple implementations are possible for same BADI
- All Active implementations will be triggered by application progra
- No sequence Control

### Filter BAD

- Implemented based on filter values
- Filter typemust be specified while defining the enhancement
- It can be Single Value or set of Values
- All methods implements Filter
   Val FLT\_VAL as their importing parameter



## BADI FOR PURCHASE ORDER PROCESSING

ME\_PROCESS\_PO\_CUST

## Thank You