

Introducing: “Enhancement Framework”

– The cool new way to enhance your ABAP systems



Applies To:

ABAP, SAP NW App Server, SAP NetWeaver, BADI, Enhancement Framework.

Summary

Learn the future of how ABAP systems can be enhanced using the new ‘Enhancement Framework’ available in SAP NetWeaver 2004s (and thus, mySAP ERP 2005 and mySAP Business Suite 2005 applications).

By : Venky Varadadesigan
Company : SAP Labs, LLC.
Date : 21 Dec 2005
Email : venkata.varadadesigan@sap.com

Table of Contents

Applies To:	1
Summary	1
Table of Contents	1
Overview	2
The current state of enhancement	3
What is Enhancement Framework?	4
Enhancement Framework Technologies	4
Source Code enhancement	5
Implicit enhancement option	6
Explicit enhancement option	6
Function Group enhancement	7
Class enhancement	8
Kernel-BADI enhancement	9
Benefits / Value Proposition	11
One final word	11
Author Bio	11
Disclaimer & Liability Notice	12

Introducing: “Enhancement Framework”

– The cool new way to enhance your ABAP systems



Overview

SAP is known for delivering business software easily adaptable by customers for their specific needs. Typically the software (for example, mySAP ERP) can be adapted by one of the following techniques: Customizing (defining system behavior through standard SAP provided mechanism without coding), enhancement (adding custom code at strategic hook positions provided by SAP) or modification (modifying SAP supplied code directly – often called Core-Mod).

The way the standard SAP system is enhanced just got a lot better with the introduction of ‘Enhancement Framework’ – a new technology available in the ABAP server of SAP NetWeaver 2004s. The future ABAP enhancement functionalities starting with this release (and thus mySAP ERP 2005 and mySAP Business Suite 2005 applications, which are based on this minor release of SAP NetWeaver), will be using this new framework. (Note: The ABAP edition Sneak Preview of SAP NetWeaver 2004s is available in the ‘Downloads’ area of SDN. You can install this to play around with the new Enhancement Framework tools).

In this article we will see how the new Enhancement Framework brings together all enhancement techniques under one roof and improves the way SAP software is enhanced with multi-layer support, extensible through object-oriented inheritance concept, provides more flexible ways to adapt the system that didn’t exist previously.

The new Enhancement Framework is part of IT Practice: ‘Custom Development’ / IT Scenario: ‘Developing, Configuring, and Adapting applications’ / Scenario Variant: ‘Creating Business Applications using ABAP’.

IT Practices	IT Scenarios				
User Productivity Enablement	Running an Enterprise Portal	Enabling User Collaboration	Business Task Management	Mobilizing Business Processes	Enterprise Knowledge Management
Data Unification	Master-Data Harmonization	Master-Data Consolidation	Central Master-Data Management	Enterprise Data Warehousing	
Business Information Management	Enterprise Reporting, Query, and Analysis		Business Planning and Analytical Services	Enterprise Data Warehousing	
Business Event Management	Business Event Resolution			Business Task Management	
End-to-End Process Integration	Enabling Application-to-Application Processes	Enabling Business-to-Business Processes	Business Process Management	Enabling Platform Interoperability	Business Task Management
Custom Development	Developing, Configuring, and Adapting Applications			Enabling Platform Interoperability	
Unified Life-Cycle Management	Software Life-Cycle Management			SAP NetWeaver Operations	
Application Governance & Security	Authentication and Single Sign-On			Integrated User and Access Management	
Consolidation	Enabling Platform Interoperability	SAP NetWeaver Operations	Master-Data Consolidation	Enterprise Knowledge Management	
Enterprise Service Architecture – Design & Deployment	Enabling Enterprise Services				

Positioning: IT Practice / Scenario for Enhancement Framework

Introducing: “Enhancement Framework”

– The cool new way to enhance your ABAP systems



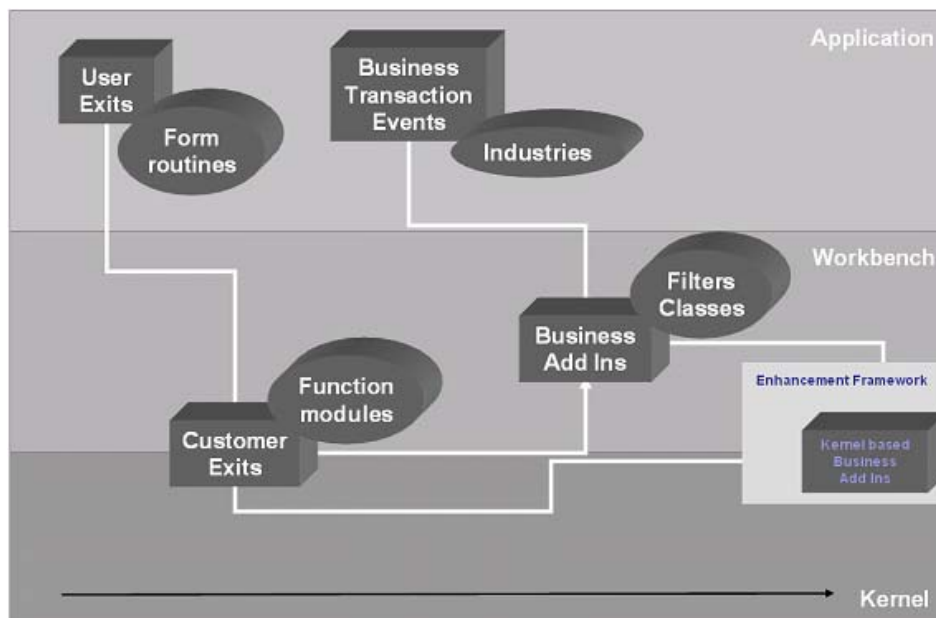
The current state of enhancement

Let us first take a look at how the enhancement technique has evolved so far in SAP.

‘*User-Exit*’ is one of the very first mechanisms provided by SAP to execute custom code in between the standard SAP control flow. This is implemented as subroutine call (PERFORM xxx). A classical example for User-Exit is MV45AFZZ include in order processing module of SAP R/3. Though this include object doesn’t fall under customer namespace, the object doesn’t get overwritten during upgrade.

‘*Customer-Exit*’ is better than the user-exit, in the sense that it is implemented using Function Modules and so has a well defined parameter interface. Also since the custom coding done as part of these customer-exits is located away from the original SAP code, the maintenance is easier than user-exits.

The ‘*BADI-s*’ (Business Add-Ins), as they exist in pre NW04s releases are now called old classic-BADI’s. This was the first object-oriented way to enhance the ABAP system. This, to a certain extent, allows multiple implementations with limited filter support. The classic-BADI’s are implemented using ABAP Objects.



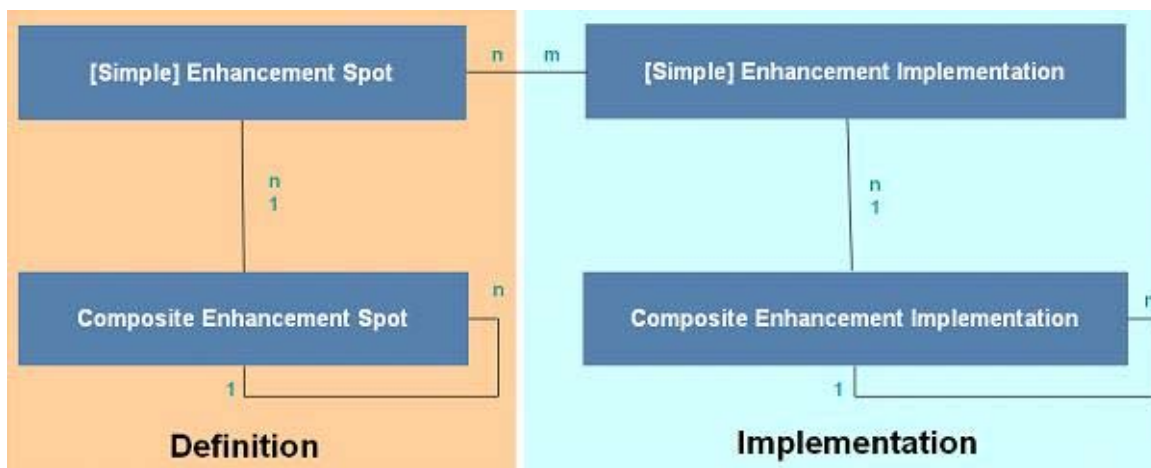
The roadmap of enhancement techniques

What is Enhancement Framework?

Enhancement Framework is the new paradigm to bring all enhancement techniques under one roof. It can also be switched using Switch Framework. The following are different enhancement technologies available under this framework.

- Source Code Enhancement
- Function Group Enhancement
- Class Enhancement
- Kernel-BADI Enhancement

Semantically related enhancement options are grouped under a ‘Simple Enhancement Spot’. ‘Composite Enhancement Spot’ contains one or more of Simple and other Composite Enhancement Spots. On the other hand (implementation side), various related implementation elements are grouped under a ‘Simple Enhancement Implementation’. A ‘Composite Enhancement Implementation’ can contain one or more of Simple and other Complex Enhancement Implementations. The important point to note here is these four entities form the basis of transportable objects in the Enhancement Framework.



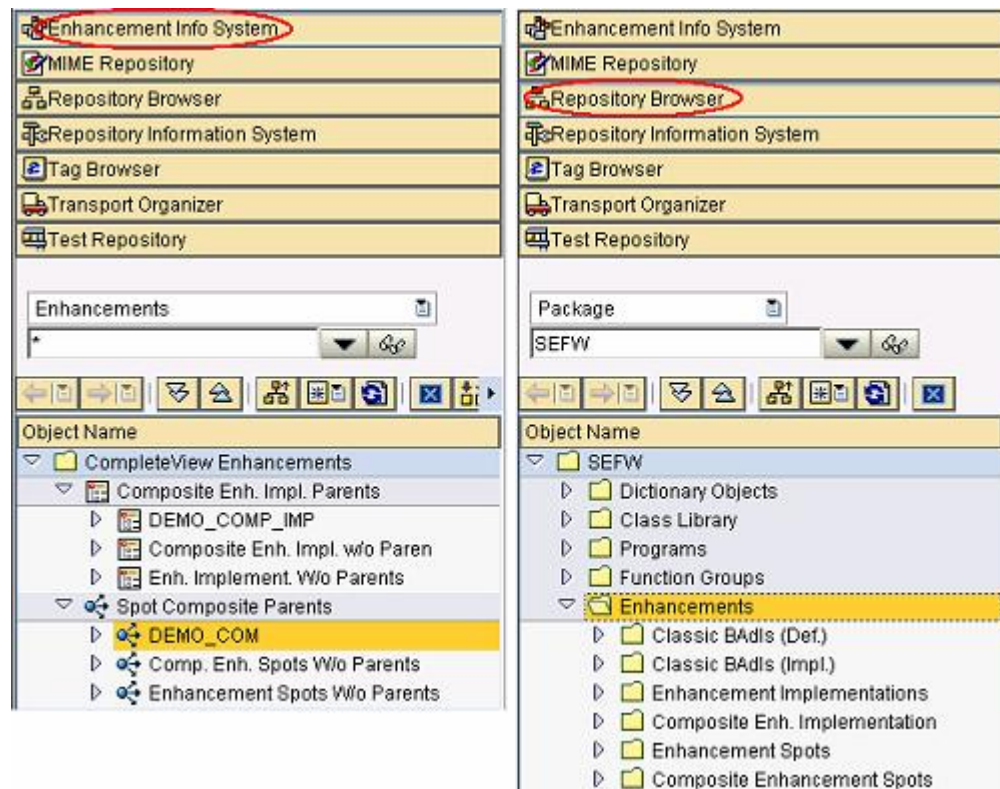
Enhancement concept: Defining the relationship

Enhancement Framework Technologies

Enhancement framework tools are tightly integrated into the ABAP workbench. There is a new browser called Enhancement Information System available in SE80 to centrally manage all the enhancement definitions and implementations available in the system. The enhancement objects can also be individually managed in the respective Package under the Repository Browser of SE80.

Introducing: “Enhancement Framework”

– The cool new way to enhance your ABAP systems



Enhancement browser integrated into SE80

As mentioned earlier, the four technologies available as part of the Enhancement Framework are,

- Source Code enhancement
- Function Group enhancement
- Class enhancement
- Kernel-BADI enhancement

The first three methods, viz., Source Code enhancement, Function Group enhancement, and Class enhancement are brand new ways to enhance the ABAP system. The final one Kernel-BADI is an improvement of the old classic-BADI now integrated into the Enhancement Framework. Let us look at each one of these and understand what they are and how and when to use them. Remember that all of these techniques are considered enhancing and not modifying the system.

Source Code enhancement


Whenever enhancement needs to be incorporated directly into the ABAP source code, this technology shall be provided. Implementing this technology is also called as Source Code Plug-In. There are two types of Source Code enhancements possible.

- *Implicit enhancement option*
- *Explicit enhancement option*

Introducing: “Enhancement Framework”

– The cool new way to enhance your ABAP systems



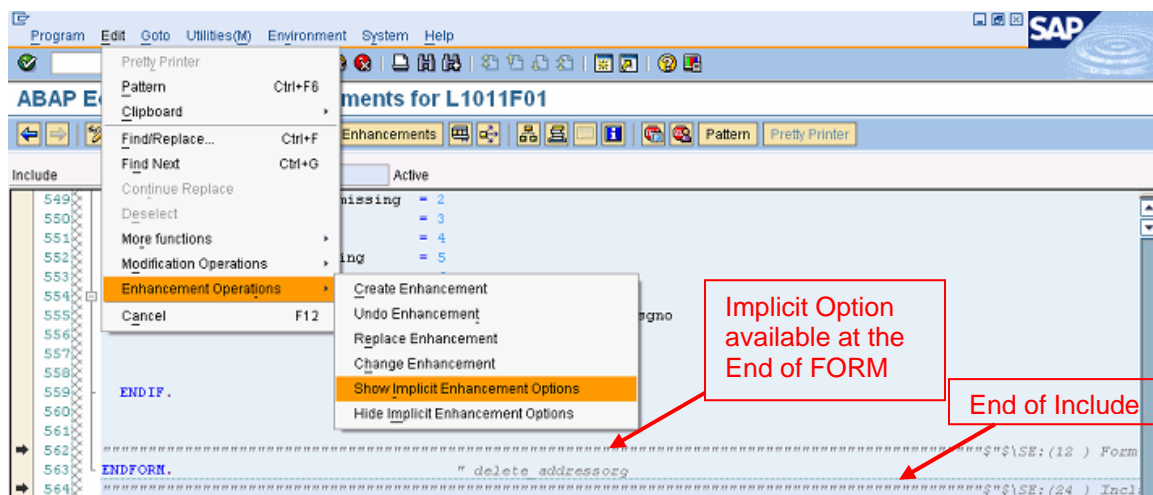
Note that, in order to implement any of these Source code enhancements, you need to be in the ‘Change Enhancement mode’ (the spiral icon  available in the editor). Technically the source code plug-in implementations are stored in a separate include program and not as part of the original source program.

Implicit enhancement option

Throughout the ABAP system, enhancement options are automatically available at certain pre-defined places. Some of the implicit options are:

- At the end of all the programs (Includes, Reports, Function pool, Module pool, etc.), after the last statement
- At the beginning and end of all FORM subroutines
- At the end of all Function Modules
- At the end of all visibility areas (public, protected and private) of local class

To view all the implicit options available in a source code, choose ‘Edit -> Enhancement Operations -> Show Implicit Enhancement Options’ from the editor.



Implicit Source Code Enhancement Options

You don’t need to have an explicitly defined enhancement spot in order to implement these enhancements. Just position the cursor on any of these implicit points and choose ‘Create Enhancement’ from the menu to implement it.

Explicit enhancement option

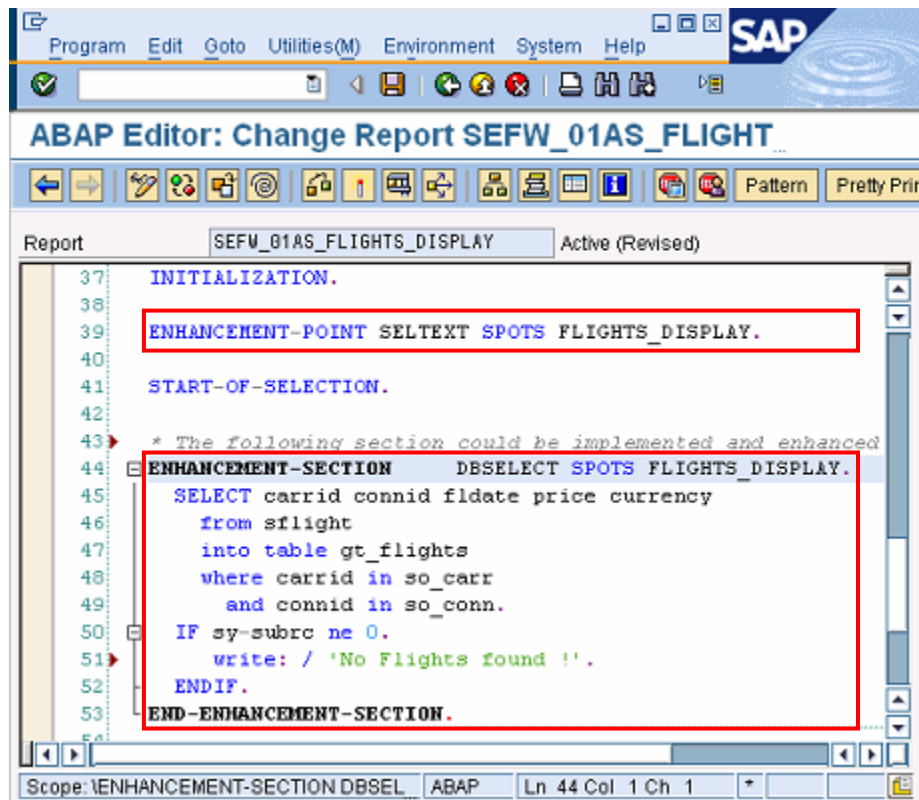
The Implicit enhancement options are provided at specific source code places explicitly by SAP (Note that these enhancement definitions can also be created by partners and customers in their code).

There are two types of Explicit Enhancement options available. One which can be provided at a specific place - called Enhancement Point, and another which can be used to replace a set of statements – called Enhancement Section. For this, we now have two new ABAP statements, viz.

- ENHANCEMENT-POINT
- ENHANCEMENT-SECTION

Introducing: “Enhancement Framework”

– The cool new way to enhance your ABAP systems



Explicit Source Code Enhancement Options

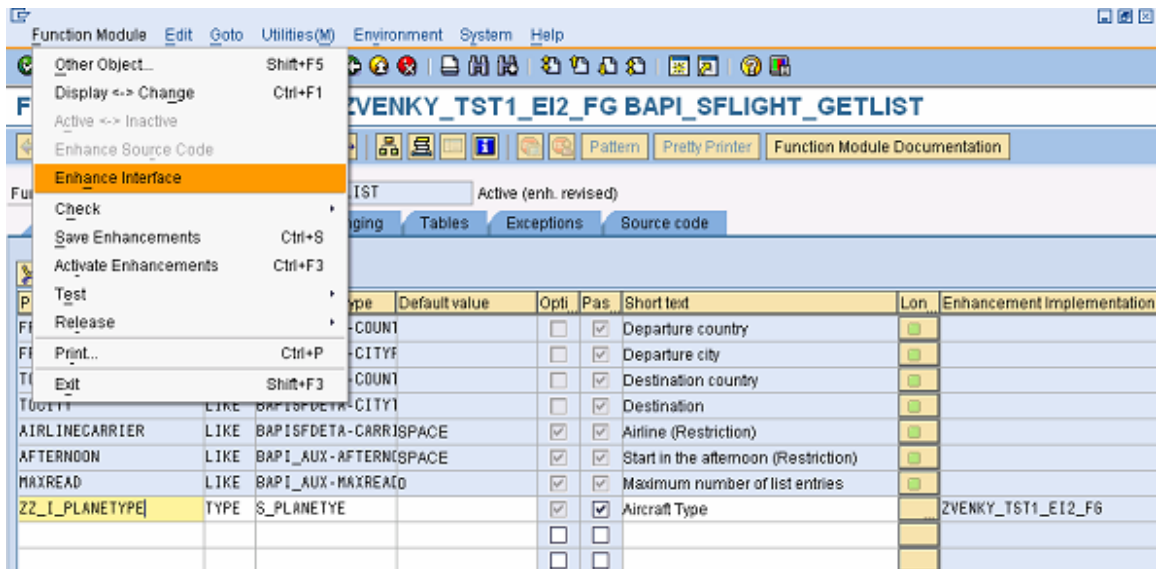
When the Enhancement-Section is implemented, only the implementation gets executed and the original code doesn't get executed. This is a new technique, which didn't exist previously in any of the old ways of enhancing, to exclude any standard SAP code from execution. Because of this, there can be only one active implementation of an Enhancement-Section. On the other hand, there can be multiple active implementations of an Enhancement-Point, in which case all the implementations will be executed with no guarantee in the order of execution.

Function Group enhancement

All application function modules can be enhanced by adding parameters to the standard function module interface. These parameters must be 'optional' in nature, since adding a mandatory parameter will require all calls to be changed. Additionally any function module that is part of the Central Basis can not be enhanced (for example: function module 'POPUP_TO_CONFIRM'). From the menu, choose 'Function module -> Enhance interface' to add optional parameters to a function module.

Introducing: “Enhancement Framework”

– The cool new way to enhance your ABAP systems



Enhancing Function Module interface

Think about all the steps that customers currently go through for adding new parameters to a BAPI. Cloning of BAPI is an option but maintaining and syncing up during upgrade could be a nightmare. Now with a combination of Function Group enhancement and Source code plug-in, BAPI's can easily be enhanced to add new parameters and add custom logic to process those parameters.

Class enhancement

The global Classes and Interfaces can be enhanced as follows:

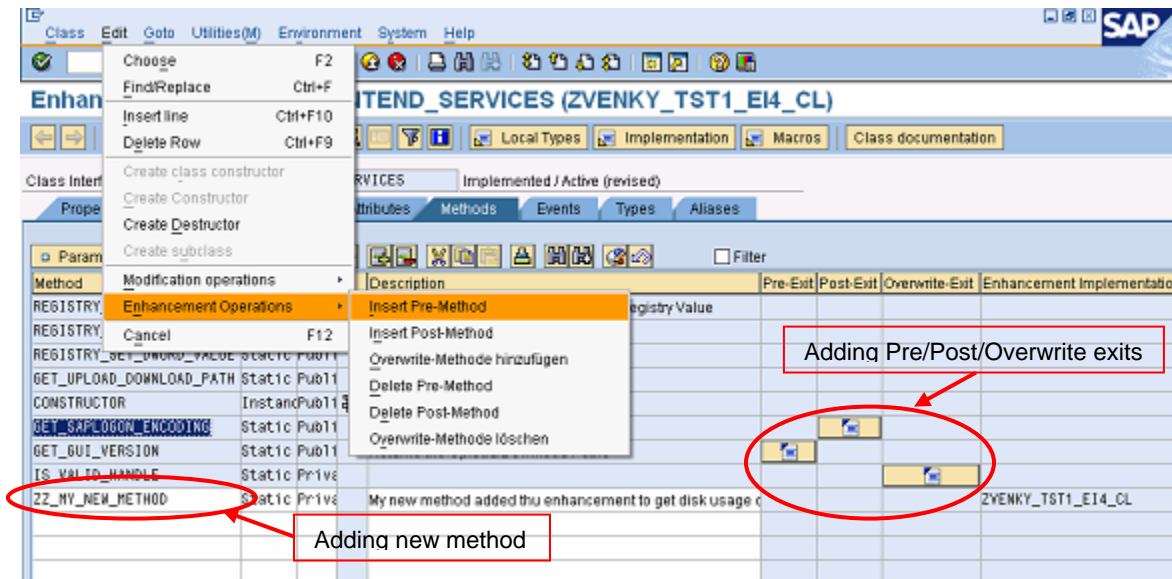
- Adding optional parameters to existing Methods
- Adding new Methods to the global Class / Interface
- Adding Pre-exit, Post-exit or Overwrite-exit to an existing Method

The Pre and Post exits get executed before and after invoking the respective method. These are achieved by an automatically generated local class. All methods are stored as part of this local class. (Point to ponder: Are private and protected attributes accessible from the Pre and Post exits?).

Choose menu option 'Class -> Enhance' to add new methods or parameters. Choose menu option 'Edit -> Enhancement operations' to add or delete the Pre/Post/Overwrite exit methods.

Introducing: “Enhancement Framework”

– The cool new way to enhance your ABAP systems



Enhancing global Classes / Interfaces

Kernel-BADI enhancement

The old classic-BADI's are implemented purely at the ABAP Workbench level; that is, both the definition and implementation are realized as workbench repository objects (global classes and interfaces). The new Kernel-BADI takes it to the ABAP language level. Because of this, the new Kernel-BADI's are much faster compared to the old classic-BADI's.

There are two new ABAP statements available now to support the Kernel-BADI's, namely GET BADI and CALL BADI.

Example:

```
data bd_hdl type ref to badi_name.  
GET BADI bd_hdl filters filt_1 = 'VALUE'.  
CALL BADI bd_hdl->method  
exporting param_1 = 10.
```

The old classic-BADI used to mix both implementation selection and method call in the same CALL METHOD statement. The implementations could only be chosen at run-time because of the above reason and due to the fact that the BADI handle could only be gotten from another method call by passing the BADI name.

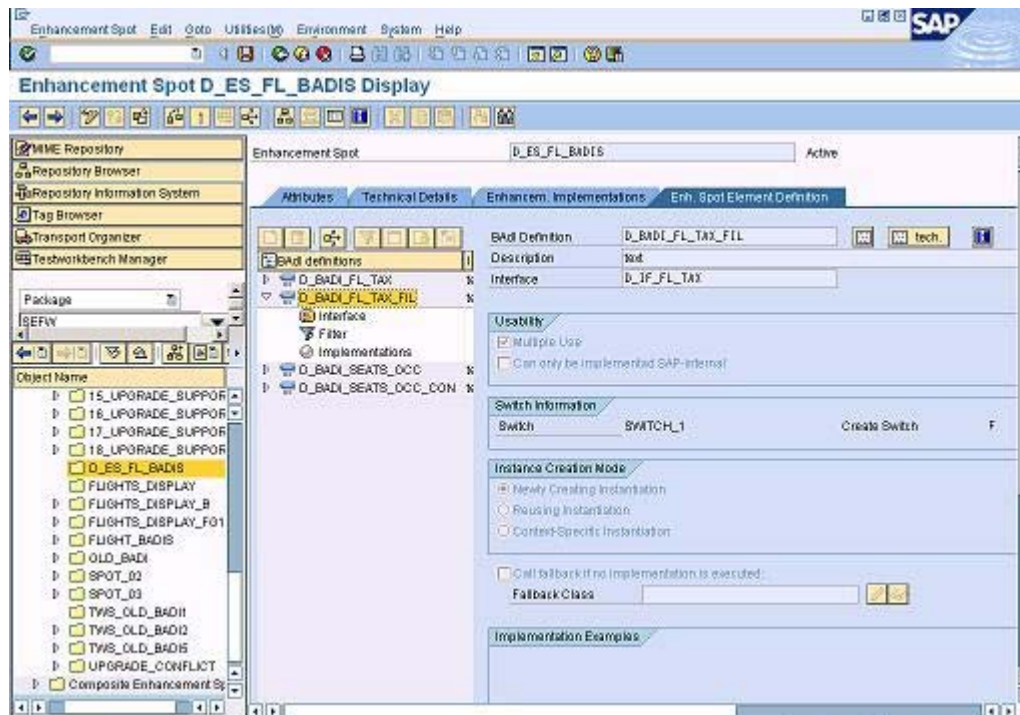
Whereas in the new Kernel-BADI, the active BADI implementations are included into the load of BADI handle **at compile time** and the filter criterion are expanded as IF statements. This is the reason the new Kernel-BADI's are much faster than classic-BADI's.

Some of the other new features of Kernel-BADI's are,

- Improved Filters with complex filter condition editor
- Stateful BADI support and Context based lifetime control
- Possibility to inherit the implementations
- Switchable using Switch Framework

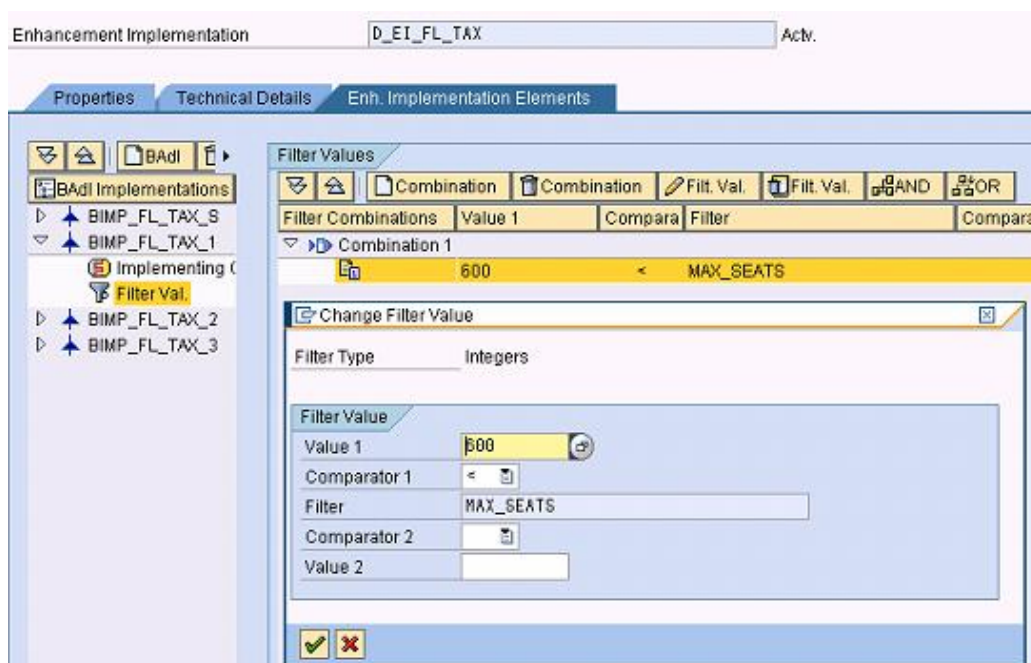
Introducing: “Enhancement Framework”

– The cool new way to enhance your ABAP systems



Creating BADI definition in SE80

Since Kernel-BADI is yet another enhancement technology under Enhancement Framework, we need to create Enhancement Spot in order to hold the BADI definitions. Similarly, the BADI implementations must also be assigned to Enhancement Implementations. The BADI definition and implementation editors are integrated into SE80.



Implementing BADI – Filter condition editor

Introducing: “Enhancement Framework”

– The cool new way to enhance your ABAP systems



Benefits / Value Proposition

By now you would have understood and appreciated the benefits this new technology brings to the table. Here is the summary.

- All enhancement technologies are now under a single roof of Enhancement Framework, and integrated into the ABAP workbench. Result: Enhancement is easily manageable.
- The Enhancement Framework can be switched using Switch Framework through package assignment.
- With new enhancement technologies (like Source Code, Function & Class enhancements), the options to enhance standard SAP code have increased tremendously. This cuts down on the need to do Core-mod, which requires more effort than enhancement
- Kernel-BADI's are much faster compared to the old classic-BADI's
- Implementing classes of Kernel-BADI's are re-usable through inheritance
- The lifecycle of BADI implementation instances are better managed through advanced options (Stateful BADI's through Context based and reusable instantiation)
- During the upgrade, SPAU_ENH transaction lets you create enhancement implementations to hold the BADI implementations, which guarantees the Return On Investment (ROI) for the old classic-BADI implementations.

One final word

With these new enhancement features (especially Implicit source code technology), the enhancement options are pretty much omnipresent. Like any technology which gives more power to the users of the system, this technology too if not used properly could lead to disorder. For example, just because more enhancement options are available it should not be over-used. Customers' business requirements for the need to adapt should still be the yard stick.

“With great power comes great responsibility”! So, choose the options wisely. My best-practice recommendation for the order in which the enhancement options should be considered and used is:

- Use a BADI; if there is no BADI to suite your need then,
- Try to solve it using Explicit Source code, Function and Class enhancements; and,
- Implicit Source code enhancement should be the last option to choose

Of course, if none of the above could solve your adaptation needs and if your business demands it, modification should be the last option.

Now that the Sneak preview ABAP edition of SAP NetWeaver 2004s is available in [SDN Downloads section](#), go ahead and try these out to equip and prepare for the future.

Author Bio



Venky Varadadesigan is a Product Manager for SAP NetWeaver with SAP Labs - in charge of Application Server topics like ABAP (including Web Dynpro for ABAP, Enhancement Framework, Switch Framework) and Systems (Monitoring & Scheduling). He comes from a consulting background with rich experience in Exchange Infrastructure and Portal.

Email: venkata.varadadesigan@sap.com

Personal: venkynarayanan@gmail.com

Mobile: 832-287-2135

Introducing: “Enhancement Framework”

– The cool new way to enhance your ABAP systems



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.