

SAP ABAP AND WEBDYNPRO TUTORIALS

SAP System , SAP ABAP tutorials , WebDynpro Tutorials and English to Hindi , English Grammar Exercises ,Manual Testing , Java, Selenium.

How to Implement SAP ABAP Implicit and Explicit Enhancement Options



- March 14, 2017



How to Implement SAP ABAP Implicit and Explicit Enhancement Options-ABAP

Implicit and [Explicit Enhancement](#) options are most advanced than [BADI's](#) and User and Customer exits. These techniques come into the picture to bring all enhancement techniques under one Enhancement Framework. These options are available since the SAP ECC 6.0 Version. These techniques are two types which are implicit and explicit options, which are used to add custom source code for standard SAP applications. The following are different enhancement technologies available under this framework.

Source Code Enhancement

Function Group Enhancement

Class Enhancement

Kernel-BADI Enhancement

Here, an important one is Source Code Enhancement which is used to add external hooks(that are additional source code plug-ins)for SAP standard applications. There are two types of Source Code enhancement techniques.

Implicit enhancement option

Explicit enhancement option

What are implicit and Explicit Enhancement Options?

Implicit Enhancement Options

Implicit enhancements are predefined enhancements provided by SAP in the source code, it is predefined and it is available at the following places:

Begin/End of an include.

Begin/End of Method/Function Module/Form Routine

End of a structure

End of Private/Protected/Public Section of a local class

Explicit Enhancement in SAP ABAP

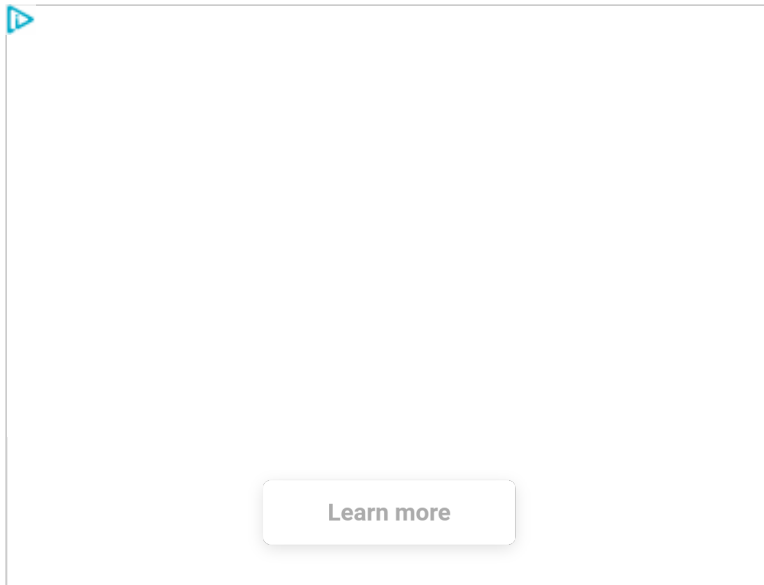
We can create custom explicit enhancement options in the ABAP program source code. Explicit enhancements can be implemented by customers or partners.

There are two types of Explicit Enhancement options available.

1. Enhancement point (Syntax - ENHANCEMENT-POINT)

We are assigning source code plug-ins, defined between ENHANCEMENT and ENDENHANCEMENT in an ABAP program, ENHANCEMENT-POINT. These source code plug-ins are inserted into the code unchanged if the switch is in the relevant position.

2. Enhancement section (Syntax - ENHANCEMENT-SECTION)



The enhancement section is used to replace a set of code or statements with the customer (custom code). In this technique the original source code does not get executed but, the customer implementation (custom code) gets executed.

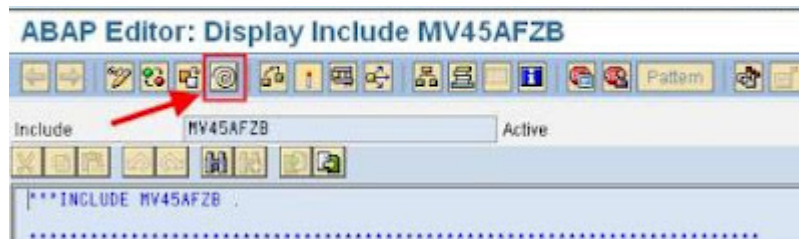
Note - Explicit enhancements though can be placed anywhere in the source code but, not just anywhere except some areas where SAP would allow.

How to implement implicit enhancement options

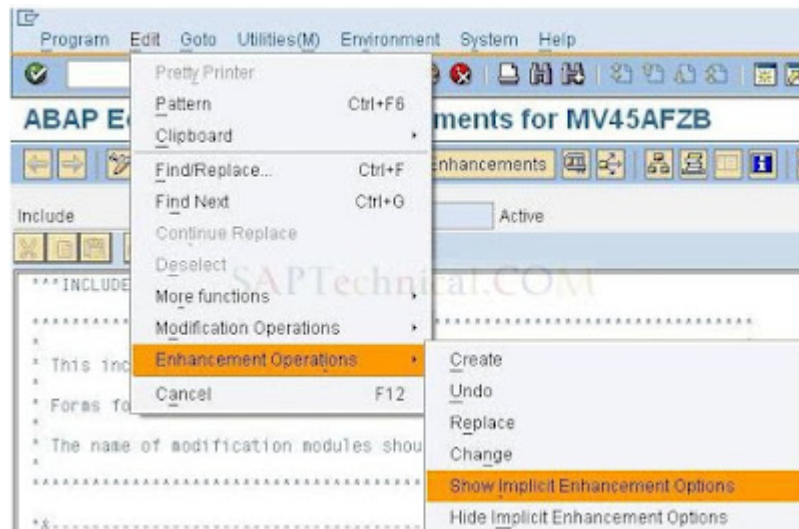
For implementing implicit enhancement options, First of all, we have to find the required spot then we have to add our custom source code in between **enhancement** ...and **endenhancement** in the ABAP program.

Step1: Go to SE38 and Open include of any program(SAP standard application), In order to implement any of these Source code enhancements, you need to be in '**change Enhancement mode**' (**spiral icon**).

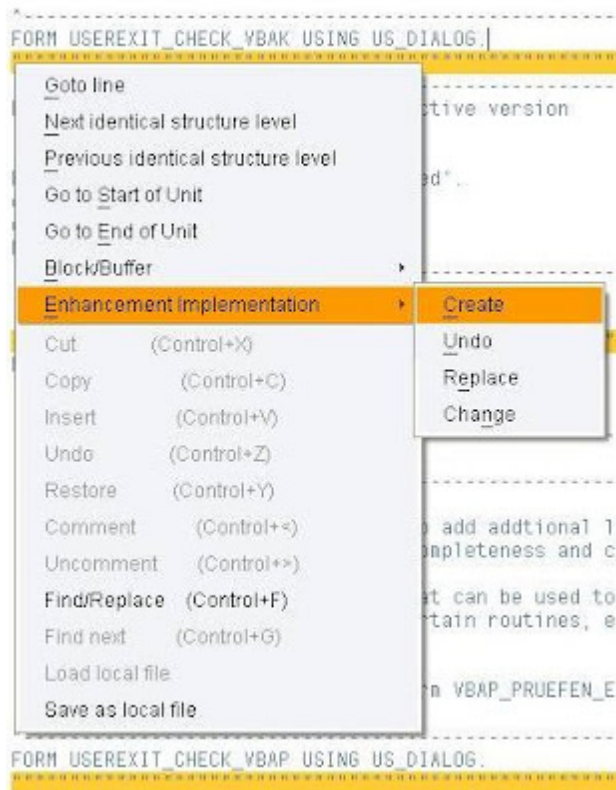




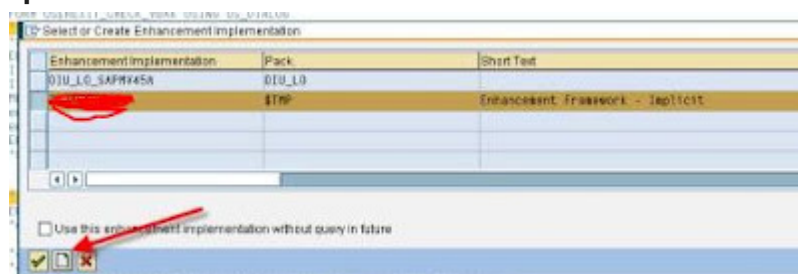
Step2: Go to the **EDIT** option in the menu and choose Enhancement Operations -> Show implicit Enhancement Options



Step4: Place the cursor on the **yellow line** and choose to create.



Step5: Give the **Implementation name** and then create.



Step6: Here Editor will be enabled to write the code. You have to write your own code between enhancement and endenhancement block in the ABAP program, after writing source code, then save and activate it.

Enhancement.

Endenhancement.

How to implement explicit enhancement options

For implementing explicit enhancement options, First of all, we have to create our own custom explicit options, then create the custom program from which we can access our created custom explicit enhancement options. Next, we have to add our source code in between:

Enhancement-spot 'spot name'.

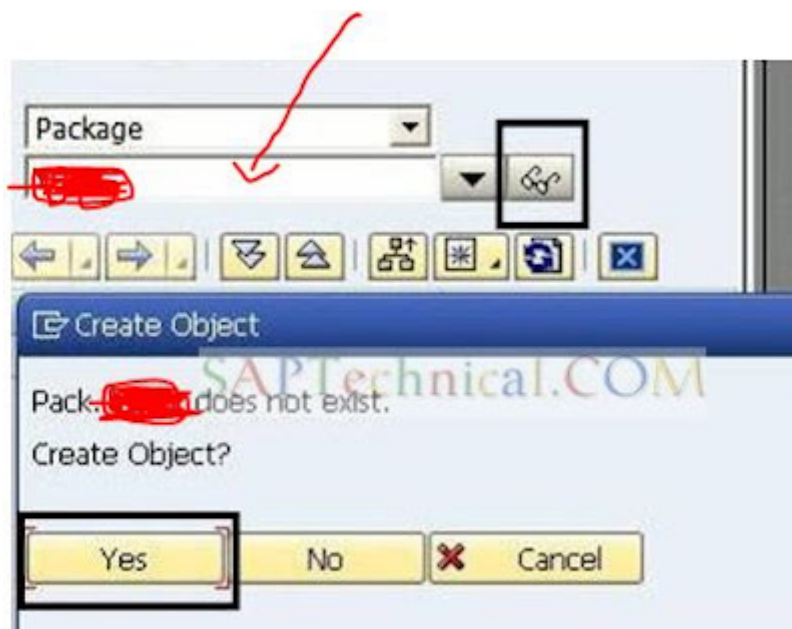
Endenhancement.

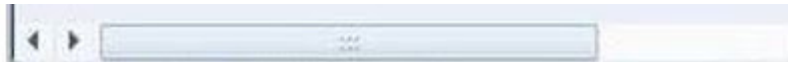
or

Enhancement-section "name"spots "name".

End-Enhancement-section.

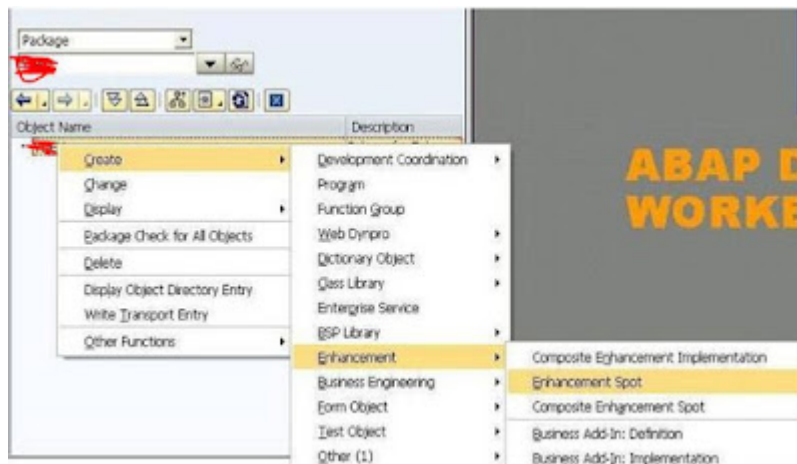
STEP 1: Create a package in transaction SE80 (Object navigator) Name package



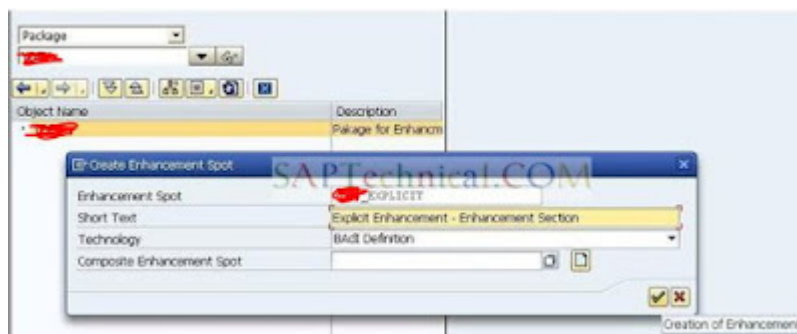


STEP 2: Navigate to 'Enhancements' folder of your package.

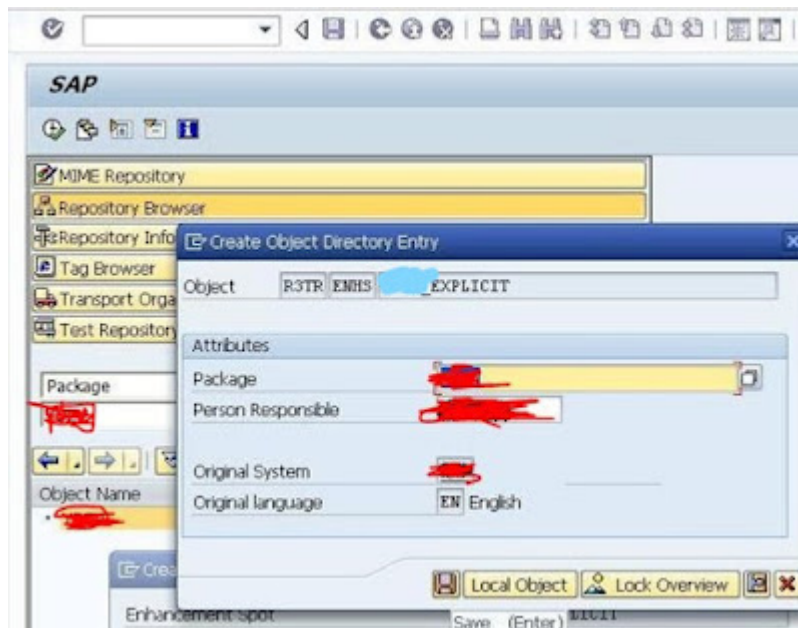
Right-click the 'Enhancements' and click on Create à Enhancement Spot'. Fill in the details in the 'Create Enhancement Spot' dialog. And save it into the created package. check the enhancement spot created under the 'Enhancement Spots' folder.



STEP 3: 'Right Click' the spot created and create an Implementation. Fill in all the details in the 'Create Enhancement Implementation' dialog.



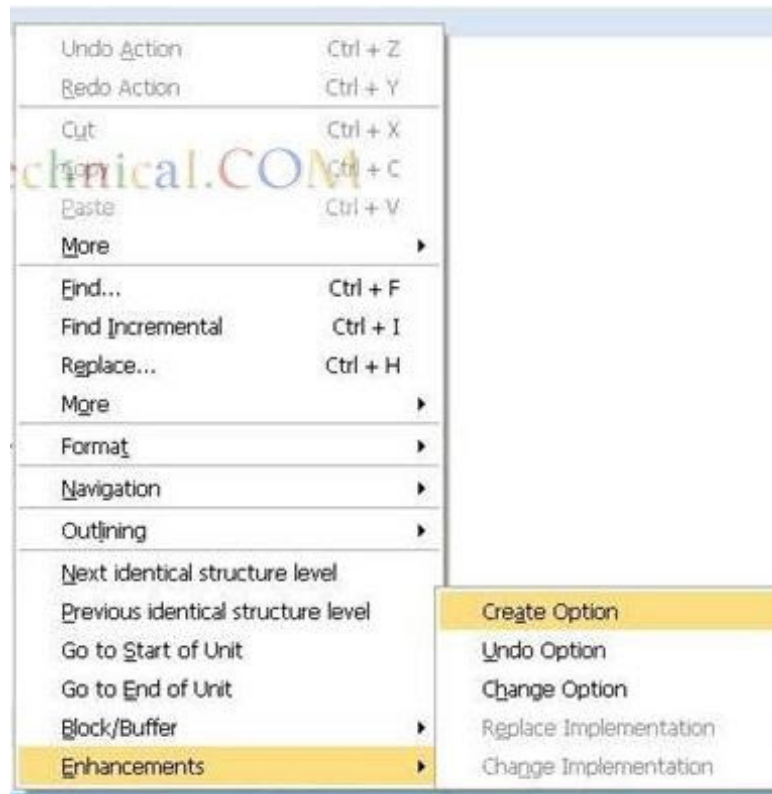
STEP 4: Now, we need to 'Activate' the enhancement spot.



STEP 5: Suppose, you want to apply enhancements to a CUSTOM program, not a standard program, then follow the below steps:

Create a program Zprogram in which we have to implement an enhancement section option for that after creating the program. suppose, you want to add our external code at the place of the start-of-selection event within the ABAP program for that

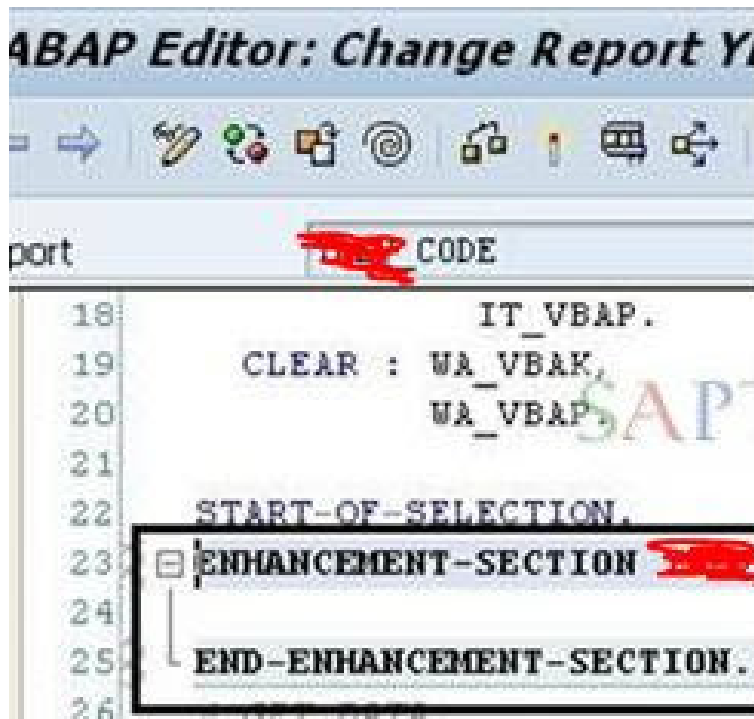
Place the cursor beginning of the start-of-selection event. then right-click and select the enhancements option, next click on the creates option.



Now give enhancement section name and spot name .then click on the implement(create) button. The system will create the

'ENHANCEMENT-SECTION...'

'END-ENHANCEMENT-SECTION'. within this block write the required source code.



Finally, SAVE and ACTIVATE the program. At last, we have learned what are implicit and explicit enhancement options and their definitions and implementation methods.

Conclusion

Please comment below if you know more about SAP ABAP Implicit and Explicit Enhancement Options, or share this post to help others.



BADI

Enhancements

SAP ABAP



Enter your comment...



Popular posts from this blog

BADI Interview Questions in SAP ABAP

- April 13, 2017

BADI Interview Questions in SAP ABAP A BAdI is an object-oriented enhancement option,

which makes it the most sophisticated enhancement type. The main characteristic of a BAdI is that it provides a mechanism to change the functionality of a well-defined business object. ...

[READ MORE](#)

Sample SAP ABAP Programming Examples for Practice

- July 16, 2017

Sample SAP ABAP Programming Examples for Practice Are you a beginner at ABAP Language, you're looking for example programs for practice. Here I collected some sample programs from various objects in ABAP useful for beginners to improve Technical skills. To ...

[READ MORE](#)

Module Pool Programming Interview Questions and Answers in SAP ABAP

- May 12, 2017

Module Pool Programming(Dialog Programming) Interview Questions and Answers in SAP ABAP As an ABAP Consultant, you should be strong in Module pool programming techniques as well as when you face the real-time or fresher interview, Questions, and answers ...

[READ MORE](#)

SAP ABAP Interview Questions and Answers for 10 Years Experienced

- January 07, 2019

SAP ABAP Interview Questions and Answers for 10 Years Experienced This post will be helpful for SAP ABAP experienced candidates who may have 3,4 5, 6,7, and 10 years of experience, all of them can follow this interview pattern for their successful interview. SAP Interview ...

[READ MORE](#)

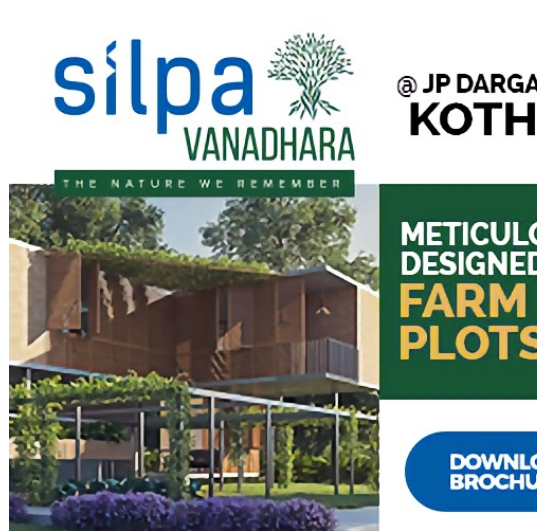
Step by Step tutorial on BDC Session Method Program in SAP ABAP

- June 20, 2017

Step by Step tutorial on BDC Session Method Program in SAP ABAP BDC Session Method is used to upload data from Non-SAP to SAP System. Using

Session method is used to upload data from NON-SAR to SAR system. Using Session method, we can transfer data through more than one ...

[READ MORE](#)



 Powered by Blogger

Theme images by [Michael Elkan](#)



SRINIVAS B

[VISIT PROFILE](#)

Archive



Labels



Report Abuse