

Authorization Objects – A Simple Guide

Applies to:

Netweaver 2004s Web Application Server SPS7

Summary

This guide is intended to demonstrate how to create and use the Authorization Concept in the most simplest of conditions. It is based on the Netweaver Web Application Server, and will utilize a table available in all R/3 systems.

Author(s): Glen Spalding

Company: gingle Ltd

Created on: 24th May 2006

Author Bio



There did not appear to be any simple guide and explanation on how to create authorizations in the most simplest manner. Therefore, I thought I would create this basic guide that, I hope, explains the main principles and tasks needed for the beginner.

This Authorization Object method should only be utilized in the most basic of uses.

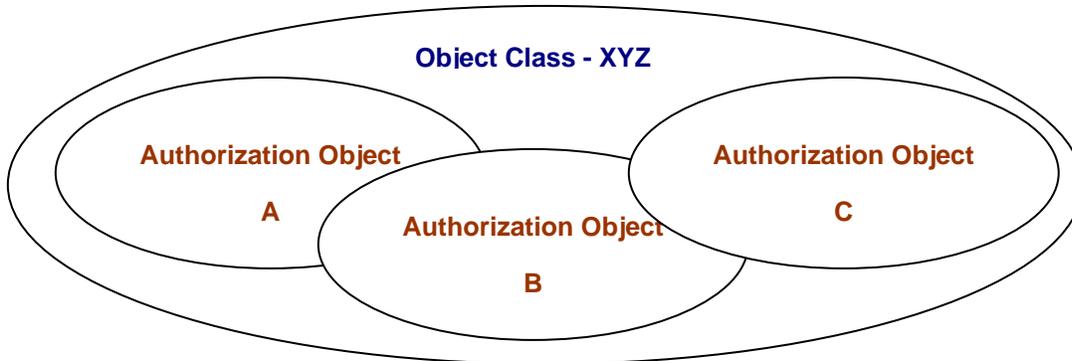
Table of Contents

Introduction	3
Scenario	4
Create Authorization Field	5
Create Authorization Class & Object	6
Create Role, Profile & Authorization	9
Assign Role to User	12
Code the Authorization Check	13
Testing	14
Copyright.....	15

Introduction

The Authorization Object mechanism is used to inspect the current user's privileges for specific data selection and activities from within a program.

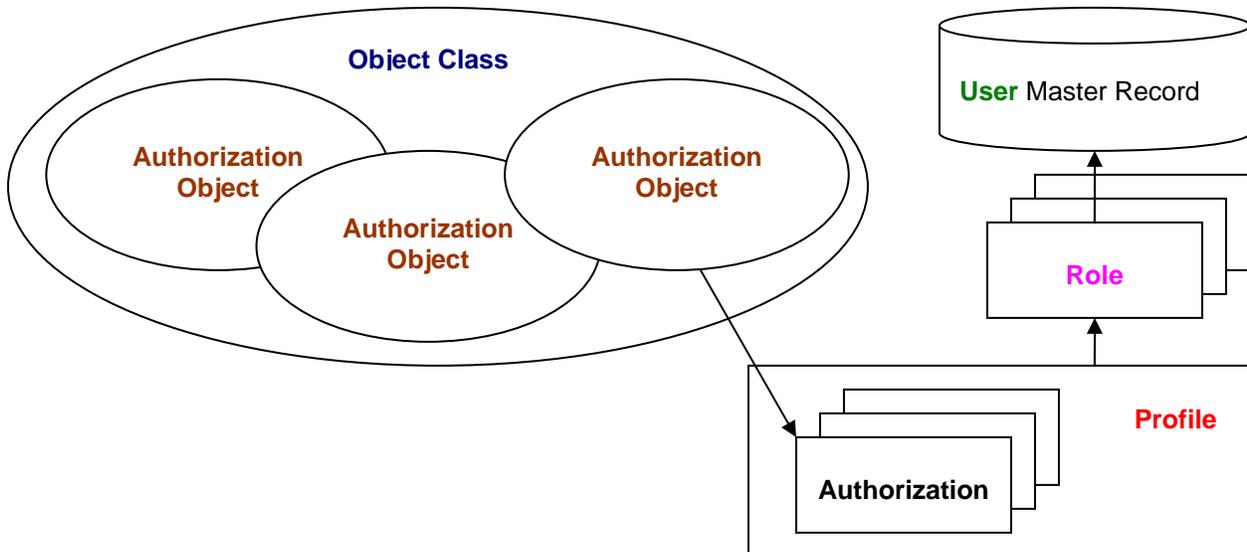
An **Object Class** contains one or more **Authorization Objects**.



The **Authorization Object** is where Permitted Activity configurations are performed against specific *fields*. E.g. Change (being the activity) the material's text – MAKTX (being the specific *field*), or Read (being the activity) a certain Customer (using Customer Number – KUNNR, as the specific *field*).

Before a **User** can be granted permission by the **Authorization Object**, the **User's** Master Record is assigned a **Role**, which includes a **Profile**.

The **Profile** contains what is simply called the **Authorization** and is where the specific data for the **Authorization Object's** *field* is assigned to the configured Permitted Activity. E.g. Allow changes to any Material Text, or read Customers between the ranges "AA100" & "BB999".



Finally the calling of the **Authorization Object** can be performed in code.

Scenario

We will be using table "TSTC" – Transaction Codes, which should exist in any R/3 version. The screen shots are taken from the SAP Netweaver 2004s Release 7.

We will demonstrate the selection of a record from this table, and due to the privileges revoked from the user, via an Authorization Object, the selection will be denied.

We will create a specific Authorization Field for which the check will be made against.

Then the Authorization Class and Authorization Object, in which the Field previously mentioned is added.

A new Role and Profile will contain the actual Authorization for data.

The Role will be assigned to the User Master Data.

Finally the Authorization Object will be called in Code.

Create Authorization Field

Transaction – SU20

Create a new Authorization Field by clicking on the Create button.

List of Authorization Fields



Enter “ZTCODE” and “TCODE” in the Field Name and Data Element, then Enter.

Authorization field

Field Name	ZTCODE
Data element	TCODE
Transaction Code	

Search Help for Auth. Values in the Profile Generator

Table Name	
Or DTEL Search Help	
Or DOMA Value Table	TSTC
<input type="checkbox"/> Or Fixed Domain Values	

Data element

Package	SZZ
Pers.Responsib.	SAP

Domain

Domain	TCODE
Length	20
Conversion Routine	

Use in Authorization Objects

Object Class	Authorization Object
--------------	----------------------

Notice the “Use in Authorization Objects” area at the bottom of the display.

Naturally, as we have just created this Field, it is not yet utilized in any Authorization Object.

Save, a “Local Object” will suffice.

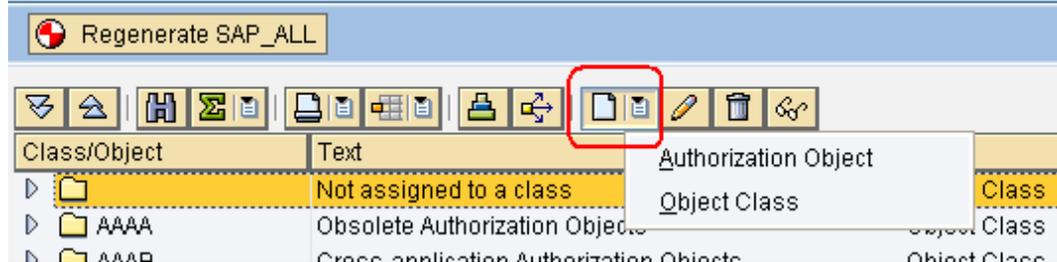
The Field has now been created for use in any Authorization Object.

Create Authorization Class & Object

Transaction – SU21

Create a new Authorization Class (Object Class) by clicking on the Create button's drop down icon, then select "Object Class".

Maintain the Authorization Objects



Enter the new Object Class name, give it a description and Save.

Object class	ZTRN
Text	TSTC Transaction code check
Author	

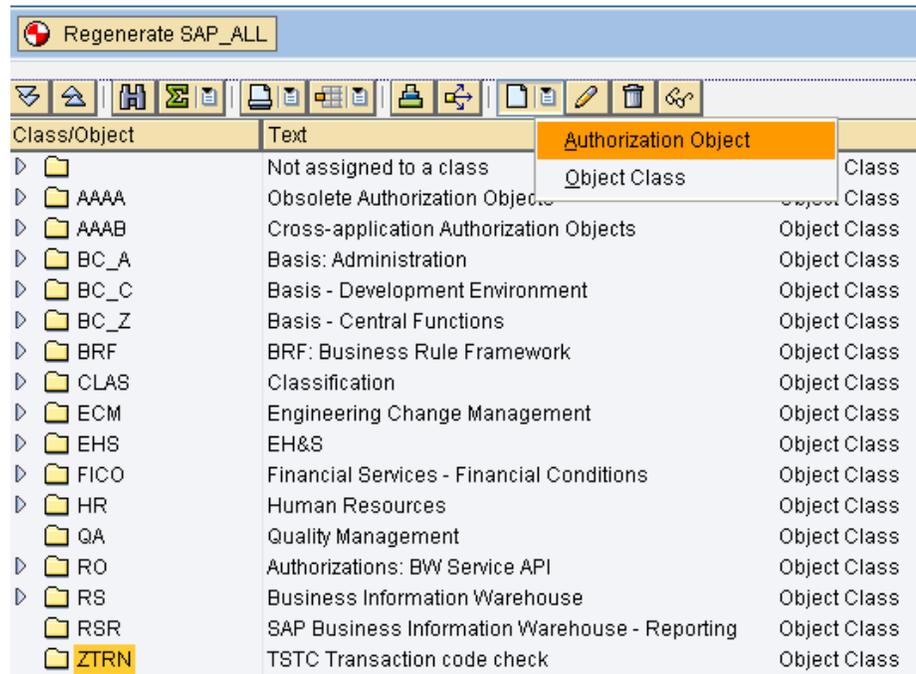
Save (floppy disk icon) Cancel (X icon)

Again, saving as a "Local Object" will suffice.

We now have the Object Class to add the new Authorization Object.

Select the newly created Object Class, and perform a similar action to before. Click on the Create button's drop down, this time selecting "Authorization Object".

Maintain the Authorization Objects



In the Authorization Object's create screen, enter a Name, and description.

Under the section "Authorization fields" enter two Field names. One being "ACTVT", this is going to be responsible for the activities that will be permissible, and the other "ZTCODE" which is the Authorization Field, created earlier.

Note: If a suitable Authorization Field already exists, it is possible to re-use it. However, for this example, we are assuming it did not, to give exposure to all necessary tasks involved when dealing with Authorization Objects.

Object	Z_TCODE
Text	Check for Transaction Code
Class	ZTRN TSTC Transaction code check
Author	

Authorization fields	
Field name	Heading
ACTVT	Activity
ZTCODE	Transaction Code

Be careful when navigating this screen, as it is a bit buggy.

Now press the "Permitted activities" button, at the bottom of the Create Authorization Object screen, to begin configuring what actions can be taken against our new field ZTCODE.

Save when prompted.

At the next popup, simply press the tick button to continue.

Now we should be at the Define Values for the ACTVT field, where we will select 01, 02, and 03.

Save and exit.

All our Authorization Objects have now been created. Back out **ALL THE WAY** and check the creation and configuration in display mode.

Object	Z_TCODE	Check for Transaction Code
Field name	ACTVT	Activity

Activities		
Sel.	Activity	Text
<input checked="" type="checkbox"/>	01	Create or generate
<input checked="" type="checkbox"/>	02	Change
<input checked="" type="checkbox"/>	03	Display
<input type="checkbox"/>	04	Print, edit messages

Having assigned the Authorization Field to the Authorization Object just created. Return back to the Authorization Field – SU20, and check that the Field is actually assigned.

Double click in the “ZTCODE” Authorization Field line

List of Authorization Fields



Authorization check fields	
Authorization fld	Data element
XMIEXTPGM	XMILOGPROG
XMILOGACC	XMILOGACC
ZTCODE	TCODE

On the next screen in the “Use in Authorization Objects” section, see the assignment.

Field Name: ZTCODE
 Data element: TCODE
 Transaction Code

Search Help for Auth. Values in the Profile Generator

Table Name:
 Or DTEL Search Help
 Or DOMA Value Table: TSTC
 Or Fixed Domain Values

Data element

Package: SZZ
 Pers.Responsib.: SAP

Domain

Domain: TCODE
 Length: 20
 Conversion Routine:

Use in Authorization Objects

Object Class	Authorization Object
ZTRN	Z_TCODE

Create Role, Profile & Authorization

Transaction – PFCG

We now have to create a Role, in which a new Profile will be added, and also an Authorization will be added that is responsible for permitting activities against specific data(fields) in the database – the actual authorization.

Role	ZTRANSCODE	  	 Role
Name			

Enter a Role name and press the create Role button, then supply a description and Save.



Create Roles

  Other role  

Role	ZTRANSCODE
Description	Transaction role

On the Authorizations tab, in the “Maintain Authorization Data and Generate Profiles” area, press the “Change Authorization Data” button.

On the next popup screen – “Choose Template”, select the “Do not select templates” option.



Created by		Last Ch	
User		User	
Date		Date	
Time	00:00:00	Time	

Information About Authorization Profile

Profile Name		
Profile Text		
Status	No authorization data exists	

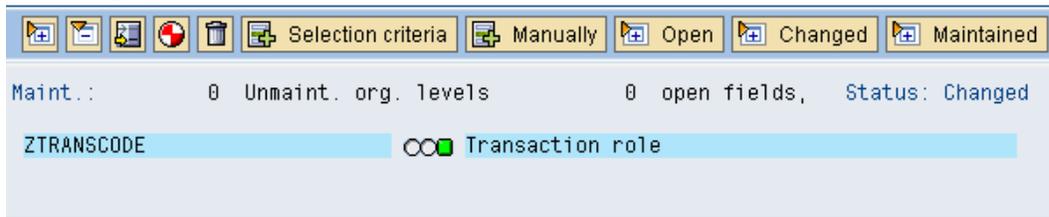
Maintain Authorization Data and Generate Profiles

 Change Authorization Data

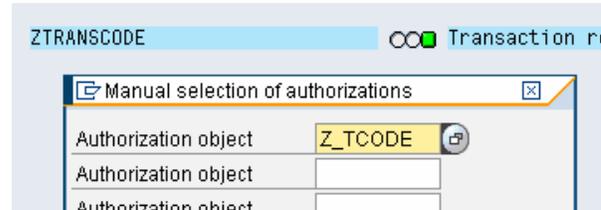
 Expert Mode for Profile Generation

We are now in the Authorizations area where we will add specific activities to field data.

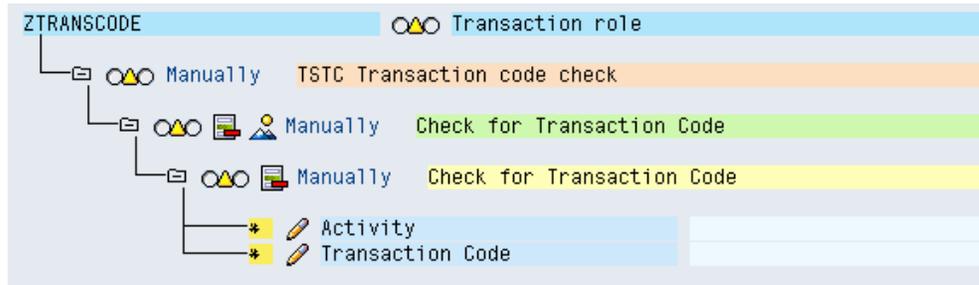
Change role: Authorizations



Press the button “Manually”, (Ctrl + Shift + F9) and enter the Authorization Object “Z_TCODE” created earlier. Select the Tick button to continue.



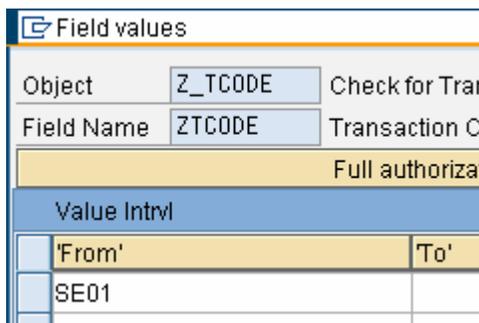
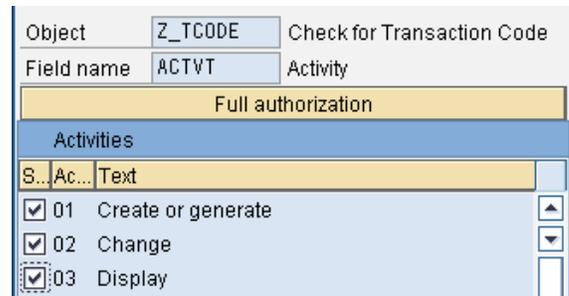
Expand all nodes.



Press the edit icon or line, of the “Activity” entry.



Select all three Activities that were earlier permitted, and Save.



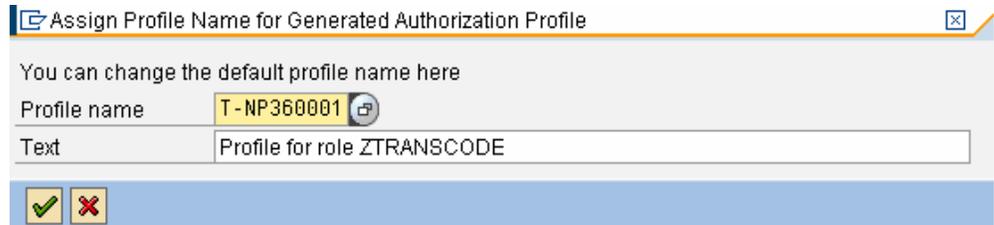
Now select the edit icon or line, of the “Transaction Code” entry, and enter “SE01” in the “From” field, and Save.

This current Authorization will permit a Create, Change, or Read activity against 'SE01' data, in the field ZTCODE, which is based on the Data Element TCODE. So wherever TCODE is used, e.g. in table TSTC, we can now being to utilize the Authorization Object Z_TCODE.

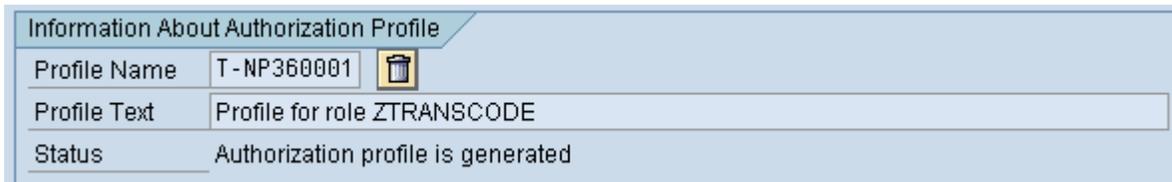
Generate the Authorization using the generation button.



Accept the default values for the Profile which will be created.



Return to the previous Role screen, and notice that we now have a Profile assigned to our Role, in the "Information About Authorization Profile" area.



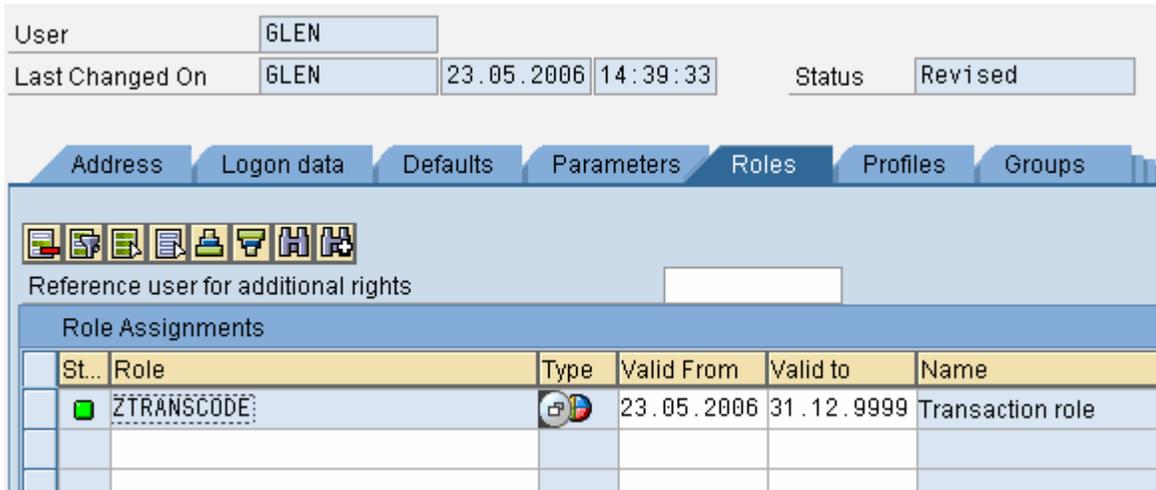
That completes the Role, Profile, and Authorization creation and configuration.

Assign Role to User

Transaction – SU01

Note: It is not in scope to explain how to create a user, so either, create a suitable user now, or select an appropriate one, so that the Role can be assigned. Also, make sure the user is able to execute a program in SE38, as this is how the Authorization will be tested and demonstrated.

Choose the User, and in Edit mode, select the Roles tab.



The screenshot shows the SAP SU01 'Roles' tab for user GLEN. The user's last changed on date is 23.05.2006 at 14:39:33, and the status is Revised. The 'Roles' tab is selected, and the 'Role Assignments' table is visible. The table has columns for Status, Role, Type, Valid From, Valid to, and Name. One role is assigned: ZTRANSCODE, which is a Transaction role, valid from 23.05.2006 to 31.12.9999.

St...	Role	Type	Valid From	Valid to	Name
<input checked="" type="checkbox"/>	ZTRANSCODE		23.05.2006	31.12.9999	Transaction role

Assign the Role recently created, press Enter and Save.

Note: If the User is currently logged on, the User will have to log off and on again before the new Role assignment can be utilized.

Code the Authorization Check

Create the program as seen below to test the Authorization.

Note the Authorization check with the syntax beginning AUTHORITY-CHECK and the checking of the sy-subrc. Also, the '03' literal that is being passed into the Object check field "ACTVT" which denotes a "read", and the p_tcode parameter being passed into the Object check field "ZTCODE" which represents the actual data, wishing to "read".

```
REPORT   zauth_check_demo.

DATA: wa_tstc TYPE tstc.
PARAMETERS: p_tcode TYPE tcode.

AUTHORITY-CHECK OBJECT 'Z_TCODE'
              ID 'ACTVT' FIELD '03'           " read access
              ID 'ZTCODE' FIELD p_tcode.     " actual value

IF sy-subrc EQ 0.                               " check authorization

* fetch record
  SELECT SINGLE *
    FROM tstc
    INTO wa_tstc
    WHERE tcode EQ p_tcode.

  WRITE:/ wa_tstc-tcode,
          wa_tstc-pgmna,
          wa_tstc-dypno,
          wa_tstc-menue,
          wa_tstc-cinfo,
          wa_tstc-arbgb.

ELSE.

* bad authorization
  WRITE:/ 'Bad Authorization'.

ENDIF.
```

Testing

Transaction – SE38 with appropriate Test User, that has been given the Role previously created.

Execute the program above, in this case ZAUTH_CHECK_DEMO.

Enter an permitted value and run the program.

Authorization Test Demo



A screenshot of the SAP transaction SE01 input screen. The title bar shows a green checkmark icon. Below the title bar, the text "Transaction Code" is followed by a yellow input field containing the value "SE01".

Result

Authorization Test Demo

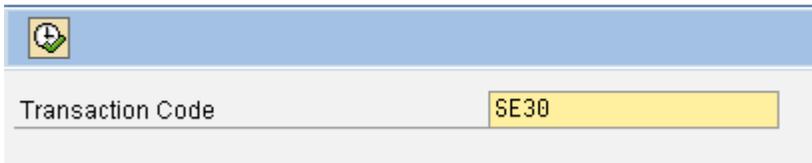


A screenshot of the SAP authorization test result for transaction SE01. The title bar shows a green checkmark icon. Below the title bar, the text "Authorization Test Demo" is displayed. A table with four columns shows the results: "SE01", "RDDM0001", "0200", and "0".

Authorization Test Demo			
SE01	RDDM0001	0200	0

Now enter any other value, and see the difference

Authorization Test Demo



A screenshot of the SAP transaction SE30 input screen. The title bar shows a green checkmark icon. Below the title bar, the text "Transaction Code" is followed by a yellow input field containing the value "SE30".

Result

Authorization Test Demo



A screenshot of the SAP authorization test result for transaction SE30. The title bar shows a green checkmark icon. Below the title bar, the text "Authorization Test Demo" is displayed. A table with one row shows the result: "Bad Authorization".

Authorization Test Demo
Bad Authorization

Copyright

© Copyright 2006 SAP AG. All rights reserved.

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, Informix, i5/OS, POWER, POWER5, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated 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.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" 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 not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. 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.

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.