



The Enhancement Framework

for Nordic User group

Michael Acker

Development Architect, NW AS ABAP Workbench, SAP AG

Dr. Roland Krämer

Senior Developer, NW AS ABAP Workbench, SAP AG

This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice. SAP assumes no responsibility for errors or omissions in this document.

As a result of this workshop, you will be able to

- **Understand the fundamental idea of the Enhancement Framework and Switch Framework**
- **Reduce TCO by using enhancement technologies instead of modifications**
- **Enhance SAP standard objects**
- **Create your own Enhancement Spots for your own source code**
- **Work with enhancements during an upgrade or import of a support package**



Enhancement Framework Overview

Source Code Plugin – Technology

Function Group Enhancement – Technology

Class Enhancement – Technology

BAdI – Technology

Upgrade Adjustment

Switch Framework

Summary



Enhancement Framework Overview

Source Code Plugin – Technology

Function Group Enhancement – Technology

Class Enhancement – Technology

BAdI – Technology

Upgrade Adjustment

Switch Framework

Summary

One of the advantages of SAP software is the possibility to adapt the software to own requirements and the possibility of keeping the adaptations during upgrade.

Ways of adaptation:

- **Customizing**
- **Enhancement**
- **Modification**

Reducing TCO

- **Enhancing objects instead of modifying them reduces the effort for adjustment during SP import or upgrade.**

Disadvantages of modifications

- **No support for multiple users or projects**
- **No grouping**
- **No support for parallel developments**
- **Will appear much more often in adjustment tools**
- **Higher adjustment effort**

Integration of several enhancement types

- **BAdIs**
- **Functiongroup Enhancement**
- **Class/Interface Enhancement**
- **Source Code Plugins**
- **WebDynpro Enhancement**
- ...

into the Enhancement Framework

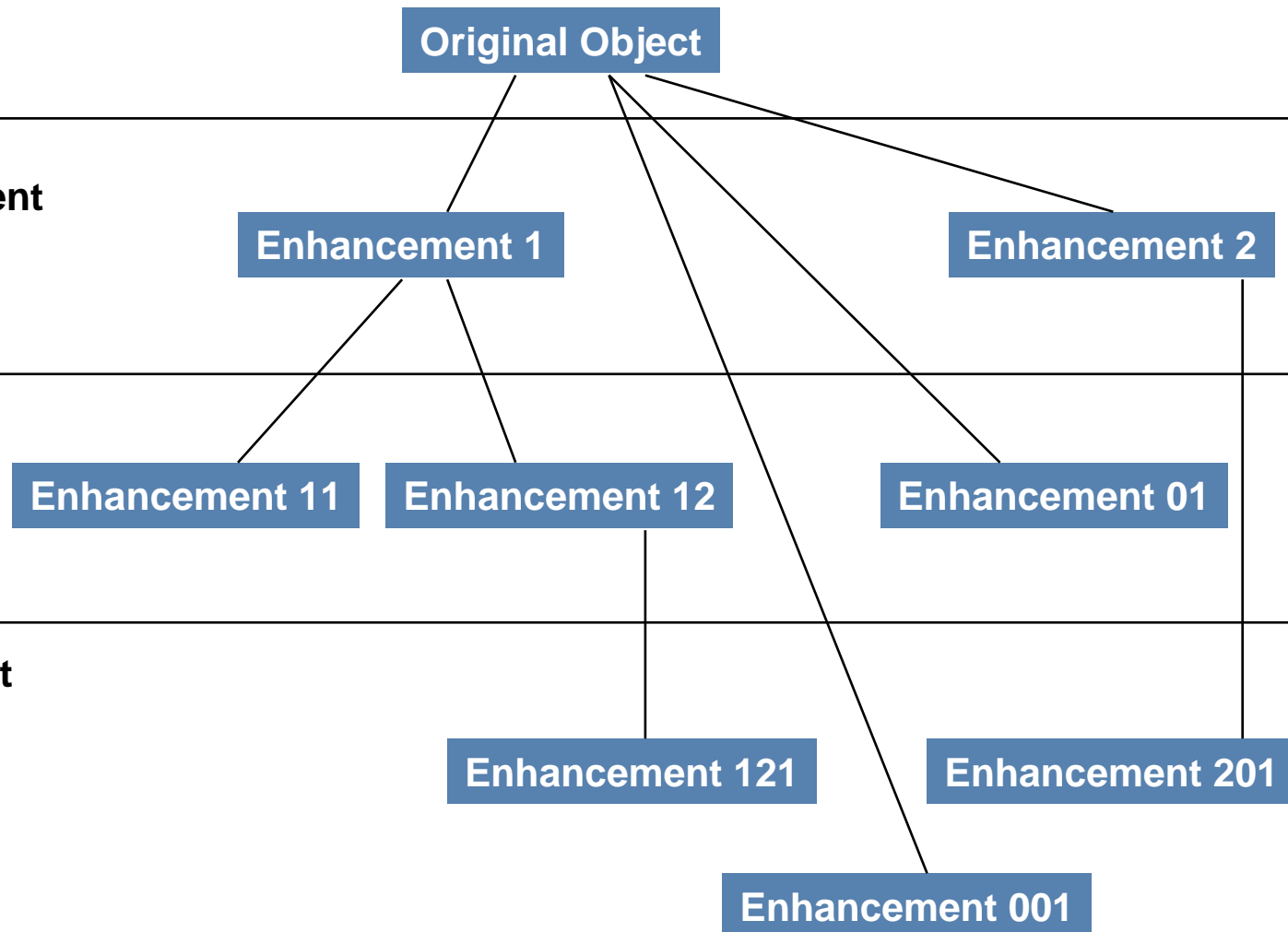
- **Switchable by Switch Framework**
- **Enhancement Browser**
- **Upgrade support**
- **Possibility to document and group enhancements**
- **Multilayer support**

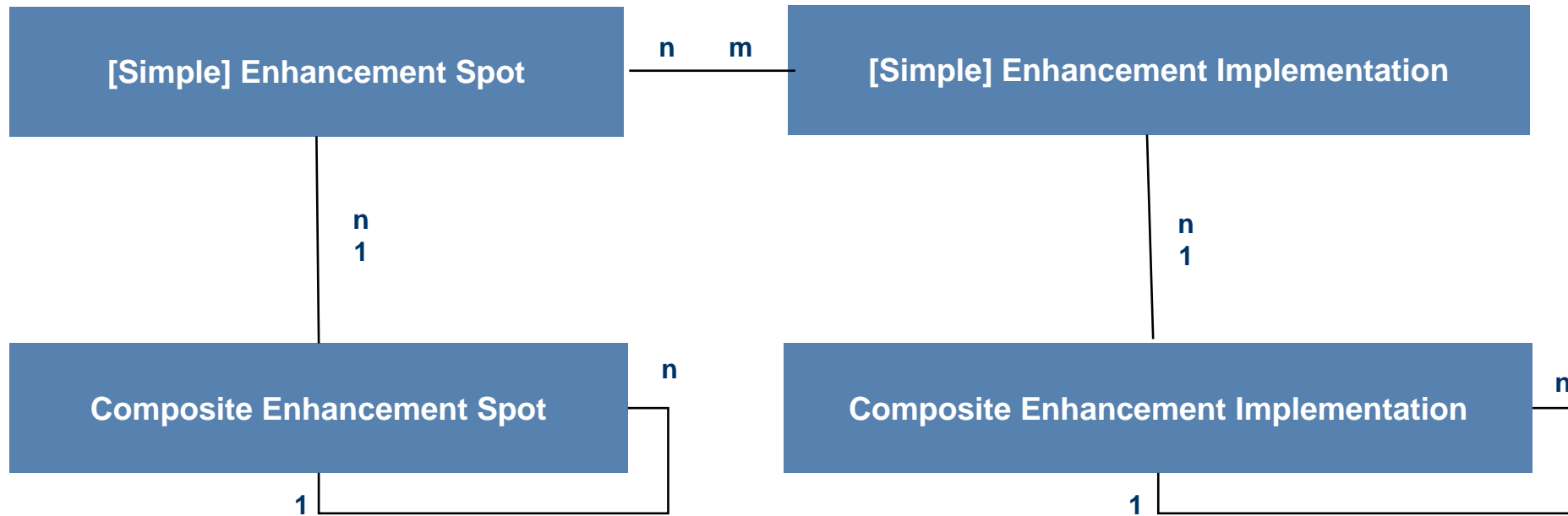
Core Development

Application Development

Add On Development

Customer Development





Composite Enhancement Spot

[Simple] Enhancement Spot

**Enhancement Elements:
Such as a
BAdI-Definition**

Composite Enhancement Spots

- **Container Objects**
- **Can contain**
 - **Other Composite Enhancement Spots**
 - **[Simple] Enhancement Spots**

[Simple] Enhancement Spots

- **Container Objects**
- **Can contain Enhancement Elements**

Explicit Enhancement Options

- **Enhancement Definitions**

Composite Enhancement Implementation

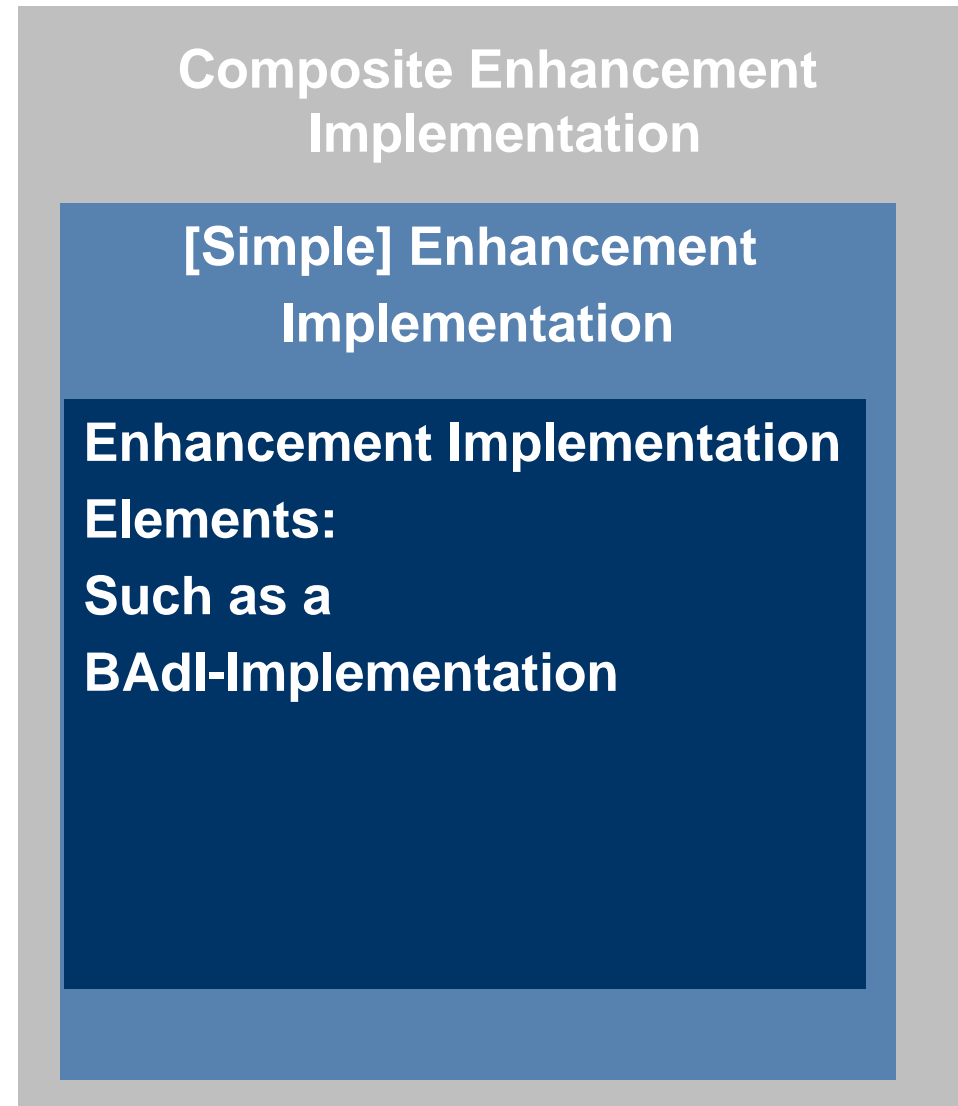
- **Container Objects**
- **Can contain**
 - **Other Composite Enhancement Implementations**
 - **[Simple] Enhancement Implementations**

[Simple] Enhancement Implementation

- **Can contain Enhancement Implementation Elements**

Enhancement Implementation Elements

- **Implementations**




Enhancement Spot Editor




Enhancement Spot: FLIGHTS_DISPLAY_00 Active

Attributes Technical Details **Enhancem. Implementations** Enhancement Options

Dev. Obj. that Can Be Enhanced: Z_00_EXERCISE1



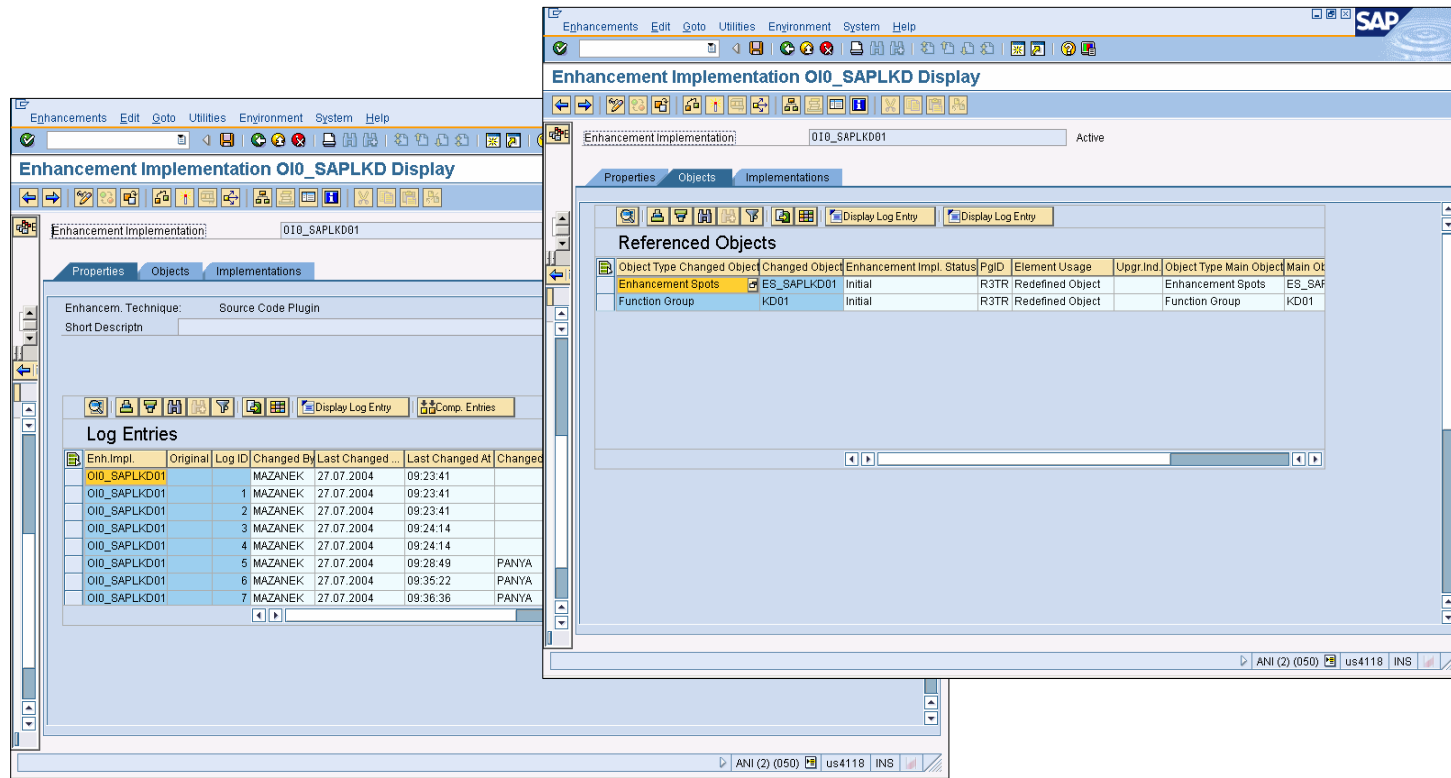
List of Enhancement Options

	Enhancement Implementation Type	Type of Modification Unit	Include-Bound	Name of Modification Unit
 \PR:Z_00_EXERCISE1\EX:DECLARATION	Static Enhancement Point	Before First Unit		N->Z_00_EXERCISE1
 \PR:Z_00_EXERCISE1\EX:SELTEXT	Dynamic Enhancement Point	Event		INITIALIZATION
 \PR:Z_00_EXERCISE1\EX:DBSELECT	Dynamic Enhancement Section	Event		START-OF-SELECTION

◀ ▶

- **Editor for Predefined Enhancement Possibilities (Source Code Enhancements & BAdIs)**
- **Integrated in Object Navigator (SE80)**
- **common tabs for all Enhancement Spots**
- **Tab 3 ➔ dependent on enhancement technology: BAdIs or Source Code Enhancements**

Enhancement Implementation Editor



- Editor for Enhancement Implementations
- Integrated in Object Navigator (SE80)
- Tab Properties & Objects → common for all enhancement types
- Tab 3 → dependent on enhancement technology:
e.g. BAdI-Implementation or Source Code Enhancements

Composite Enhancement Implementation

Composite Enh. Implementation CIMPL_01 Active

Short Text TEST

General Data

Created By	DANGERS	Last Changed By	DANGERS
Created On	15.06.2005	Changed On	15.06.2005
Original Language	EN English	Paket	\$TMP

Enhancement Implementations Composite Enhancement Implementation

Enhancement Implementations

Enhancement Implementation	Status	Short Text	Enhancement Type
IMPL_01	✔	test	Redefinition

**Contained
Composite
Enhancement
Implementations**

**contains [Simple]
Enhancement
Implementations**

Composite Enhancement Spot

ERP

Composite Enhancement Spot

SD

Composite Enhancement Spot

Invoice

Composite Enhancement Spot

Pricing



Enhancement Spot
Source Code Enhancement
FUGR V60A

Enhancement Spot
BAdI Definition
PRICING_BADI1

Composite Enhancement Implementation

IS_OIL

Composite Enhancement Implementation

Upstream

Composite Enhancement Impl.

OIL_PRICING

Enhancement Implementation
Functiongroup Enhancement
FUGR V60A

Enhancement Implementation
Source Code Plugin
FUGR V60A

Enhancement Implementation
BAdI Implementation
OIL_PRICING_IMPL1



Features of explicit enhancement options

- **More stable, pre-defined**
- **Few changes in definition to expect**
- **Only at valid source code locations**

Features of implicit enhancement options

- **enhancement of „arbitrary“ objects**
- **No enhancement spots necessary**

Search for

- Enhancements possibilities
- Existing Enhancement Implementations
- Enhancement Implementations to be adjusted after upgrade

Enhancement Browser

Composite Enhancement Implementation: Display CIMPL_01

Enhancement Info System

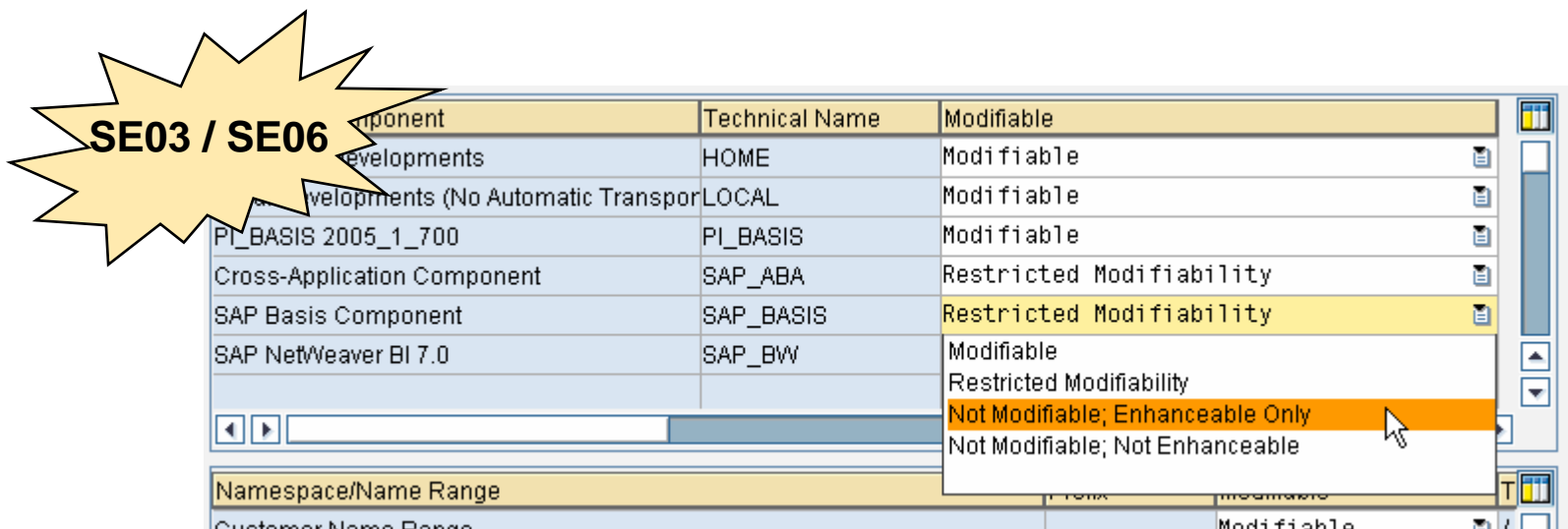
Composite Enh. Implementation: CIMPL_01 Active

Created By: DANGERS Last Changed By: DANGERS
Created On: 15.06.2005 Changed On: 15.06.2005
Original Language: EN English Paket: \$TMP

Enhancement Implementations

Enhancement Implementation	Status	Short Text	Enhancement Type
IMPL_01	✓	test	Redefinition

System Change Option – Enhanceable



New option:

Enhanceable

➔ Only creation of enhancement implementations is allowed



Enhancement Framework Overview

Source Code Plugin – Technology

Function Group Enhancement – Technology

Class Enhancement – Technology

BAdI – Technology

Upgrade Adjustment

Switch Framework

Summary

```
PROGRAM p1.
```

```
WRITE 'Hello World'.
```

```
ENHANCEMENT-POINT ep1 SPOTS  
s1.
```

```
..  
..  
..
```

```
ENHANCEMENT-SECTION ep2  
SPOTS s1.  
    WRITE 'Original'.  
END-ENHANCEMENT-SECTION.
```

```
ENHANCEMENT 1.  
    WRITE 'Hello  
Paris'.  
ENDENHANCEMENT
```

```
ENHANCEMENT 2.  
    WRITE 'Hello  
London'.  
ENDENHANCEMENT.
```

```
ENHANCEMENT 3.  
    WRITE 'Enhanced'.  
ENDENHANCEMENT.
```

Modification-free enhancement of source code

Explicit Enhancement Option

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

Implicit Enhancement Option

- **At common enhancement places, implicit Enhancement options are available. Examples:**
 - Beginning/End of Include
 - Beginning/End of Method/Function Module/Form Routine
 - End of a structure
 - End of Private/Protected/Public Section of a local class
 - ...

Explicit Enhancement Options

* Selection screen

SELECT-OPTIONS:

so_carr FOR gv_carrid,

so_conn FOR gv_connid.

ENHANCEMENT-POINT FLIGHTS_DECLARATION SPOTS FLIGHTS_DISPLAY STATIC.

START-OF-SELECTION.

* Select Data

ENHANCEMENT-SECTION FLIGHTS_DBSELECT SPOTS FLIGHTS_DISPLAY.

SELECT carrid connid fldate price currency

from sflight

into table gt_flights

where carrid in so_carr

and connid in so_conn.

END-ENHANCEMENT-SECTION.

* Listoutput

LOOP AT gt_flights INTO gs_flights.

WRITE: / gs_flights-carrid,

gs_flights-connid,

gs_flights-fldate.

Implicit Enhancement Options

The screenshot shows the SAP IDE interface. The 'Enhancement Operations' menu is open, and 'Show Implicit Enhancement Points' is highlighted. The background code is ABAP, showing a list output loop and a method definition for GET_PROPERTY. Red ovals highlight the code sections where implicit enhancement points are shown.

```
* Listoutput
LOOP AT gt_flights INTO gs_
WRITE: / gs_flights-carri
gs_flights-connid,
gs_flights-fldate,
gs_flights-price,
gs_flights-currency.
ENDLOOP.
```

*****"\$\SE:(1) Include ZSEFW_00_FLIGHTS_DISPLA

```
METHOD GET_PROPERTY.
*****"$\SE:(4 ) Klasse CL_GUI_OBJECT, Methode 6
* (PROPERTY, P1, P16, P COUNT, QUEUE_ONLY; => VALUE)
* check handle
DATA : STRPROPNAME TYPE STRING.
DATA : STRVALUE TYPE STRING.
DATA : QUEUE_EMPTY TYPE CHAR01.

**
*****"$\SE:(3 ) Klasse CL_GUI_CONTROL, PUBLIC S
*****"$\SE:(2 ) Klasse CL_GUI_CONTROL, PROTECTE
*****"$\SE:(1 ) Klasse CL_GUI_CONTROL PRIVATE
```

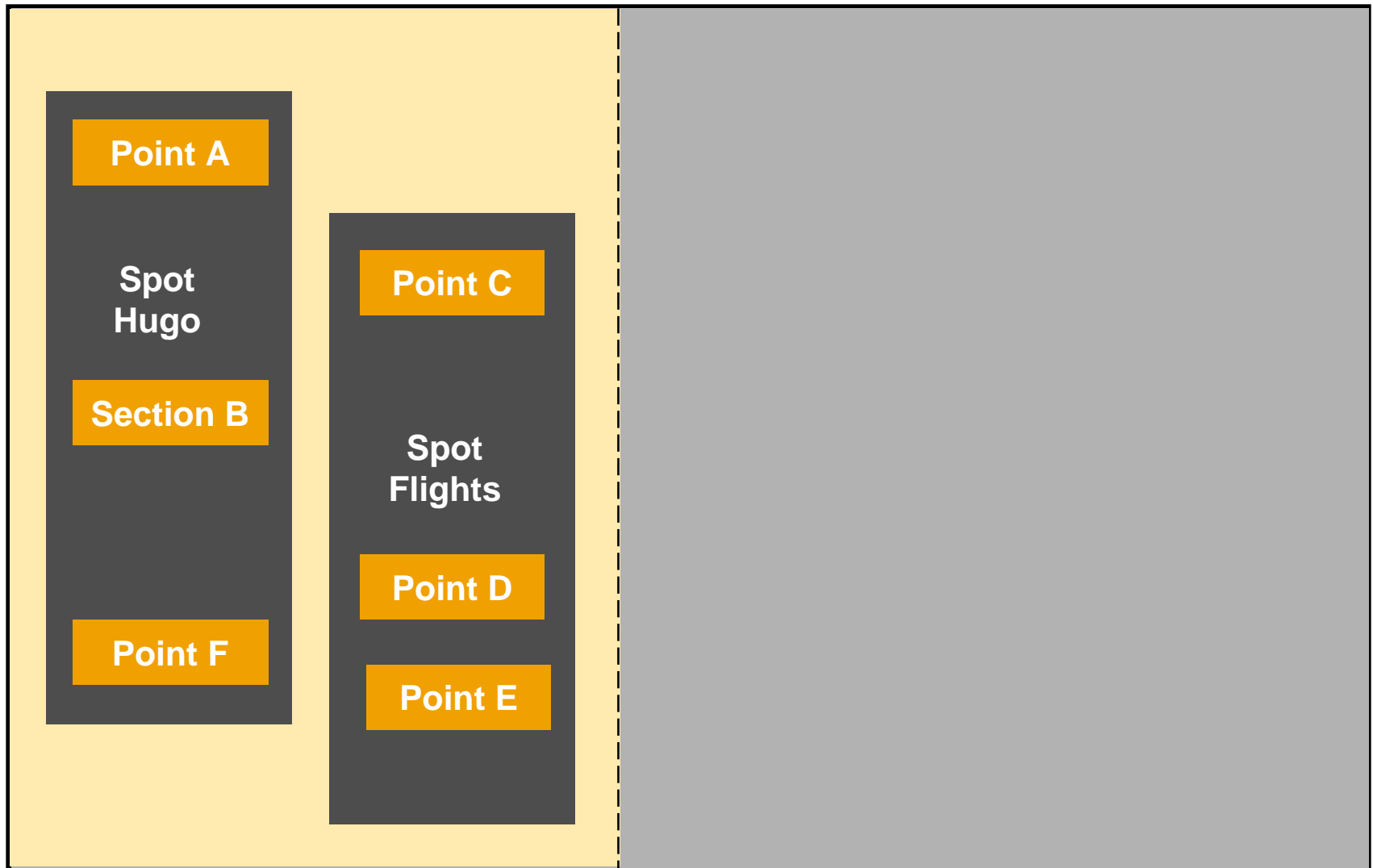

ENHANCEMENT-POINT <name> SPOTS <spot1> [<spot2>] ..
[STATIC]

- **Static** – e.g. additional data declaration
- **Dynamic** – e.g. additional source code

ENHANCEMENT-SECTION <name> SPOTS <spot1> [<spot2>] ..
[STATIC]

- **Static** – e.g. replace an existing data declaration
- **Dynamic** – e.g. replace source code

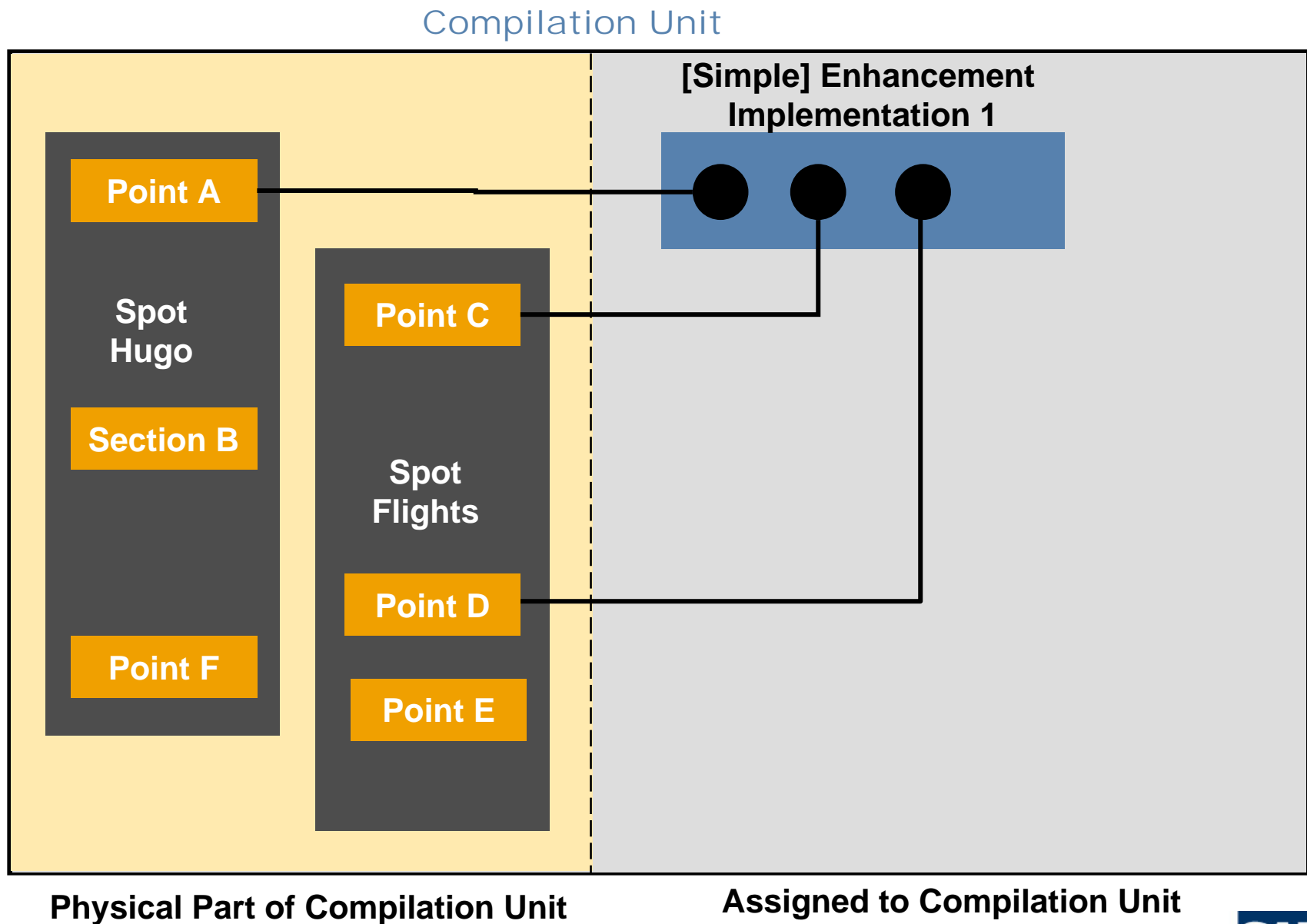
Compilation Unit



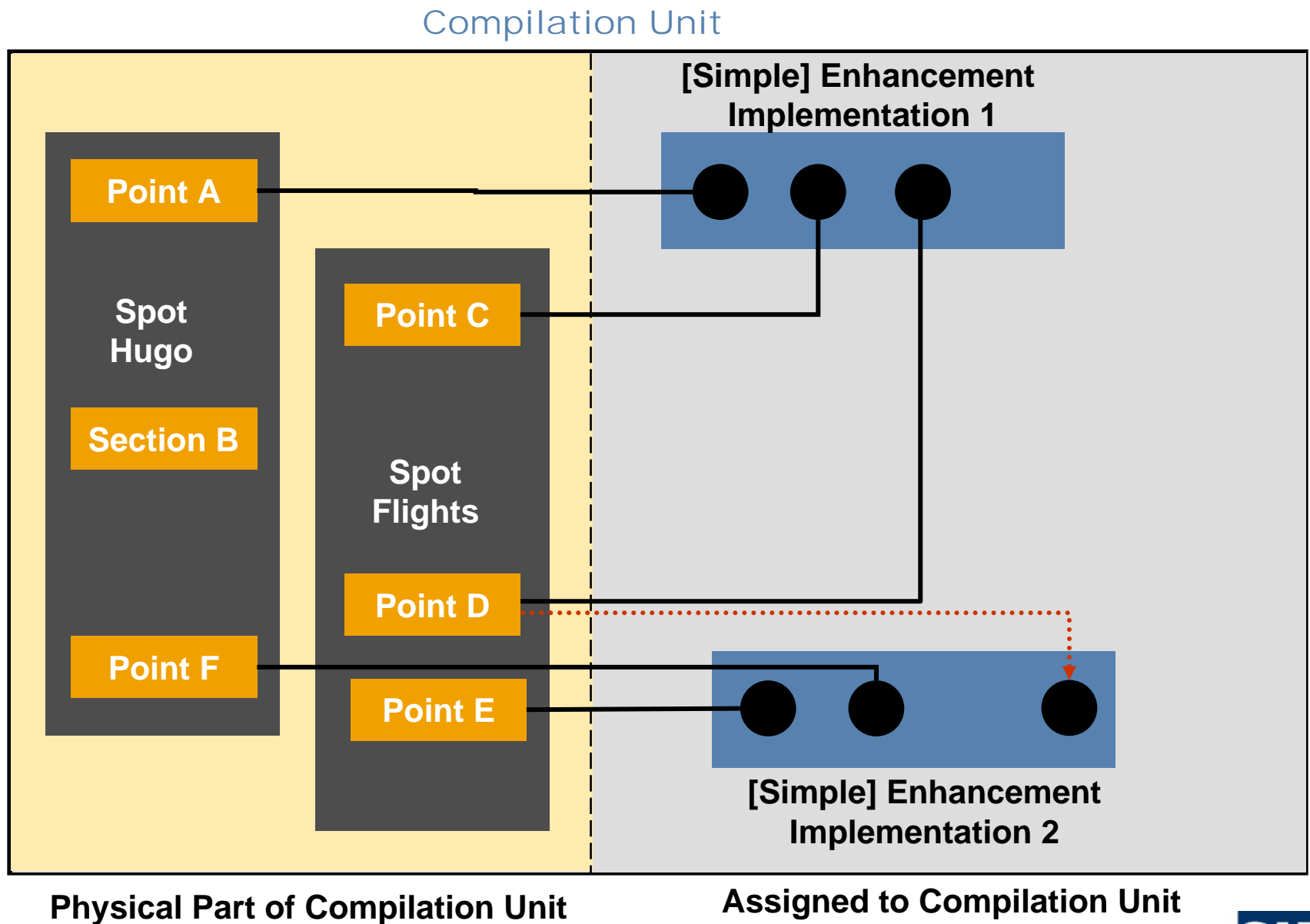
Physical Part of Compilation Unit

Assigned to Compilation Unit

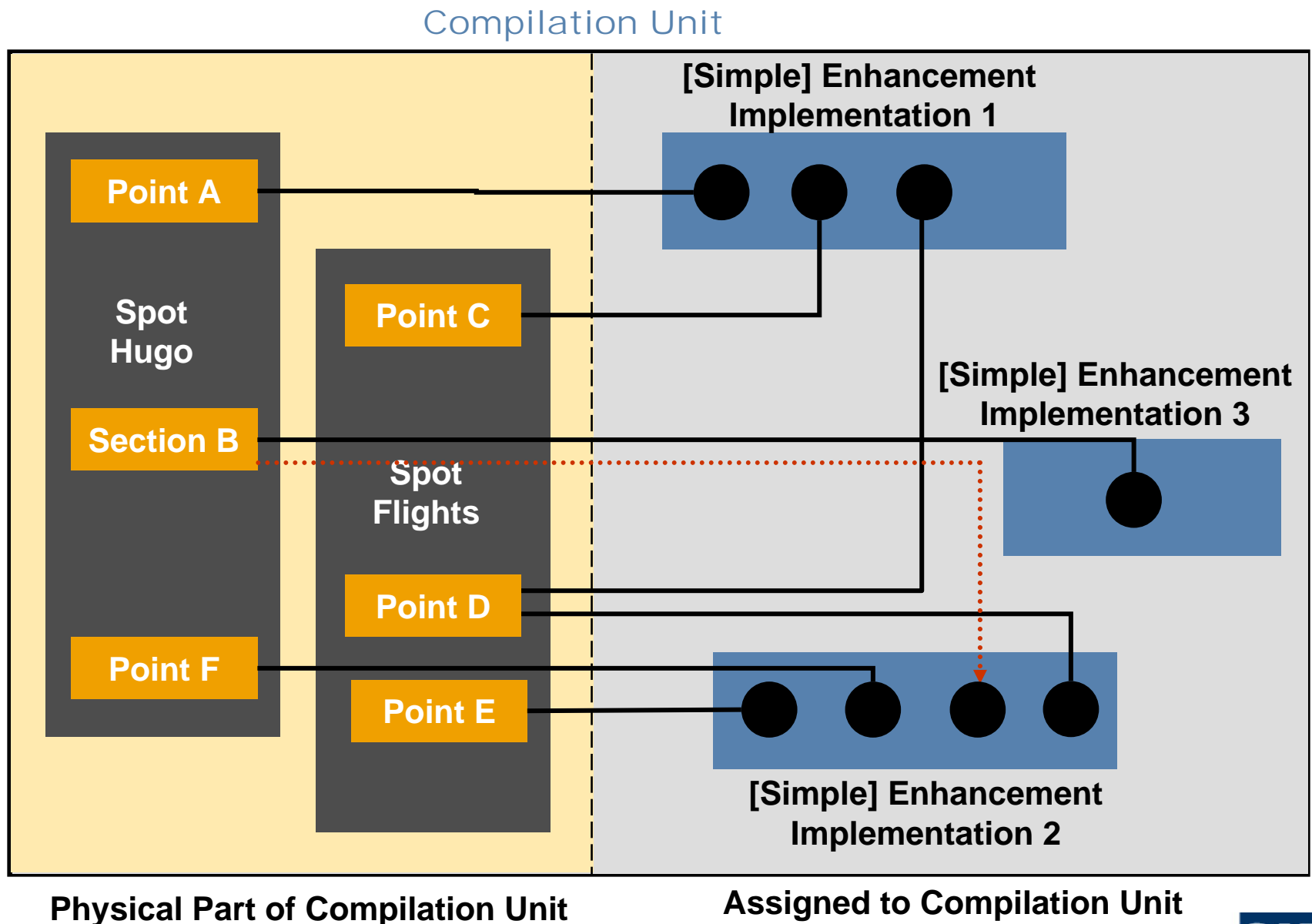
Enhancement Spots & Implementations



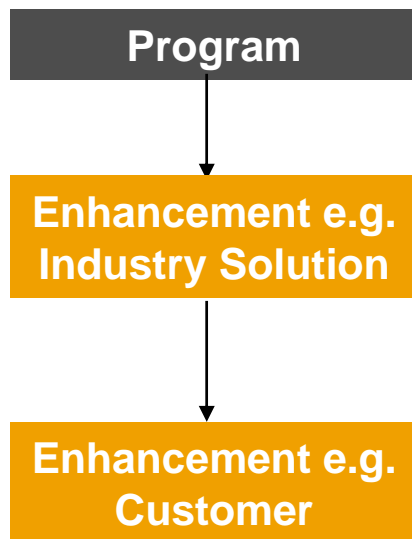
Enhancement Spots & Implementations



Enhancement Spots & Implementations



Existing Enhancements must not be modified. Enhance the Enhancement instead!



Example:

An SAP Program is enhanced by an Industry Solution.

The source code of the Industry Solution enhancement is exchanged by a customer enhancement.

Use **Change Mode** for creating enhancement points & sections.

- use button  "Display <-> Change" to switch to change mode.

Use **Enhancement Mode** for creating enhancement implementations.

- use button "Change Enhancements"  to switch to Enhancement mode
- use button "Display <-> Change"  to leave Enhancement mode



DEMO



Enhancement Framework Overview

Source Code Plugin – Technology

Function Group Enhancement – Technology

Class Enhancement – Technology

BAdI – Technology

Upgrade Adjustment

Switch Framework

Summary

Function Group Enhancements allow

- Adding new optional parameters to existing function modules

Function Builder: Display RV_XVBUP_MAINTAIN

Function module: RV_XVBUP_MAINTAIN Active

Attributes Import Export Changing Tables Exceptions Source code

Parameter Name	Type...	Associated Type	Default value	Opt...	Pa...	Short text	Lo...	Enhancement
F_VBELN	LIKE	VBAK-VBELN		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Document number		
F_XVBAP_SORTIERT				<input type="checkbox"/>	<input checked="" type="checkbox"/>	X = items table is sorted (XVBAP)		
F_XVBUP_BERUECKSICH...			X'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	X = Determine completeness status. ...		
F_FPLNR	LIKE	VBKD-FPLNR	SPACE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
F_XFPLT_BERUECKSICH...			SPACE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
F_REQ_BLOCK	LIKE	TVLS-SPEBE	SPACE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
F_BESTA_ERMITTELN			SPACE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
F_VORGAENGER				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
F_DCSTA_ERMITTELN	LIKE	TVAK-DELCF		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
F_LIFSK	LIKE	VBAK-LIFSK		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Lieferscheinsperre Belegkopf		/SAPHT/SW_LIC_BU_VE_SAPLV4
F_GBL_BUMP	LIKE	VBUP-BUMP	SPACE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	High-Tech-Bump-Kennzeichen		/SAPHT/SW_LIC_BU_VE_SAPLV4
F_BOS_PACKAGE_NO	LIKE	VBKD-PACKNO		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Paketnummer		AD_BOS_01_SAPFW45P_I
F_BOS_VBRK_FKTYP	LIKE	VBRK-FKTYP		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Fakturaty		AD_BOS_01_SAPFW45P_I
				<input type="checkbox"/>	<input type="checkbox"/>			
				<input type="checkbox"/>	<input type="checkbox"/>			
				<input type="checkbox"/>	<input type="checkbox"/>			

ANI (2) (050) us4118 INS

Add optional parameters to existing function modules

Function module

BAPI_SFLIGHT_GETDETAIL

Aktiv(Erweiterung aktiv)

Attributes

Import

Export

Changing

Tables

Exceptions

Source code

Parameter name	Type...	Reference type	Default value	Opt...	Pa...	Short text	Lo...	Enhancement
AIRLINECARRIER	LIKE	BAPISFDETA-CAR...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Carrier ID		
CONNECTIONNUMBER	LIKE	BAPISFDETA-CON...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Connection number		
DATEOFFLIGHT	LIKE	BAPISFDETA-FLD...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Departure date		
PARAMETER_1	TYPE	STRING		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			CD_FB_1
				<input type="checkbox"/>	<input type="checkbox"/>			
				<input type="checkbox"/>	<input type="checkbox"/>			



Enhancement Framework Overview

Source Code Plugin – Technology

Function Group Enhancement – Technology

Class Enhancement – Technology

BAdI – Technology

Upgrade Adjustment

Switch Framework










Summary

Class/Interface Enhancements allow addition of:

- **optional parameters to existing methods**
- **methods**
- **events and event handlers**
- **references to interfaces**
- **types***
- **exits to existing methods**
 - **Pre-Exit – Called at the beginning of a method**
 - **Post-Exit – Called at the End of a method**
 - **Overwrite-Exit – Replaces the original method**
 - **access to private and protected attributes of the original class***

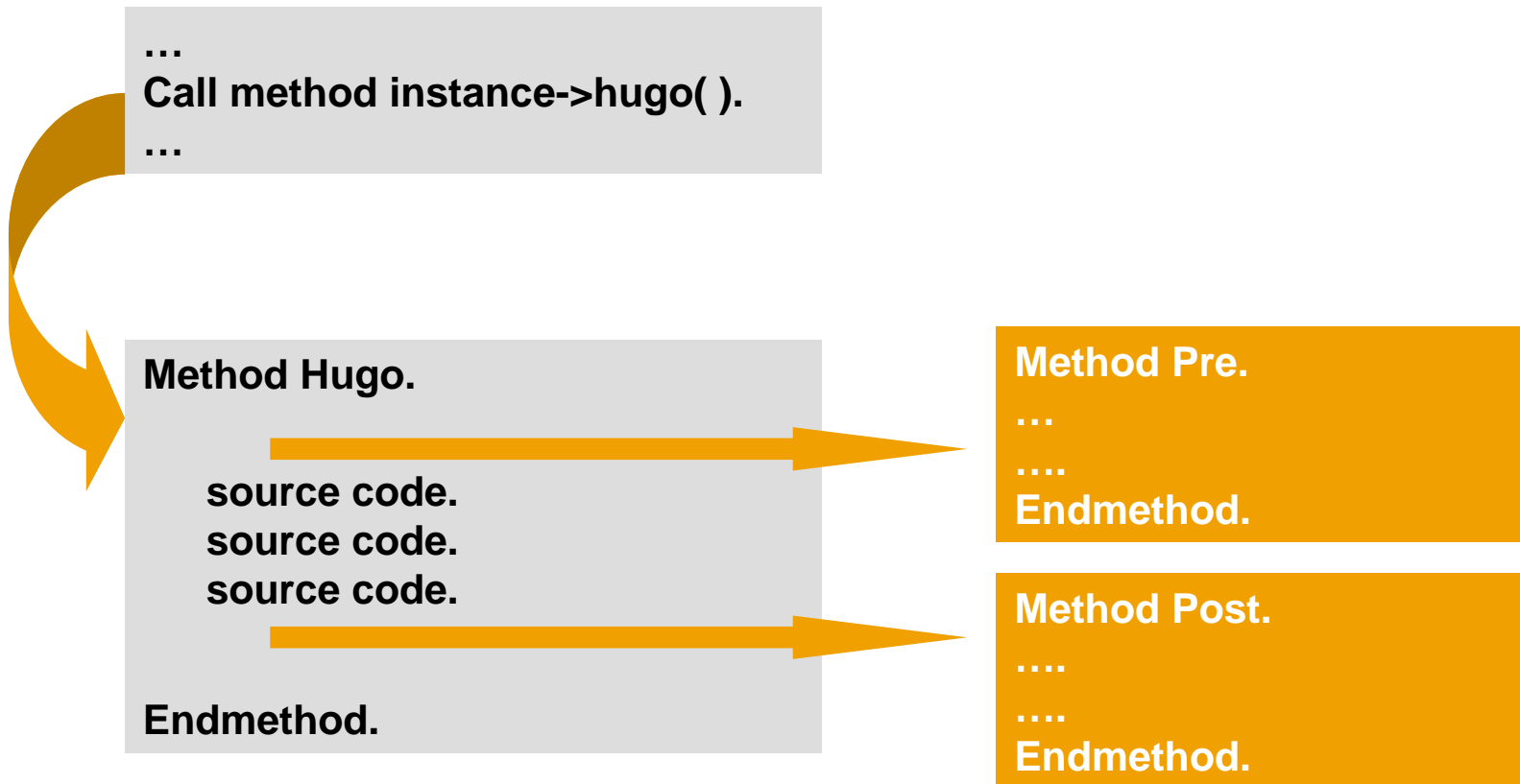
* as from next release

Add optional parameters to existing methods

Method parameters									
AFTER_IMPORT									
 Methods  Exceptions       									
Parameter	Type	P...	O...	Typing M...	Associated Type	Default value	Description	Enhancement	
OBJ_NAME	Importin	<input type="checkbox"/>	<input type="checkbox"/>	Type	TROBJ_NAME		Object Name in Object Directory		
PROTOCOL	Changir	<input type="checkbox"/>	<input type="checkbox"/>	Type	SPROT_U_TAB		Table Type for SPROT_U (Log In		
MY_ADD_PARAM	Importin	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	CHAR10		Characterfeld der Länge 10	MATECHED2005_001	
MY_ADD_PARAM_EXP	Exportin	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Type	INT4		Natürliche Zahl	MATECHED2005_001	

Add new methods

Method	Level	Visibility	M...	Description	PreExit	PostExit	Enhancement
AFTER_IMPORT	Static	Public					
UPDATE	Static	Public					
READ	Static	Public					
WRITE	Static	Public					
ADD_METHOD_ONE	Instant	Private		additional functionality public			MATECHED2005_001
ADD_METHOD_PRIVATE	Instant	Private		additional functionality private			MATECHED2005_001





DEMO



Enhancement Framework Overview

Source Code Plugin – Technology

Function Group Enhancement – Technology

Class Enhancement – Technology

BAdI – Technology

Upgrade Adjustment

Switch Framework

Summary

A BAdI

- is an anticipated **point of extension** – these points act like sockets and exist in the original source code
- is a **predefined anchor** for an Object PlugIn
- has a **well-defined interface** in contrast to source code plugIns and is therefore more stable to changes in the original source code
- has **switchable** implementation(s) (by switches introduced by the Switch Framework)

The calculation depends on the filter value. Only one active implementation is allowed for one filter value.

```
GET BADI mytaxbadi
  FILTERS LANGUAGE = sy-
    langu.
  ..

CALL BADI mybadi->add_tax
  EXPORTING
    value = my_value
  IMPORTING
    result = my_result.
  ..
write / my_result.
```

language = ,XY'

language = ,DE'

```
METHOD if_mybadi~add_tax.
  result = value +
    myfunc(value).
ENDMETHOD.
```

```
METHOD if_mybadi~add_tax.
  result = value * 1.16.
ENDMETHOD.
```

The BAdI allows to perform an additional step inside a standard process. The core functionality doesn't need any implementation, but the implementation may react on a core event. Several implementations may be called in a sequence.

```
GET BADI mybadi.  
..  
CALL BADI mybadi->value_changed  
  EXPORTING  
    value = my_value  
..  
..
```

```
METHOD  
  if_mybadi~value_changed.  
    insert value ..  
  ENDMETHOD.
```

```
METHOD  
  if_mybadi~value_changed.  
    write value ..  
  ENDMETHOD.
```

Use the advantages (performance, upgrade adjustment) to implement a registry of classes.

```
GET BADI myregbadi.  
  FILTERS reg = 'DEF'  
..  
CALL BADI myregbadi->do  
  
..
```

ABC	
DEF	
XYZ	
MER	
FGH	

```
METHOD  
if_mybadi~get_result.  
  do ...  
ENDMETHOD.
```

```
METHOD  
if_mybadi~get_result.  
  do ..  
ENDMETHOD.
```

New BAdIs – New Features

- **Are integrated directly in the ABAP Language/Runtime**
- **Improved filter support allows non-character filter types (packed, numeric, string) and complex filter conditions**
- **Enable reusable implementation instances**
- **Different kinds of default implementations**
- **Control of the lifetime of implementations (BAdI-context)**
- **Allow for inheritance of implementations**

Comparison: Usage of Old BAdIs vs. New BAdIs

With Classic BAdI

```
DATA: bd TYPE REF TO if_intf.  
DATA: flt TYPE flt.  
  
CALL METHOD cl_exithandler=>  
get_instance  
EXPORTING  
    exit_name = `BADI_NAME`  
CHANGING  
    instance = bd.  
  
flt-lang = `D`.  
CALL METHOD bd->method  
EXPORTING  
    x          = 10  
    flt_val    = flt.
```

selecting implementations and issuing calls is mixed

calls are redirected over a proxy class

With New BAdI

```
data bd type ref to badi_name.  
get badi bd filters lang = `D`.  
call badi bd->method  
    exporting x          = 10.
```

Selection occurs when the handle is requested

Implementations are called directly (without a proxy)

The new BAdI evaluates as much information as possible during compile time.

Better Performance/Lower Memory consumption

- **Database access only at compile time**
- **Statically typed comparisons at runtime**
- **Internal handle-class integrated in SAP Kernel**

→ 40-600 times faster than Classic BAdIs

Active implementations are evaluated at compile time and included in the load of the BAdI-handle.

- **Switch Framework** – only implementations, that are switched on in at least one client are considered.
- **Constant Filter values** – Implementations that do not match the filter are excluded
- **Active Flag** – Only active implementations are considered

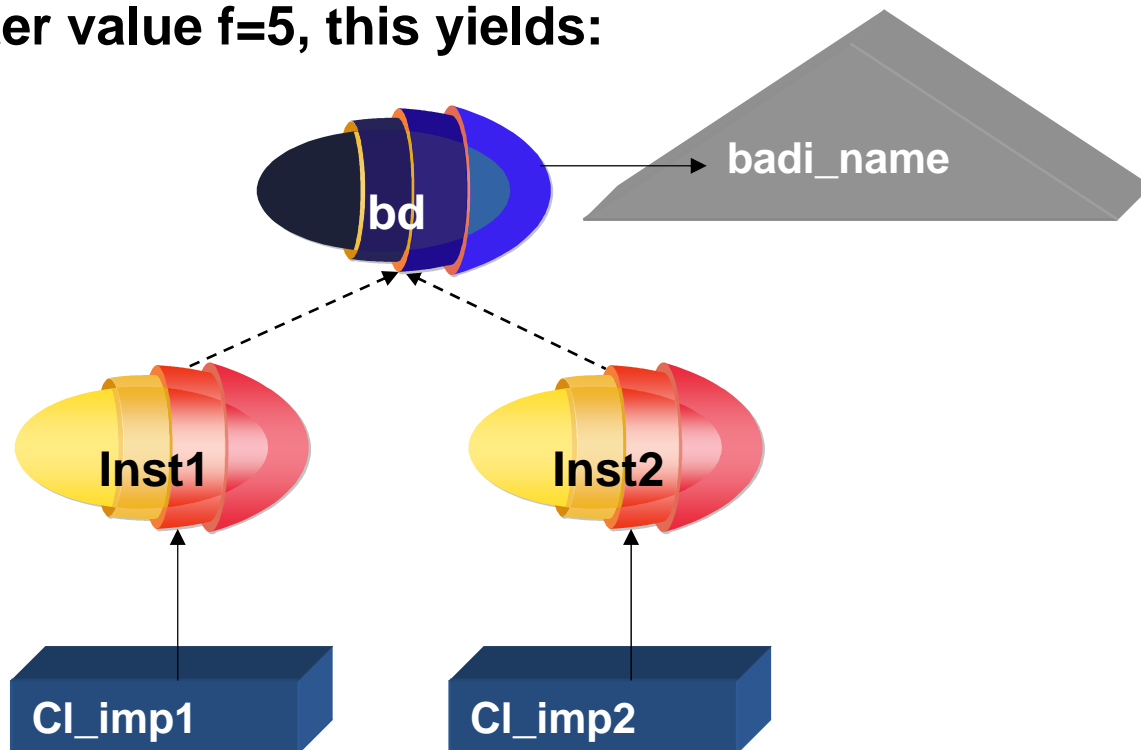
Special optimizations for BAdIs with

- **no active implementations:**
 - ‘GET BADI’ is ignored while compiling
 - ‘CALL BADI’ just needs the time for a simple if-statement.
- **one active implementation**
 - Direct call of that implementation

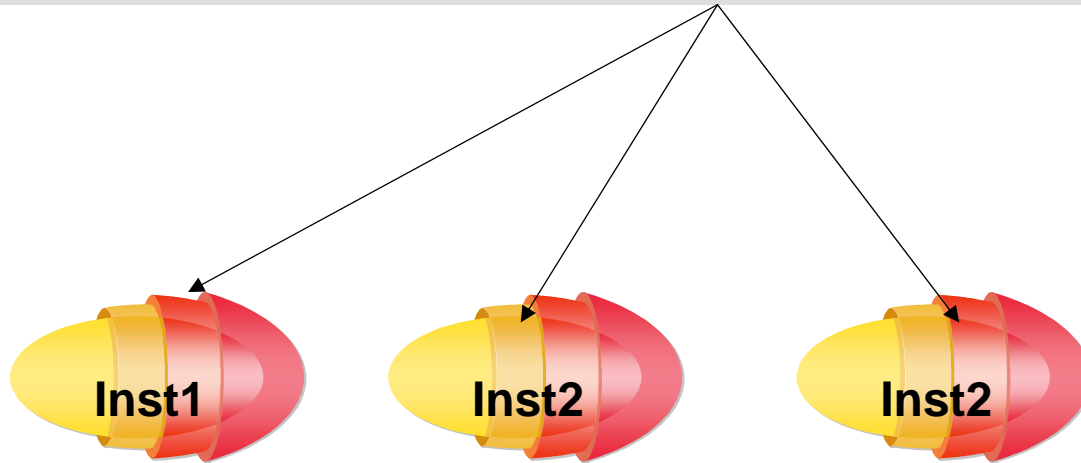
BAdIs are represented by a reference to BAdI-Handles:

```
DATA bd type ref to badi_name.  
GET BADI bd FILTER f = 5.
```

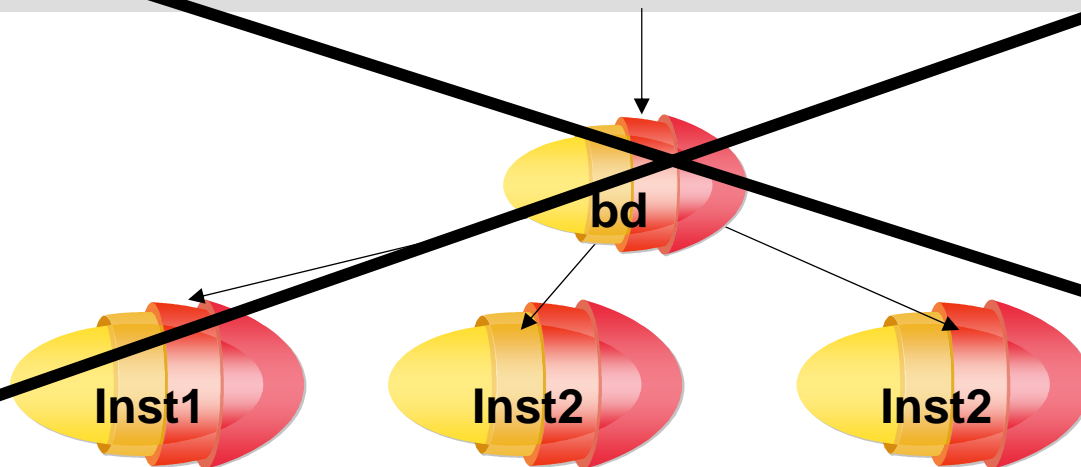
If there are two implementations of badi_name that are selected for the filter value f=5, this yields:

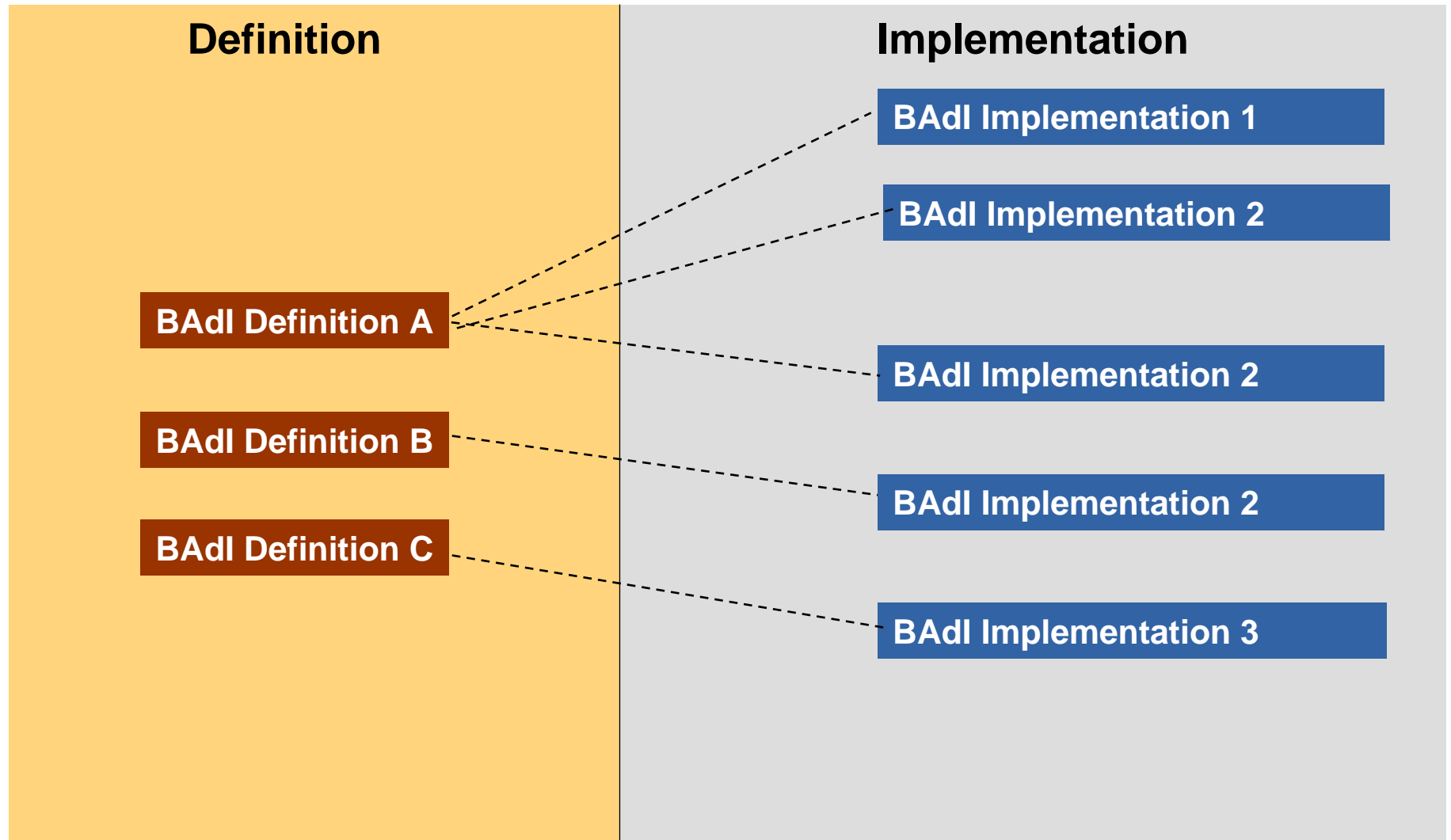


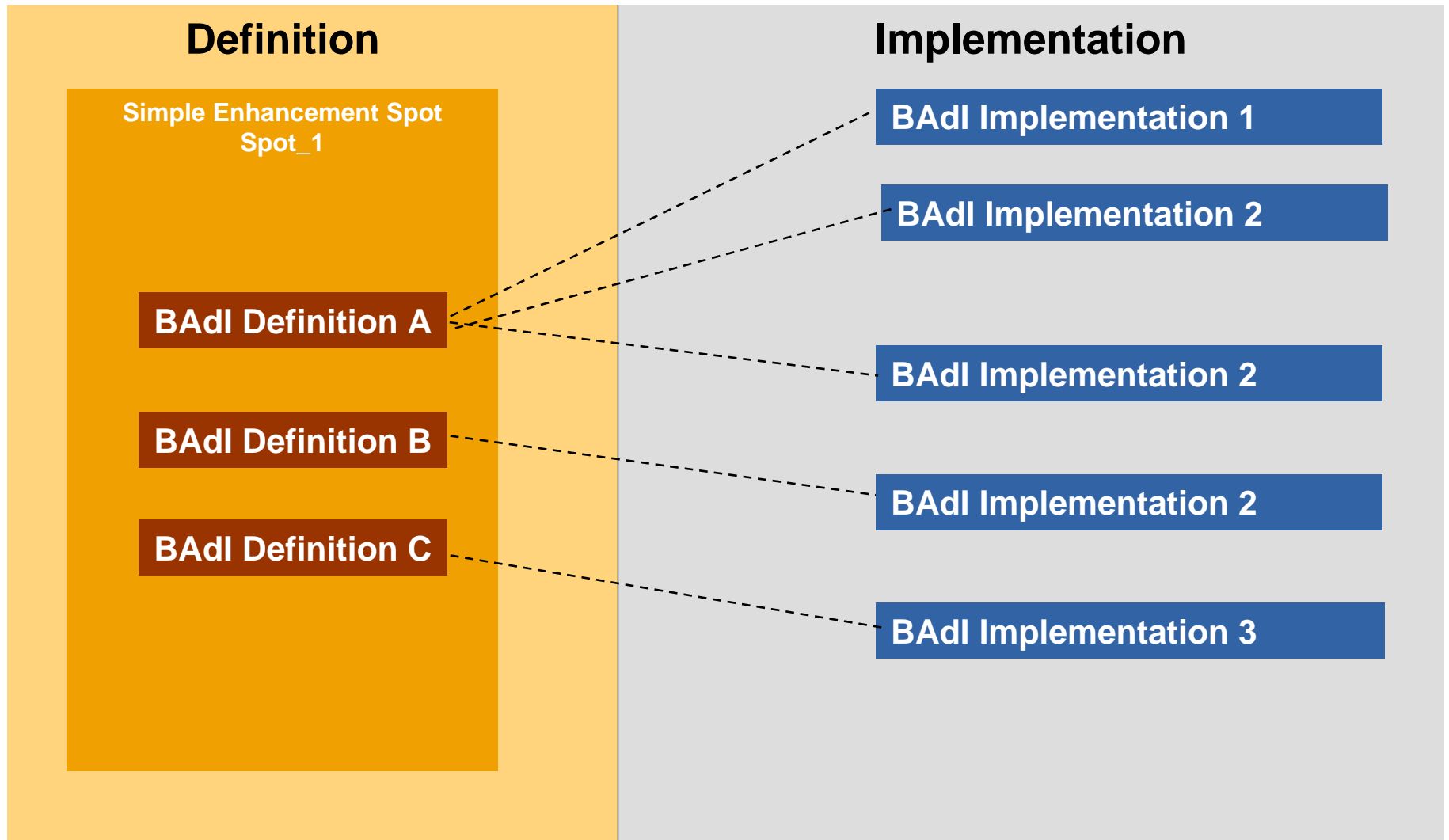
CALL BADI bd->m exporting ... importing ...

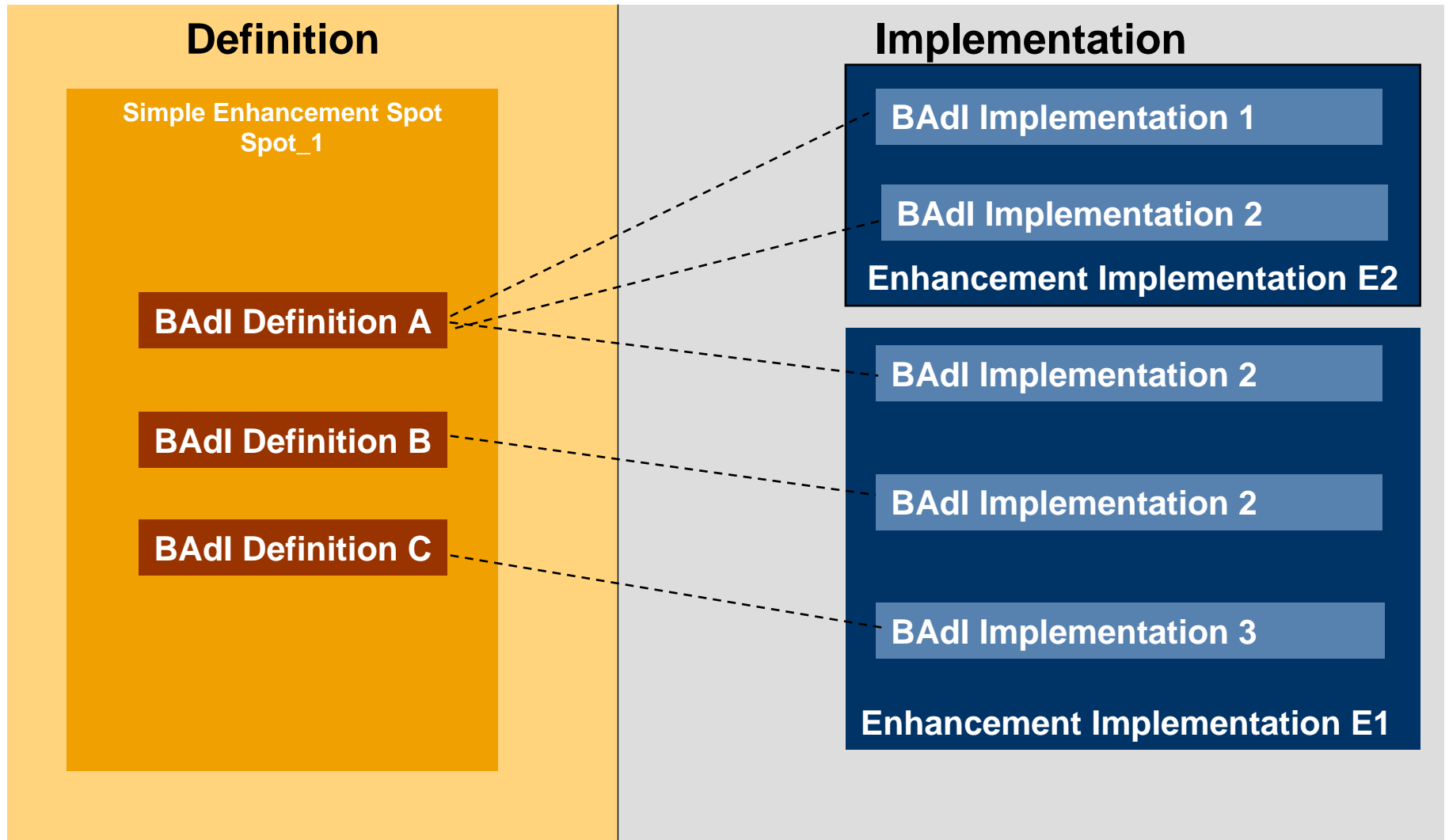


~~**CALL BADI bd->m exporting ... importing ...**~~









Definition

Simple Enhancement Spot
Spot_1

BAdI BADI_A

BAdI BADI_B

BAdI BADI_C

Implementation

BAdI Implementation 1

Simple Enhancement Implementation SEI1

BAdI Implementation 2

Package A

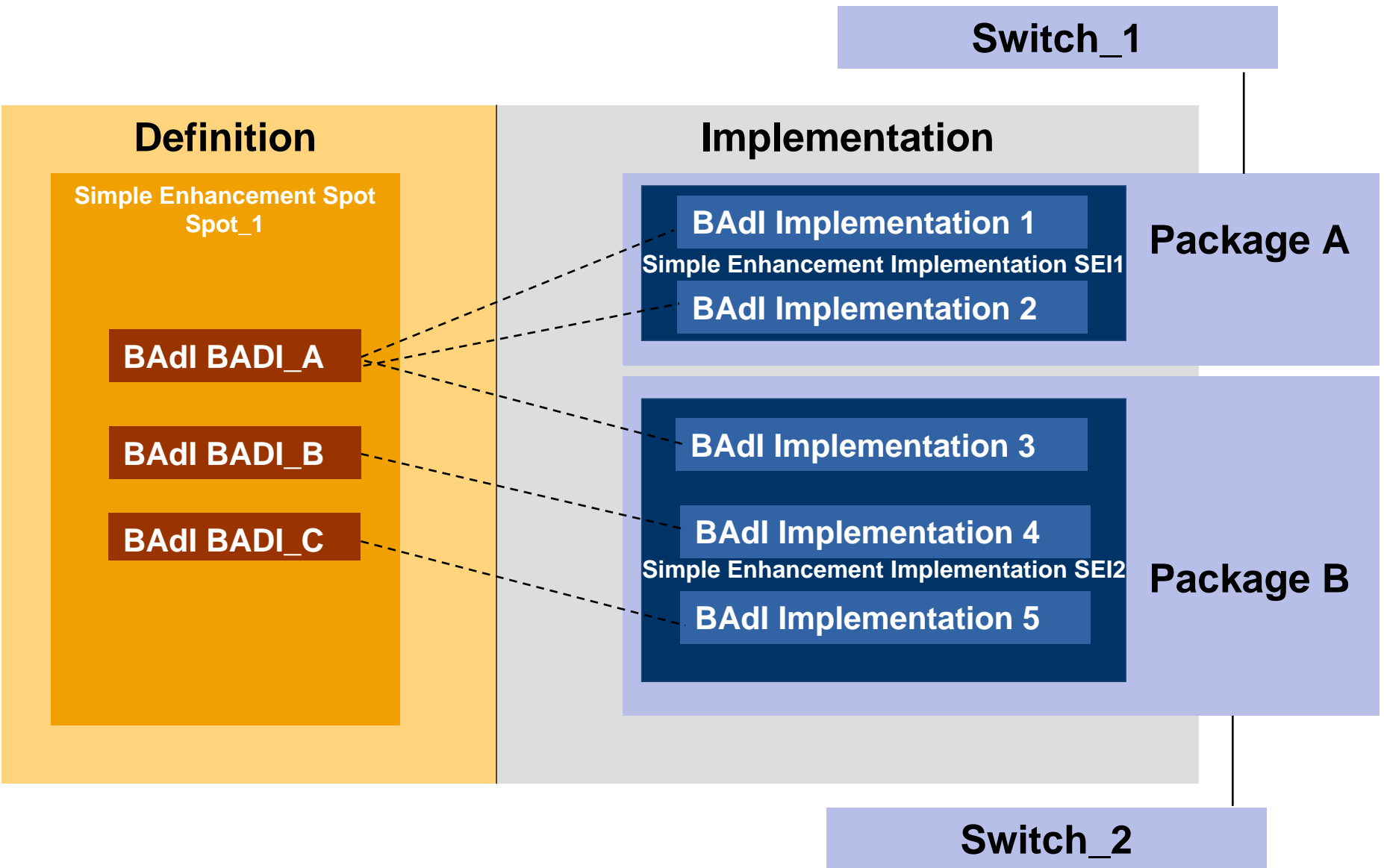
BAdI Implementation 3

BAdI Implementation 4

Simple Enhancement Implementation SEI2

BAdI Implementation 5

Package B



Definition

Simple Enhancement Spot
Spot_1

BAdI BADI_A

BAdI BADI_B

BAdI BADI_C

Simple Enhancement Spot
Spot_2

BAdI BADI_E

Implementation

BAdI Implementation 1

Simple Enhancement Implementation SEI1

BAdI Implementation 2

BAdI Implementation 3

BAdI Implementation 4

Simple Enhancement Implementation SEI2

BAdI Implementation 5

BAdI Implementation 6

Types of Implementations

- **A BAdI-Definition may have an associated fallback class**
- **A BAdI-Implementation is either default or non-default**

Selection Procedure: (during GET BADI)

- 1. Apply selection to all non-default implementations**
- 2. if nothing has been selected, apply selection to all default implementations**
- 3. if still nothing has been selected and there is a fallback class, take the default class**

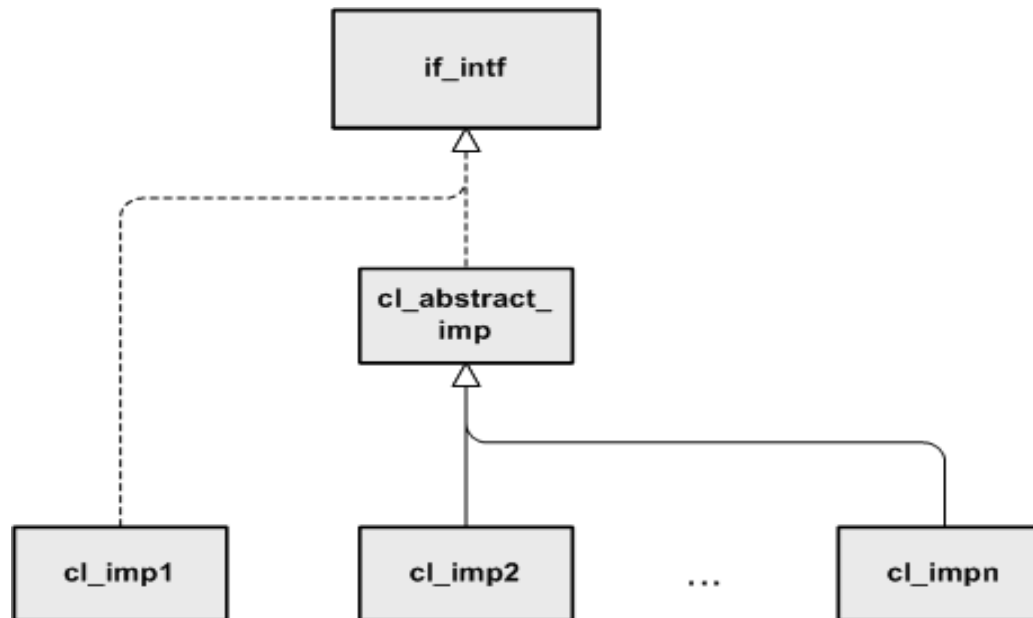
Implementation Inheritance

Together with the BAdI, abstract or normal implementation classes may be supplied

cl_imp1 has to implement all interface methods

cl_imp2, ... implement methods not supplied by cl_abstract_imp or override some methods

It is also allowed to inherit from the default class or example class, if they are not defined as final

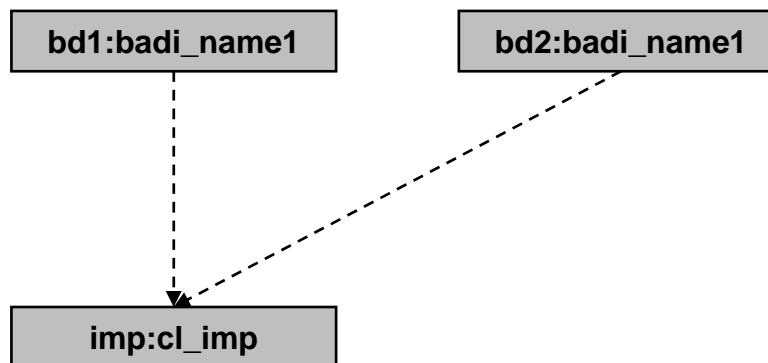
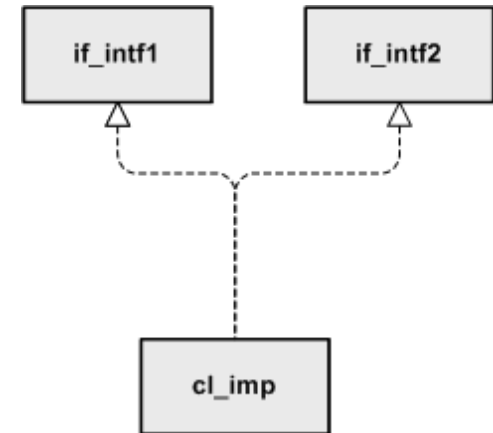


There are two BAdIs `badi_name1` and `badi_name2`, with interfaces `if_intf1` and `if_intf2`, respectively

`cl_imp` implements both of these interfaces

```
DATA: bd1 TYPE ref to badi_name1,  
      bd2 TYPE ref to badi_name2.
```

```
GET BADI bd1.  
GET BADI bd2.
```



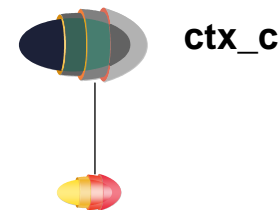
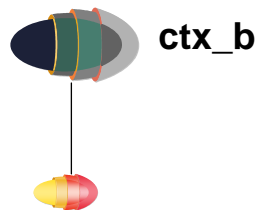
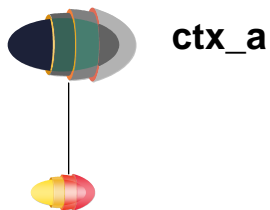
ever new instances are created (Scenario 1)



per implementation class there is one instance (Scenario 2)

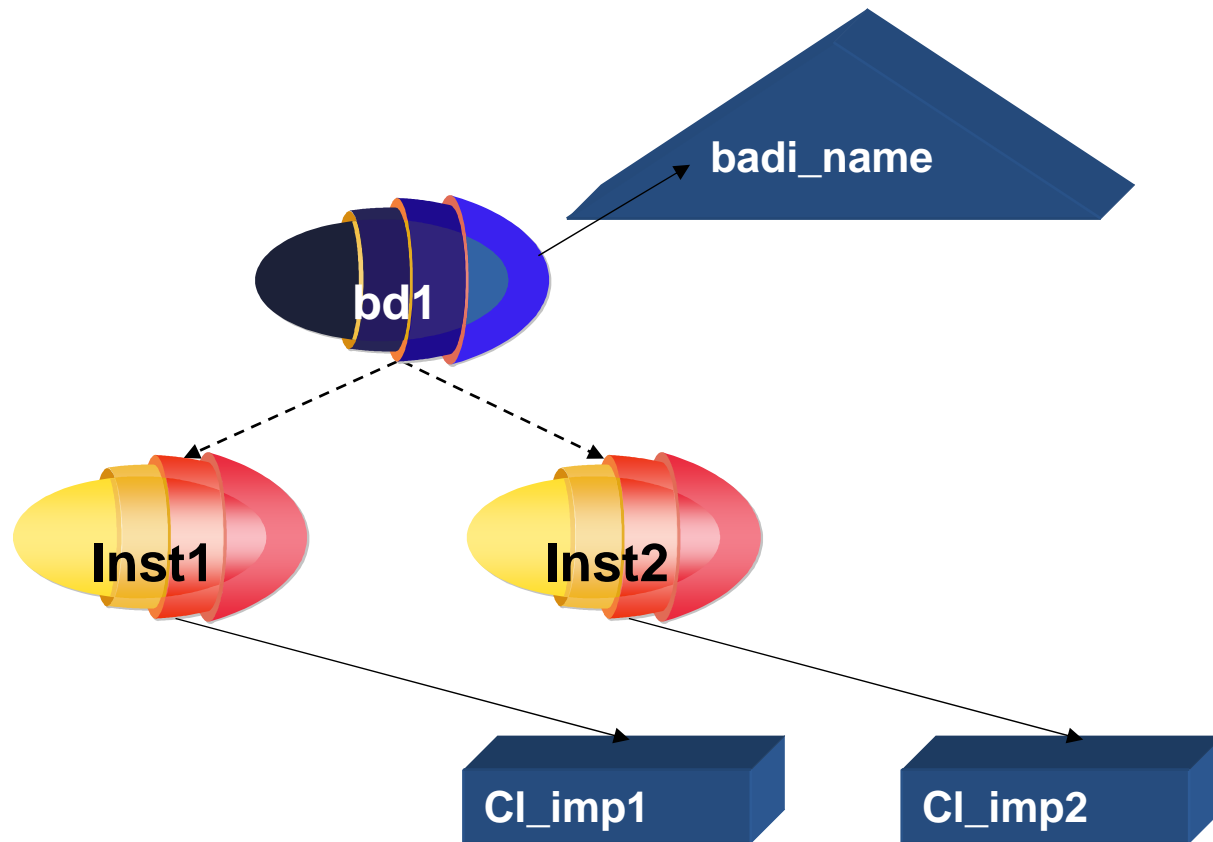


per *context reference* and implementation class there is one instance



DATA bd TYPE ref to badi_name.

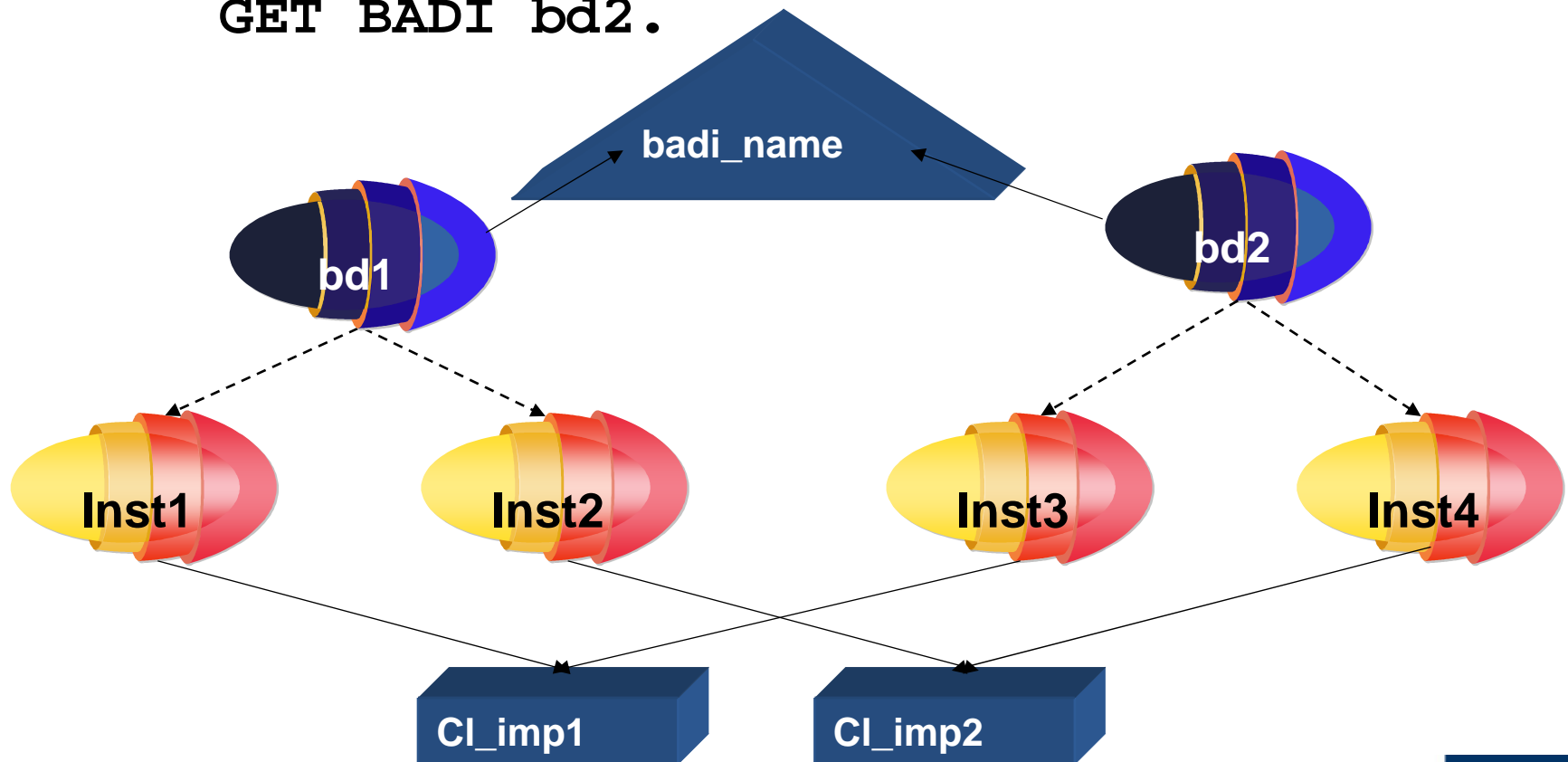
GET BADI bd1.



```
DATA: bd1 TYPE ref to badi_name,  
      bd2 TYPE ref to badi_name.
```

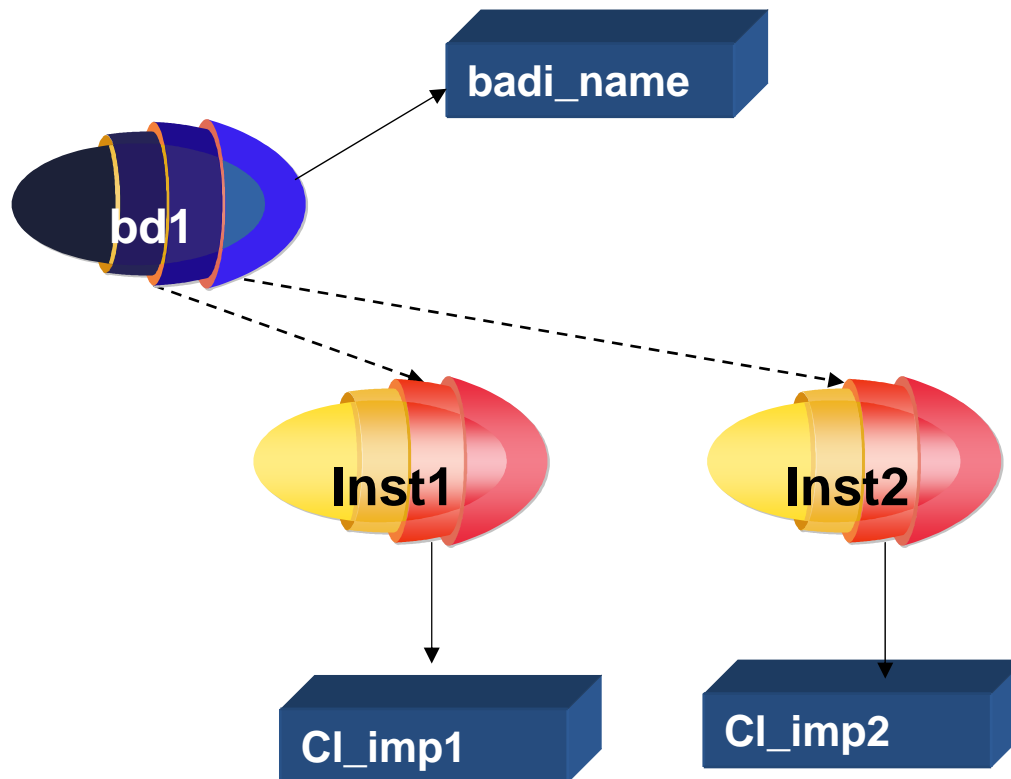
```
GET BADI bd1.
```

```
GET BADI bd2.
```



DATA: bd1 TYPE ref to badi_name.

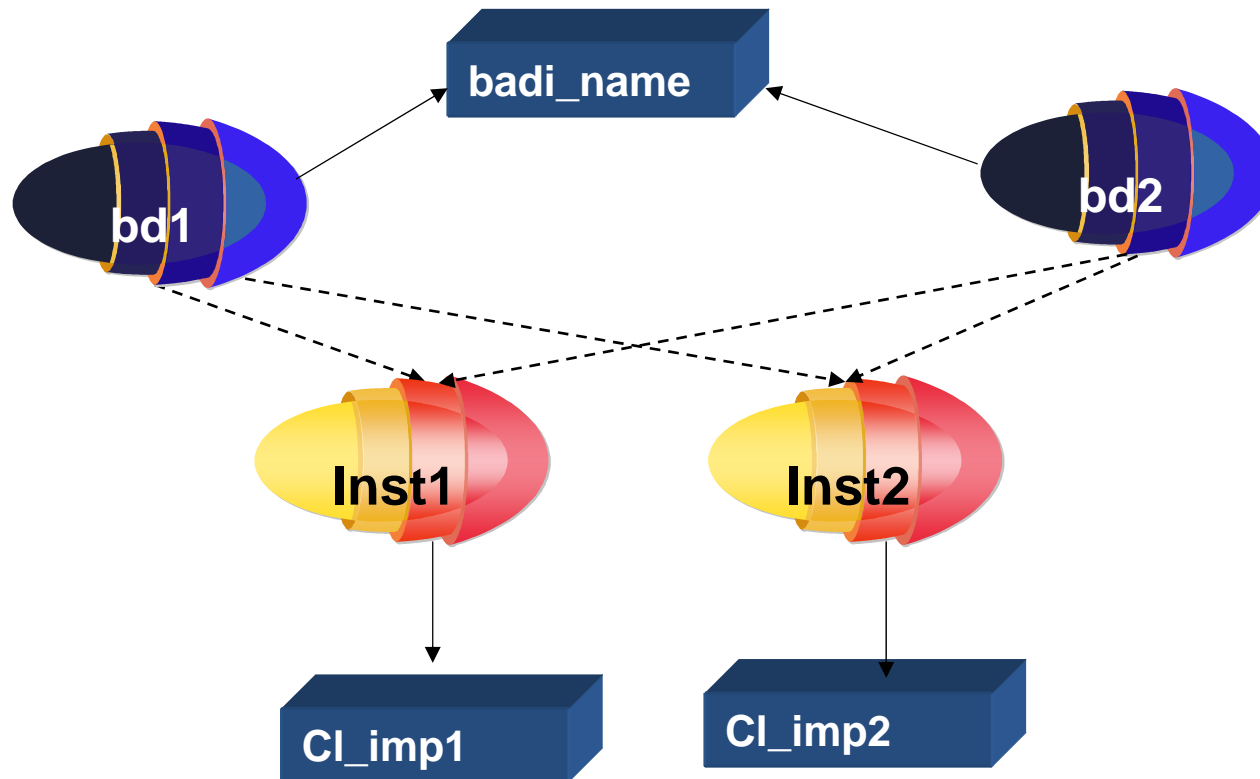
GET BADI bd1.



```
DATA: bd1 TYPE ref to badi_name,  
      bd2 TYPE ref to badi_name.
```

```
GET BADI bd1.
```

```
GET BADI bd2.
```



With a context there is one instance per *context reference* and implementation class.

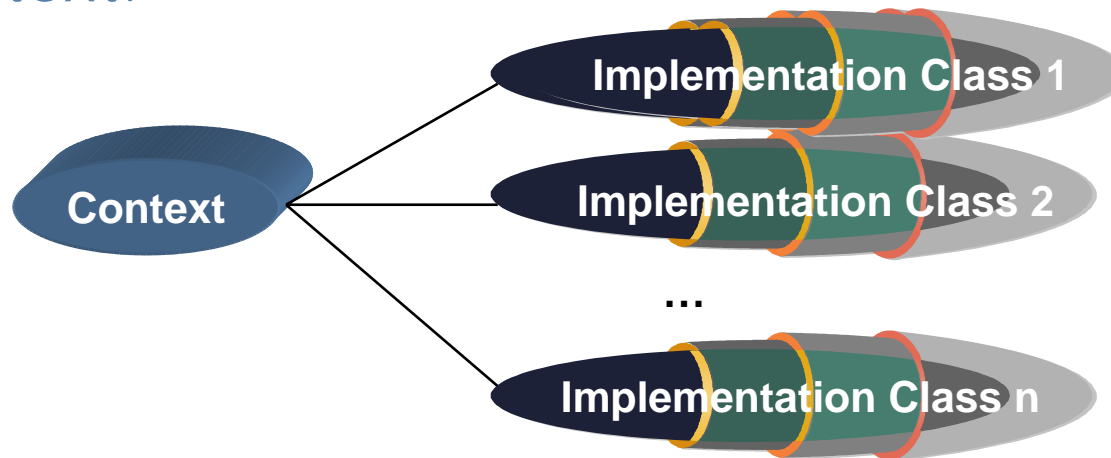
A *context class* is a class implementing `if_badi_context`, a *context reference* is a reference to an instance of a context class

In ABAP:

```
DATA ctx TYPE REF TO cl_myctx.  
...  
GET BADI bd CONTEXT ctx.
```

Passing the same context yields identical implementation instances

BAdI Implementations assign themselves to a given BAdI Context:

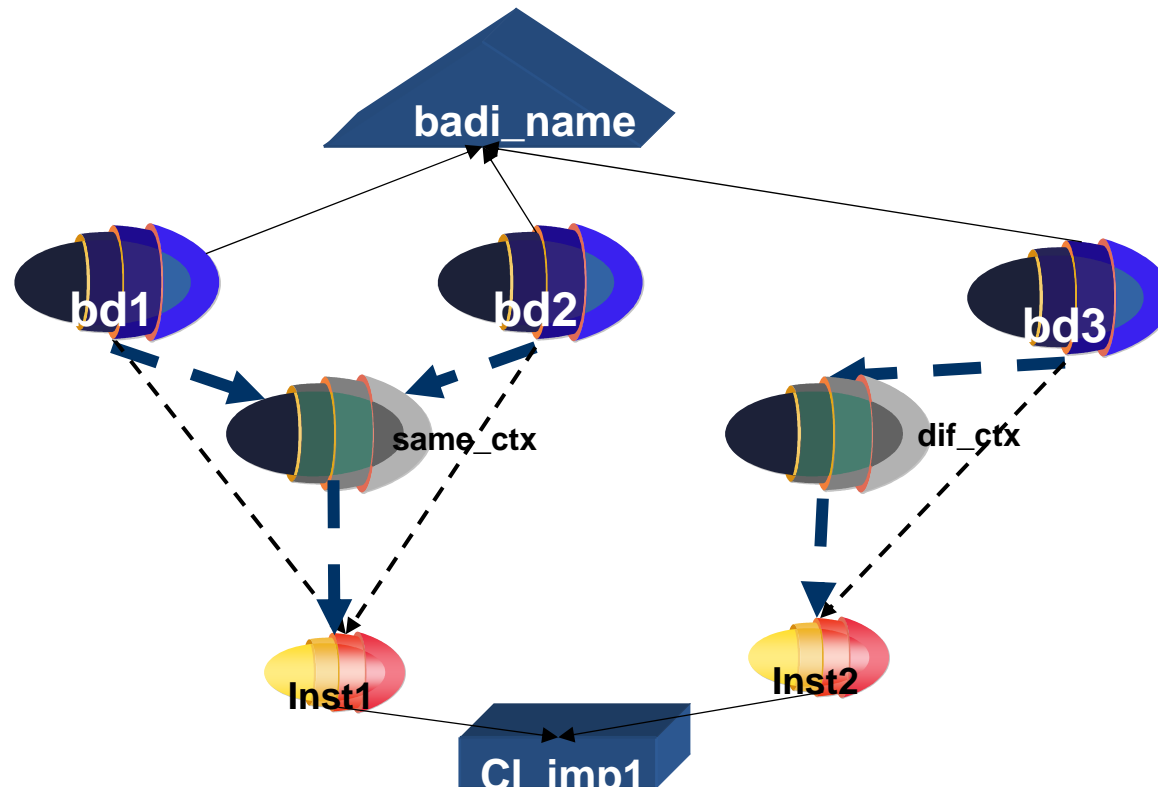


- Lifetime of implementations is bound to the lifetime of the context.
- Whenever 'GET BADI' is called with the same context and a filter value which leads to the same implementation implementation class, **the already created instance is used.**
- Performance improvement
- Holds data over different method calls or even BAdIs.

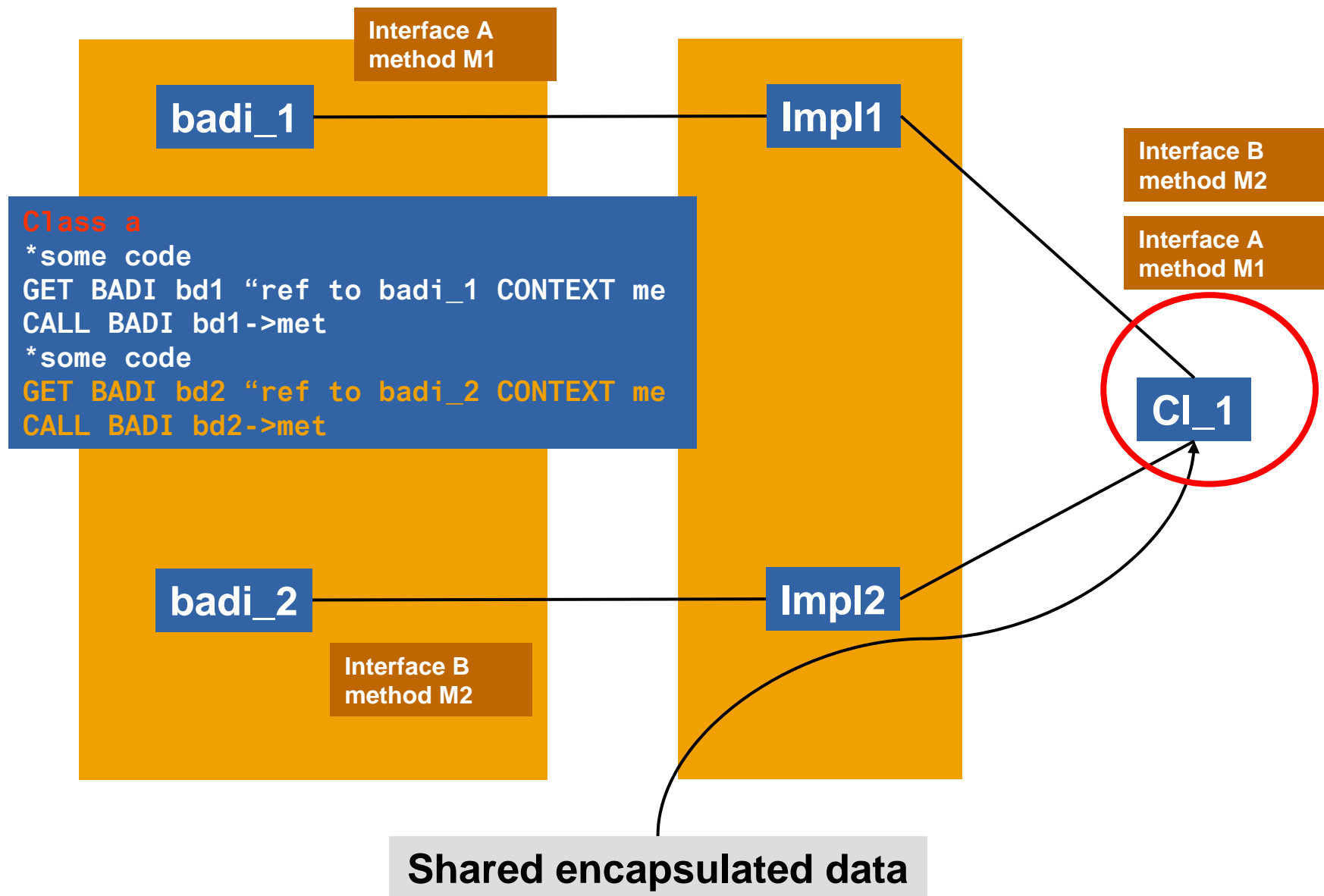
Instantiations With Context

DATA: bd1 TYPE ref to badi_name,
 bd2 TYPE ref to badi_name,
 bd3 TYPE ref to badi_name.

GET BADI bd1 CONTEXT same_ctx.
GET BADI bd2 CONTEXT same_ctx.
GET BADI bd3 CONTEXT dif_ctx.



Encapsulating Data for Two BAdIs Within One Class



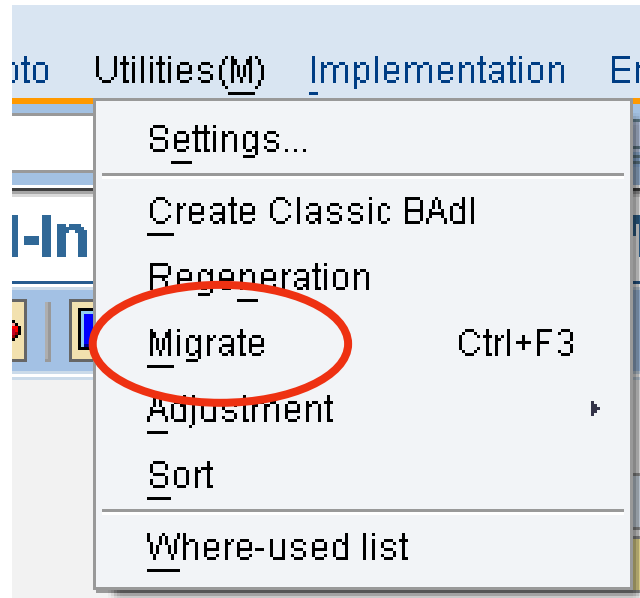
Automatic migration by selecting utilities → migration from BAdI Builder (SE18)

- Specify Enhancement Spot for BAdI Definition
- Specify Enhancement Implementation for BAdI Implementation

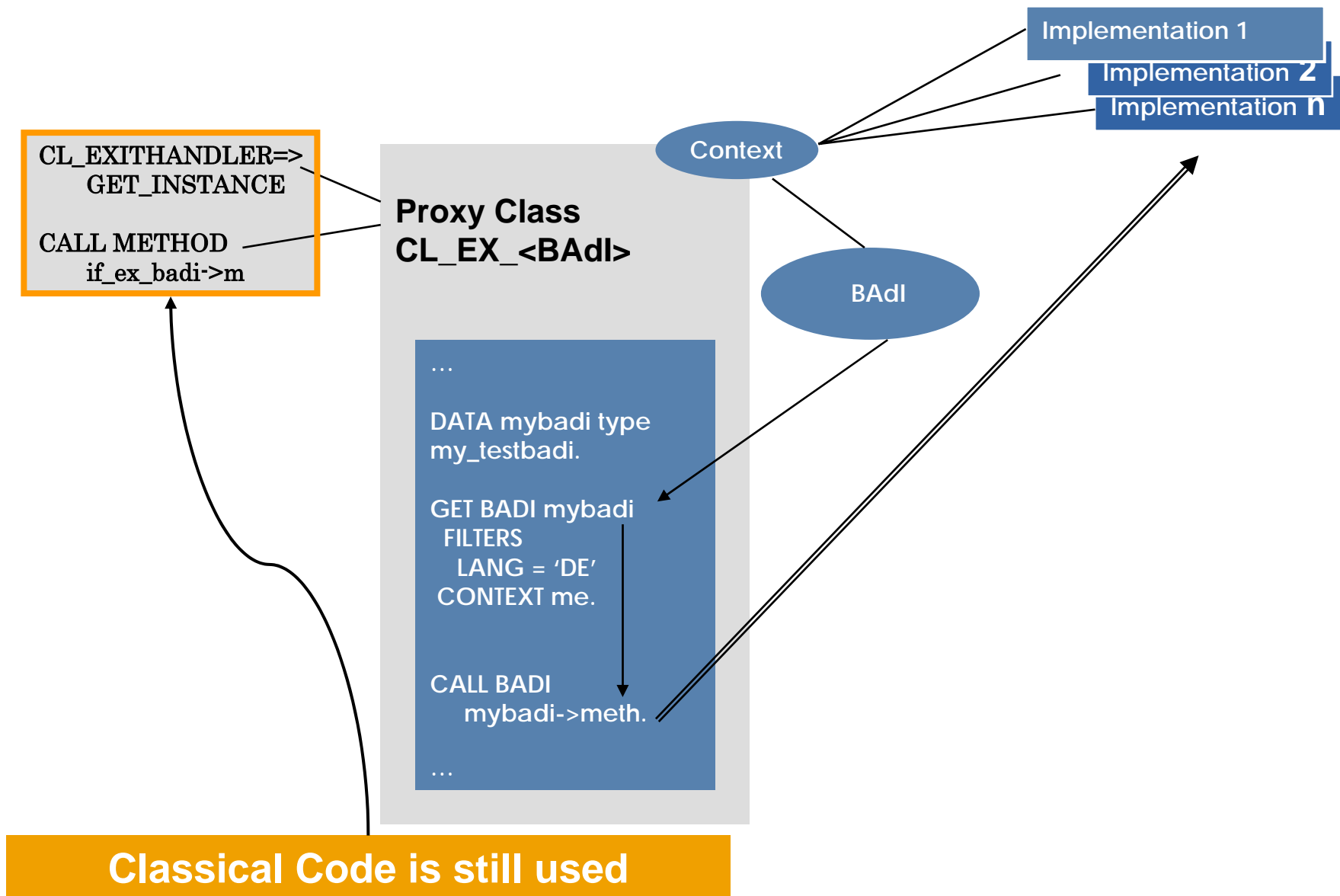
→ no special knowledge necessary

→ effort: 5 minutes per BAdI.

Automatic migration by selecting utilities → migration from BAdI Builder (SE18)



Automatic (Partial) BAdI Migration

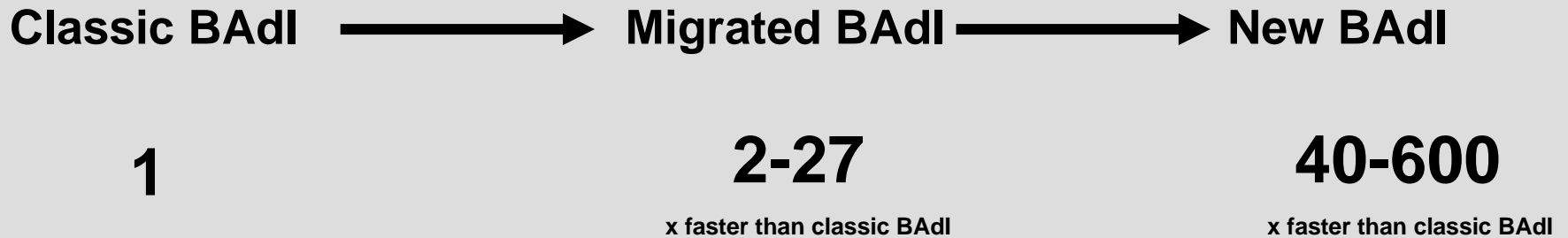


Complete migration:

- Delete the classic BAdI.
- Change Context Settings if you need no context.
- Find all calls of the classic BAdI by GET_INSTANCE and reprogram the BAdI call using the new commands 'GET BADI' and 'CALL BADI'.
- For BAdIs that are called more than once in one program the context settings may be changed to improve the performance.

→ Expert knowledge of the application necessary

→ Effort: from some minutes up to some days per BAdI



A BAdI call is ca. 7.5% slower than a method call!

The more implementations defined, the higher is the improvement on performance

Normally BAdI definition and implementations are defined in different systems.



- The after import method of a BAdI definition writes 'Call transaction SPAU' in the transport log.
- After import or upgrade perform a manual migration of BAdI implementations by using transaction SPAU.

Syntax: bd is either a concrete BAdI-Handle or a generic BAdI-Handle (i.e. TYPE REF TO CL_BADI_BASE)

```
GET BADI { badi [FILTERS f1 = x1 f2 = x2 ...] }  
    | { badi TYPE (name)  
        [ {FILTERS f1 = x1 f2 = x2 ...}  
          | {FILTER-TABLE ftab}} ] }  
[CONTEXT con].
```

```
CALL BADI { badi->meth parameter_list }  
    | { badi->(meth_name) {parameter_list  
        |parameter_tables} }.
```

BAdI is restricted to exactly one filter and the implementation only allows '=' and 'OR' in the filter definition.

Use case

1. BAdIs used in Frameworks often have only one key to select BAdI implementations e.g.: ESI

→ A special implementation for these BAdIs will allow to keep efficiency even when thousands of BAdI-Implementations for one BAdI exist in a system

2. You want to program a registry application. Some registered code is called depending on a registry key.

→ Reuse the BAdI Framework for a fast registry implementation including upgrade support. A method for evaluation all implemented filter values is:

`CL_ENH_BADI_RUNTIME_FUNCTIONS=>GET_IMPLS_FOR_LIMITED_BADI(name)`



DEMO



Enhancement Framework Overview

Source Code Plugin – Technology

Function Group Enhancement – Technology

Class Enhancement – Technology

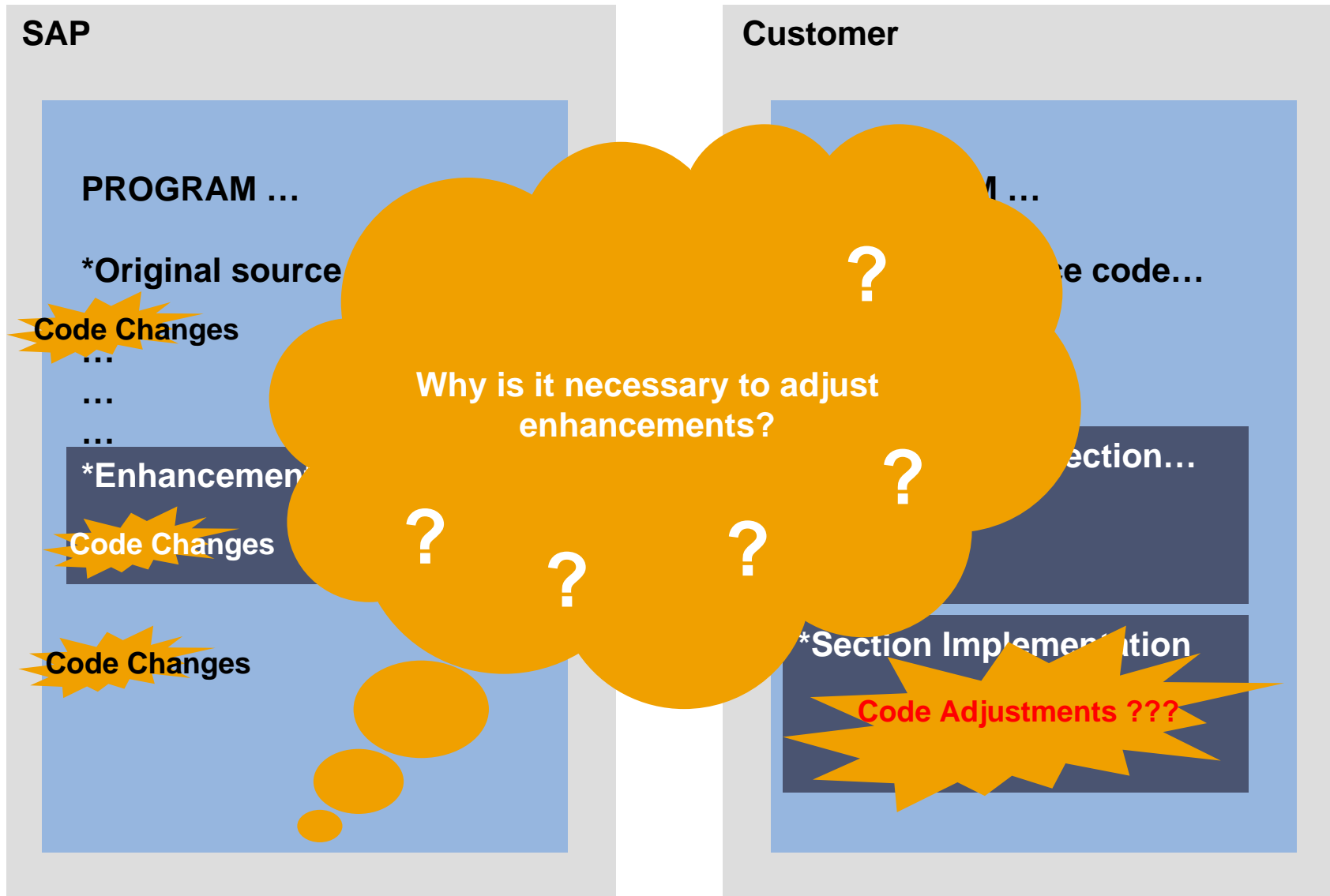
BAdI – Technology

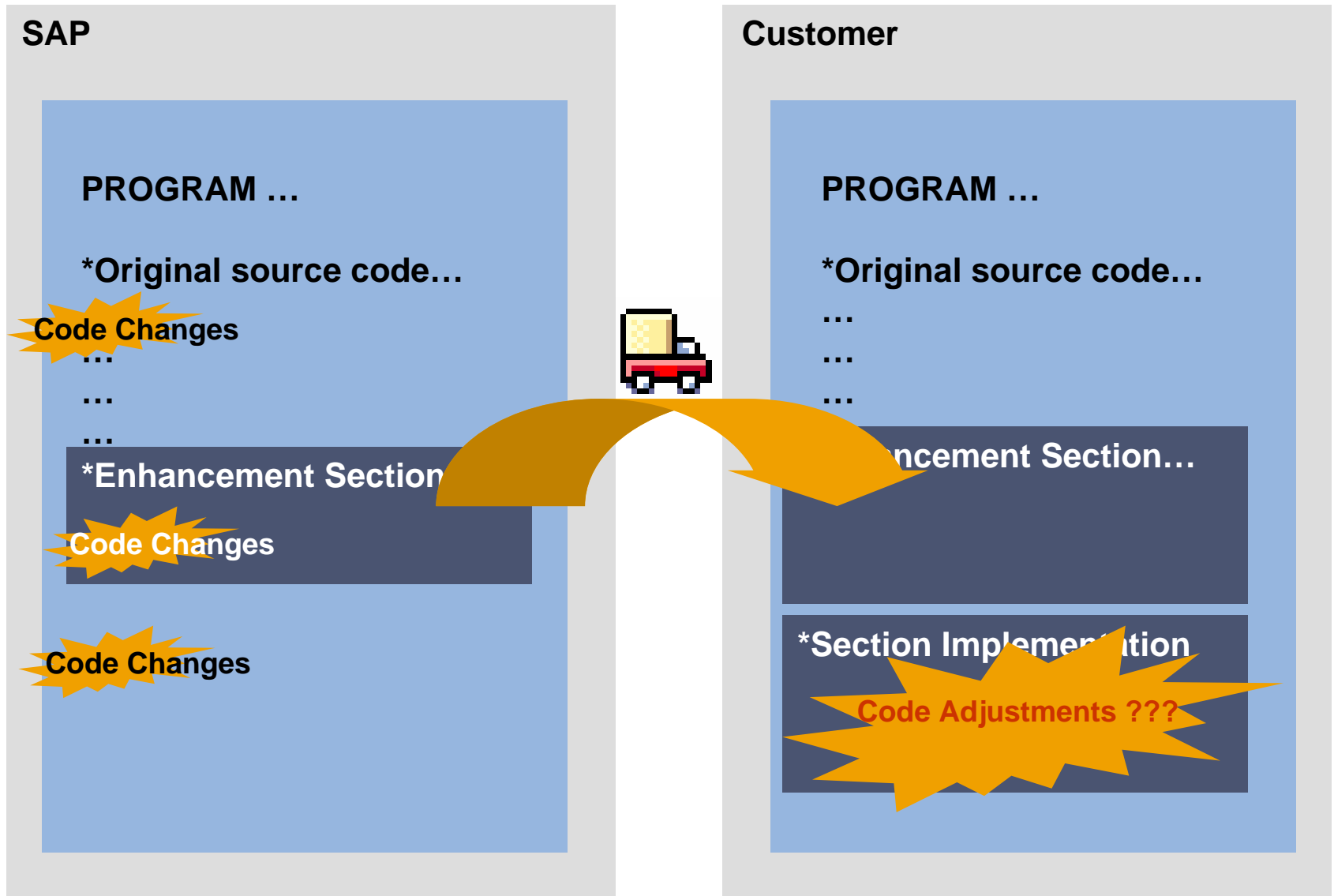
Upgrade Adjustment

Switch Framework

Summary

Reasons for Upgrade Support I





ABAP Source code with enhancements

- **The standard source code between ENHANCEMENT-SECTION and END-ENHANCEMENT-SECTION has changed.**

Function modules with enhancements

- **A new importing parameter was declared that has the same name as an importing parameter introduced by an enhancement.**

Classes and interfaces with enhancements

- **An enhanced method was deleted.**

BAdIs

- **BAdI interface was changed or a filter was deleted**

How to Recognize Necessary Adjustments?

Import Log

Start Import LIMUREPSCD_TEST_ENH_UPGRADE2 ...


! 1 conflict with enhancement objects occurred (ENHOBJ PROG CD_TEST_ENH_UPGRADE2) - please call transaction SPAU_ENH
1 0





Editor

Enhancement Implementation: Adjustment Information

Enhancement Implementation: MDS_BUPA_LOCK

Adjustment Status: OO

 The enhancement implementation is in adjustment mode. Before the enhancement implementation can be edited, it must be adjusted.

  Adjust  



SPAU_ENH

✓ ISH_LV61A536

Composite Enh. Impl. w/o Pa

ICWM/COMPOSITE

- ✓ ICWM/APPL_MM_RMVCON CWM_RM
- ✓ ICWM/APPL_MM_RVBBIN
- ICWM/APPL_MM_SALMBG
- OO ICWM/APPL_MM_RM07
- ✓ ICWM/APPL_MM_RM07MS
- ✓ ICWM/APPL_MM_RM07ML
- OO ICWM/APPL_MM_SAPLI
- OO ICWM/APPL_MM_SAPLI/ICWM/APP



 + 

SPAU_ENH

1. Double-Click on the Enhancement Implementation you want to adjust
2. Switch to change mode

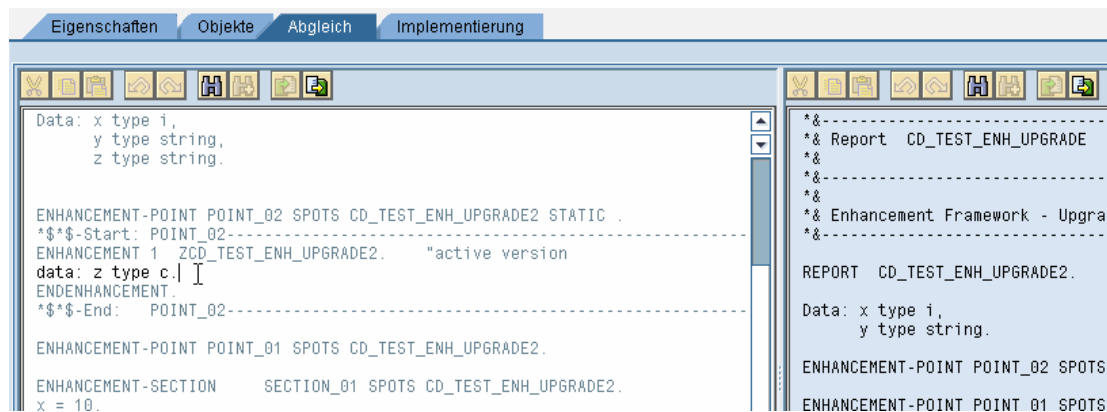


3. Go to tab „Adjustment“
4. Double-Click on list entry to adjust it


Status	Enhancement Fullname	Program name	Includename
	\PR:CD_TEST_ENH_UPGRADE2\EX:POINT_02\EI	CD_TEST_ENH_UPGRADE2	CD_TEST_ENH_UPC
	\PR:CD_TEST_ENH_UPGRADE2\EX:SECTION_01\EI	CD_TEST_ENH_UPGRADE2	CD_TEST_ENH_UPC

SPA_U_ENH

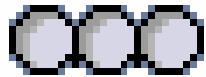
5. Adjust Enhancement, e.g. in Splitscreen Editor



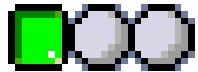
6. Set adjustment status to 

7. Press Button  Adjust Enhancement Implementation

8. Activate Enhancement 



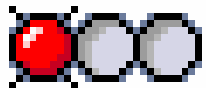
Semantic Changes



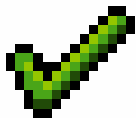
Automatic Adjustment



Tool-Aided Adjustment



Manual Adjustment



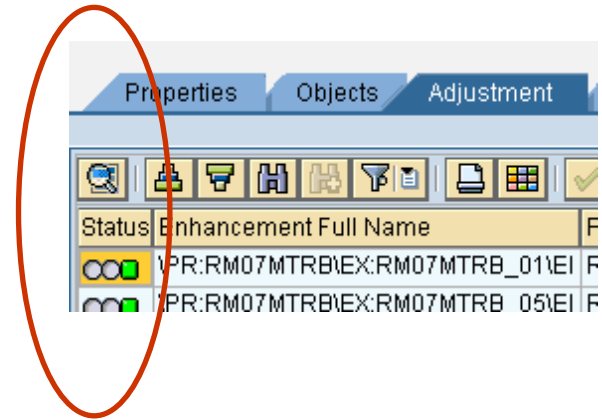
Enhancement is adjusted



Enhanced Object was deleted



Implementation is empty





DEMO



Enhancement Framework Overview

Source Code Plugin – Technology

Function Group Enhancement – Technology

Class Enhancement – Technology

BAdI – Technology

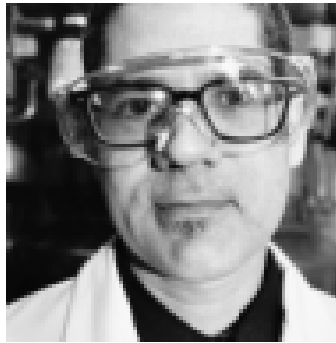
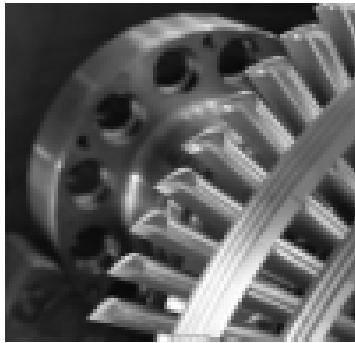
Upgrade Adjustment

Switch Framework

Summary

- **Shipment delay compared to core 6-12 months**
- **Latest technology stack can not be utilized**
- **Missing release synchronization leads to upgrade delay and implementation hurdles**
- **Delayed provision of legal requirements**
- **Industries are implemented as modifications**
- **No reuse of functionality of other Industry Solutions**

Goals and Benefits



Goals and Benefits

- Reduce **TCO** for **customer**
- **Timely** provision of **legal requirements**
- **Synchronization** of release cycles and release planning
- Boost the **attractiveness of ERP** by providing already existing industry solutions to the entire SAP-Community
- **Speed-up** the introduction of SAP **NetWeaver**

Synergies of Industry Solutions in mySAP ERP 2005

Exclusiv Activation	Multiple Activation
SAP ECC Industry Extension Healthcare 6.0	SAP ECC Enterprise Extension Consumer Products 6.0
SAP ECC Industry Extension Chemicals 6.0	SAP ECC Enterprise Extension Defense Forces & Public Security 6.0
SAP ECC Industry Extension Contract Accounting 6.0	SAP ECC Enterprise Extension Financials 6.0
SAP ECC Industry Extension Discrete Industries & Mill Products 6.0	SAP ECC Enterprise Extension FERC: Regulatory Reporting 6.0
SAP ECC Industry Extension Insurance 6.0	SAP ECC Enterprise Extension Financial Services 6.0
SAP ECC Industry Extension Media 6.0	SAP ECC Enterprise Extension Global Trade 6.0
SAP ECC Industry Extension Mining 6.0	SAP ECC Enterprise Extension Human Capital Management 6.0
SAP ECC Industry Extension Oil & Gas 6.0	SAP ECC Enterprise Extension Incentive & Commission Management 6.0
SAP ECC Industry Extension Public Services 6.0	SAP ECC Enterprise Extension Industry-specific Sales Enhancements 6.0
SAP ECC Industry Extension Retail 6.0	SAP ECC Enterprise Extension Joint Venture Accounting 6.0
SAP ECC Industry Extension Telecommunications 6.0	SAP ECC Enterprise Extension PLM 6.0
SAP ECC Industry Extension Utilities, Waste & Recycling 6.0	SAP ECC Enterprise Extension Public Sector Management 6.0
SAP ECC Industry Extension Catch Weight Management 6.0	SAP ECC Enterprise Extension Retail 6.0
	SAP ECC Enterprise Extension SCM 6.0
	SAP ECC Enterprise Extension Travel Management 6.0



Single Activation



Multiple Usage

Switches control the visibility of repository objects at runtime

Benefits:

- **Industry Solutions are available with every release and SP without delay (i.e. timely provision of legal requirements), CRT's are no longer necessary for add-on systems**
- **Industry Solutions can be enriched by generic functions from other industries**
- **Synchronization of release cycles and planning**

Switchable Objects...

...by package assignment

- **Appends, SI-, CI-cludes for dictionary structures**
- **Fixed value appends to domains**
- **Secondary Indexes**
- **Append Search Helps**
- **Enhancement Implementations**
- **Switch Business Configuration Sets (Switch BC-Sets)**

How to Determine if an Object is Switched?

Dictionary: Display Extension Index

Navigation icons: Back, Forward, Edit, Copy, Paste, Print, etc.

Index Name	MARA	M02	Switched Off	Switch	ISM_MEDIA_BASIS_MODIF
Short description	Übergeordnete Medienproduktfamilie				
Last changed	WENZLER	04.02.2005	Original language	DE	German
Status	New	Saved	Package	JCOREMOD	

⚠ Index does not exist in database system DB6

Dictionary: Display Append Structure

Navigation icons: Back, Forward, Edit, Copy, Paste, Print, etc.

Hierarchy Display Append Structure...

Append Structure:	ISH_MARA_APP	New	Switched Off
Short Description	IS-H MM: Hospital-Specific Fields		

Attributes Components Entry help/check Currency/quantity fields

Last changed on/by	GROTHUS	20.01.2005
Package	NBAS	Appl. development Hospital System
Original language	DE	German
Appending table	MARA	
Switch	ISH_MAIN	⚠ SAP Patient Management

Switchable Objects ...

... by direct assignment

- **Screen elements & Flow logic**
- **Menu entries & functions**
- **IMG nodes**
- **Customizing**

Switch dependent module

Screen Painter: Display Screen for SAPLCOK01

Screen number: 250 Active

Attributes | Element list | Flow logic

MODULE HEADER_POSITION_FILL.
* Ändern nach mitgegebener Struktur
MODULE SET_CHANGES_DARK.
* Fertigungssteuerungsprofiltext füllen
MODULE GET_SFPCPF_TEXT.
* Linienhierarchie und -text füllen
MODULE GET_LINEHIER_TEXT.

* TDP: Handling type Text lesen
MODULE OIH_LOOKUP_OIH5T_HEADER SWITCH OIH_EXCISE_DUTY.

PROCESS AFTER INPUT.

* Validation of Handling type
FIELD CAUFVD-OIHANTYP
MODULE OIH_VALIDATE_HANTYP ON REQUEST. "SWITCH OIH_EXCISE_DUTY.

* Abbrechen
MODULE PROCESSING_EXIT AT EXIT-COMMAND.

* Include: Bunter Header

Li 7, Co 18 Ln 7 - Ln 28 of 38 lines

Screen Painter – Element List

Switch dependent screen elements in the element list – disabled elements are not visible at runtime.

Screen Painter: Display Screen for SAPLCOK01

Screen number: 250 Active

Attributes | Element list | Flow logic

General attr. | Texts/ I/O templates | Special attr. | Display attr. | Mod. groups / functions | References

H	M Name	Type	Gr	Gr	Gr	Switch	Reaction to Sw	Entry	Valu	Input	Function
	CAUFVD-SLSBS	Text						<input type="checkbox"/>			
	CAUFVD-SLSBS	I/O						<input type="checkbox"/>			
	CAUFVD-SBMEH	Text						<input type="checkbox"/>			
	CAUFVD-SBMEH	I/O						<input checked="" type="checkbox"/>			
	CAUFVD-SBMNG	Text						<input type="checkbox"/>			
	CAUFVD-SBMNG	I/O						<input type="checkbox"/>			
	CAUFVD-AUFLD	Text						<input type="checkbox"/>			
	CAUFVD-AUFLD	I/O						<input checked="" type="checkbox"/>			
	OIH_FRAME01	Box				OIH_EXCISE_DUTY	D Display	<input type="checkbox"/>			
	CAUFVD-OIHANTYP	Text				OIH_EXCISE_DUTY	D Display	<input type="checkbox"/>			
	CAUFVD-OIHANTYP	I/O				OIH_EXCISE_DUTY	D Display	<input checked="" type="checkbox"/>			
	OIH5T-TEXT1	I/O				OIH_EXCISE_DUTY	D Display	<input type="checkbox"/>			

Switch dependent menu entries

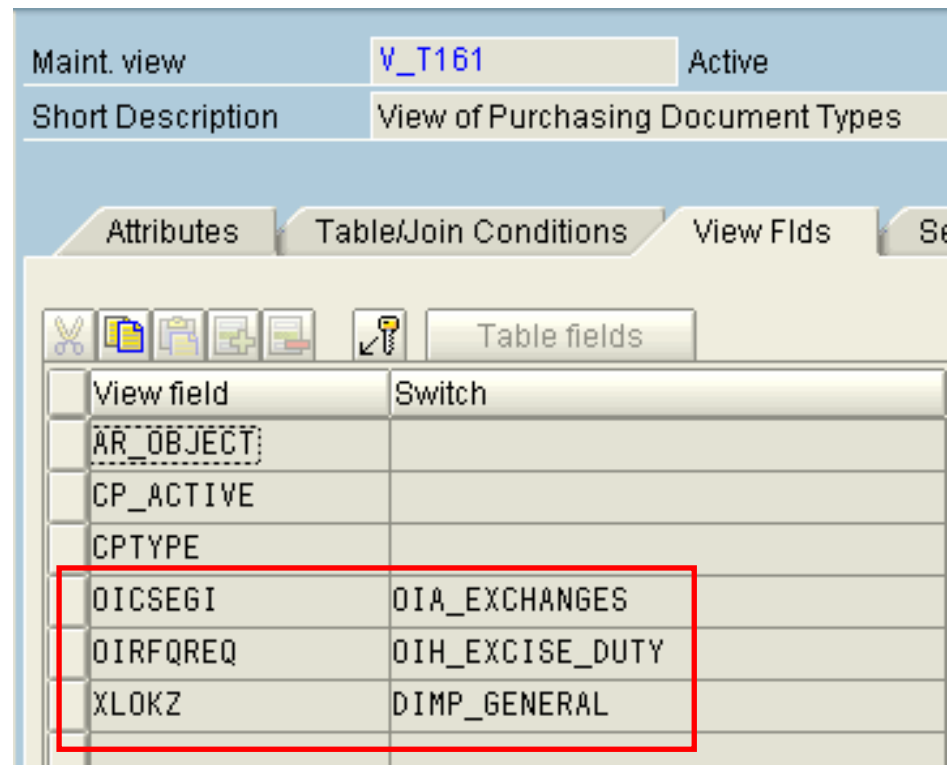
The screenshot displays the SAP Menu Painter interface. On the left, a sidebar contains a list of repository objects: MIME Repository, Repository Browser, Repository Information System, Tag Browser, Transport Organizer, and Test Repository. Below this is a 'Program' field set to 'ZPROGRAM' and a toolbar with navigation icons.

The main workspace is divided into two panes. The top pane, titled 'User Interface ZPROGRAM Active', shows a hierarchical menu structure. It includes a 'Menu Bar' and an 'Application Toolbar', both containing a function code 'ft'. Below these is a grid of menu items organized into three groups: 'Items 1 - 7', 'Items 8 - 14', and 'Items 15 - 21'. The 'Items 1 - 7' group contains a row of icons: NEXP (a right-pointing triangle), DL (a trash can), TXP (a document with a pencil), DETD (a truck), AFP (a flag), and AS (a bar chart). The 'Items 8 - 14' group contains a row of icons: LB (a document with a plus sign), DIEN (a document with a magnifying glass), and LIM (a document with a clock). The 'Items 15 - 21' group is currently empty.

The bottom pane, titled 'Function Attributes', is used to configure the selected function code 'NEXP'. It shows the 'Function Code' as 'NEXP' and the 'Functional Type' as 'Application Function'. A red rectangle highlights the 'Schalter' (Switch) field, which is set to 'FAZEEL_SWITCH', and the 'Reaktion' (Reaction) field, which is set to 'X'. Below this, the 'Static Function Texts' section contains fields for 'Function Text' (Next item), 'Icon Name' (ICON_COLUMN_RIGHT), 'Icon Text', 'Info. Text', and 'Fastpath' (N). At the bottom of the pane are green and red checkmark buttons.

Maintenance Views

The visibility of the fields of a table control is determined by the switch state.



The screenshot shows the SAP Maintenance View for V_T161, titled 'View of Purchasing Document Types'. The 'View Flds' tab is selected. A table lists fields and their corresponding switches. A red rectangle highlights the last three rows of the table.

View field	Switch
AR_OBJECT	
CP_ACTIVE	
CPTYPE	
OICSEGI	OIA_EXCHANGES
OIRFQREQ	OIH_EXCISE_DUTY
XLOKZ	DIMP_GENERAL

The visibility of a view within a view cluster can be controlled by switches.

Change View "Object structure": Overview

New Entries

Dialog Structure

- Header entry
- Object structure
- Field dependence
- Events

View cluster: CO_RESS_KALKSCH

Field-dependence

Object structure

View/Table	DD...	Short text	Predecess.	D...	P...	Start	Backgr	n:m	Schalter
V_T683_COR		Costing Sheets	V_T683_COR	R	1	<input checked="" type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	
V_T683S_CR		Costing Sheet Rows	V_T683_COR	S	2	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	
V_T685_COR		Costing Sheet Rows	V_T683S_CR	S	3	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
V_KONR		Resource Prices	V_T683S_CR	S	4	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	DI0_COMMON

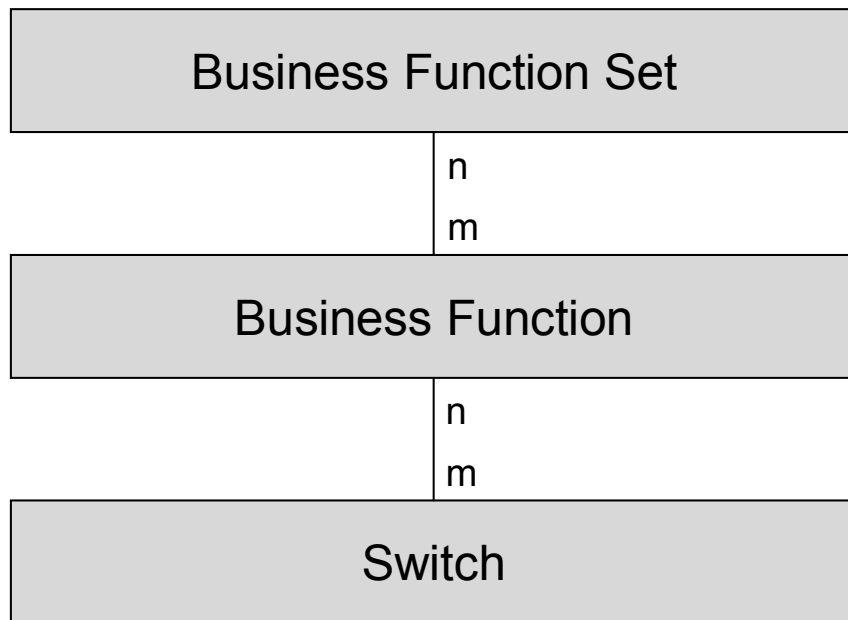
Switch dependent import of table content via BC-Sets

■ BC-Sets improvements:

- Performance improvements
- Support of deletion of customizing entries
- Tool improvements like automatic recording, switch dependent definition and browsing

Switch dependent IMG nodes

■ IMG nodes of disabled industries are not visible



Business Function Set

- **Pool of business functions**
- **Represents one industry solution**
- **Max. 1 can be active**
- **Use Transaction SFW5 to switch a on BFS.**
- **Transaction SFW3**

Business Function

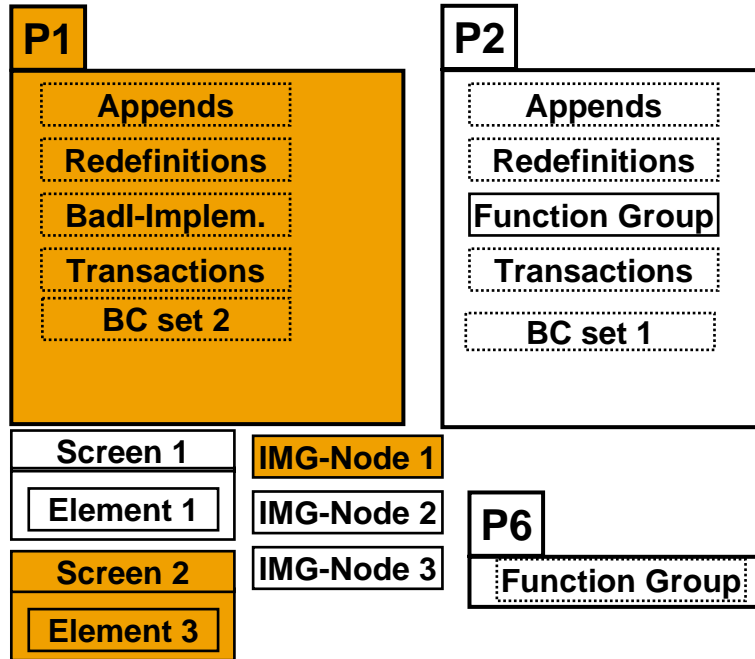
- **Represents a piece of business functionality**
- **Contains switches**
- **Transaction SFW2**

Switch

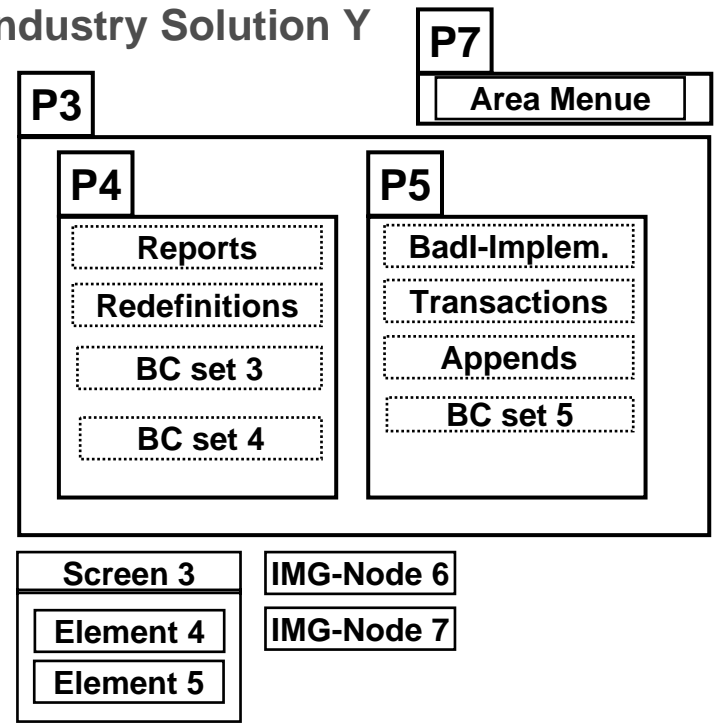
- **Calculated states: ON, OFF, STANDBY**
- **Transaction SFW1**

Mapping Switches and Architecture I

Industry Solution X



Industry Solution Y

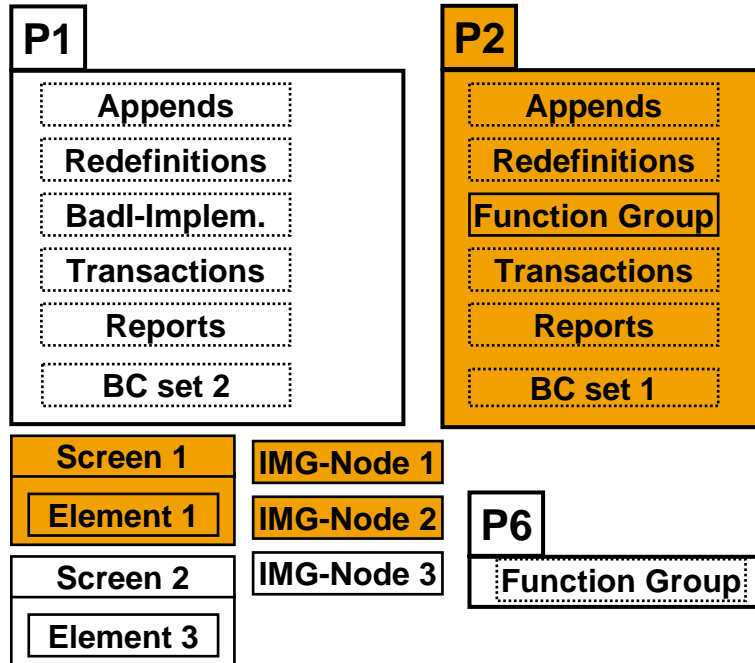


Bus Fct Set 1

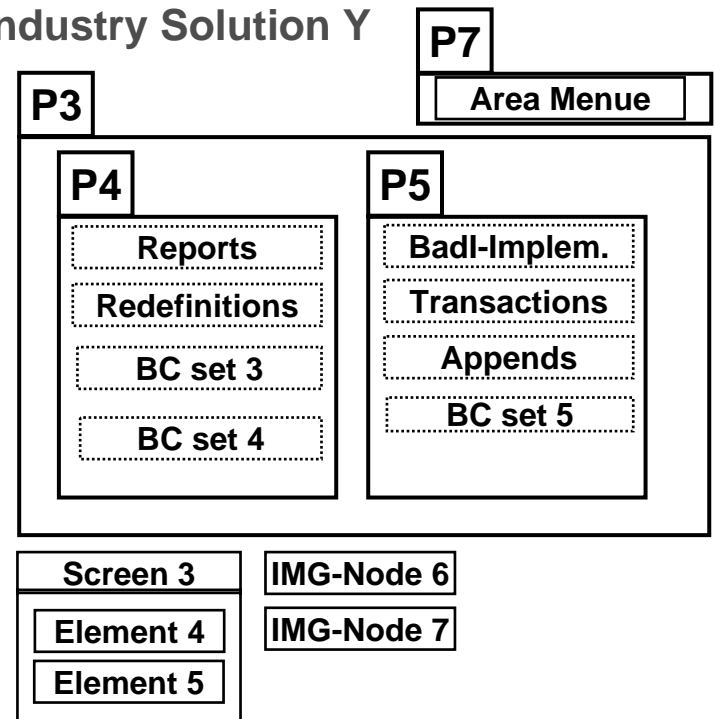
Bus Fct F1
Bus Fct F2
Bus Fct F4

Mapping Switches and Architecture II

Industry Solution X



Industry Solution Y



Bus Fct Set 1

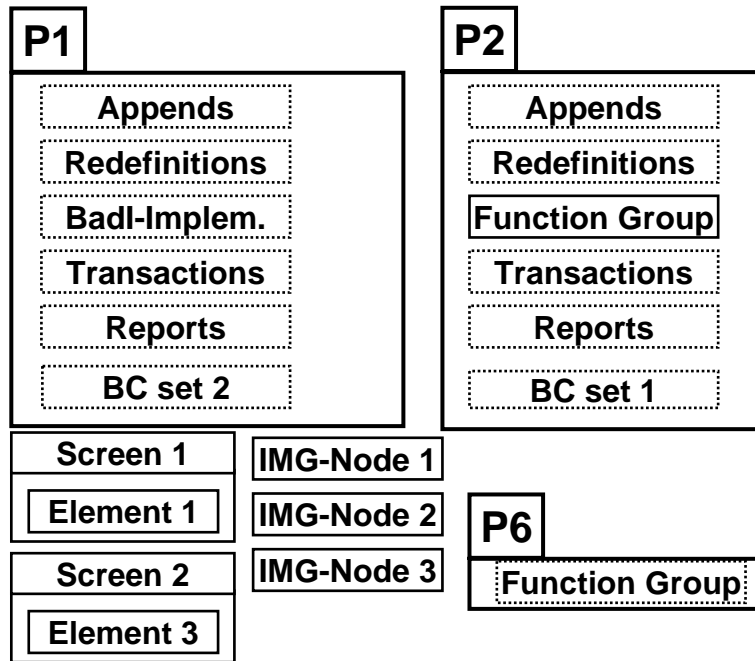
Bus Fct F1

Bus Fct F2

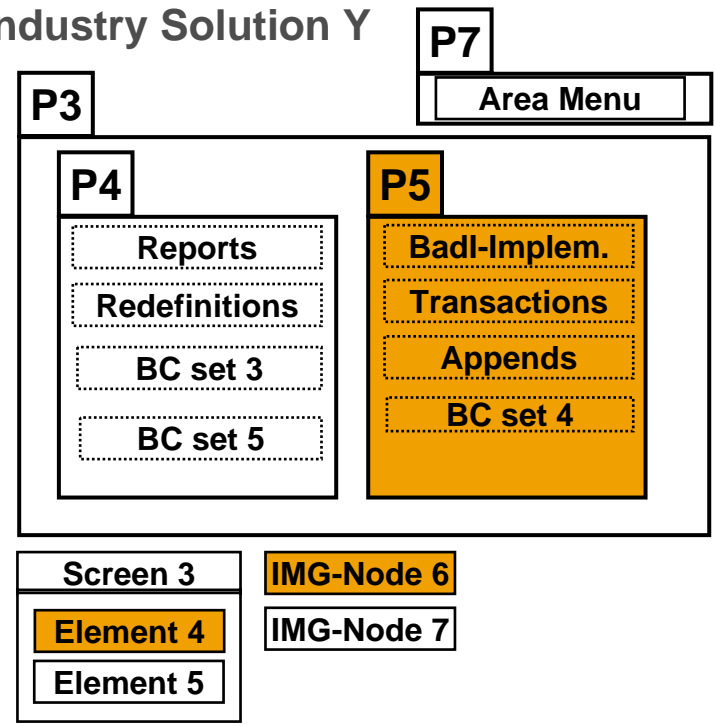
Bus Fct F4

Mapping Switches and Architecture III

Industry Solution X



Industry Solution Y



Bus Fct Set 1

Bus Fct F1
Bus Fct F2
Bus Fct F4

Log on to the Development System with authorization SAP_ALL

Disable all batch jobs

Close the system for all other users

Transaction SFW5

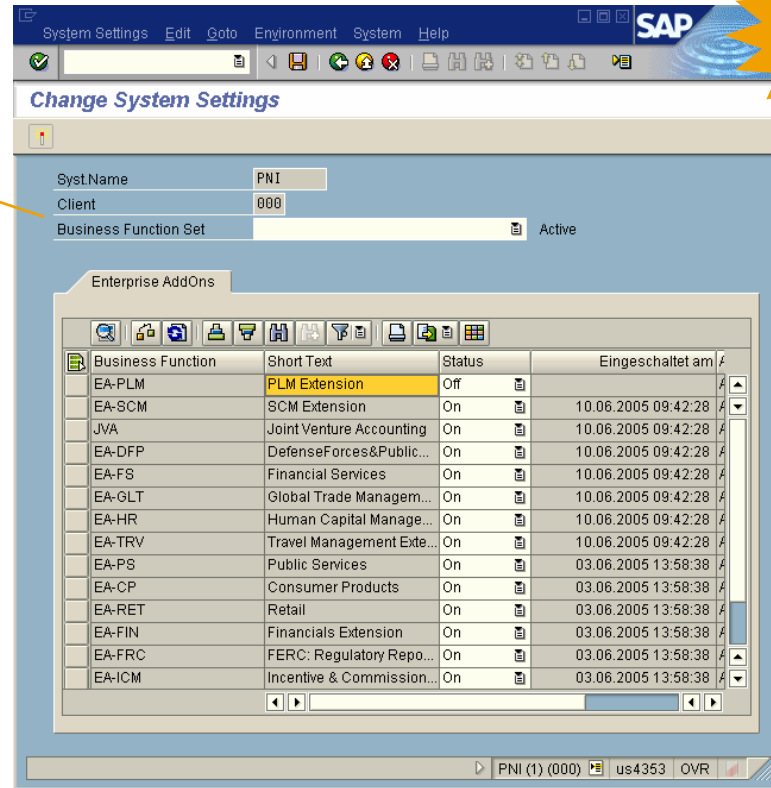
Choose the BFS in the drop down box

Select the required Business Functions that belong to the chosen BFS

Click on the button “activate”

Choosing the Business Function Set (BFS)

I. Select the BFS



Activating the Business Function Set

SFW5

II. Choose the Business Functions

III. Press activate

Systemeinstellungen Springen Umfeld System Help

Activate Business Set

Syst.name: ANI
Client: 005

Business Function Set: SAP Oil and Gas Active

SAP Oil and Gas Enterprise AddOns

Business Function	Short Text	Status	AV
COMMODITY_MGMT_&...	Commodity Mgmt & Bulk...	On	Active
ISR_RETAILSYSTEM	Retail System	Off	Inactive
OIL_PRODUCTION	Oil Production	On	Active
OIL_REVENUE_ACCO...	Oil Revenue Accounting	On	Active
REMOTE_LOGISTICS_...	Remote Logistics Mana...	On	Active

ANI (1) (005) us4121 INS



DEMO

- **The Batch Job for DDIC and BC-Set activation starts immediately.**
- **This job takes about 30-120 min**
- **The background activities are comparable to installing an Add On manually**
- **Reschedule batch jobs in the development system and unlock users**

Caution:
Only one Business
Function Set can be
activated

Caution:
There is no possibility,
to deactivate a BFS

Enhancements for the objects of the Switch Framework

■ **Switches**

Packages can be added by enhancements. Beneficial for distributed development of switchable solutions. This enhancement is not switchable itself. When switching on a switch, all enhancements of this switch will be considered.

■ **Business Functions**

Switches can be added by enhancements.

■ **Business Function Sets**

Business Functions can be added by enhancements. Thus it is possible for ISVs to add their functionality to SAP Business Function Sets without modifying the original object.

■ **Switching off Business Functions**

For test purposes only



Enhancement Framework Overview

Source Code Plugin – Technology

Function Group Enhancement – Technology

Class Enhancement – Technology

BAdI – Technology

Upgrade Adjustment

Switch Framework

Summary

- **The Enhancement Framework offers new possibilities to extend the SAP Standard instead of modifying it.**
 - **Source Code PlugIns**
 - **Function Group Enhancements**
 - **Class Enhancements**
 - **New BAdIs**
- **The new BAdIs are more flexible and faster than the classic ones.**
- **The Enhancements offered by Enhancement Framework and several other object types can be switched by the Switch Framework.**



Public Web

<http://help.sap.com> → Documentation → SAP Netweaver → Application Platform → ABAP technology → ABAP Workbench → Enhancement Framework

<http://sdn.sap.com> → Weblogs by author Thomas Weiss



THANK YOU FOR YOUR
ATTENTION !

QUESTIONS – SUGGESTIONS – DISCUSSION

- No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.
 - Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.
 - Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.
 - IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.
 - Oracle is a registered trademark of Oracle Corporation.
 - UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.
 - Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.
 - HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.
 - Java is a registered trademark of Sun Microsystems, Inc.
 - JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.
 - MaxDB is a trademark of MySQL AB, Sweden.
 - SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.
-
- The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP AG.
 - This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice.
 - SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.
 - SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.
 - The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages.