



ABAP News - Design-time Tools



Venky Varadadesigan

Product Manager

SAP NetWeaver US PM - SAP Labs

Venkata.varadadesigan@sap.com



Positioning

The New ABAP Editor

Dynamic Programming & RTTS

Code Inspector

ABAP Unit

Summary



Positioning

The New ABAP Editor

Dynamic Programming & RTTS

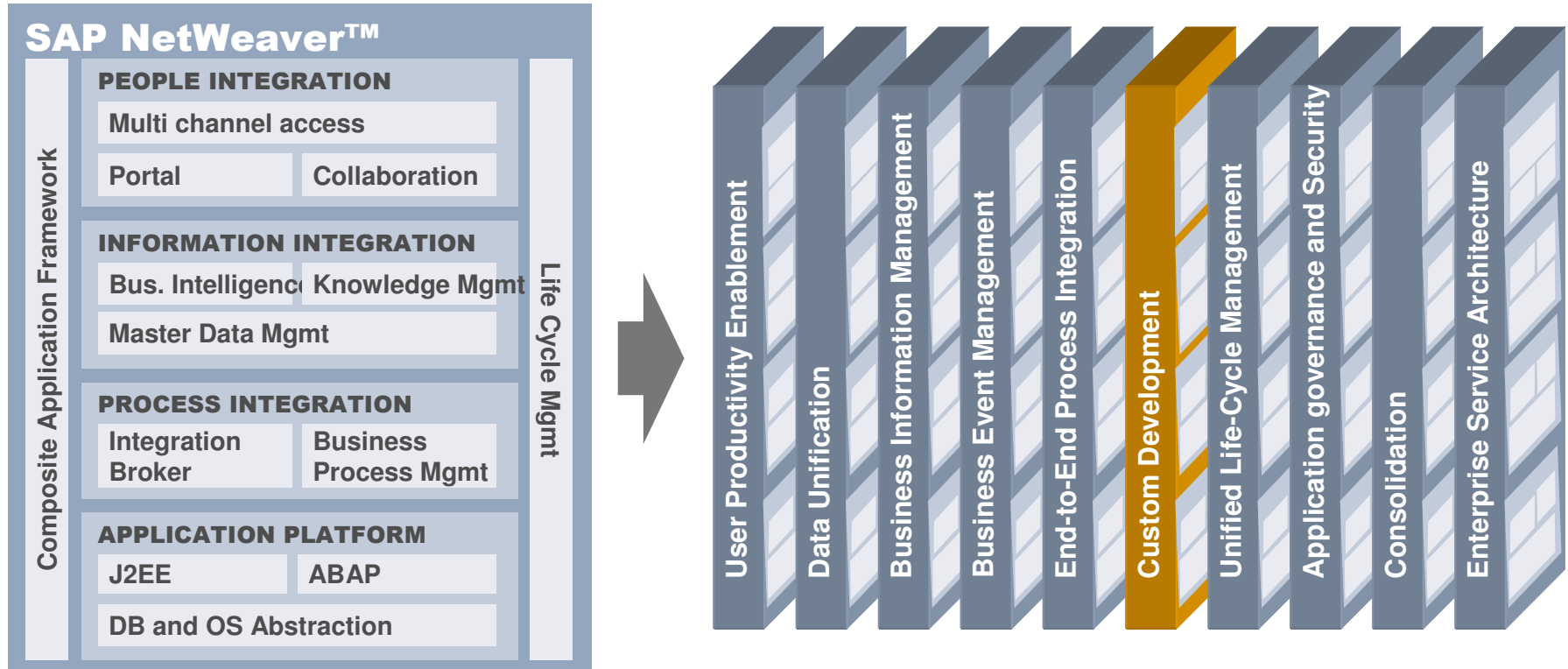
Code Inspector

ABAP Unit

Summary

Positioning: IT Practices and IT Scenarios

IT practices “slice” SAP NetWeaver to directly address key issues ... and help enterprises find the right starting point



Different capabilities used jointly in each “slice”

Positioning: SAP NetWeaver Technology Map

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

The New ABAP Editor

Dynamic Programming & RTTS

Code Inspector

ABAP Unit

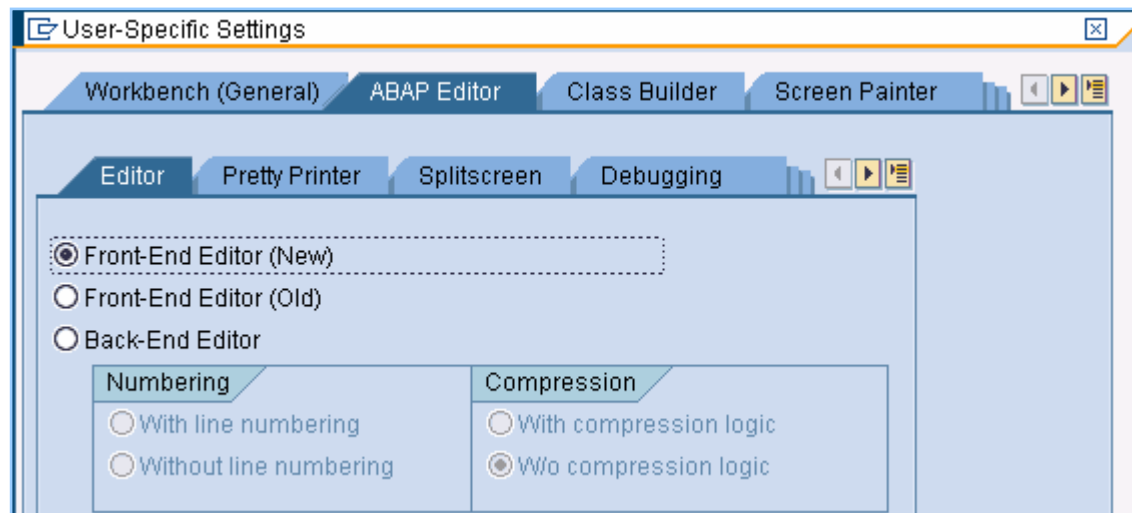
Summary

Introducing the New ABAP Editor

```
Report      ZVENKY_SDN_DEMO1      Active
1  *~-----*
2  *~ Report  ZVENKY_SDN_DEMO1
3  *~-----*
4  REPORT  ZVENKY_SDN_DEMO1.
5
6  tables: sflight.
7
8  data: begin of itab occurs 0,
9         carrid like sflight-carrid,
10        connid type sflight-connid,
11        fldate type sflight-fldate,
12        end   of itab.
13
14  start-of-SELECTION.
15  perform populate_flight_data.
16  perform print_results.
17
18  *~-----*
19  *~      Form  populate_flight_data
20  *~-----*
21  FORM populate_flight_data .
22
23  select * from sflight
24         where carrid eq 'LH'.
25         move-corresponding sflight to itab.
26         append itab.
27  endselect. "sflight
28
29  ENDFORM.           " populate flight data
30
31
Scope: \FORM populate_flight_data\SELECT
```

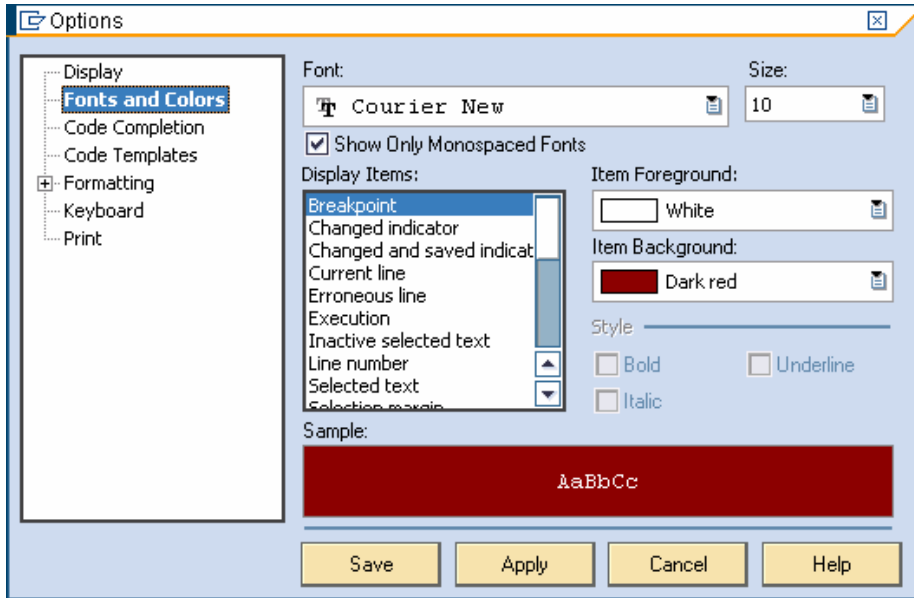

New ABAP 'Front-End Editor'

- The new ABAP 'Front-End Editor' available as of **SAP NetWeaver 2004s (AS-ABAP)** with **SAPGUI 6.40 Patch >= 10**
- Uses the Code Editor **ActiveX control**
- Used in all ABAP editing transactions (SE80, SE38, SE37, SE24...)
- **Integrated with the New ABAP Debugger**



Editor -> Utilities -> Settings

New ABAP Editor – Features (1)



Syntax Highlighting

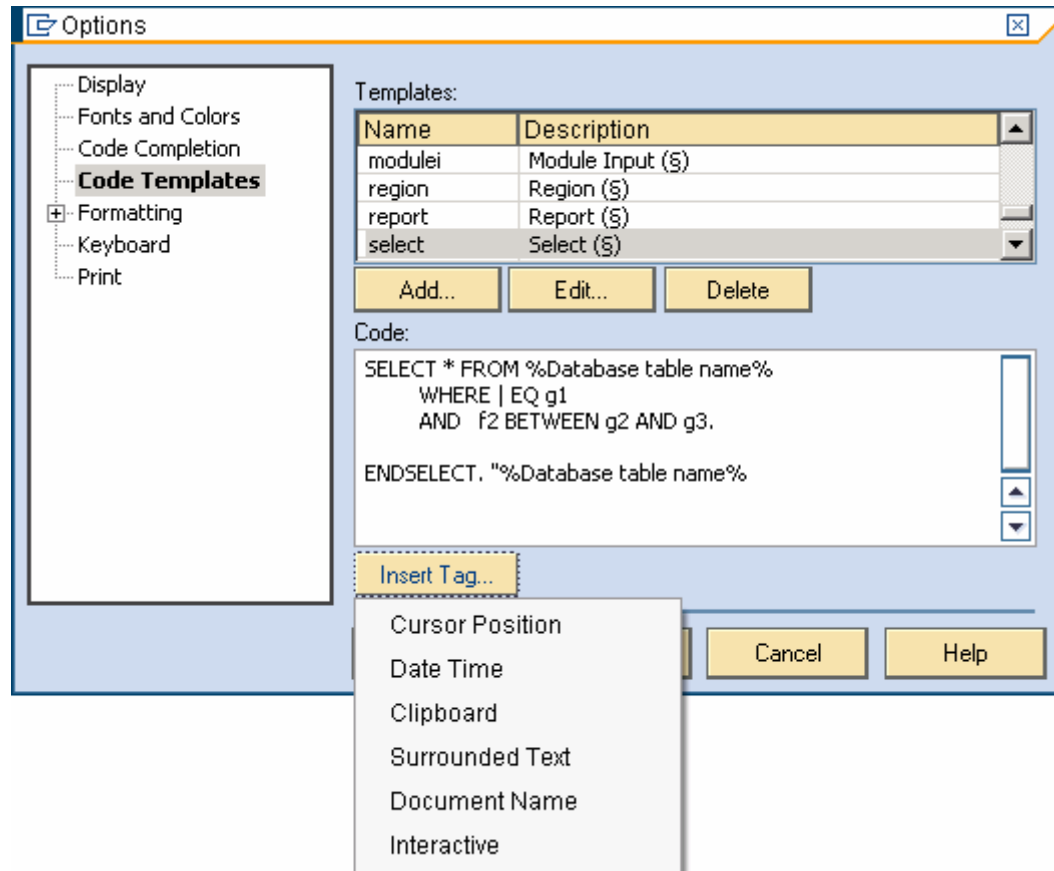
User defined colors for different items

```
8  + data: begin of itab occurs 0,  
13 data: begin of itab occurs 0,  
14   carrid like sflight-carrid,  
15   connid type sflight-connid,  
16   fldate type sflight-fldate. flight_data.  
17   end of itab. _results.  
18  
19  *&-----  
20  *&      Form populate_flight_data  
21  *&-----  
22  FORM populate_flight_data .  
23  
24  select * from sflight  
25     where carrid eq 'LH'.  
26     move-corresponding sflight to itab.  
27     append itab.  
28  endselect. "sflight  
29  
ENDFORM. " populate flight data
```

Outlining

Collapse and Expand block of code & Scope

New ABAP Editor – Features (2)



Code Templates

User defined code snippets (hot key: <Ctrl-Enter>)

New ABAP Editor – Features (3)

```
14: start-of-SELECTION.  
15: perform populate_flight_data.  
16:   
17: PERFORM  
18: per  
19:
```

Enable Code Hints

Display Hint When Saving at Least Chars

Automatically Hide Tooltips After msec

Code Hints

Hints for auto completion & auto correction

Replace

Find what: Find Next

Replace with: Replace

Replace All

Mark All

Cancel

Options

Match whole word

Match case

Regular Expression

Search

Search up

Search hidden text

Replace in selection

Find/Replace

Extended functionality for Find and Replace

Demo

New ABAP Editor

ABAP Editor – Quick reference Overview

The screenshot shows the SAP ABAP Editor interface with the following callouts and features:

- Click on margin to set breakpoint:** A red circle icon in the left margin next to line 586.
- Notice your changes by Changed line Indicator:** A red horizontal bar highlights line 587.
- Use bookmark for fast navigation:** A blue bookmark icon in the left margin next to line 595.
- Collapse /Expand block:** A square icon in the left margin next to line 597.
- See where you are by Current Scope:** A vertical line in the left margin indicates the current position at line 590.
- Call Template with Ctrl-Enter:** A callout pointing to the right margin next to line 590.
- Split view to see different parts of the document in same time:** A callout pointing to the vertical scrollbar on the right.
- Scroll to see Scroll Info:** A callout pointing to the scroll bar area.
- Double click on status to change indicator or call Go To Line dialog:** A callout pointing to the status bar at the bottom right.
- Press button to check user settings:** A callout pointing to a gear icon in the bottom right corner.

The code editor displays the following ABAP code:

```
586: * Some "Special" stuff
587: IF ev_ext_type IS INITIAL.
588:     CASE iv_int_type.
589:         WHEN mc_wt_attribute_all.
590:             ev_ext_type = mc_ext_attribute.
591:         WHEN mc_wt_method_all.
592:             ev_ext_type = mc_ext_method.
593:         END CASE.
594:     ENDIF.
595: ENDIF.
596:
597: * Internal to external
598: IF ev_int_type IS REQUESTED AND iv_ext_type IS SUPPLIED.
599:     SEARCH mc_translate_ids FOR iv_ext_type.
600:     IF sy-subrc EQ 0.
601:         ev_int_type = sy-fdpos.
602:     ELSE.
603:         * Some "Special" stuff
604:         CASE iv_ext_type.
605:             WHEN mc_ext_attribute.
606:                 ev_int_type = mc_wt_attribute_all.
607:             WHEN mc_ext_method.
```

The status bar at the bottom shows: Scope: TCLASSIC1_ab4ed1tor\METHOD convert_types\IFNIFCASE ABAP4 DOB Ln 590 Col 44 Ch 44





Positioning

The New ABAP Editor

Dynamic Programming & RTTS

Code Inspector

ABAP Unit

Summary

Challenging Development Request

Senior management wants a generic table display tool where they can display joined tables freely.

e.g.
**Flight connection and Carrier IDs or
 Flights and bookings or ...**



Cit	ID	No.	Cty	Depart. city	Air...	Cty	Arrival city	Apt	Flight t...	Departure	Arrival	Distance	Dis. ...	D...	Airline	
0	AA	17	US	NEW YORK	JFK	US	SAN FRANCISCO	SFO	6:01	11:00:00	14:01:00	2.572,0000	MI	0	American Airlines	
000	AZ	555	IT	ROME	FCO	DE	FRANKFURT	FRA	2:05	19:00:00	21:05:00	845,0000	MI	0	Alitalia	
000	AZ	789	JP	TOKYO	TYO	IT	ROME	FCO	15:40	11:45:00	19:25:00	6.130,0000	MI	0	Alitalia	
000	DL	106	US	NEW YORK	JFK	DE	FRANKFURT	FRA	7:55	19:35:00	09:30:00	3.851,0000	MI	1	Delta Airlines	
000	JL	407	JP	TOKYO	NRT	DE	FRANKFURT	FRA	12:05	13:30:00	17:35:00	9.100,0000	KM	0	Japan Airlines	
000	JL	408	DE	FRANKFURT	FRA	JP	TOKYO	NRT	11:15	20:25:00	15:40:00	9.100,0000	KM	X	1	Japan Airlines
000	LH	400	DE	FRANKFURT	FRA	US	NEW YORK	JFK	7:24	10:10:00	11:34:00	6.162,0000	KM	0	Lufthansa	
000	LH	401	US	NEW YORK	JFK	DE	FRANKFURT	FRA	7:15	18:30:00	07:45:00	6.162,0000	KM	1	Lufthansa	
000	LH	402	DE	FRANKFURT	FRA	US	NEW YORK	JFK	7:35	13:30:00	15:05:00	6.162,0000	KM	X	0	Lufthansa
000	LH	2402	DE	FRANKFURT	FRA	DE	BERLIN	SXF	1:05	10:30:00	11:35:00	555,0000	KM	0	Lufthansa	
000	QF	5	SG	S												
000	SQ	2	SG	S												
000	SQ	15	US	S												
000	UA	941	DE	F												

Cit	ID	No.	Flight Date	Airfare	Curr.	Plane Type	Capacity	Occupied	Booking total
0	AA	17	27.08.2004	1.186,69	USD	747-400	660	43	48.203,37
000	AA	17	27.08.2004	1.186,69	USD	747-400	660	43	48.203,37
000	AA	17	27.08.2004	1.186,69	USD	747-400	660	43	48.203,37
000	AA	17	27.08.2004	1.186,69	USD	747-400	660	43	48.203,37
000	AA	17	27.08.2004	1.186,69	USD	747-400	660	43	48.203,37
000	AA	17	27.08.2004	1.186,69	USD	747-400	660	43	48.203,37
000	AA	17	27.08.2004	1.186,69	USD	747-400	660	43	48.203,37

Selecting the data by Joining different Tables

e.g. Join of SPFLI and SCARR

SELECT

**SPFLI~MANDT SPFLI~CARRID ...
SCARR~CURRCODE SCARR~URL ...**

FROM

SPFLI join SCARR

ON

**SPFLI~MANDT = SCARR~MANDT AND
SPFLI~CARRID = SCARR~CARRID**

e.g. Join of SFLIGHT and SBOOK

SELECT

**SFLIGHT~MANDT SFLIGHT~CARRID ...
SBOOK~BOOKID SBOOK~CUSTOMID ...**

FROM

SFLIGHT join SBOOK

ON

**SFLIGHT~MANDT = SBOOK~MANDT AND
SFLIGHT~CARRID = SBOOK~CARRID AND
SFLIGHT~CONNID = SBOOK~CONNID.**

Problem Solution Step 1



How shall I cover all these different
SELECTS in one program ?

Use Dynamic Open SQL:

```
SELECT  
(select_clause_it)  
FROM  
(from_clause).
```



Dr. ABAP

Problem Solution Step 2



Nice, but where do I get the DB table components from ?

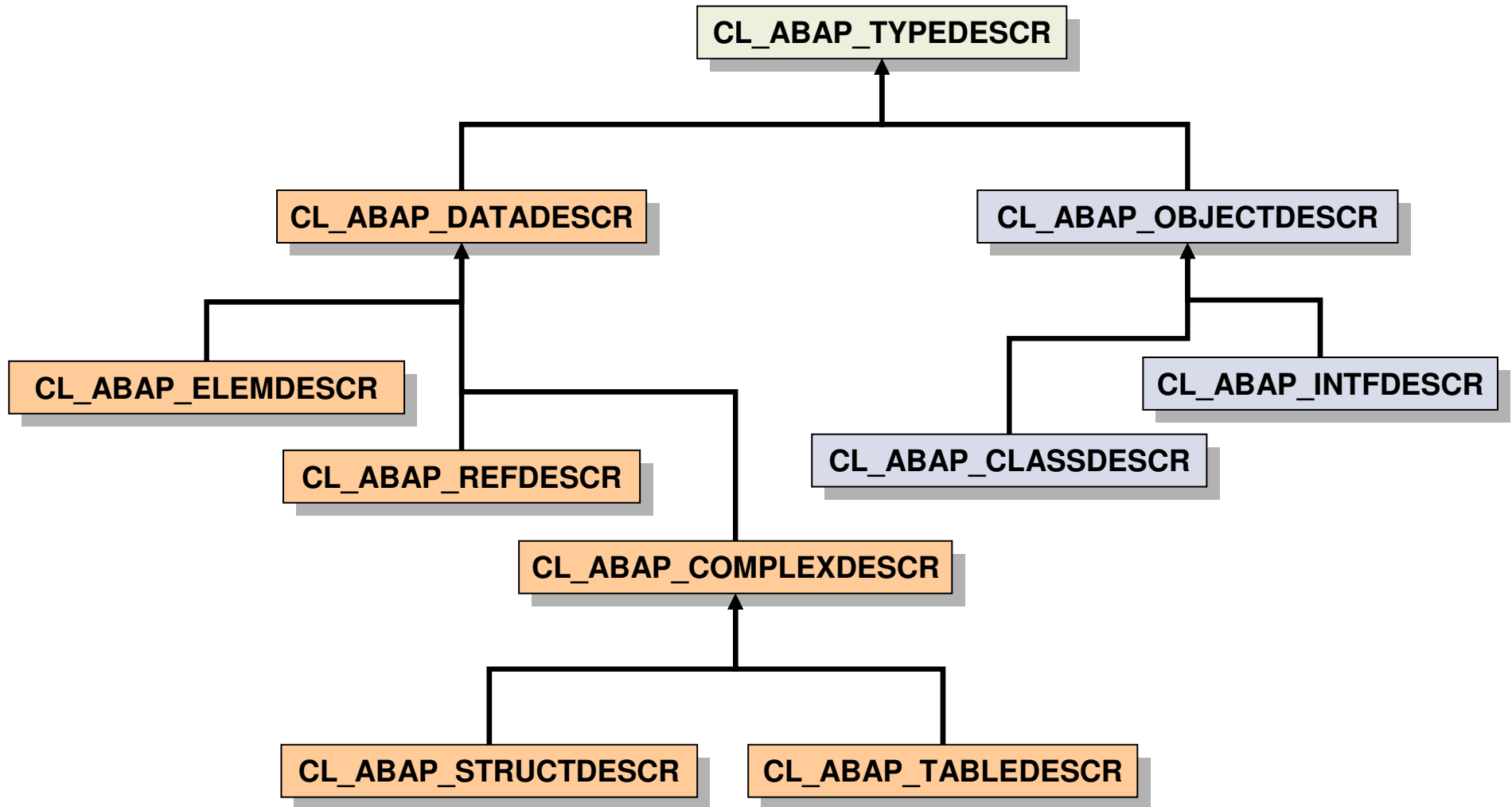


Dr. ABAP

Use RTTS (Run Time Type Services):

```
struct_type ?= cl_abap_typedescr=>describe_by_name( dbtable ).  
components = struct_type->get_components( ).
```

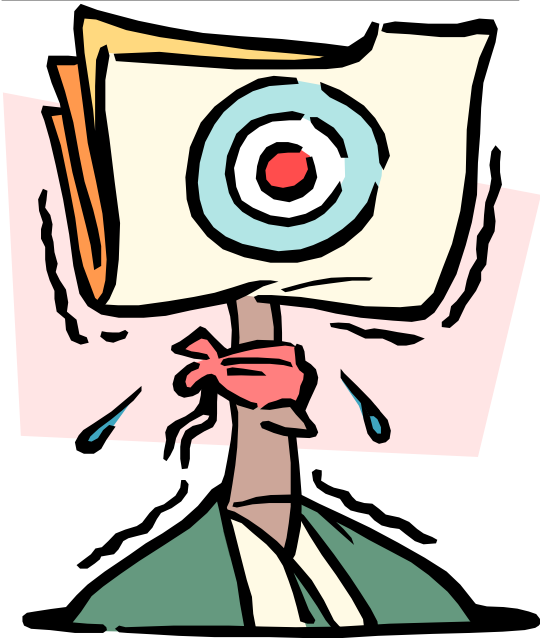
RTTS Class Hierarchy



- Type identification and description at run time (formerly RTTI)
- Dynamic Type creation (RTTC)
- Implemented as system Classes

Problem Solution Step 3

```
SELECT  
(select_clause)  
INTO TABLE ITAB  
FROM  
(from_clause).
```

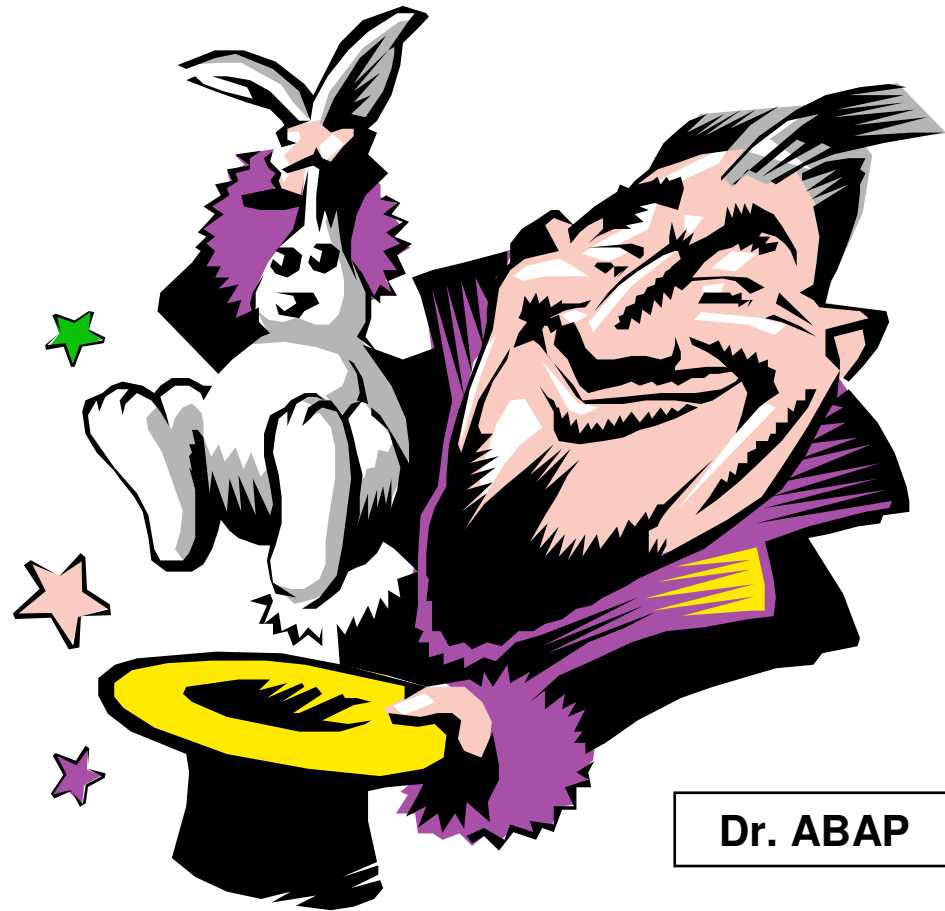


Perfect, but now we are lost !

For each DB table combination I need a result table **ITAB** with totally different components!

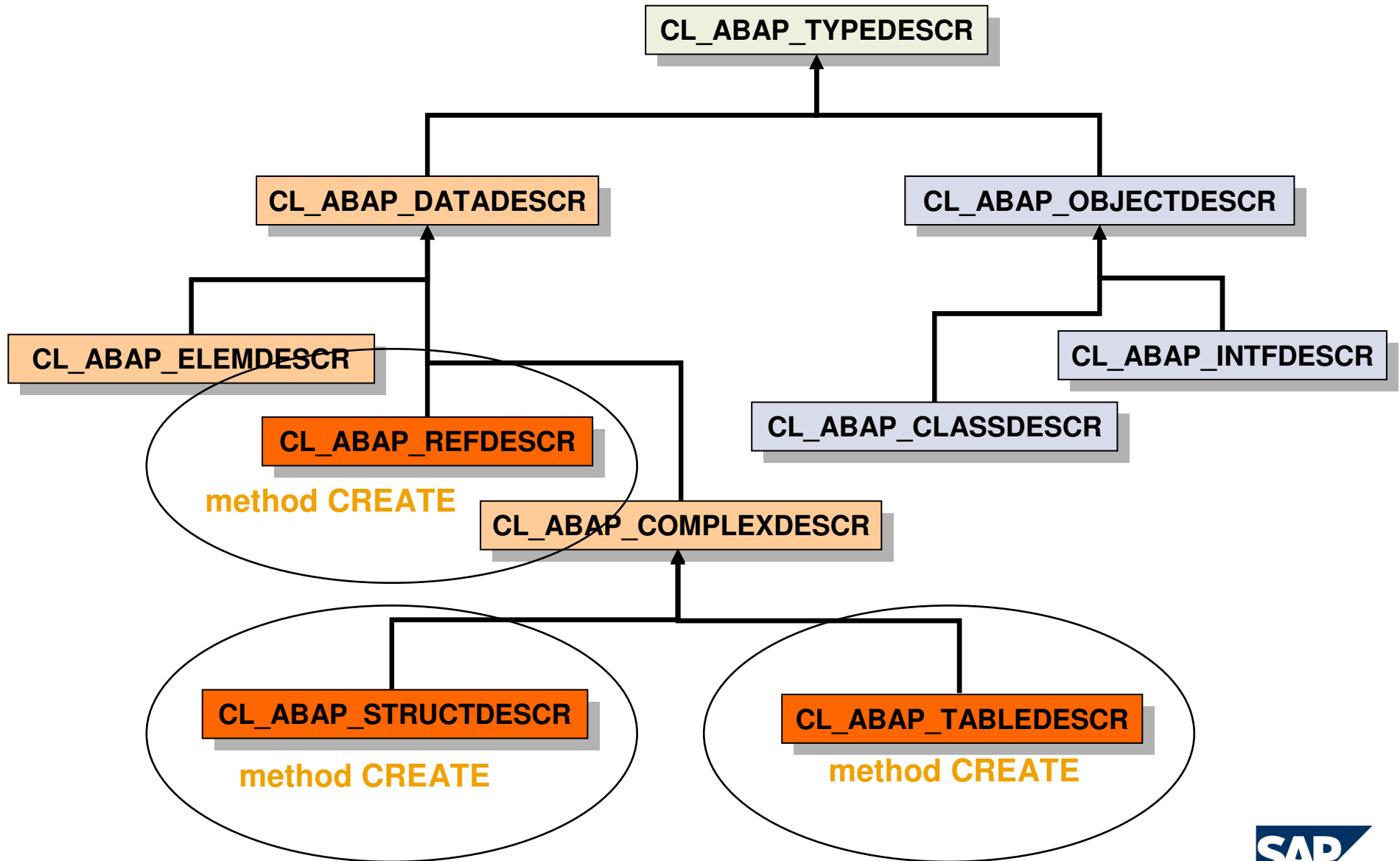
Problem Solution Step 3

**Create the internal table you need
during runtime !**



Dr. ABAP

RTTS Class Hierarchy



Demo **Run Time Type Services**



Positioning

The New ABAP Editor

Dynamic Programming & RTTS

Code Inspector

ABAP Unit

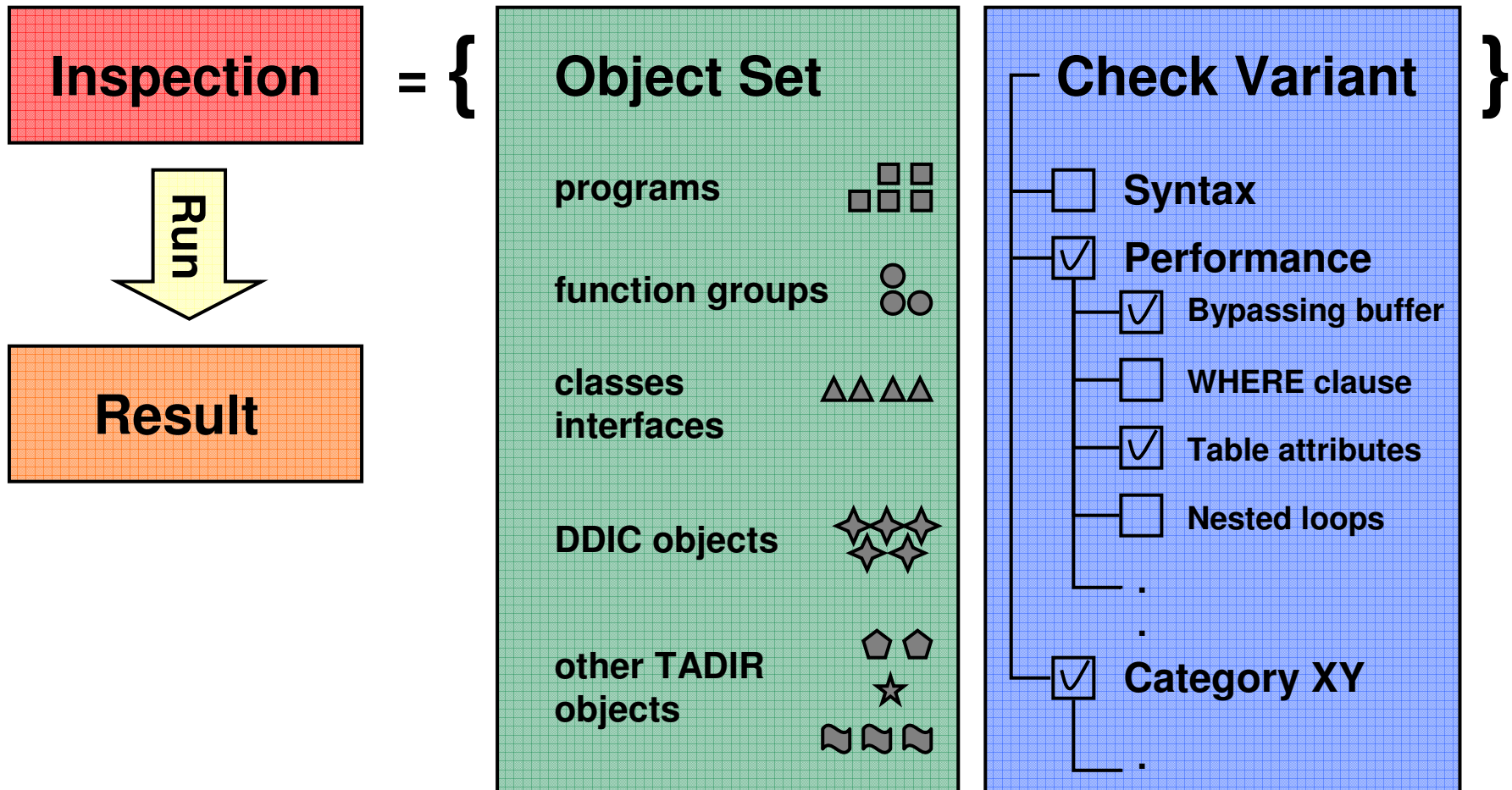
Summary

Code Inspector: a new test tool

- Available as of SAP NW AS 6.10
Downport to 4.6C available (Note 543359)
- Analysis of '**static**' ABAP code and other repository objects
- Framework with test driver and configurable set of checks
- **Single and mass** tests possible

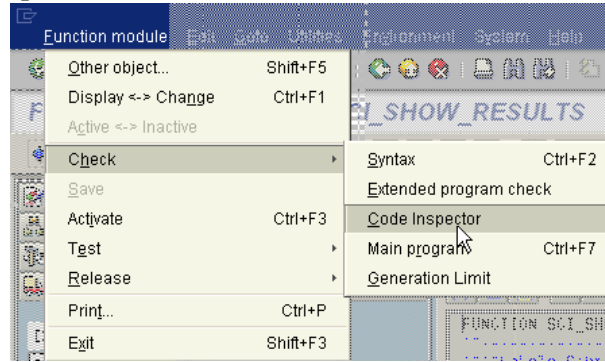
Target Group: Developers and Quality manager

Elements of the Code Inspector



How to Call the Code Inspector

- Call from workbench transactions SE38, SE37, SE24, or SE80 with **Object** → **Check** → **Code Inspector** to check a single report, function group, class, ...



- OR, Call transaction '**SCI**' (Code Inspector) to define Object Sets, Check Variants, Inspections.

A screenshot of the 'Code Inspector: Initial Screen' in SAP. The screen has a blue header with the title 'Code Inspector: Initial Screen'. Below the header, there is a 'Person Responsible' field with the value '11805032'. There are three main sections: 'Inspection', 'Object Set', and 'Check Variant'. Each section has a 'Name' field and a 'Vers.' field. The 'Inspection' section has a 'Name' field with a lock icon and a 'Vers.' field. Below it are icons for search, create, edit, delete, and refresh. The 'Object Set' section has a 'Name' field with a lock icon and a 'Vers.' field. Below it are icons for search, create, edit, delete, and refresh. The 'Check Variant' section has a 'Name' field with a lock icon. Below it are icons for search, create, edit, delete, and refresh.

Step-1: Object Set Selection from TADIR Characteristics

Deleted On:
Deletion date automatically
set to 50 days after creation !

Tab 'Select Object Set'

Object Assignment

logical AND

Object Selection

Selections Only:
'Calculate' Object List now
or just in time ?


The screenshot shows the SAP Code Inspector interface for an object set. The title bar reads "Code Inspector: Object set". The main area is divided into several sections:

- Header Information:** Person Responsible: HOLMES, Changed On: 09.10.2002, Object Set: TEST_SET, Vers.: 001, Deleted On: 28.11.2002, Description: TEST_SET__001, Number of Elements: 5.
- Navigation Tabs:** Select Object Set (active), Edit Object Set, Object Set fr. Result, Obj.Set from Request.
- Object Assignment Section:** A table with columns for field name, value, and a selection arrow. The value "PA-MA*" is entered in the Component ID field.
- Object Selection Section:** A table with columns for field name, value, and a selection arrow. The value "RH_*" is entered in the Function group field.
- Footer:** A status bar at the bottom indicates "The Object set has been saved" and shows system information like "XB4 (1) (900)", "p39826", and "INS".

Step-2: Select Check Variant

A single check is added to the check variant by simply clicking the checkbox

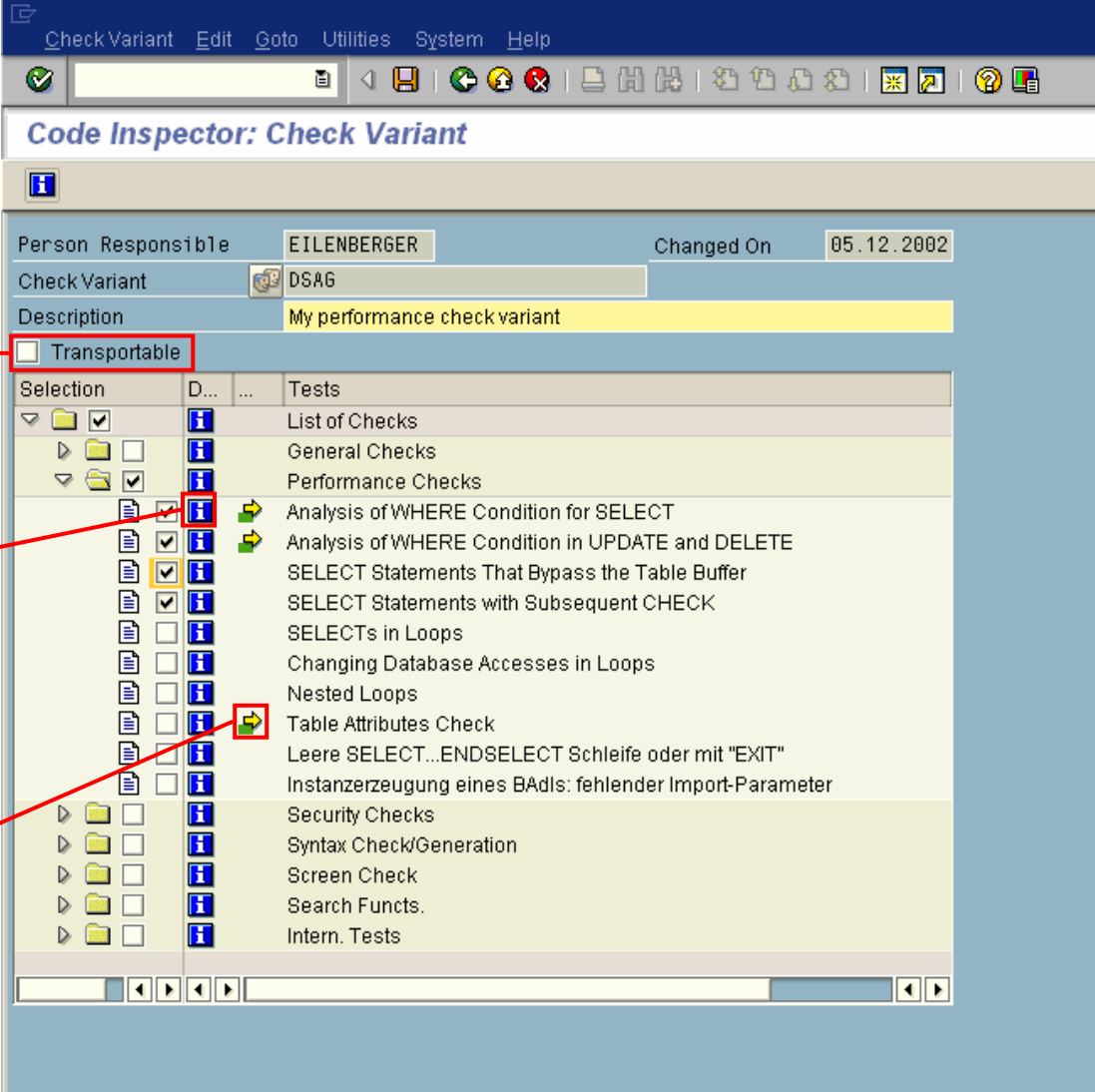
Transportable

Global  check variants can be transported

Information  about check

Attributes of a check



















-  - settings are ok
-  - set attributes



The screenshot shows the SAP Code Inspector interface for a check variant. The title bar reads "Code Inspector: Check Variant". The main area displays the following information:

- Person Responsible: EILENBERGER
- Changed On: 05.12.2002
- Check Variant: DSAG
- Description: My performance check variant

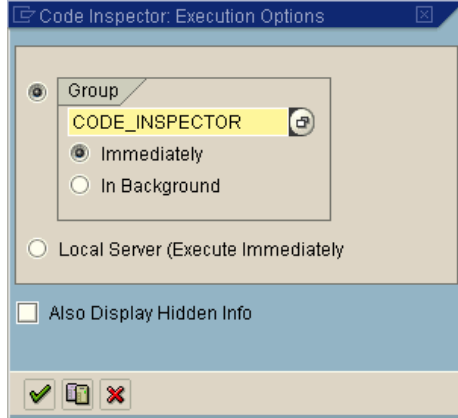
Below this information is a table of checks with columns for Selection, D..., and Tests. The "Transportable" checkbox is checked. The "Table Attributes Check" is highlighted with a red box, and its status is indicated by a green arrow icon.

Selection	D...	Tests
<input checked="" type="checkbox"/>		List of Checks
<input type="checkbox"/>		General Checks
<input checked="" type="checkbox"/>		Performance Checks
<input checked="" type="checkbox"/>		Analysis of WHERE Condition for SELECT
<input checked="" type="checkbox"/>		Analysis of WHERE Condition in UPDATE and DELETE
<input checked="" type="checkbox"/>		SELECT Statements That Bypass the Table Buffer
<input checked="" type="checkbox"/>		SELECT Statements with Subsequent CHECK
<input type="checkbox"/>		SELECTs in Loops
<input type="checkbox"/>		Changing Database Accesses in Loops
<input type="checkbox"/>		Nested Loops
<input type="checkbox"/>		Table Attributes Check
<input type="checkbox"/>		Leere SELECT...ENDSELECT Schleife oder mit "EXIT"
<input type="checkbox"/>		Instanzerzeugung eines BADs: fehlender Import-Parameter
<input type="checkbox"/>		Security Checks
<input type="checkbox"/>		Syntax Check/Generation
<input type="checkbox"/>		Screen Check
<input type="checkbox"/>		Search Funct.
<input type="checkbox"/>		Intern. Tests






Step-3: Inspection – Before Execution

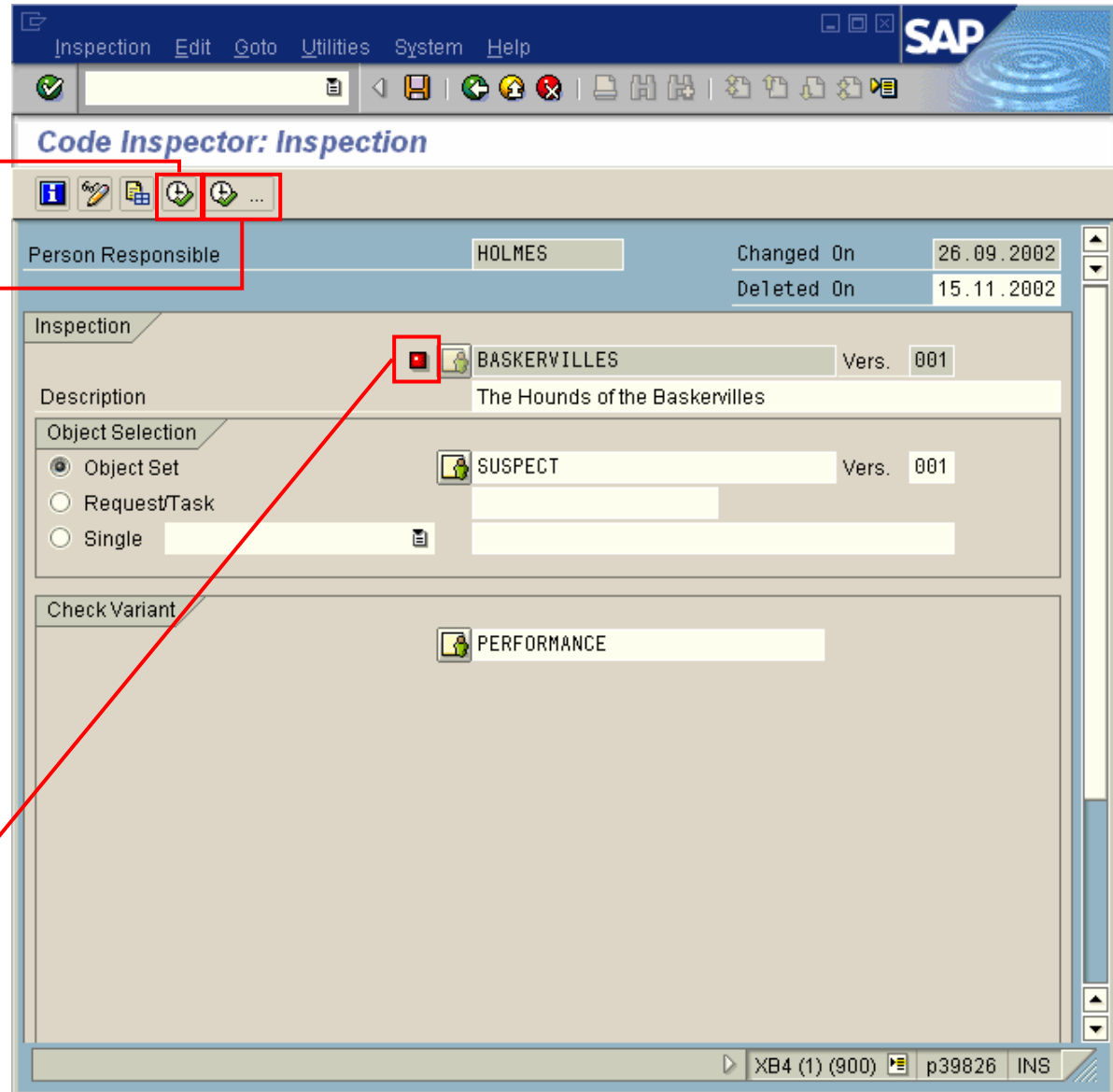
 **Run immediately on local server**

 **Execution options**



Inspection state

-  **not yet executed**
-  **broken, try to restart**
-  **scheduled in background job**
-  **being executed**
-  **already executed**








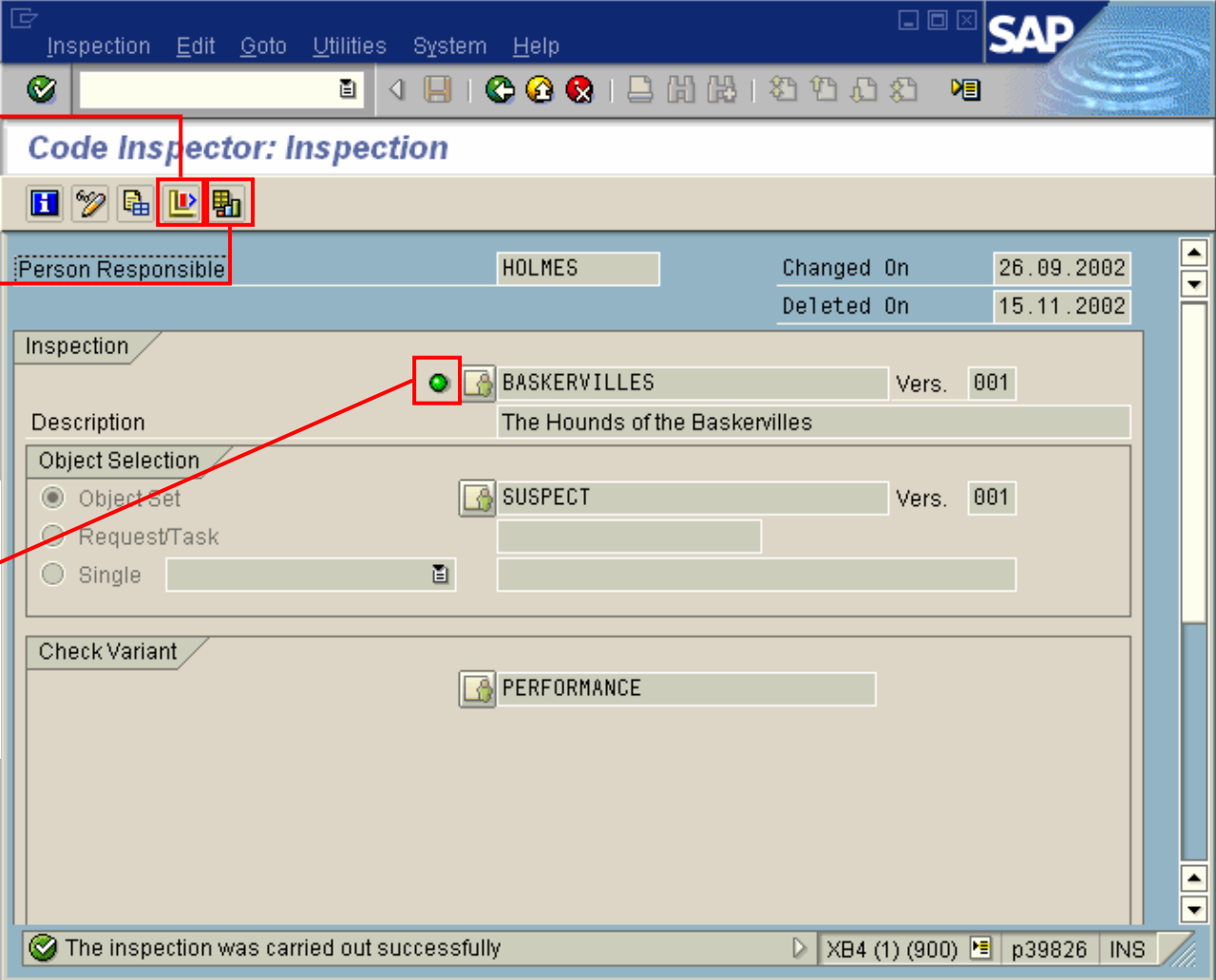
Step-3: Inspection – After Execution

 Show results

 Show statistics

Inspection state

-  not yet executed
-  broken, try to restart
-  scheduled in background job
-  being executed
-  already executed



The screenshot shows the SAP Code Inspector interface. The title bar reads "Code Inspector: Inspection". The main area displays the following information:

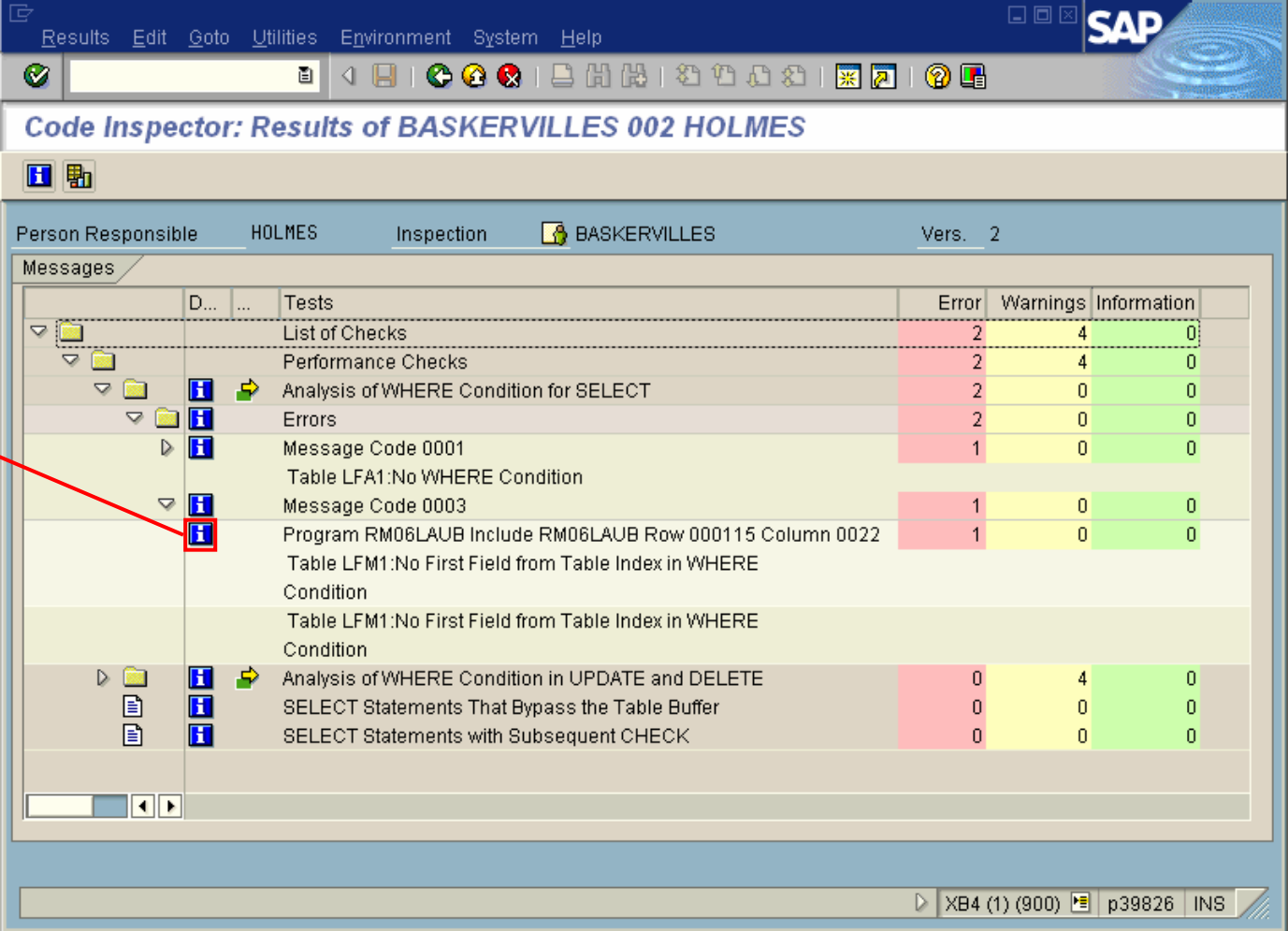
- Person Responsible:** HOLMES
- Changed On:** 26.09.2002
- Deleted On:** 15.11.2002
- Inspection:** BASKERVILLES (Vers. 001)
- Description:** The Hounds of the Baskervilles
- Object Selection:** Object Set (selected), Request/Task, Single
- Check Variant:** PERFORMANCE

A status bar at the bottom indicates: "The inspection was carried out successfully". The status bar also shows "XB4 (1) (900)", "p39826", and "INS".

Inspection Results

Double-click message text to display and change object

Information about check message 



Code Inspector: Results of BASKERVILLES 002 HOLMES

Person Responsible: HOLMES Inspection: BASKERVILLES Vers. 2

Messages	D...	Tests	Error	Warnings	Information
[-]		List of Checks	2	4	0
[-]		Performance Checks	2	4	0
[-]		Analysis of WHERE Condition for SELECT	2	0	0
[-]		Errors	2	0	0
[+]		Message Code 0001	1	0	0
		Table LFA1:No WHERE Condition			
[-]		Message Code 0003	1	0	0
		Program RM06LAUB Include RM06LAUB Row 000115 Column 0022	1	0	0
		Table LFM1:No First Field from Table Index in WHERE Condition			
		Table LFM1:No First Field from Table Index in WHERE Condition			
[+]		Analysis of WHERE Condition in UPDATE and DELETE	0	4	0
		SELECT Statements That Bypass the Table Buffer	0	0	0
		SELECT Statements with Subsequent CHECK	0	0	0

XB4 (1) (900) p39826 INS



Demo Code Inspector



Positioning

The New ABAP Editor

Dynamic Programming & RTTS

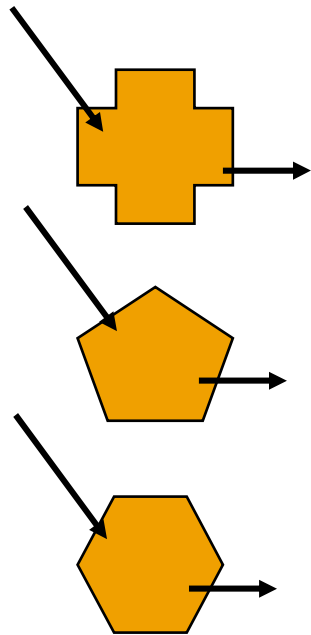
Code Inspector

ABAP Unit

Summary

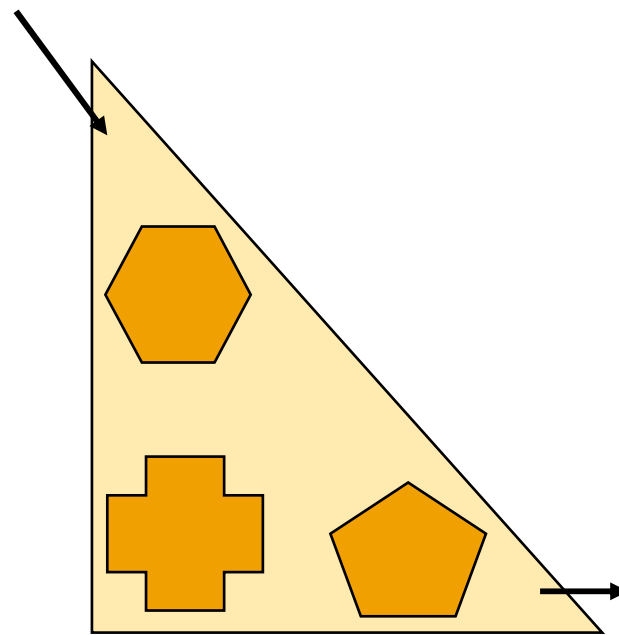
Test Scopes (OO-Languages)

Unit Tests



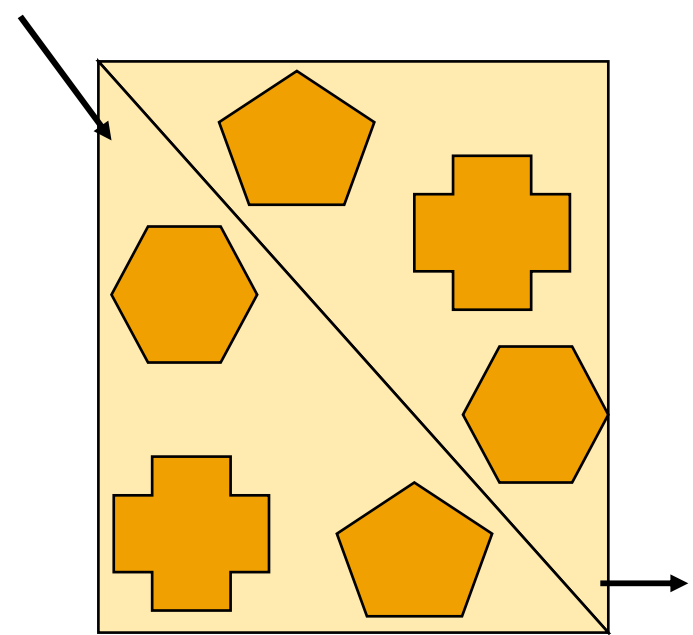
Tests on single programs / classes

Integration Tests



Tests on subsystems of programs / classes

System Tests



Tests on applications

ABAP Unit – What Should I Know ?

What is ABAP Unit?

ABAP Unit is the ABAP framework for module/unit tests.

What is an Unit?

An unit can be considered as a non-trivial, accessible code portion (method, function or form) where a given input or action causes a verifiable effect.

Ideally it is the smallest code part which can be tested in isolation.

How does an ABAP Unit test looks like?

The ABAP Unit tests are realized as methods of a local class (with the addition “FOR TESTING”).

This local class is part of the class, function group or program you want to test.

Example – Test Class Definition

```
class wallet_test definition for testing.  
  private section.  
    methods:  
      test for testing.  
endclass.
```

Identify class as test class

Identify method as test method

ABAP UNIT – What Should I Know ?

Why is the test class part of the productive code?

- ABAP Unit tests and the linked production code are in sync
- In a productive system the ABAP Unit tests are not part of the productive program load.
(-> No performance or security drawbacks)

Which services are provided by ABAP UNIT?

ABAP Unit provides a service class **CL_AUNIT_ASSERT**, which contains static methods (e.g. **ASSERT_EQUALS**) to compare e.g. strings or internal tables in order to verify test results.

Example – Test Class Implementation

class wallet_test implementation.

method test.

data: l_wallet type ref to wallet.

create object l_wallet.

l_wallet->put_in(euros = '12.50').

cl_aunit_assert=>assert_equals(

act = l_wallet->liquidity exp = '12.50'

msg = 'As many euros as just put in').

endmethod.

endclass.

Test code

Verify Test
Assertion

Result Presentation

Ergebnisanzeige von ABAP Unit

Ergebnisanzeige von ABAP Unit

Programm/Klasse/Methode	Zust	Pr.	A	L	W
CL_AUST_FAILURES=====	⊖	6	2	0	2
LCL_PRUEFE_ADDITION	⊖	1	0	0	0
PRUEFE_2_PLUS_4	⊖	0	0	0	0
PRUEFE_3_PLUS_4	⊖	1	0	0	0
LCL_TEST1_FAILURE_TYPES	⊖	3	1	0	0
TEST1_SUCCESS	⊕	0	0	0	0
TEST2_FAILURE_ASSERT	⊖	2	0	0	0
TEST3_FAILURE_EXCEPT	⊖	0	1	0	0
TEST4_FAILURE_QUIT_CL	⊖	1	0	0	0
LCL_TEST2_WARNINGS	⊖	0	0	0	1
LCL_TEST3_WARNINGS	⊖	0	0	0	1
LCL_TEST4_ABSTRACT	⊖	0	1	0	0
LCL_TEST5_WRONG_TESTS	⊖	1	0	0	0
TEST_ABSTRACT	⊖	1	0	0	0
Z_LAST	⊕	0	0	0	0
LCL_ZY_FINALLY_QUIT_PRO	⊖	1	0	0	0
TEST_FAILURE_QUIT_PR	⊖	1	0	0	0
CL_AUST_SHORT_DUMP=====	⊖	0	0	1	0
TEST_SHORT_DUMP	⊖	0	0	1	0
CL_AUST_TESTSAMPLE=====	⊖	4	3	0	0
LCL_TEST	⊖	1	0	0	0
TEST_SAMPLE	⊖	1	0	0	0
TEST1	⊖	1	0	0	0
TEST11	⊖	1	0	0	0
Z_LAST	⊕	0	0	0	0
TEST2	⊖	2	0	0	0
TEST11	⊖	1	0	0	0
TEST21	⊖	1	0	0	0
TEST_ERROR	⊖	0	1	0	0
TEST_ZERO_DIVISION	⊖	0	1	0	0
TEST_SETUP_ERROR	⊖	0	1	0	0
TEST_TEARDOWN_ERROR	⊖	0	1	0	0
TEST	⊕	0	0	0	0
RS_AUST_ASSERT	⊖	21	0	0	0
TEST_ASSERT	⊖	21	0	0	0
TEST_ASSERT_BOUND	⊖	3	0	0	0
TEST_ASSERT_CHARACT	⊖	2	0	0	0

Art der...	Meldungstext
⚠	Kritischer Prüffehler: '3 + 4 != 8'
⚠	Kritischer Prüffehler: 'failure, but quit not'
⚠	Kritischer Prüffehler: 'failure, quit method'
⚠	Ausnahmefehler <COMPUTE_INT_ZERODIVIDE>
⚠	Fataler Prüffehler: 'failure, quit class'
⚠	'SETUP'-Methode muss öffentlich (public) sichtbar sein.
⚠	Konstruktor der Testklasse 'LCL_TEST3_WARNINGS' muss parameterlos sein.
⚠	Ausnahmefehler <CREATE_OBJECT_CLASS_ABSTRACT>
⚠	Toleranter Prüffehler: 'CORRECTLY executed: test redefinition of abstract test method.'
⚠	Kritischer Prüffehler: 'failure, quit program'
⚠	Laufzeitfehler <LOAD_PROGRAM_NOT_FOUND>
⚠	Kritischer Prüffehler: 'float'
⚠	Kritischer Prüffehler: 'fail: test1/test11'

Einzelheiten/...

Infos

verschiedene Werte:

erwartet [S-Table[2x110] of \TYPE=TSDIR].

tatsächlich [S-Table[1x110] of \TYPE=TSDIR].

Tabellenwert Index [2] der erwarteten Tabelle nicht in tatsächlicher Tab

erwarteter Wert [[ABC |1234]

Stack

in RS_AUST_ASSERT (Zeile:283)

BIN (1) (000) Is0347 INS



Demo ABAP Unit

ABAP Unit Assertion Class

```
CLASS CL_AUNIT_ASSERT ...
```

```
CLASS-METHODS:
```

```
fail IMPORTING
```

```
msg ...
```

```
level ...
```

```
quit ...
```

```
assert_equals
```

```
IMPORTING
```

```
exp ...
```

```
act ...
```

```
msg ...
```

```
level ...
```

```
quit ...
```

```
tol ...
```

```
assert_initial ...
```

```
assert_not_initial ...
```

```
assert_bound ...
```

```
assert_not_bound ...
```

```
...
```

Method "fail":
Executes test failure
immediately

Methods "assert_*":
Verifies condition. If false,
executes test failure.

Choice of "assert*" methods

ABAP Unit Assertion Class

```
CLASS CL_AUNIT_ASSERT ...  
  
CLASS-METHODS:  
  fail IMPORTING  
    msg ...  
    level ...  
    quit ...  
  assert_equals  
    IMPORTING  
      msg ...  
      act ...  
      exp ...  
      tol ...  
      quit ...  
      level ...  
  assert_initial ...  
  assert_not_initial ...  
  assert_bound ...  
  assert_not_bound ...  
  ...
```

Failure message for
framework

Severity of failure, indicated
by framework:
TOLERANT, CRITICAL, FATAL

Consequences for test case:

- NO:
 continue testing
- METHOD:
 quit test case
- CLASS:
 quit test class
- PROGRAM:
 quit all tests for current
 program

ABAP Unit Assertion Class

```
CLASS CL_AUNIT_ASSERT ...
```

```
CLASS-METHODS:
```

```
fail IMPORTING  
    msg ...  
    level ...  
    quit ...  
assert_equals  
    IMPORTING  
    msg ...  
    act ...  
    exp ...  
    tol ...  
    level ...  
    quit ...  
assert_initial ...  
assert_not_initial ...  
assert_bound ...  
assert_not_bound ...  
...
```

Actual value

Expected value

the tolerance when
comparing floating
point numbers



Positioning

The New ABAP Editor

Dynamic Programming & RTTS

Code Inspector

ABAP Unit

Summary

Q & A



Venky Varadadesigan

Product Manager
SAP NetWeaver

SAP Labs, LLC

Westchase Corporate Center
10111 Richmond Ave., Suite 600
Houston, TX 77042

T 832.287.2135

E venkata.varadadesigan@sap.com



Stay Tuned...

March 24, 2005 (Friday)

Web Dynpro for ABAP

- 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®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® 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®, Informix and Informix® Dynamic Server™ are trademarks of IBM Corporation in USA 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®, the Citrix logo, ICA®, Program Neighborhood®, MetaFrame®, WinFrame®, VideoFrame®, MultiWin® and other Citrix product names referenced herein are trademarks of Citrix Systems, Inc.
- HTML, DHTML, XML, XHTML 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.
- MarketSet and Enterprise Buyer are jointly owned trademarks of SAP AG and Commerce One.
- SAP, SAP Logo, R/2, R/3, mySAP, mySAP.com 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 trademarks of their respective companies.

- Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.
- Die von SAP AG oder deren Vertriebsfirmen angebotenen Softwareprodukte können Softwarekomponenten auch anderer Softwarehersteller enthalten.
- Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® und SQL Server® sind eingetragene Marken der 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®, Informix und Informix® Dynamic Server™ sind Marken der IBM Corporation in den USA und/oder anderen Ländern.
- ORACLE® ist eine eingetragene Marke der ORACLE Corporation.
- UNIX®, X/Open®, OSF/1® und Motif® sind eingetragene Marken der Open Group.
- Citrix®, das Citrix-Logo, ICA®, Program Neighborhood®, MetaFrame®, WinFrame®, VideoFrame®, MultiWin® und andere hier erwähnte Namen von Citrix-Produkten sind Marken von Citrix Systems, Inc.
- HTML, DHTML, XML, XHTML sind Marken oder eingetragene Marken des W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.
- JAVA® ist eine eingetragene Marke der Sun Microsystems, Inc.
- JAVASCRIPT® ist eine eingetragene Marke der Sun Microsystems, Inc., verwendet unter der Lizenz der von Netscape entwickelten und implementierten Technologie.
- MarketSet und Enterprise Buyer sind gemeinsame Marken von SAP AG und Commerce One.
- SAP, SAP Logo, R/2, R/3, mySAP, mySAP.com und weitere im Text erwähnte SAP-Produkte und -Dienstleistungen sowie die entsprechenden Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und anderen Ländern weltweit. Alle anderen Namen von Produkten und Dienstleistungen sind Marken der jeweiligen Firmen.