



Enhancement Framework - Class Enhancements - Pre-exit, Post-exit and Overwrite-exit methods - Concept and Simple Scenarios



Anonymous
Feb 26, 2009

Watch

Enhancement Framework → Class Enhancements → Pre-exit, Post-exit and Overwrite-exit methods

1 Class Enhancements (Klasse Verbesserungen) -

We can take the advantage of enhancing the global parameters and interface of a class with the help of the new enhancement framework to achieve the expected business functionality.

What things are possible when we can think of enhancing the features or functionality of a class?

- Adding new methods to the existing ones.
- Adding new, but optional global parameters to the existing method parameters.
- Enhancing (Customer defined business functionality) the way the methods are triggered or executed with the help of Pre-exit, Post-exit and Overwrite methods respectively with customer logic.

The best way to utilize the functionality would be the combined use of enhancing the global parameters as well as applying the 'Pre-Post-Overwrite exit' methods.

In Short,

1 Pre-exit methods -

Pre-exit methods are the enhanced methods which can be invoked at runtime before the original method code is supposed to be executed or triggered.

A pre-exit can be utilized for certain validations or passing some data from the pre-exit to the original method.

Note: In addition to the pre-exit some source code enhancements can be well used to adapt the changes from the enhanced 'pre-exit' method to the original method.

1 Post-exit methods -

Post-exit methods are the enhanced methods which can be invoked at runtime after the original method code has been executed or triggered.

The post-exit can be used to add or enhance the data flowing from the original method and satisfy the custom business functionality.

1 Overwrite-exit methods -

Overwrite-exit methods are the enhanced methods which act as a replacement at run time for the original method.

[Open in app](#)

s a replacement at run time for the original

Point to correlate -

If the things are observed closely then it reveals that the 'Pre-exit' and 'Post-exit' methods act like 'Enhancement-point' as in source code enhancements.

Also, the 'overwrite-exit' method acts like 'Enhancement-section'.

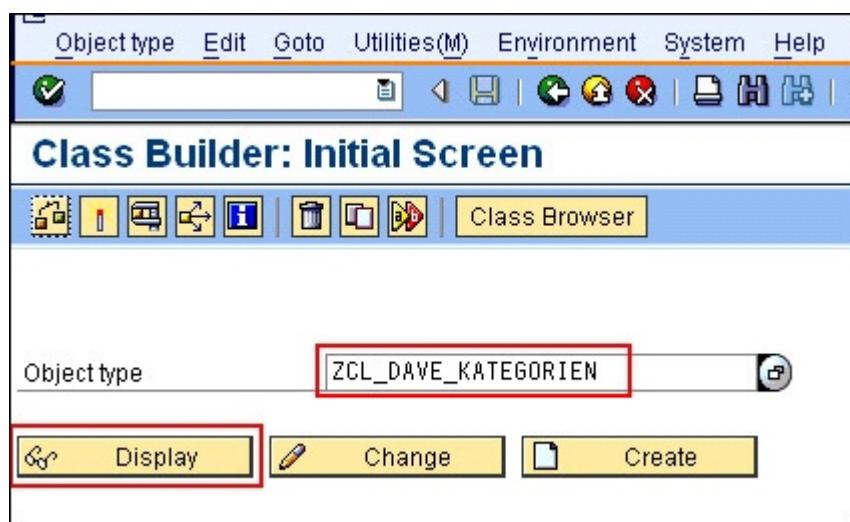
 **Note:** This is the biggest function of the framework which knows what needs to be invoked for that particular 'Exit' method applied instead or along with the original code.

To understand these powerful techniques here is a practical demonstration of them by means of some basic scenarios.

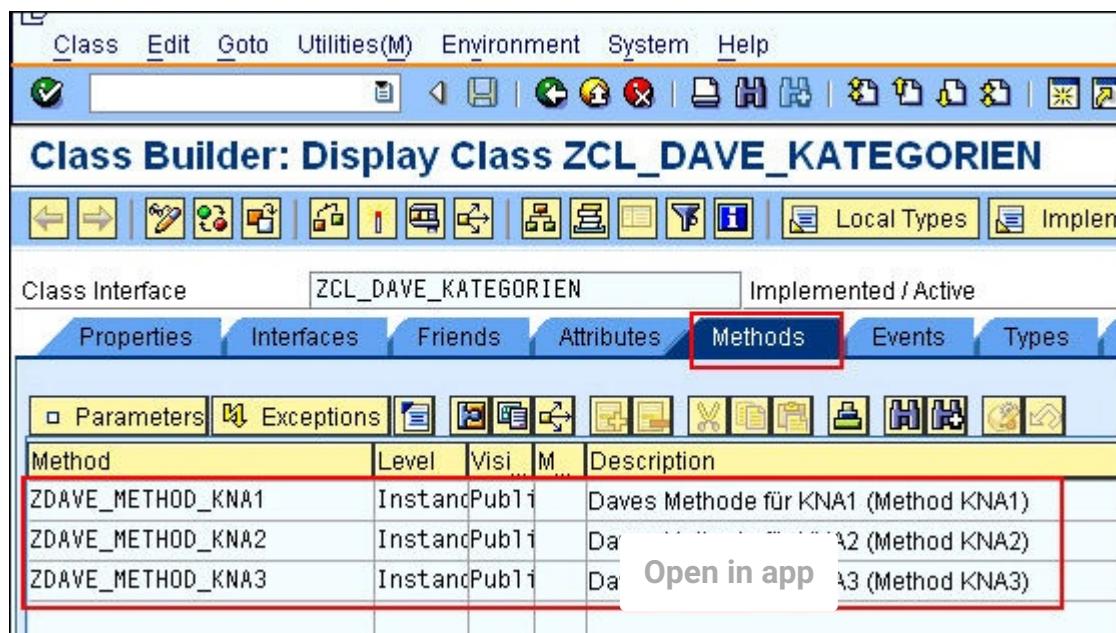
 **Note:** Here in this scenario the class used is in customer name space ('Z' class).

With respect to the scenario, three methods are used to demonstrate the functionalities of the three enhancement methods (Pre, Post and Overwrite Exit methods).

Access the class in 'Display' mode which needs to be enhanced.



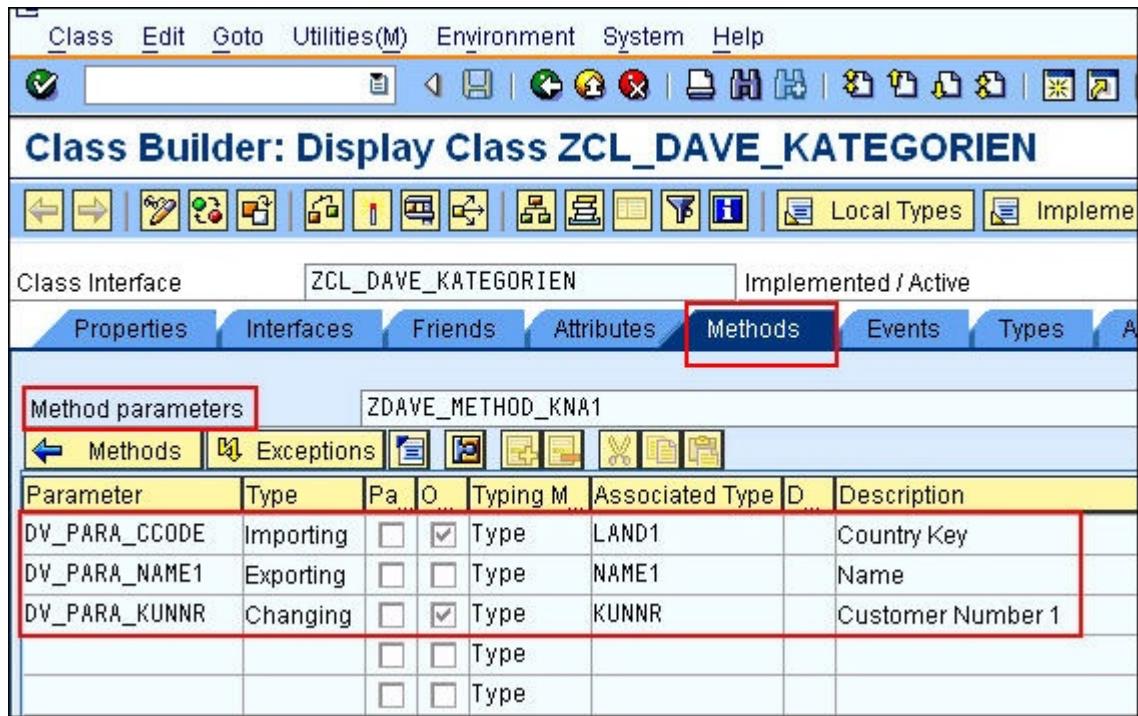
These are three methods which need to be enhanced,



Method	Level	Visi	M...	Description
ZDAVE_METHOD_KNA1	Instanc	Publi		Daves Methode für KNA1 (Method KNA1)
ZDAVE_METHOD_KNA2	Instanc	Publi		Daves Methode für KNA2 (Method KNA2)
ZDAVE_METHOD_KNA3	Instanc	Publi		Open in app Daves Methode für KNA3 (Method KNA3)

→  Scenario 1: Applying 'pre-exit' for enhancing method 'ZDAVE_METHOD_KNA1'.

Considering the first method 'ZDAVE_METHOD_KNA1', observe the global parameters



Class Builder: Display Class ZCL_DAVE_KATEGORIEN

Class Interface: ZCL_DAVE_KATEGORIEN Implemented / Active

Properties Interfaces Friends Attributes Methods Events Types A

Method parameters: ZDAVE_METHOD_KNA1

Parameter	Type	P	O	Typing M	Associated Type	D	Description
DV_PARA_CCODE	Importing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	LAND1		Country Key
DV_PARA_NAME1	Exporting	<input type="checkbox"/>	<input type="checkbox"/>	Type	NAME1		Name
DV_PARA_KUNNR	Changing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	KUNNR		Customer Number 1
		<input type="checkbox"/>	<input type="checkbox"/>	Type			
		<input type="checkbox"/>	<input type="checkbox"/>	Type			

Check the source code for the original method,

Open in app

The screenshot shows the SAP Class Builder interface for class ZCL_DAVE_KATEGORIEN. The title bar reads "Class Builder: Class ZCL_DAVE_KATEGORIEN Display". The toolbar includes standard icons for file operations, search, and help. Below the toolbar is a parameter table:

Ty.	Parameter	Type spec.	Description
»	DV_PARA_CCODE	TYPE LAND1 OPTIONAL	Country Key
»	DV_PARA_NAME1	TYPE NAME1	Name
»	DV_PARA_KUNNR	TYPE KUNNR OPTIONAL	Customer Number 1

The method list shows one active method: ZDAVE_METHOD_KNA1. The code editor displays the following ABAP code:

```
1  method zdave method kna1.
2    data:
3      begin of fs_kna1,
4        kunnr type kunnr,
5        land1 type land1,
6        name1 type name1,
7        end of fs_kna1.
8    data:
9      t_kna1 like
10     standard table
11       of fs_kna1.
12   select kunnr
13     land1
14     name1
15   from kna1
16   into table
17     t_kna1
18   where kunnr = dv_para_kunnr
19     and land1 = dv_para_ccode.
20   loop at t_kna1 into fs_kna1.
21   dv_para_name1 = fs_kna1-name1.
22   endloop.
23 endmethod.
```

A red box highlights the inner data block and the SELECT statement.

This method can be called into an application (say) like a report program which will offer the standard functionality,

[Open in app](#)

Program Edit Goto Utilities(M) Environment System Help

ABAP Editor: Change Report ZDAVE_KATEGORIEN_ENHC

Report ZDAVE_KATEGORIEN_ENHC Active

```
1 REPORT zdaive_kategorien_enhc.
2
3 DATA:
4     zdaive_enhc TYPE REF TO zcl_dave_kategorien,
5     w_kunnr TYPE kunnr,
6     w_name1 TYPE name1.
7
8
9 PARAMETERS:
10    p_kunnr TYPE kunnr,
11    p_ccode TYPE land1.
12
13    w_kunnr = p_kunnr.
14
15 CREATE OBJECT zdaive_enhc.
16
17 CALL METHOD zdaive_enhc->zdaive_method_knai
18     EXPORTING
19         dv_para_ccode = p_ccode
20     IMPORTING
21         dv_para_name1 = w_name1
22     CHANGING
23         dv_para_kunnr = w_kunnr.
24
25 WRITE:/5 'Customer' COLOR 5 INTENSIFIED.
26 SKIP.
27 WRITE:/5 w_name1 COLOR 5 INVERSE.
```

Execute the report, to check out the standard functionality

✓ Test case I -

Program Edit Goto System Help

Enhancement Framework - Class Enhancements (Kategorienverbesserungen)

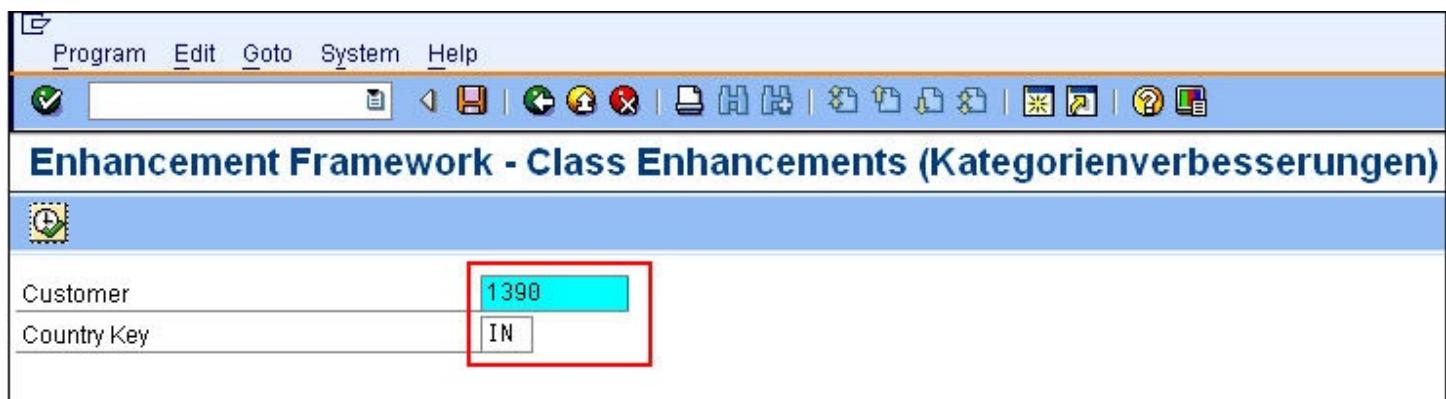
Customer Country Key

Observe the outcome,

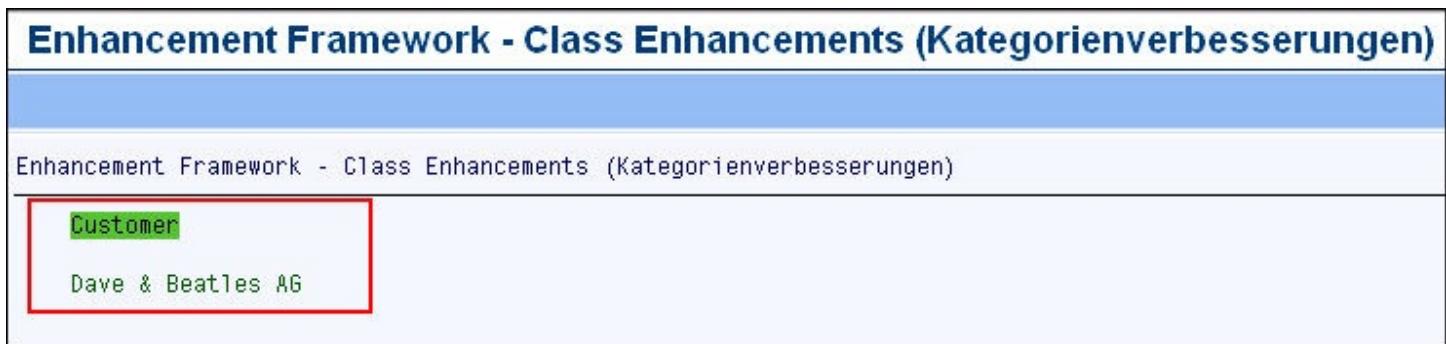
Open in app



✓ Test case II -

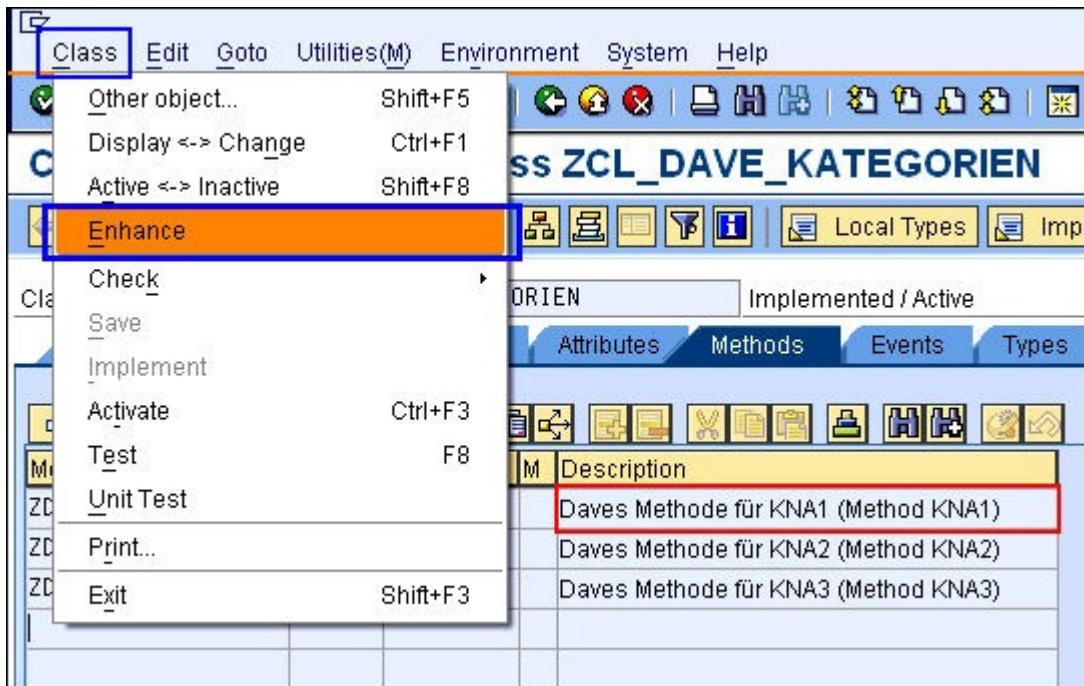


Observe the outcome,



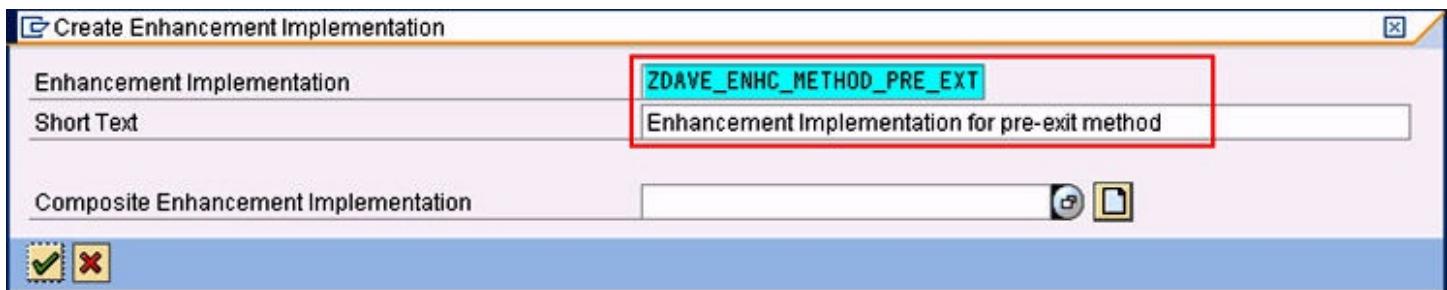
Now, we need to enhance this method. To access the enhancement mode,

Open in app



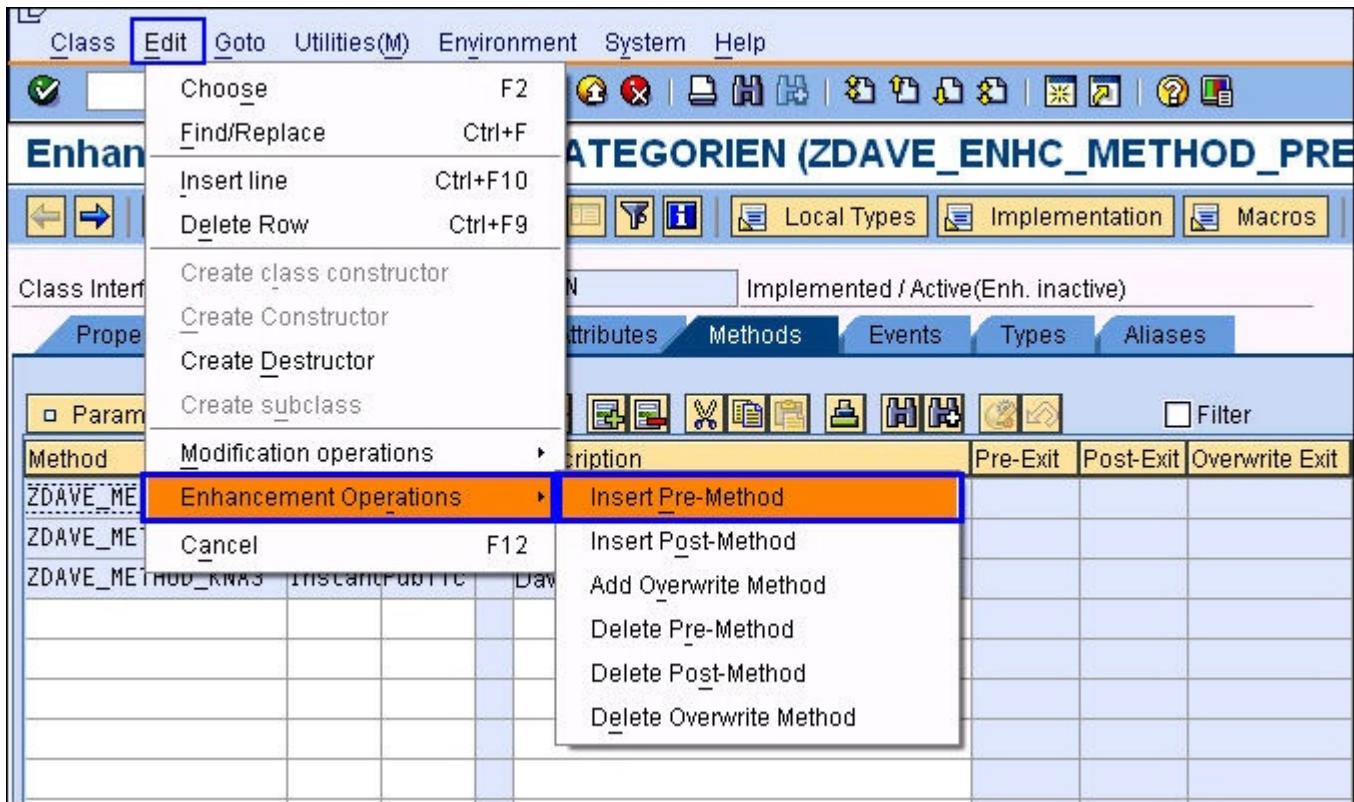
As expected it is needed to create an implementation,

Note - It is suggested to create separate implementations for all the three 'Exit method' enhancements.



After selecting the implementation access the enhancement operation option for creating a 'Pre-exit' method enhancement.

Open in app



Observe the post-exit created and click on that indication to navigate to enter the custom code.
In this case there are some validations and operations performed in this 'Pre-exit' method,



Activate the 'Pre-exit' method and call the 1st method 'ZDAVE_METHOD_KNA1' into the customer report program 'ZDAVE_KATEGORIEN_ENHC'.

Test the functionality achieved for different test cases,

[Open in app](#)

Program Edit Goto Utilities(M) Environment System Help

ABAP Editor: Change Include ZDAVE_ENHC_METHOD_PRE_EXT=====EIMP

Include ZDAVE_ENHC_METHOD_PRE_EXT===== Active

```

23  ENDMETHOD.
24
25  METHOD ipr_zdave_enhc_method_pre_ext~zdave_method_kna1.
26  **
27  ** Declaration of PRE-method, do not insert any comments here please!
28  **
29  **methods ZDAVE_METHOD_KNA1
30  ** importing
31  **   DV_PARA_CCODE type LAND1 optional
32  ** changing
33  **   DV_PARA_KUNNR type KUNNR optional .
34  **
35  IF dv_para_kunnr BETWEEN '0000000001' AND '00000000201'.
36
37  MESSAGE i001(zdave_kategorien).
38  dv_para_kunnr = '0000000101'.
39
40  ELSEIF dv_para_kunnr BETWEEN '00000001000' AND '00000099999'.
41
42  MESSAGE i002(zdave_kategorien).
43  dv_para_kunnr = '0000001000'.
44
45  ENDIF.
46  ENDMETHOD.          "IPR ZDAVE ENHC METHOD PRE EXT-ZDAVE METHOD KNA1
47  ENDCCLASS.
```

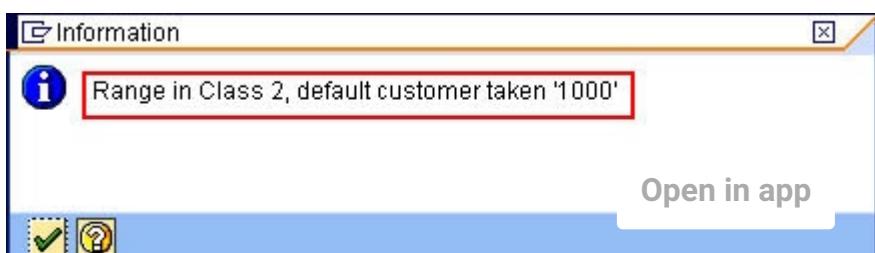
As some validation for customer range was carried out, observe the 'Pre-exit' method has triggered

Program Edit Goto System Help

Enhancement Framework - Class Enhancements (Kategorienverbesserungen)

	Customer	1390
	Country Key	IN

✓ Observe the difference,



Another test case,

The screenshot shows the SAP Enhancement Framework interface. The title bar reads "Enhancement Framework - Class Enhancements (Kategorienverbesserungen)". The main area displays a test case for the "Customer" class, which is highlighted with a green background. The details shown are "Dave & Beatles group".

Validations are processed for the above test case,

The screenshot shows the SAP Enhancement Framework interface after validation. The title bar reads "Enhancement Framework - Class Enhancements (Kategorienverbesserungen)". The validation results are displayed in a table:

Customer	201
Country Key	IN

The value "201" in the Customer row is highlighted with a red box.

✓ Finally, observe the difference in the output of the report as the 'Pre-exit' method of the class 'ZDAVE_METHOD_KNA1' has been invoked or triggered.

The screenshot shows an SAP information message dialog titled "Information". The message content is: "Range in Class 1, default customer taken '101'". The message is highlighted with a red box.

→ + Scenario 2: Applying 'post-exit' for enhancing method 'ZDAVE_METHOD_KNA2'.

Open in app



Check the parameters for the method 'ZDAVE_METHOD_KNA2',

Class Builder: Display Class ZCL_DAVE_KATEGORIEN

Class Interface: ZCL_DAVE_KATEGORIEN Implemented / Active

Properties Interfaces Friends Attributes Methods Events Types Alias

Method parameters: ZDAVE_METHOD_KNA2

Parameter	Type	P	O	Typing M	Associated Type	D	Description
DV_PARA_KUNNR	Importing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	KUNNR		Customer Number 1
DV_PARA_CCODE	Importing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	LAND1		Country Key
DV_PARA_NAME1	Exporting	<input type="checkbox"/>	<input type="checkbox"/>	Type	NAME1		Name
DV_PARA_NAME2	Exporting	<input type="checkbox"/>	<input type="checkbox"/>	Type	NAME2		Name 2
DV_PARA_STRAS	Exporting	<input type="checkbox"/>	<input type="checkbox"/>	Type	STRAS		House number and street
		<input type="checkbox"/>	<input type="checkbox"/>	Type			

Original source code for method 'ZDAVE_METHOD_KNA2',

Open in app

Method Edit Goto Utilities(M) Environment System Help

Class Builder: Class ZCL_DAVE_KATEGORIEN Display

Parameter Type spec. Description

DV_PARA_KUNNR	TYPE KUNNR OPTIONAL	Customer Number 1
DV_PARA_CCODE	TYPE LAND1 OPTIONAL	Country Key

Method ZDAVE_METHOD_KNA2 Active

```

3 begin of fs_kna2,
4   kunnr type kunnr,
5   land1 type land1,
6   name1 type name1,
7   name2 type name2,
8   stras type stras,
9   end of fs_kna2.
10
11 data:
12   t_kna2 like
13   standard table
14   of fs_kna2.
15 select kunnr
16   land1
17   name1
18   name2
19   stras
20   from kna1
21   into table
22   t_kna2
23   where kunnr = dv_para_kunnr
24     and land1 = dv_para_ccode.
25 loop at t_kna2 into fs_kna2.
26   dv_para_name1 = fs_kna2-name1.
27   dv_para_name2 = fs_kna2-name2.
28   dv_para_stras = fs_kna2-stras.
29 endloop.
```

After calling the method in the application report program, execute the report

Test case I -

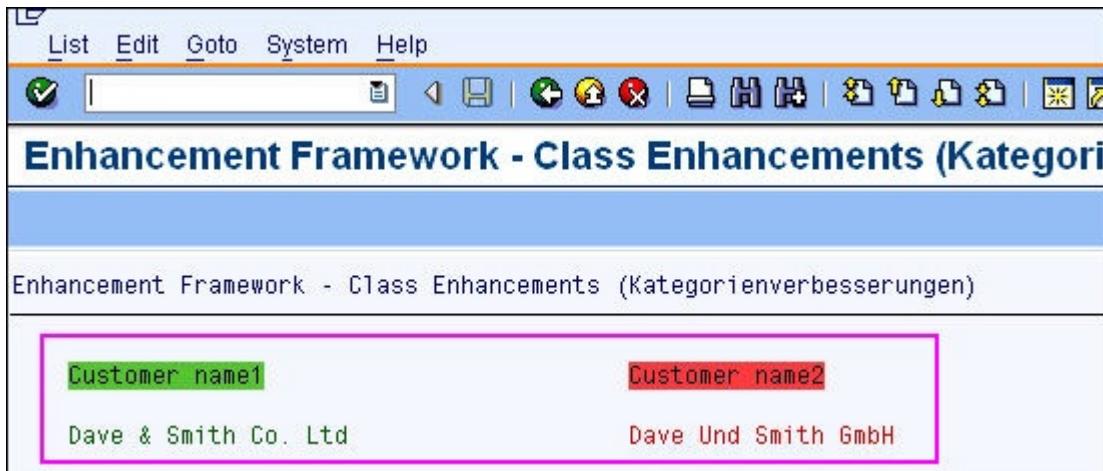
Program Edit Goto System Help

Enhancement Framework - Class Enhancements (Kategorienverbesserungen)

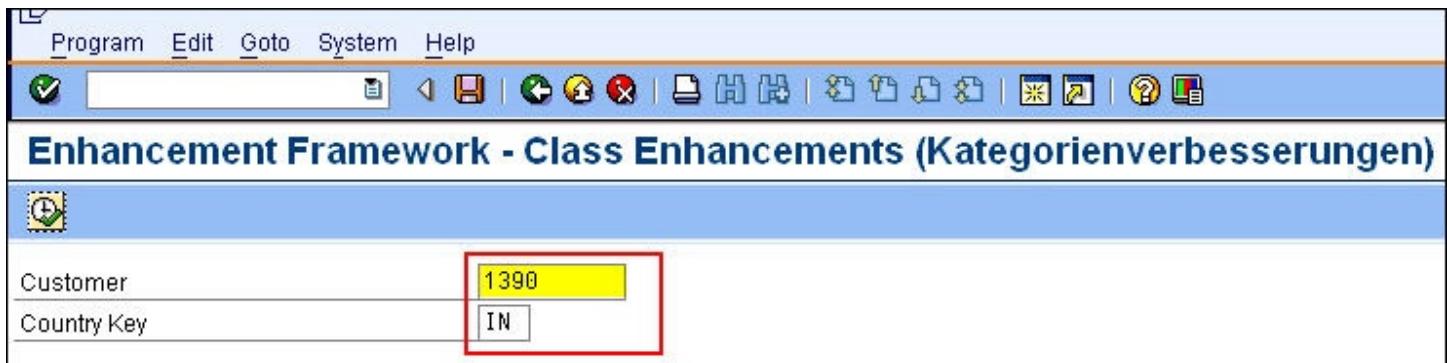
Customer	201
Country Key	IN

Observe the output,

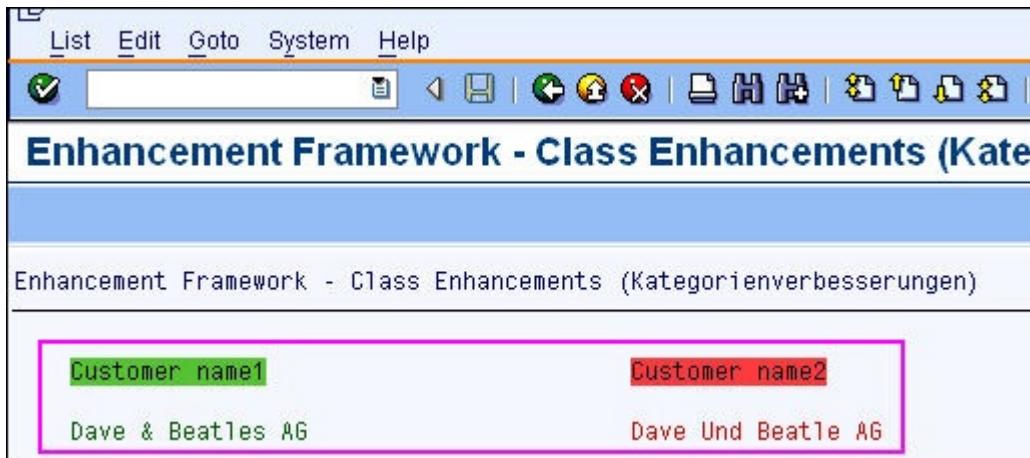
Open in app



✓ Test case II -



Observe the output,



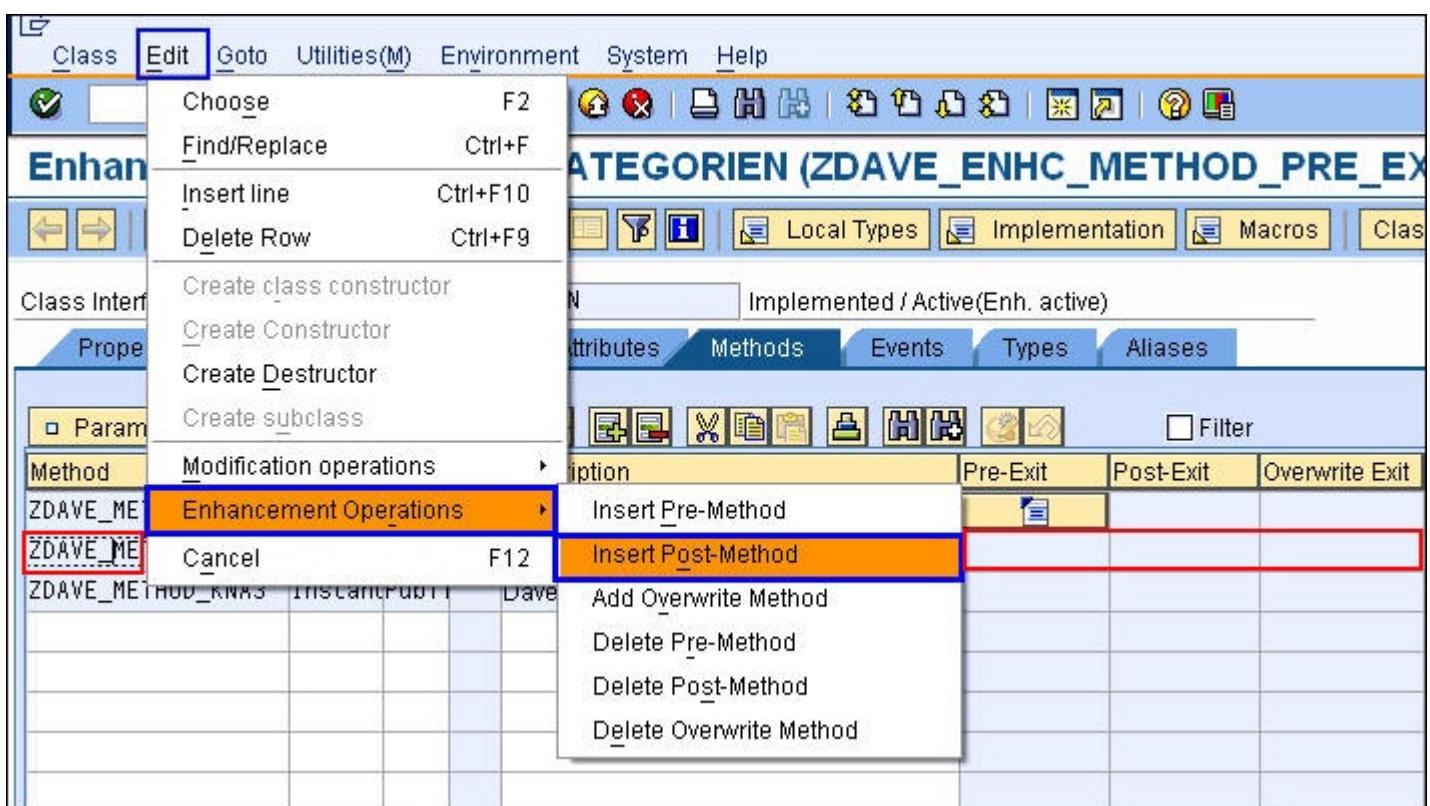
Access the 'Enhance' path (Enhancement Mode) 'Method' → 'Enhance'.

Create an Implementation for the method 'ZDAVE_METHOD_KNA2',

Open in app

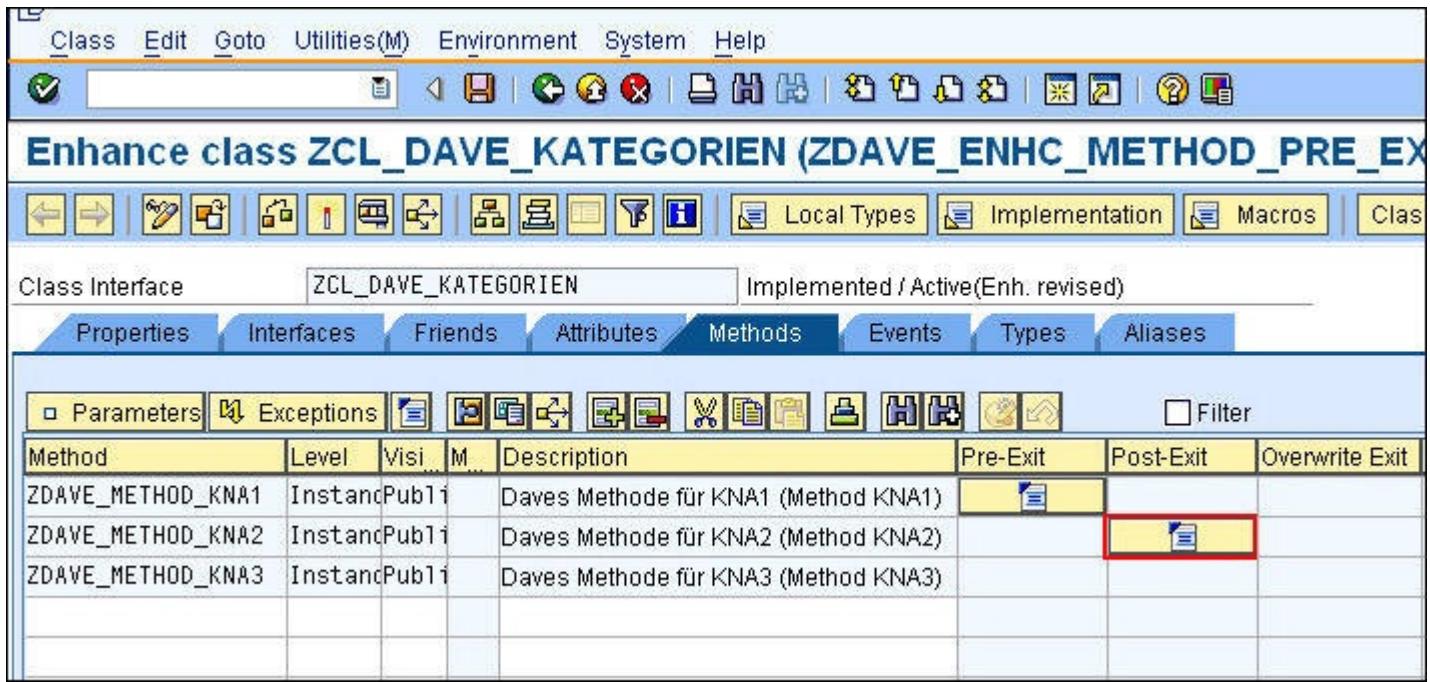
Select or Create Enhancement Implementation		
Enhancement Implementation	Pack.	Short Text
ZDAVE_ENHC_METHOD_POS_EXT	YH1167	Enhancement Implementation for post-exit method
ZDAVE_ENHC_METHOD_PRE_EXT	YH1167	Enhancement Implementation for pre-exit method
<input type="checkbox"/> Use this enhancement implementation without query in future		
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Create a 'Post-exit' method by accessing the path '**Enhancement operations**' → 'Insert Post-method'



Observe the 'Post-exit' method created and navigate inside to write the enhancement code,

Open in app



After the original method does its job the 'Post-exit' method will get invoked and it will allow to enhance the application by adding one more parameter which will be displayed by the application (here report program), here's the code.

[Open in app](#)

```
46      ENDMETHOD.  
47      METHOD ipo_zdave_enhc_method_pre_ext~zdave_method_kna2.  
48      *-----------------------------------------------------------------*  
49      ** Declaration of POST-method, do not insert any comments here please!  
50      **  
51      **methods ZDAVE_METHOD_KNA2  
52      **  importing  
53      **    DV_PARA_KUNNR type KUNNR optional  
54      **    DV_PARA_CCODE type LAND1 optional  
55      **  changing  
56      **    DV_PARA_NAME1 type NAME1  
57      **    DV_PARA_NAME2 type NAME2  
58      **    DV_PARA_STRAS type STRAS .  
59      **-*  
60      DATA:  
61      BEGIN OF fs_postexit,  
62          kunnr TYPE kunnr,  
63          land1 TYPE land1,  
64          name1 TYPE name1,  
65          name2 TYPE name2,  
66          stras TYPE stras,  
67      END OF fs_postexit.  
68      DATA:  
69          t_postexit LIKE  
70          STANDARD TABLE  
71          OF fs_postexit.  
72          SELECT kunnr  
73              land1  
74              name1  
75              name2  
76              stras  
77          FROM kna1  
78          INTO TABLE  
79          t_postexit  
80          WHERE kunnr = dv_para_kunnr  
81              AND land1 = dv_para_ccode.  
82          LOOP AT t_postexit INTO fs_postexit.  
83              dv_para_stras = fs_postexit-stras.  
84          ENDLOOP.
```

Now, call the second method 'ZDAVE_METHOD_KNA2' in the report program,

Open in app

Program Edit Goto Utilities(M) Environment System Help

ABAP Editor: Change Report ZDAVE_KATEGORIEN_ENHC

Report ZDAVE_KATEGORIEN_ENHC Active

```
5 zdale_enhc TYPE REF TO zcl_dave_kategorien,
6 w_kunnr TYPE kunnr,
7 w_name1 TYPE name1,
8 w_name2 TYPE name2,
9 w_stras TYPE stras.
10
11 PARAMETERS:
12   p_kunnr TYPE kunnr,
13   p_ccode TYPE land1.
14
15 w_kunnr = p_kunnr.
16
17 CREATE OBJECT zdale_enhc.
18
19 CALL METHOD zdale_enhc->zdale_method_kna2
20   EXPORTING
21     dv_para_kunnr = p_kunnr
22     dv_para_ccode = p_ccode
23   IMPORTING
24     dv_para_name1 = w_name1
25     dv_para_name2 = w_name2
26     dv_para_stras = w_stras.
27
28 WRITE:/5 'Customer' COLOR 4 INTENSIFIED,
29           40 'Customer 2' COLOR 5 INTENSIFIED,
30           75 'Street' COLOR 7 INTENSIFIED.
31 SKIP.
32 WRITE:/5 w_name1 COLOR 4 INVERSE,
33           40 w_name2 COLOR 5 INVERSE,
34           75 w_stras COLOR 7 INVERSE.
```

Execute and enter the desired data,

Program Edit Goto System Help

Enhancement Framework - Class Enhancements (Kategorienverbesserungen)

Customer	1390
Country Key	IN

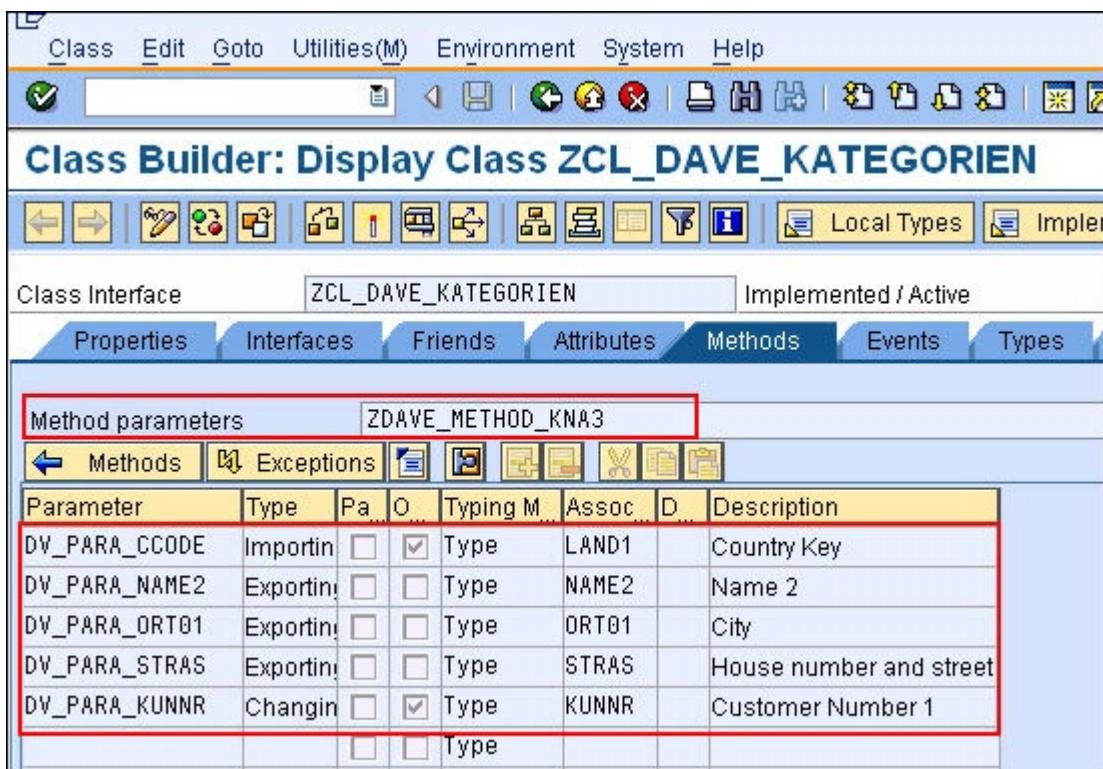
✓ Observe the difference,

Open in app



→ + Scenario 3: Applying 'Overwrite-exit' method to the method 'ZDAVE_METHOD_KNA3',

Check the parameters of the method 'ZDAVE_METHOD_KNA3',



Observe the source code of the original method,

[Open in app](#)

Method Edit Goto Utilities(M) Environment System Help

Class Builder: Class ZCL_DAVE_KATEGORIEN Display

Ty.	Parameter	Type spec.	Description
<input checked="" type="checkbox"/>	DV_PARA_CCODE	TYPE LAND1 OPTIONAL	Country Key

Method ZDAVE_METHOD_KNA3 Active

```

2   data:
3     begin of fs_kna3,
4       kunnr type kunnr,
5       land1 type land1,
6       name2 type name2,
7       ort01 type ort01,
8       stras type stras,
9     end of fs_kna3.
10    data:
11      t_kna3 like
12        standard table
13          of fs_kna3.
14    select kunnr
15      land1
16      name2
17      ort01
18      stras
19    from knal
20    into table
21      t_kna3
22    where kunnr = dv_para_kunnr
23      and land1 = dv_para_ccode.
24    loop at t_kna3 into fs_kna3.
25      dv_para_name2 = fs_kna3-name2.
26      dv_para_ort01 = fs_kna3-ort01.
27      dv_para_stras = fs_kna3-stras.
28    endloop.

```

Call the method in the application report program and test it (Execute)

✓ Test case I -

Program Edit Goto System Help

Enhancement Framework - Class Enhancements (Kategorienverbesserungen)

<input checked="" type="checkbox"/>	Customer	201
	Country Key	IN

Output -

[Open in app](#)

List Edit Goto System Help

Enhancement Framework - Class Enhancements (Kategorienverbesserungen)

Enhancement Framework - Class Enhancements (Kategorienverbesserungen)

Customer name2	City	Street
Dave Und Smith GmbH	Pune	5 Strawberry Fields

✓ Test case II -

Program Edit Goto System Help

Enhancement Framework - Class Enhancements (Kategorienverbesserungen)

Customer	1390
Country Key	IN

Output -

List Edit Goto System Help

Enhancement Framework - Class Enhancements (Kategorienverbesserungen)

Enhancement Framework - Class Enhancements (Kategorienverbesserungen)

Customer name2	City	Street
Dave Und Beatle AG	Pune	10 Mos Eisely

Now, to replace this functionality with the customer's functionality, we make use of 'Overwrite-exit' method
As usual access the path '**Method**' → '**Enhance**' to enable the enhancement mode.
Create an Implementation,

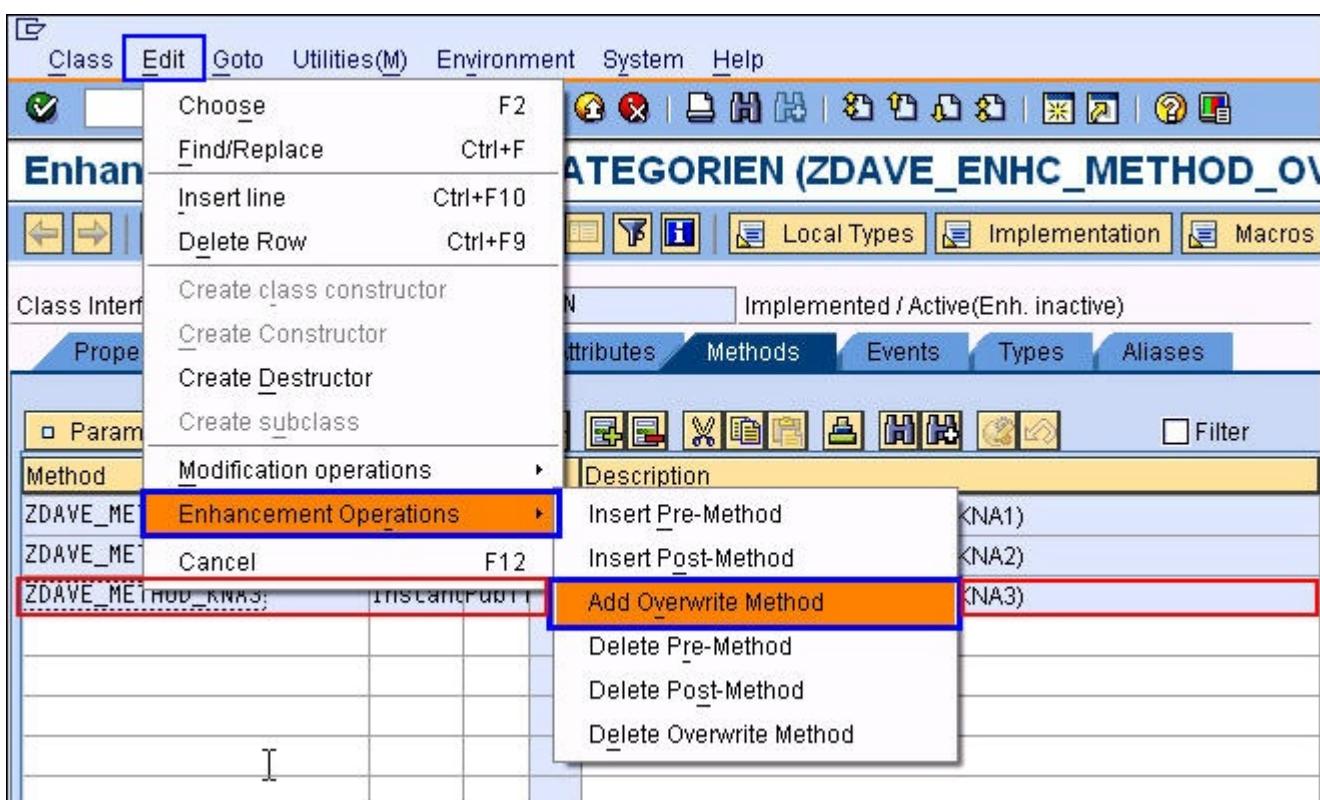
[Open in app](#)

Select or Create Enhancement Implementation

Enhancement Implementation	Pack.	Short Text
ZDAVE_ENHC_METHOD_OVR_EXT	YH1167	Enhancement Implementation for overwrite-exit method
ZDAVE_ENHC_METHOD_POS_EXT	YH1167	Enhancement Implementation for post-exit method
ZDAVE_ENHC_METHOD_PRE_EXT	YH1167	Enhancement Implementation for pre-exit method

Use this enhancement implementation without query in future

Add an 'Overwrite-method' accessing the path,



Observe that the method has been created and follow the indication to code the functionality which will replace the original one,

Open in app

The screenshot shows the SAP ABAP Class Editor interface. The title bar reads "Enhance class ZCL_DAVE_KATEGORIEN (ZDAVE_ENHC_METHOD_OVR)". The "Methods" tab is selected in the navigation bar. A table below lists methods with columns for Method, Level, Visi, M, Description, Pre-Exit, Post-Exit, and Overwrite Exit. The "Overwrite Exit" column for ZDAVE_METHOD_KNA3 is checked and highlighted with a red border.

Method	Level	Visi	M	Description	Pre-Exit	Post-Exit	Overwrite Exit
ZDAVE_METHOD_KNA1	Instanc	Publi		Daves Methode für KNA1 (Method KNA1)			
ZDAVE_METHOD_KNA2	Instanc	Publi		Daves Methode für KNA2 (Method KNA2)			
ZDAVE_METHOD_KNA3	Instanc	Publi		Daves Methode für KNA3 (Method KNA3)			<input checked="" type="checkbox"/>

Observe the coding logic,

Open in app

```

23  ENDMETHOD.
24
25  METHOD iow_zdave_enhc_method_ovr_ext-zdave_method_kna3.
26
27  ** Declaration of Overwrite-method, do not insert any comments here please!
28
29  ** methods ZDAVE_METHOD_KNA3
30  **   importing
31  **     /DV_PARA_CCODE type LAND1 optional
32  **   exporting
33  **     /DV_PARA_NAME2 type NAME2
34  **     /DV_PARA_ORT01 type ORTO1
35  **     /DV_PARA_STRAS type STRAS
36  **   changing
37  **     /DV_PARA_KUNNR type KUNNR optional .
38
39  DATA:
40    BEGIN OF fs_overwrite,
41      kunnr TYPE kunnr,
42      land1 TYPE land1,
43      name2 TYPE name2,
44      ort01 TYPE ort01,
45      stras TYPE stras,
46    END OF fs_overwrite.
47  DATA:
48    t_overwrite LIKE
49    STANDARD TABLE
50      OF fs_overwrite.
51  IF dv_para_kunnr BETWEEN '0000000019' AND '0000000201'.
52    MESSAGE s001(zdave_kategorien).
53  ELSEIF dv_para_kunnr = '0000000101'.
54    dv_para_kunnr = '00000001000'.
55    MESSAGE s002(zdave_kategorien).
56    dv_para_kunnr = '0000001000'.
57  ENDIF.
58  SELECT kunnr
59    land1
60    name2
61    ort01
62    stras
63  FROM knai
64  INTO TABLE
65    t_overwrite
66  WHERE kunnr = dv_para_kunnr
67    AND land1 = dv_para_ccode.
68  LOOP AT t_overwrite INTO fs_overwrite.
69    dv_para_name2 = fs_overwrite-name2.
70    dv_para_ort01 = fs_overwrite-ort01.
71    dv_para_stras = fs_overwrite-stras.
72  ENDLOOP.
73  ENDMETHOD.

```

"IOW ZDAVE ENHC METHOD OVR EXT-ZDAVE METHOD KNA3

Activate and call the method in the application report program,

Open in app

Program Edit Goto Utilities(M) Environment System Help

ABAP Editor: Change Report ZDAVE_KATEGORIEN_ENHC

Report ZDAVE_KATEGORIEN_ENHC Active

```

7   w_name1 TYPE name1,
8   w_name2 TYPE name2,
9   w_stras TYPE stras,
10  w_ort01 TYPE ort01.
11
12 PARAMETERS:
13   p_kunnr TYPE kunnr,
14   p_ccode TYPE land1.
15
16  w_kunnr = p_kunnr.
17
18  CREATE OBJECT zdale_enhc.
19
20  CALL METHOD zdale_enhc->zdale_method_kna3
21    EXPORTING
22      dv_para_ccode = p_ccode
23    IMPORTING
24      dv_para_name2 = w_name2
25      dv_para_ort01 = w_ort01
26      dv_para_stras = w_stras
27    CHANGING
28      dv_para_kunnr = w_kunnr.
29
30  WRITE:/5 'Customer name2' COLOR 5 INTENSIFIED,
31    35 'City' COLOR 6 INTENSIFIED,
32    55 'Street' COLOR 7 INTENSIFIED.
33  SKIP.
34  WRITE:/5 w_name2 COLOR 5 INVERSE,
35    35 w_ort01 COLOR 6 INVERSE,
36    55 w_stras COLOR 7 INVERSE.
37

```

Now, its time to test the application,

✓ First test case,

Program Edit Goto System Help

Enhancement Framework - Class Enhancements (Kategorienverbesserungen)

Customer	201
Country Key	IN

✓ Observe that the overwrite-exit method has been triggered,

Range in Class 1, default customer taken '101'

Open in app

✓ To confirm, observe the output,

Customer name2	City	Street
Dave & Schiff Unternehmensgruppe	Pune	5 Bespin Cloud City

✓ Second, test case

Customer	1390
Country Key	IN

✓ Again, observe that the overwrite-exit method has been triggered,

✓ Range in Class 2, default customer taken '1000'

✓ Confirm with the output, 😊

Customer name2	City	Street
Dave & Beatle Unternehmensgruppe	Pune	5 Genesis Spire

You can refer to the SAP documentation on Enhancing the Components of Classes or Interfaces.

Like

Open in app



Unknown User (106601wsx)

This article is really great. Can you also tell me how to access the private attributes of the class in the overwrite-exit?

Jul 31, 2009

[Reply](#) · [Like](#)

[Open in app](#)