

# Integrating BW Process Chains with BusinessObjects Reports



## Applies to:

SAP BW 7.0 EHP1, BusinessObjects BOE XI 3.1. For more information, visit the [Business Objects homepage](#).

## Summary

Integrate the process chain completion on BW to execution of reports on BOBJ side by creating a process type that can execute an ABAP program to create BusinessObjects file events and also return the status of file creation with Red or Green status to ensure correct triggering of BOBJ events and reports.

At present, there is no standard functionality available in SAP NetWeaver 2004s BW/BOBJ that offers this integration and automation.

This has been implemented at one of our Max Attention customer as part of RUNSAP like a Factory project.

**Authors:** Srinivas Rapthadu, Bhavani Doppalapudi and Indraneel Chatterjee

**Company:** SAP America Inc, SAP Labs India

**Created on:** 01 June 2011

## Author Bio



Srinivas Rapthadu works as Solution Architect at SAP America with over 15 years of experience. His core competency is in the field of Enterprise Information Management and Business Intelligence portfolio including SAP NetWeaver, BW, BW Accelerator, BusinessObjects, BO Explorer, BPC, and in-memory solutions.



Bhavani Doppalapudi works as a Senior Software Engineer in SAP Labs India for three years with an overall IT experience of five years. Her core competencies include BW Warehouse management.



Indraneel Chatterjee is an SAP Certified Associate. He has 11 years of experience in SAP and overall IT experience of 14 years. His core competency is Business Intelligence (BI), BusinessObjects, and BusinessObjects Explorer. He has also worked on BW, Databases, Data Warehouse Design, UNIX and web technologies. He is currently working with SAP as a Senior Support Engineer.

## Table of Contents

Introduction .....	3
File system mapping at OS Level .....	4
BW Steps .....	4
Creation of Custom Process Type .....	5
Event Trigger ABAP Program .....	8
Create Variants for Event Trigger files .....	9
Attach the Custom Process Type to Process Chains/Meta Chain .....	10
BusinessObjects Steps .....	12
Creation of File Based Event .....	12
Attach the Event to the Report .....	14
Related Code .....	16
Process Type methods code .....	16
IF_RSPC_EXECUTE~EXECUTE .....	16
IF_RSPC_EXECUTE~GIVE_CHAIN .....	16
IF_RSPC_CALL_MONITOR~CALL_MONITOR .....	16
IF_RSPC_CHECK~GIVE_ALL .....	17
IF_RSPC_CHECK~CHECK .....	17
IF_RSPC_GET_STATUS~GET_STATUS .....	17
IF_RSPC_GET_VARIANT~EXISTS .....	17
IF_RSPC_GET_VARIANT~WILDCARD_ENABLED .....	18
IF_RSPC_GET_VARIANT~GET_VARIANT .....	18
IF_RSPC_MAINTAIN~GET_HEADER .....	18
IF_RSPC_MAINTAIN~MAINTAIN .....	19
IF_RSPC_TRANSPORT~GET_TLOGO .....	19
IF_RSPV_TRANSPORT~GET_ADDITIONAL_OBJECTS .....	19
Event Triggering Program .....	20
Related Content .....	22
Copyright .....	23

## Introduction

Currently there is no mechanism to trigger the BusinessObjects (BOBJ) report to refresh automatically once the corresponding process chains are completed on the BW side. This often leads to one of the following:

- having old data in the BOBJ reports
- needing to manually refresh the reports after checking the status of the corresponding process chains.

To overcome this issue, this article describes the approach to create a process executing an ABAP program. The program generates file events to trigger BOBJ events and automatic report execution.

The following are the common scenarios using the approach described in this document:

- BOBJ Reports dependent on a single process chain
- BOBJ Reports dependant on multiple process chains (Meta Chain)
- BOBJ Reports depending on a single Process Chain and others on multiple process chains.

Below is a summary of the steps involved:

- Create a mount or share drive that can be accessed by BW system and BOBJ system.
- Create a Custom ABAP Process Type that returns Status Red or Green, an ABAP program that creates events on the mount or share.
- Create program variants to handle multiple reports. Attach the custom process type with specific program variant to the desired process chain.
- Create the File based Event on the BOBJ side and assign this event to the desired Report.

The desired report is triggered automatically once the associated process chain from BW is successfully finished.

These steps are outlined in detail below.

## File system mapping at OS Level

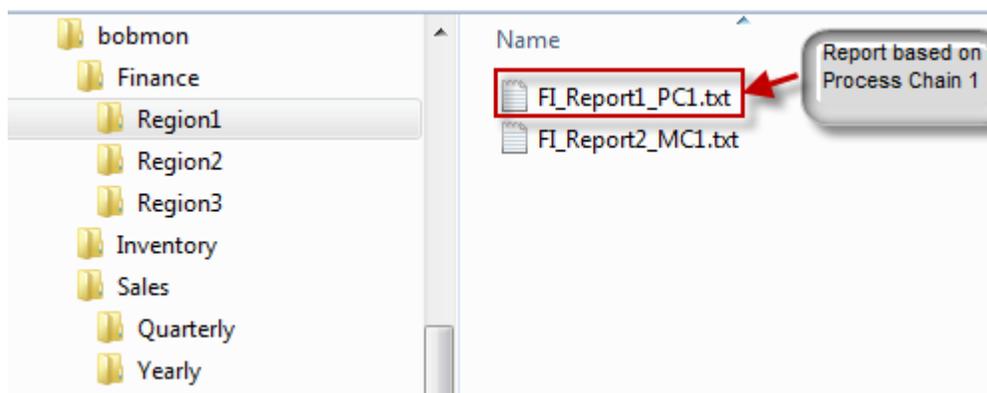
1. Create a mount or share drive that can be accessed by BW system and BOBJ system with full rights with an admin user. For example, **/bobmon** is the mount to which we can write files from the BW system. These files can be read from the BOBJ side.

/bobmon/pc\_sales\_rep2/pc\_sales\_rep2.txt

2. Define a folder structure under the mount to handle multiple Events, Process Chains and Reports.

For example, define Sales, Finance, and Inventory as the key areas. The structure of the event files can be as below. In this example, we have two reports Report1 and Report2 depending on Process Chain1 (PC1) and Meta Chain1 (MC1) respectively for Region1.

The events created from the BW side are FI\_Report1\_PC1.txt and FI\_Report2\_MC1.txt respectively. File based events on BOBJ are created based on these files to trigger the BOBJ reports Report1 and Report2.



## BW Steps

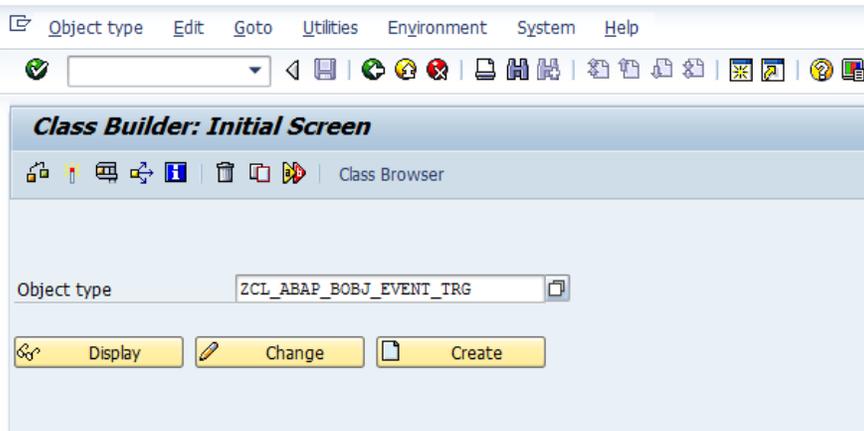
From the BW side, following needs to be completed:

- Creation of Custom ABAP Process Type that returns status of the execution (RED or GREEN).
- Creation of Event Trigger ABAP Program
- Creation of Variants for Event Trigger files
- Attach the Custom Process Type to Process Chains/Meta Chain

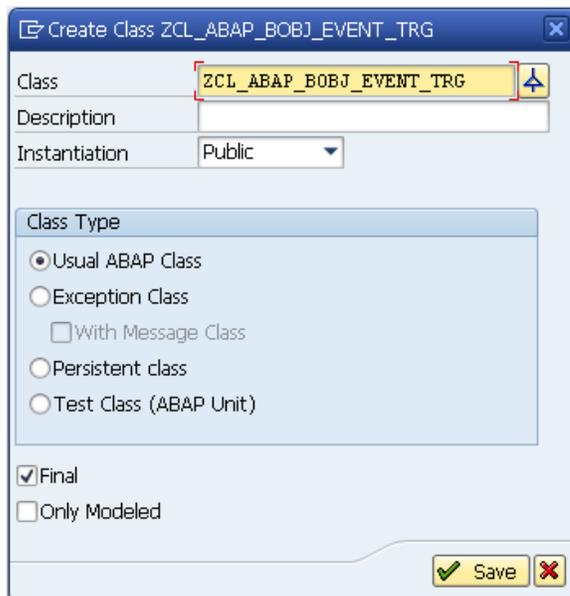
## Creation of Custom Process Type

Follow the steps below to create the Custom Process Type. Code for the implemented methods can be found in the [Related Code](#) section.

1. Enter the Class builder using SE24. Specify the name of the class. Click the **Create** button.

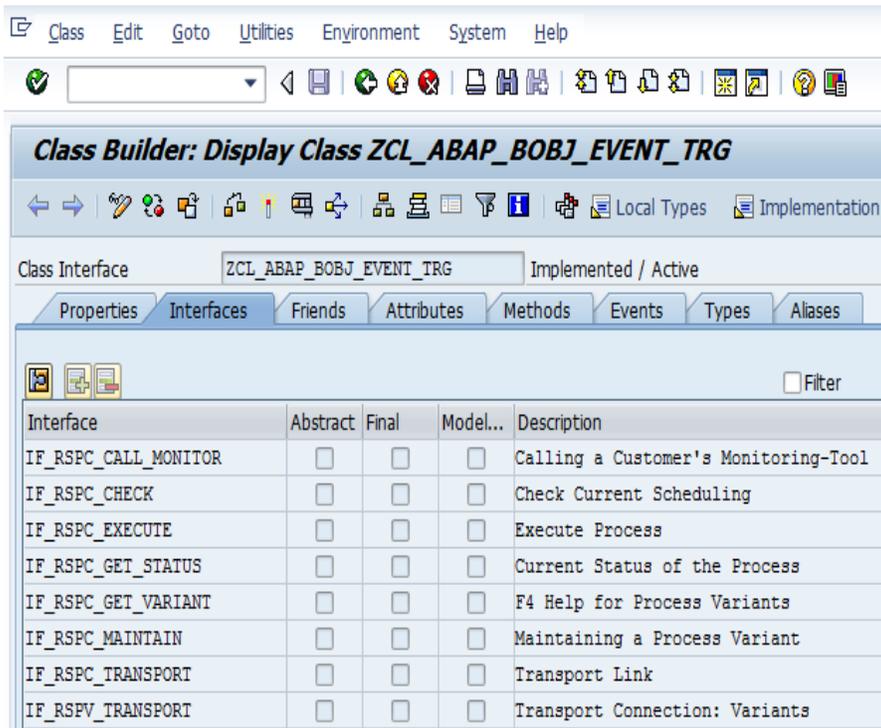


2. Choose the Class Type as **Usual ABAP Class**.

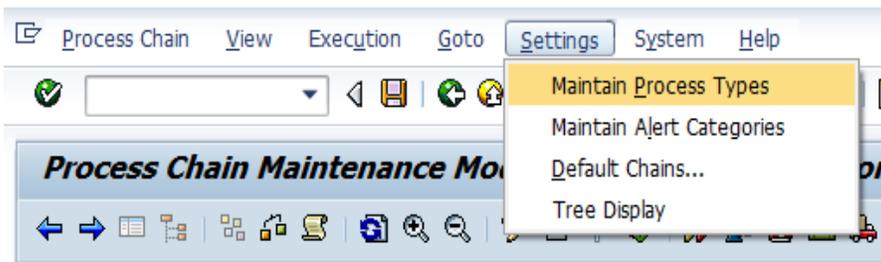


3. Switch to the Interfaces tab and specify the following interfaces:

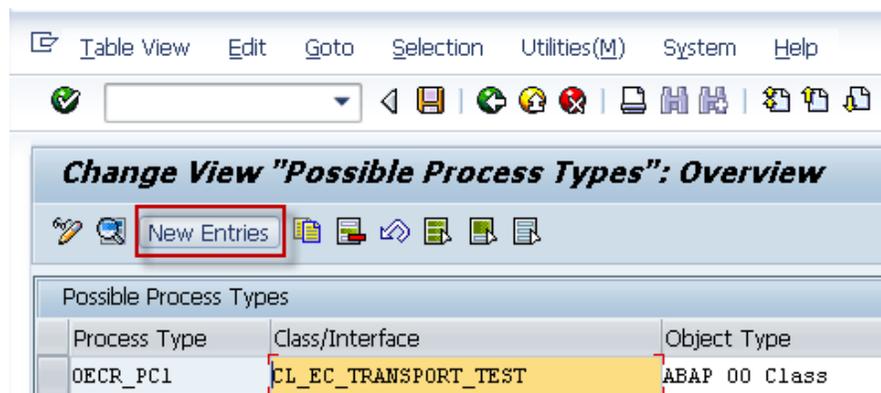
**IF\_RSPC\_CALL\_MONITOR** – Calling a Customer's Monitoring-Tool  
**IF\_RSPC\_CHECK** – Check Current Scheduling  
**IF\_RSPC\_EXECUTE** – Execute Process  
**IF\_RSPC\_GET\_STATUS** – Current Status of the Process  
**IF\_RSPC\_GET\_VARIANT** – F4 Help for Process Variants  
**IF\_RSPC\_MAINTAIN** – Maintaining a Process Variant  
**IF\_RSPC\_TRANSPORT** – Transport Link  
**IF\_RSPV\_TRANSPORT** – Transport Connection: Variants



4. In Transaction RSPC, go to **Maintain Process Types** on the menu bar.



5. In the following screen, click the **New Entries** button.



6. Give a name for the New Process type (Z\_BWBO\_TRG for example). Give the name of the class in ObjectTypeName.

**Change View "Possible Process Types": Details**

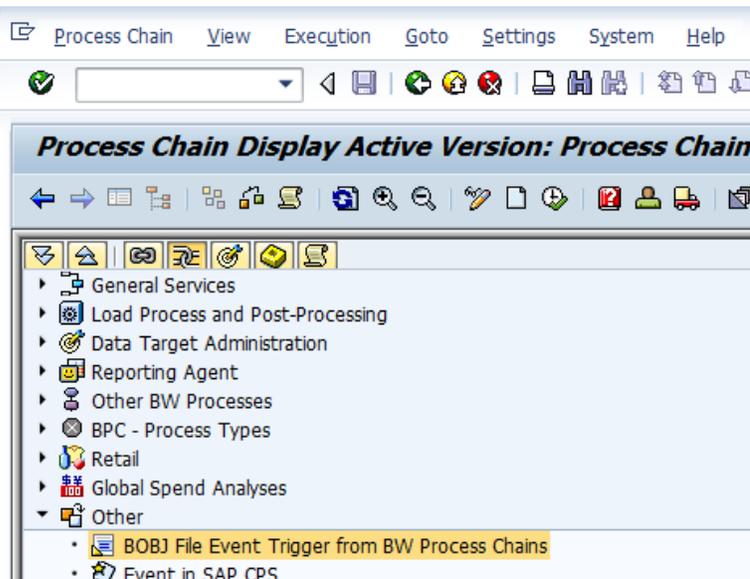
Process Type: Z\_BWBO\_TRG

Possible Process Types

Short description	BOBJ File Event Trg
Long description	BOBJ File Event Trigger from BW Process Chains
ObjectTypeName	ZCL_ABAP_BOBJ_EVENT_TRG
Object Type	ABAP OO Class
Possible Events	Process ends "successful" or "incorrect"

Repeatable  
 Repairable  
 ID: 99V9  
 Internal Name  
 Own Mail  
 Process Category: 98  
 Two-digit no.:  
 Documentation Type:  
 Docu. Object:  
 Only in BW Clients  
 Component:

7. The new process type is underneath the category specified. In this example it is **Other > BOBJ File Event Trigger from BW Process Chains**.

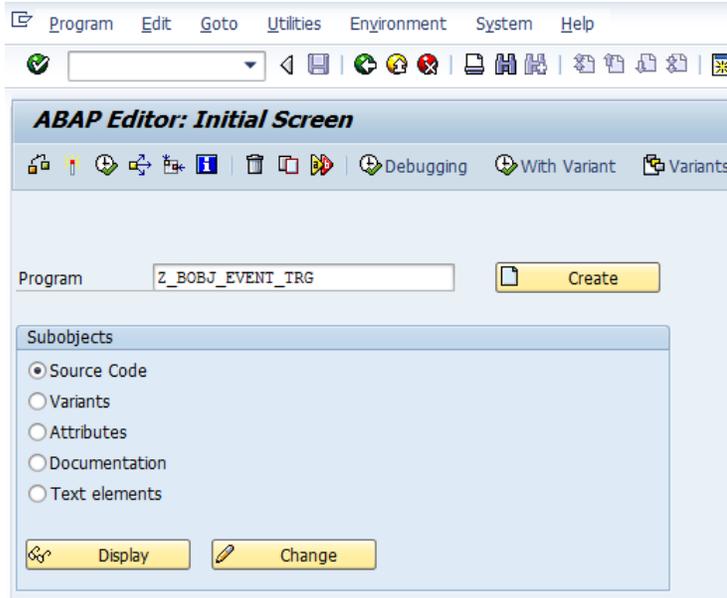


## Event Trigger ABAP Program

Follow the steps below to create the Event Trigger ABAP program. Code for the implemented program can be found in the [Related Code](#) section.

1. Create ABAP program Z\_BOBJ\_EVENT\_TRG with the logic to create an event file based on the completion of process chain or meta chain using the variant information.

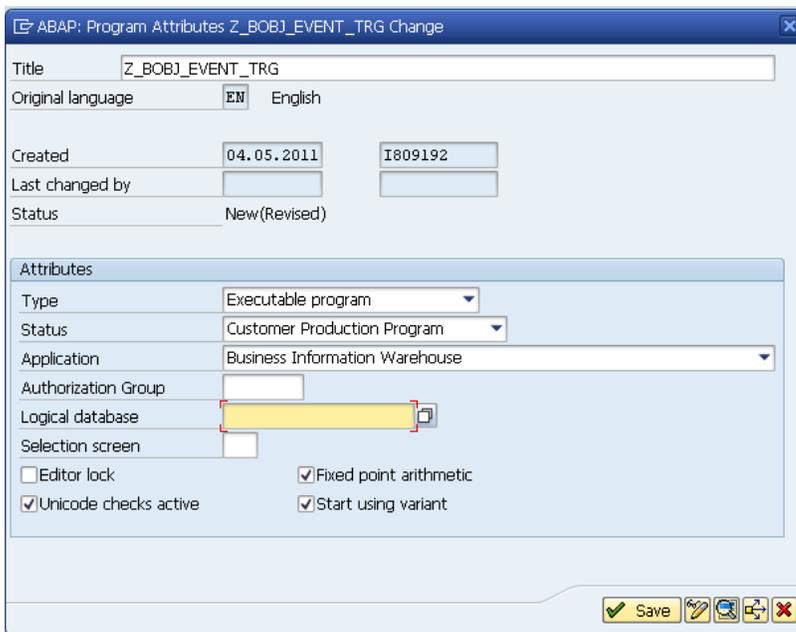
Also handle the deletion of the file if an older version already exists.



- Ensure that the users are given appropriate authorizations to access the files, so that the authorizations are not misused.
- For more sophisticated authorization checks to the files you can also use the function module FILE\_VALIDATE\_NAME. You can find more information regarding the same here at the link below.

[http://help.sap.com/saphelp\\_em70/helpdata/en/4c/8d0a9af52b3543e1000000a15822b/content.htm](http://help.sap.com/saphelp_em70/helpdata/en/4c/8d0a9af52b3543e1000000a15822b/content.htm)

2. Select the options shown for the program to be created.



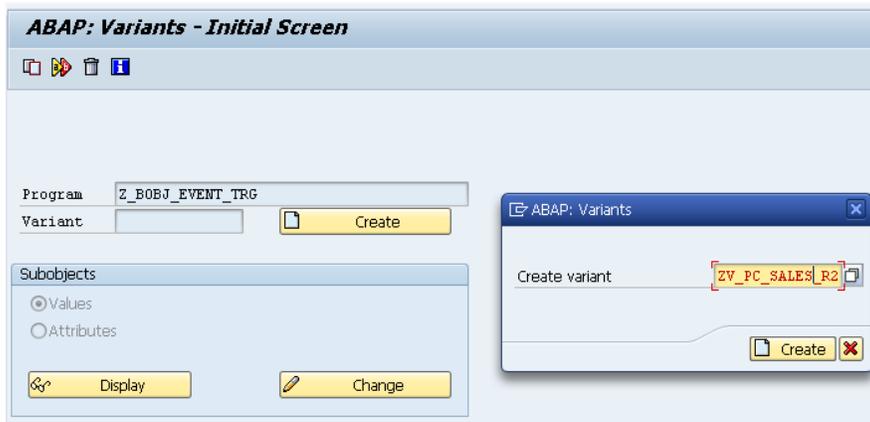
## Create Variants for Event Trigger files

Create variants to manage multiple event file creation to handle events from various process chains and meta chains.

1. Create Variants to handle multiple reports and process chains easily. In this example, two variants are created:

ZV\_PC\_SALES\_R2

ZV\_MC\_INV\_REP1



2. Specify the event names to be created on the share / mount.

/bobmon/pc\_sales\_rep2/pc\_sales\_rep2.txt

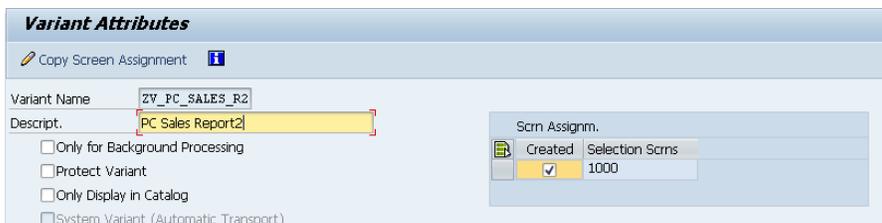
/bobmon/Meta\_Inv\_Rep1/Meta\_Inv\_Rep1.txt



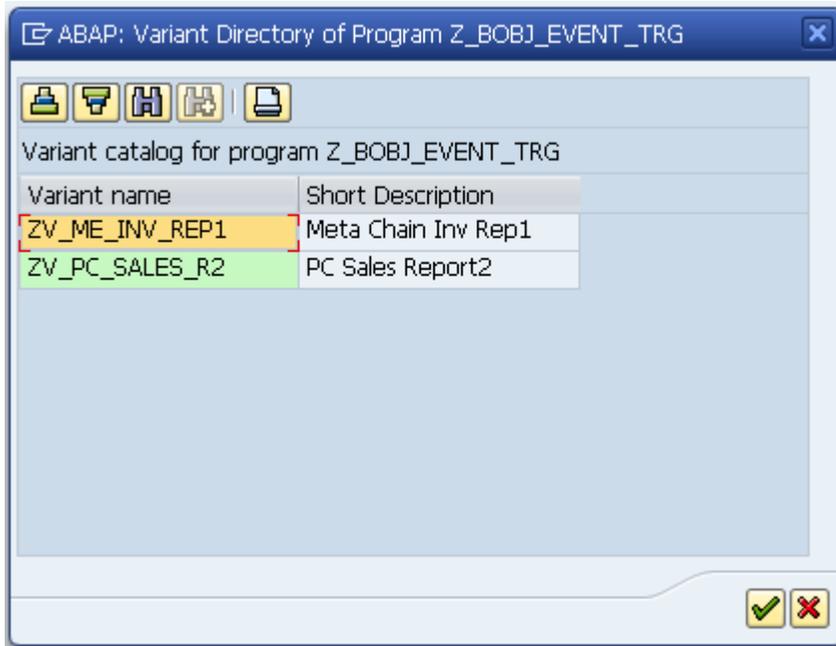
3. Enter the description of the Variant.

PC Sales Report2

Meta Chain Inv Rep1

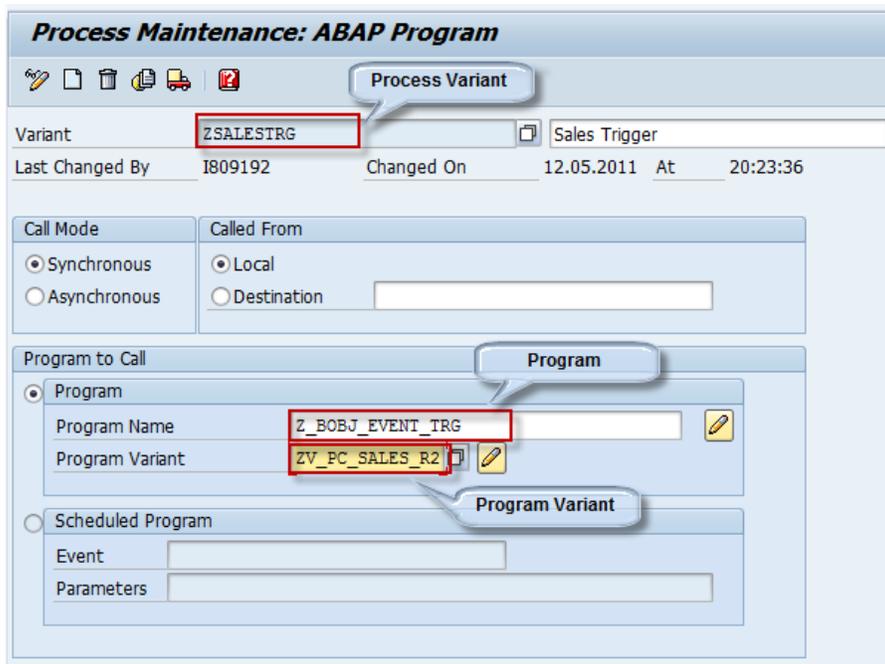


Here is a summary of the two variants created.



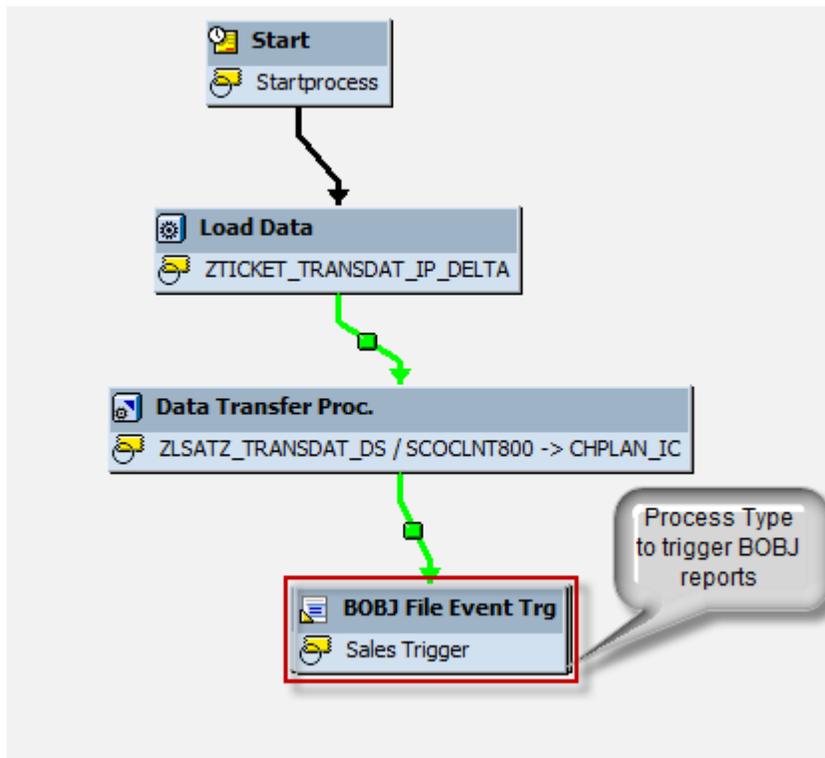
#### Attach the Custom Process Type to Process Chains/Meta Chain

1. Create the process variant **ZSALESTRG**. Attach the program **Z\_BOBJ\_EVENT\_TRG** created to trigger the event. Specify the variant created **ZV\_PC\_SALES\_R2** to trigger the file event /bobmon/pc\_sales\_rep2/pc\_sales\_rep2.txt.



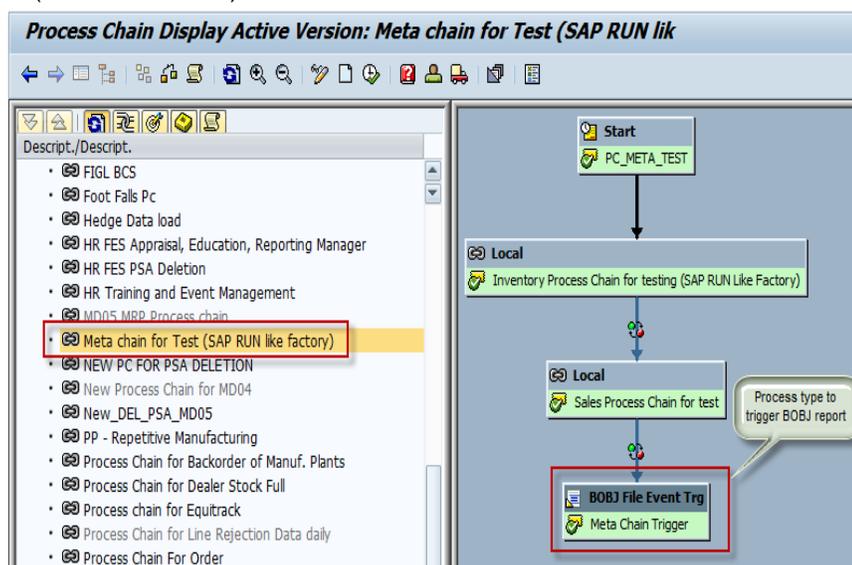
- Attach the created process variant to the Sales process chain1 that writes the file event on the share/mount to be used as file based event from BOBJ side.

Once the execution of this step is completed, the status of the process type can be seen in the RSPC (RED or GREEN).



- Another example of attaching the created custom process type to meta chain based on the second variant **ZV\_MC\_INV\_REP1**.

Once the execution of this step is completed, the status of the process type can be seen in the RSPC. (RED or GREEN).



## BusinessObjects Steps

From the BusinessObjects side, following issues need to be addressed:

- creation of the file-based event
- attaching the event to the report.

Refer to following Knowledge Base Articles for details on these settings and different settings needed for SP2 and SP3.

[1319644](#) - Event Poll Interval setting for Event Server in CMC says minutes but actually polls events in seconds in BusinessObjects Enterprise XI 3.1

[1272094](#) - Event Server Polling Interval and Clean Up Interval in BusinessObjects Enterprise XI 3.1.

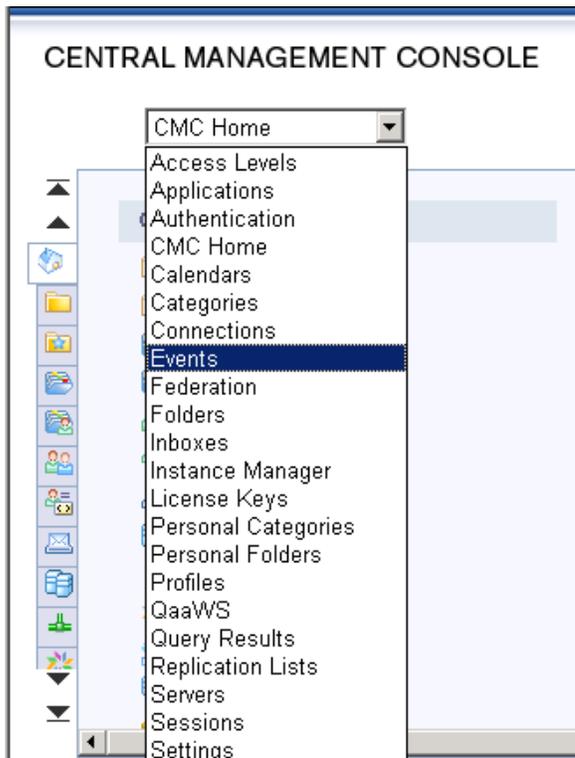
### Creation of File Based Event

1. Change the event poll settings (Event poll interval and Cleanup Interval) on the Event Server that will be used for scheduling the events.

*Event Service*

<input type="checkbox"/> Use Configuration Template	
Cleanup Interval (minutes):	5
Event Poll Interval (minutes):	1
<input type="checkbox"/> Restore System Defaults	
<input type="checkbox"/> Set Configuration Template	

2. From the Central Management Console (CMC), click **new event**. Enter the event name, specify the Event Server and file location / name as show below. Click **OK**.



3. Create a new Event based on **File**.

**New Event...**

Type: Custom

Event Name: Custom  
File  
Schedule

Description:

4. Specify the File Name which is created at the end of the successful Process Chain execution.

/bobmon/pc\_sales\_rep2/pc\_sales\_rep2.txt

**CENTRAL MANAGEMENT CONSOLE**

**New Event...**

Type: File

Event Name: Sales\_Report2\_Event

Description: Sales\_Report2\_Event

Server: BOBJPROD.EventServer

File Name: \\bobjprod\bobmon\pc\_sales\_rep2\pc\_sales\_rep2.txt

The file-based event is successfully created.

**Properties: Sales\_Report2\_Event**

▼ Properties

General Properties

Event Type

User Security

**Event Type**

Type: File

Server: BOBJPROD.EventServer

File Name: \\bobjprod\bobmon\pc\_sales\_rep2\pc\_sales\_rep2.txt

Created: Mar 16, 2011 10:53:42 AM

The following is a list of all the events created. The **Sales\_Report2\_Event** is the one created.

CENTRAL MANAGEMENT CONSOLE

Events Welcome: Administrator | Hk

Manage Actions Search title

Title	Type	Description
Inventory_Report1_Event	File	Inventory Report1 Event
Sales_Report2_Event	File	Sales_Report2_Event

### Attach the Event to the Report

1. From InfoView, right-click the **report to schedule**.

SAP BUSINESSOBJECTS INFOVIEW

Home | Document List | Open | Send To | Dashboards

New | Add | Organize | Actions

All

- My Favorites
- Inbox
- Public Folders
  - Administration Tools
  - Auditor
  - Feature Samples
  - Report Conversion To
  - Report Samples
    - Demonstration
    - Feature Samples
    - Financial
    - Report Data
    - Navigation Packag
  - Search Program
  - Voyager Demo

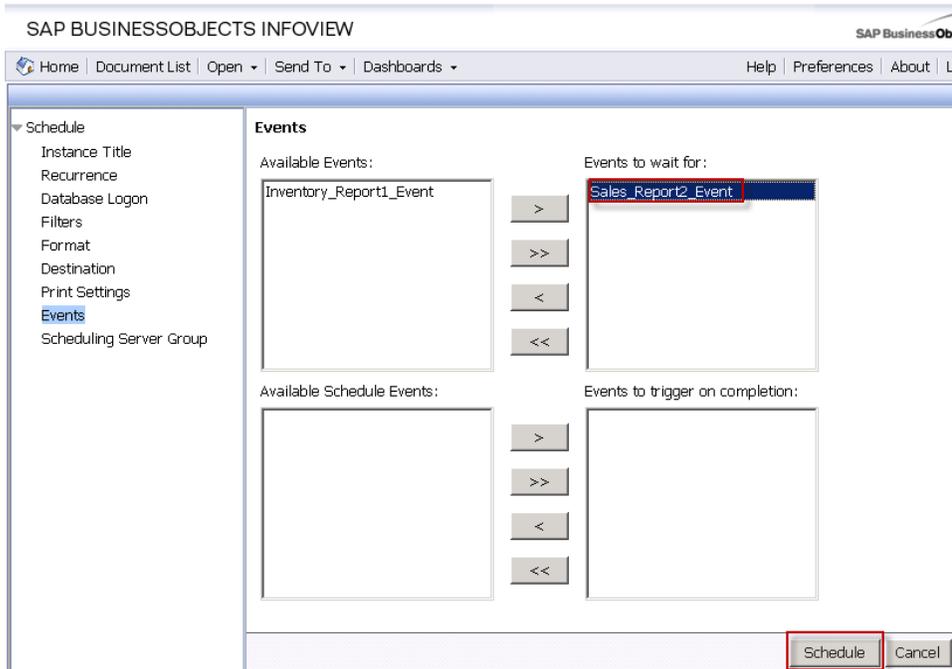
Title
Comparative Income Statement
Order Processing Efficiency Dashboard This report demonstrates the usage of Crystal's C
World Sales Report Top 5 Countries' Sales with

**View**

- Properties
- Categories
- Schedule
- History
- New ▶
- Add ▶
- Organize ▶

2. In the Schedule screen, click **Events**. Click the event previously created. Move it to the **Event to wait for** pane.

3. Enter the recurrence. Click **Schedule**. Once the file is created from BW at the location based on earlier steps), the report will show running status.



Below is the summary of the reports with Scheduled Events. These will be triggered automatically once the process chain on BW completes successfully.

Instance Time	Title	Run By	Parameters	Format	Status
May 12, 2011 1:33 PM	World Sales Report	Administrator	No Parameters	Crystal Reports	Pending

## Related Code

This section provides the coding for the implemented methods. Perform further changes as required.

### Process Type methods code

In the **public section**, enter the following constant information:

```
constants SUCCESS type RSPC_STATE value 'G'. "#EC NOTEXT
constants FAILED type RSPC_STATE value 'R'. "#EC NOTEXT
constants STATUS_IDX_ID type CHAR25 value 'PC_ABAP_STATUS'. "#EC NOTEXT
```

### IF\_RSPC\_EXECUTE~EXECUTE

```
METHOD if_rspc_execute~execute.

    cl_rspc_abap=>if_rspc_execute~execute(
        EXPORTING
            i_variant = i_variant
            i_event_start = i_event_start
            i_eventp_start = i_eventp_start
            i_t_processlist = i_t_processlist
            i_logid = i_logid
            i_t_variables = i_t_variables
            i_synchronous = i_synchronous
            i_simulate = i_simulate
            i_repair = i_repair
        IMPORTING
            e_instance = e_instance
            e_state = e_state
            e_eventno = e_eventno
            e_hold = e_hold
    ).

    IMPORT e_state TO e_state FROM DATABASE indx(bo) ID
    zcl_abap_bobj_event_trg=>status_idx_id.

ENDMETHOD.
```

### IF\_RSPC\_EXECUTE~GIVE\_CHAIN

```
METHOD if_rspc_execute~give_chain.

    return = cl_rspc_abap=>if_rspc_execute~give_chain( i_variant ).

ENDMETHOD.
```

### IF\_RSPC\_CALL\_MONITOR~CALL\_MONITOR

```
METHOD if_rspc_call_monitor~call_monitor.

    cl_rspc_abap=>if_rspc_call_monitor~call_monitor(
        i_variant = i_variant
        i_instance = i_instance
    ).

ENDMETHOD.
```

**IF\_RSPC\_CHECK~GIVE\_ALL**

```

METHOD if_rspc_check~give_all.

    return = cl_rspc_abap=>if_rspc_check~give_all( i_variant ).

ENDMETHOD.

```

**IF\_RSPC\_CHECK~CHECK**

```

METHOD if_rspc_check~check.

    cl_rspc_abap=>if_rspc_check~check(
        EXPORTING
            i_s_process = i_s_process
            i_t_chain = i_t_chain
            i_t_chains = i_t_chains
        IMPORTING
            e_t_conflicts = e_t_conflicts
    ).

ENDMETHOD.

```

**IF\_RSPC\_GET\_STATUS~GET\_STATUS**

```

METHOD if_rspc_get_status~get_status.

    cl_rspc_abap=>if_rspc_get_status~get_status(
        EXPORTING
            i_variant = i_variant
            i_instance = i_instance
            i_dont_update = i_dont_update
        IMPORTING
            e_status = e_status
    ).

ENDMETHOD.

```

**IF\_RSPC\_GET\_VARIANT~EXISTS**

```

METHOD if_rspc_get_variant~exists.

    r_exists = cl_rspc_abap=>if_rspc_get_variant~exists(
        i_variant = i_variant
        i_objvers = i_objvers
    ).

ENDMETHOD.

```

**IF\_RSPC\_GET\_VARIANT~WILDCARD\_ENABLED**

```

METHOD if_rspc_get_variant~wildcard_enabled.

    result = cl_rspc_abap=>if_rspc_get_variant~wildcard_enabled( ).

ENDMETHOD.

```

**IF\_RSPC\_GET\_VARIANT~GET\_VARIANT**

```

METHOD if_rspc_get_variant~get_variant.

cl_rspc_abap=>if_rspc_get_variant~get_variant(
EXPORTING
    i_variant = i_variant
    i_t_chain = i_t_chain
    i_t_select = i_t_select
    i_objvers = i_objvers
IMPORTING
    e_variant = e_variant
    e_variant_text = e_variant_text
EXCEPTIONS
    nothing_selected = 1
).

IF sy-subrc EQ 1.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
        WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4
        RAISING nothing_selected.
ENDIF.

ENDMETHOD.

```

**IF\_RSPC\_MAINTAIN~GET\_HEADER**

```

METHOD if_rspc_maintain~get_header.

    cl_rspc_abap=>if_rspc_maintain~get_header(
EXPORTING
    i_variant = i_variant
    i_objvers = i_objvers
IMPORTING
    e_variant_text = e_variant_text
    e_s_changed = e_s_changed
    e_control = e_control
    e_conttimestamp = e_conttimestamp
).

ENDMETHOD.

```

**IF\_RSPC\_MAINTAIN~MAINTAIN**

```

METHOD if_rspc_maintain~maintain.

    cl_rspc_abap=>if_rspc_maintain~maintain(
EXPORTING
    i_variant = i_variant
    i_t_chain = i_t_chain
IMPORTING
    e_variant = e_variant
    e_variant_text = e_variant_text
    ).

ENDMETHOD.

```

**IF\_RSPC\_TRANSPORT~GET\_TLOGO**

```

METHOD if_rspc_transport~get_tlogo.

    cl_rspc_abap=>if_rspc_transport~get_tlogo(
EXPORTING
    i_variant = i_variant
    i_objvers = i_objvers
IMPORTING
    e_tlogo = e_tlogo
    e_objnm = e_objnm
    ).

ENDMETHOD.

```

**IF\_RSPV\_TRANSPORT~GET\_ADDITIONAL\_OBJECTS**

```

METHOD if_rspv_transport~get_additional_objects.

    cl_rspc_abap=>if_rspv_transport~get_additional_objects(
EXPORTING
    i_variant = i_variant
    i_cto_mode = i_cto_mode
    i_is_content_system = i_is_content_system
IMPORTING
    e_t_cto_object = e_t_cto_object
    e_t_cto_key = e_t_cto_key
    ).

ENDMETHOD.

```

## Event Triggering Program

```
REPORT Z_BOBJ_EVENT_TRG.
```

```
DATA mesg(80) VALUE 'BW load complete. Ready for BOBJ Refresh. File Created on : '.
```

```
DATA fname(60).
```

```
DATA pc_flag TYPE rspc_state.
```

```
* Get the folder and file name from parameter
```

```
PARAMETERS p_fname TYPE string LOWER CASE.
```

```
AT SELECTION-SCREEN.
```

```
START-OF-SELECTION.
```

```
  fname = p_fname.
```

```
*Add the system date into the file.
```

```
  CONCATENATE mesg sy-datum INTO mesg.
```

```
  CALL FUNCTION 'AUTHORITY_CHECK_DATASET'
```

```
    EXPORTING
```

```
*     PROGRAM          =
       activity        = 'DELETE'
       filename        = fname
```

```
  EXCEPTIONS
```

```
    no_authority      = 1
    activity_unknown  = 2
    OTHERS             = 3
```

```
  IF sy-subrc = 0.
```

```
*Delete the file if it is already existing
```

```
  DELETE DATASET fname.
```

```
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
```

```
*     WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
```

```
  ELSE.
```

```
    MESSAGE i051(rsar) WITH 'No Authorizations to delete the file'.
```

```
  ENDIF.
```

```
  WAIT UP TO 65 SECONDS.
```

```
*Set the flag for the process chain to read if file deletion fails.
```

```
  IF sy-subrc <> 0." The ABAP program failed
```

```
    pc_flag = zcl_abap_bobj_event_trg=>failed.
```

```
  ELSE. " The ABAP program was successful
```

```
    pc_flag = zcl_abap_bobj_event_trg=>success.
```

```
  ENDIF.
```

```
CALL FUNCTION 'AUTHORITY_CHECK_DATASET'
```

```
  EXPORTING
```

```
    activity          = 'WRITE'
    filename          = fname
```

```
  EXCEPTIONS
```

```
    no_authority      = 1
    activity_unknown  = 2.
```

```
  IF sy-subrc <> 0.
```

```
    MESSAGE i051(rsar) WITH 'No Authorizations to Write to file'.
```

```
  ELSE.
```

*\*Create the file based on the input from variant including folder and file name.*

```
OPEN DATASET fname FOR OUTPUT IN TEXT MODE ENCODING DEFAULT.
```

*\*Set the flag for the process chain to read if file creation fails.*

```
IF sy-subrc <> 0.
```

```
  MESSAGE i051(rsar) WITH 'File open failed.'.
```

```
  pc_flag = zcl_abap_bobj_event_trg=>failed.
```

```
ELSE. " The ABAP program was successful
```

```
  TRANSFER mesg TO fname.
```

```
  pc_flag = zcl_abap_bobj_event_trg=>success.
```

```
ENDIF.
```

```
CLOSE DATASET fname.
```

```
ENDIF.
```

```
EXPORT e_state FROM pc_flag TO DATABASE indx(bo) ID
```

```
zcl_abap_bobj_event_trg=>status_idx_id.
```

## Related Content

[Implementing Your Own Process Type](#)

[Creating Process Chains](#)

[Creating an ABAP Process Type for Process Chains in BI](#)

Knowledge Base Article [1319644](#) -- Event Poll Interval setting for Event Server in CMC says minutes but actually polls events in seconds in BusinessObjects Enterprise XI 3.1

Knowledge Base Article [1272094](#) -- Event Server Polling Interval and Clean Up Interval in BusinessObjects Enterprise XI 3.1

[Business Intelligence homepage](#)

[Business Warehouse homepage](#)

For more information, visit the [Business Objects homepage](#).

## Copyright

© Copyright 2011 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, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

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 Oracle Corporation.

JavaScript is a registered trademark of Oracle Corporation, used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, 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 other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

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.