Object based Navigation based on Business Objects and POWL

Applies to:
SAP NetWeaver Portal 6.0 and 7.0 SP 16, SAP ERP 6.0 EhP3 and EhP4; ECC 7.1. For more information, visit the Business Objects homepage.

Summary
The objective of this document is to provide an understanding of:
- Business Objects
- POWL (Personal Object Work list) or POWER (Personal Object Work Entity Repository)
- Object Based Navigation based on Business Objects and its Operations
- Integration of BO with POWL and provide OBN to end user
The user will be able to create BO in Portal, integrate it with POWL and provide Object Based Navigation.

Author: Richa Shrivastava
Company: Accenture Services Pvt Ltd
Created on: 27 January 2011

Author Bio
Richa Shrivastava works as a Senior SAP NetWeaver Portal Consultant at Accenture Services Pvt Ltd and has extensive experience in implementation of Portal Projects specializing in Guided Procedures, SOA Webservices, Webdynpro, Business Objects, Object Based Navigation and Integrating Business Objects in Portal with POWL.
# Table of Contents

Document Objective ........................................................................................................... 3

Document Details .................................................................................................................. 3

  What are Business Objects? ............................................................................................... 3
    Definition ......................................................................................................................... 3
    Use .................................................................................................................................. 3
  Object-Based Navigation .................................................................................................. 3
    Purpose ............................................................................................................................ 3

Business Object Operations ................................................................................................ 4

Implementation by iView ...................................................................................................... 4

Example ................................................................................................................................ 4

Important Features of Object Based Navigation in EP 6.0 .................................................. 4

Benefits of Object Based Navigation .................................................................................. 5

Unification Architecture ....................................................................................................... 6

What is POWL? ...................................................................................................................... 6

  Definition ........................................................................................................................... 6
  Use ..................................................................................................................................... 6
  Features ............................................................................................................................. 7
  Example ............................................................................................................................... 7
  Concept of POWL .............................................................................................................. 7

BO and POWL Integration ................................................................................................... 8

Prerequisites ......................................................................................................................... 8

  System Administration ..................................................................................................... 8
  Content Administration ..................................................................................................... 8

Integrating Business Object, Business Operation and POWL ............................................. 17

  Block Diagram ................................................................................................................ 17

How to Create a Business Object in Portal, Add Business Operation to it, and finally integrate it with a POWL Application? .............................................................................................................................. 18

Add the operations to BO .................................................................................................... 20

Add iView / Page to BO_Operation ...................................................................................... 21

Integrating Business Object with the POWL Application .................................................... 22

  POWL screen with Action Buttons .................................................................................. 22

Following are the steps for Integrating BO_BO_Operation with the feeder Class of POWL ...................................................................................................................... 22

Benefits to Business on Implementing Business Objects ...................................................... 30

Monitory Savings to Business when Business Objects are implemented .......................... 31

SAP Extends the Power of SAP® BusinessObjects™ Explorer to Tap into All Data .......... 31

Customer and Partner Ecosystem Thrives With SAP® BusinessObjects™ Governance, Risk, and Compliance Solutions .................................................................................... 31

  Mission of SAP and Business Objects is: ........................................................................ 31

Related Content .................................................................................................................. 32

Disclaimer and Liability Notice ........................................................................................... 33
Document Objective

The objective of this document is to provide an understanding of
- Business Objects
- POWL
- Object Based Navigation based on Business Objects (and its operations)
- Integration of OBN with POWL (Personal Object Work list) or POWER (Personal Object Work Entity Repository)

The application is used to provide a centralized task list to the user and hence, reduce complexity and enhance the productivity of the end-users.
The user will be able to create Business Objects in Portal, create and add Business Operations to it, integrate it with POWL, and provide Object based Navigation to the end user.

Prerequisites
- Understanding of POWL concepts and Portal knowledge is Mandatory.
- SAP Enhancement Package 2 or above is needed as POWL is supported there-after only.

Must Read
"concept of powls.pdf" attached in this document under heading Concept of POWL.

Document Details

What are Business Objects?

Definition
Business Object is representation of a business object of the real world in the SAP System. Business objects represent concrete or abstract objects as well as activities or processes. Examples include purchase orders, contracts, customers, risk or telephone calls.

Use
Business objects are used to model and further process objects occurring in a company in the SAP System. Certain processing options, for example, display, change or send, are available for each business object.

Object-Based Navigation

Purpose
Object-based navigation (OBN) offers portal users an additional method of navigation based on business objects.

Business objects can be imported from systems in the Portal Content Directory (PCD), or imported to the Business Objects folder in the Portal Content Catalog, using the Business Object Importer. In the Business Objects folder, they can be arranged logically into sub-folders as desired.

The primary capability offered by OBN is that the data returned to the user during navigation is role-based and accessed dynamically during runtime.

While navigating in the portal, users receive different kinds of data from iViews based on business objects, according to the needs and requirements of their role in the organization.

In other words, two different users may perform the same navigation operation, and the data returned to each of them will be different because it is role dependent. This is the significance of business object operations.
Business Object Operations

The business object operation is the crux of OBN administration. It is the operation that serves as the connection between the business object and the iView that implements it during runtime.

Navigation is defined by means of operations attached to the business objects.

Since multiple operations may be assigned to a single business object, numerical priorities are assigned to each operation. The priority specifies the default operation to be implemented for the specific business object.

Implementation by iView

Just as a business object may have multiple operations attached to it, the single business object operation may have multiple implementing iViews associated with it. The implementing iView displays business information relevant to the users for whom it is authorized. In the event of multiple associated operations, the iView implemented during runtime depends on two considerations:

- The role of the user
  iViews are role-based.

- The operation priority
  For a user having more than one role, runtime may be presented with a choice of more than one iView to implement the same operation.

Note: Portal pages, as well as iViews, can be attached to business object operations.

Example

A sales representative and a sales manager both choose to display an iView based on the same business object, representing a specific account. The default operation for this business object has two role-based iViews attached to it, one showing Account Details, the other showing List of Opportunities.

The user with Role sales representative sees Account Details which was assigned the higher priority.

The user with Role sales manager sees List of Opportunities which has the higher priority.

Important Features of Object Based Navigation in EP 6.0

- The operation is directly attached to the object:
  - Optically (Visible for the User)
  - Logically (in the Portal Catalog)

- The operation is role-sensitive:
  - The Navigation options offered depend on the user’s role

User option is ergonomical

- Object-based navigation follows the style users are familiar with from www and from other software
• User can choose an operation by clicking
  - SAP delivers navigation scenarios out-of-the-box
• Customers can easily modify or create navigation scenarios
  - based navigation means: Make Business object entries navigationable
  - OBN only works for new content (not for existing content)
    - Either delivered by SAP
    - Developed by the customer

Benefits of Object Based Navigation
Same operation can render different results depending on the Role assigned to the user.

Administrator

Demon User

The two users get different results based on their two Roles: Roles Administrator and User.
What is POWL?

Definition

The Personal Object Worklist is a framework which shows list of business objects and also allows specific activities based on the objects made. Specific Activities can be performed for each business object via buttons shown in the objects and thus handling the action of buttons.

Personnel Object Worklist Framework is based on WebDynpro ABAP

Use

The Personal Object Worklist (POWL) provides us, as an Enterprise Portal user, with a general overview of our work environment and all related business objects that we would be interested to work on. It enables us to manage our work efficiently and aids in decision-making. POWLs are most suited to assignments that are not a part of a workflow.

Using POWL, we can define and store worklists that are defined as queries. We can activate these worklists so that their results are visible when we navigate to a portal page or workset containing the POWL iView. POWL is a generic tool and worklists can be designed by the user.

The following functions are provided by POWL:

- Gives us a unified and centralized way to access our work and the relevant information
- Aggregates non-workflow task items from multiple and different systems in one universal list
  - Displays additional information as required from document and object repositories, including attachments and other details such as worklist type dependent actions
  - Provides easy entry into business object handling

**Features**

Content of worklists is structured thematically and created depending on a business object. In the backend system of an Enterprise Portal the worklist type maps the information which business object can process using fixed methods.

In the Enterprise Portal the personal object worklists are designed as queries.

Portal users can create new worklists or modify existing worklists by creating and editing queries in the Enterprise Portal.

**Example**

Consider a production order-related worklist; based on a predefined selection variant the Production Supervisor (portal role) gets a worklist with planned orders for his or her section of a plant, including a workshop, which is represented by all materials belonging to the same material class.

The Production Supervisor can now select one or more planned orders, convert them to production orders, release the orders, print the work instructions, and distribute the worklists to the responsible operator groups.

**Concept of POWL**

Please refer to the following link for the additional technical information on concepts on POWL

BO and POWL Integration

Prerequisites

- Portal Knowledge
- Content Administration, System Administration and User Administration roles

Steps

These steps need to be performed by each of the super admin roles.

Please follow the following steps to access the powl corresponding to applid TSW_FSR from the portal.

System Administration

1. Configure the system from the powl application needs to be called. While setting up this system, create a s/m alias e.g. ‘SAP_SYS_ALIAS’. This shall be later used while creating the iViews.

Content Administration

2. Create a folder e.g. ‘Powl Test’ with the following sub-folders in a desired location under ‘Portal Content’.
   a. iViews
   b. Pages
   c. Worksets
   d. Roles

3. Create an iView of type ‘SAP Web Dynpro iView’ of type ABAP to render the POWL webdynpro application in the folder ‘iViews’.

Set the following properties of this iView

- Application name : Z_TSW_POWL_FSR
- Application Parameters :
- Namespace : sap
- System Alias : SAP_SYS_ALIAS
- Height Type : Full Page
4. Create an iView of type ‘SAP Transaction iView’ e.g. ‘Display Material’ to point to the transaction ‘MM03’.

Set the following properties of this iView

- Transaction: MM03
- System Alias: SAP_SYS_ALIAS
- SAP GUI Type: SAP GUI for HTML

5. Create an iView of type ‘SAP Transaction iView’ e.g. ‘Display Stocks/Requirements List’ to point to the transaction ‘MD04’.

Set the following properties of this iView

- Transaction: MD04
- System Alias: SAP_SYS_ALIAS
- SAP GUI Type: SAP GUI for HTML

6. Create an iView of type ‘SAP Transaction iView’ e.g. ‘Display Nomination’ to point to the transaction ‘O4NSN’.

Set the following properties of this iView

- Transaction: O4NSN
7. In the ‘Pages’ folder create a new page e.g. ‘POWL Page’ of type ‘Default Page Template’. While creating the page, choose the layout ‘1 Column (Full Width)’ and set this as the default Layout.

Set the following properties for the page:
- ‘Invisible in navigation areas’ : Yes
- ‘Default Entry for Folder’ : Yes

8. Edit the page created in the above step to attach all the 4 iViews created in steps 3, 4, 5 and 6 as delta links.
9. Set the visible property of only the 'powl' iView as visible. The visible property of all other iViews in this page should be unchecked.

Also set the following properties of this page:

- Default Entry for Folder: Yes
- Height Type: Full Page

10. In the 'Worksets' folder create a workset e.g. 'POWL Entry'.

11. In the 'Worksets' folder create a workset e.g. 'POWL'.

Set the following property of the workset:

- Leaf Folder: Yes
12. Open the workset created in the above step and attach the page ‘POWL Page’ as a delta link.

![Diagram](image1.png)

13. Open the workset ‘POWL Entry’. Create a folder called ‘Overview’ under this workset. Now add the workset POWL to this newly created ‘overview’ folder as a delta link as shown below.

![Diagram](image2.png)

14. In the ‘Roles’ folder, create a role e.g. ‘POWL Role’.

![Diagram](image3.png)

15. Open this role for editing and attach the workset ‘POWL Entry’

![Diagram](image4.png)
While editing the role itself, double click on workset ‘POWL Entry’ and set the property ‘Entry Point’ to true.

Next double click on the workset ‘POWL’.
Set the following properties of this workset
Leaf Folder : Yes

16. To create a new Business Object and an operation in the portal, follow the below mentioned steps. (Given below is an example)
   a. Create a folder called ‘XXXX’ and within this another folder called ‘Traders & Scheduler Workbench’ under the Business Objects folder
   b. Right click on this ‘Traders & Scheduler Workbench’ folder and click on ‘Import Business Objects’.
c. In the ‘Add Manually’ section specify the following details as shown in the fig below.

![Object based Navigation](image)

Business Object Name : Nominations
System Alias (new) : SAP_OIL_System
Business Object ID : nomination
Click on ‘Add’ and then on ‘Save’. A new business object is now created.

d. Now open Business object ‘Nominations’

Click on the ‘Add’ button under the operations table and add a new operation called ‘Display Nomination’ with id ‘display_nomination_transaction’. Now save the changes.

![Business Object Editor](image)

Open the iView ‘Display Material’ from the role itself as shown below and navigate to Display ‘Object Based Navigation’ section
1. Click on the ‘Business Objects’ folder on the left hand navigation pane. Navigate to the operation Business Objects->ERP Commons Parts->Manufacturing->Material Master->Display. Right click on this operation and click on ‘Add Operation to iView’.

The operation gets added to the iView.

Save the changes.
g. Choose the Display Option ‘Object’ next and set the following properties
   ‘Launch in New Window’ : ‘Display in Separate Headerless Portal Window.’
   ‘Process First Screen’ : Yes
   Window Features : toolbar=no, resizable=yes
   Width of External Window (Pixels) : 1000

17. Repeat steps 17a, 17b and 17c for the iViews 'Display Stocks/Requirements List' and 'Display Nominations' with the operations mentioned in the below table

<table>
<thead>
<tr>
<th>iView</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Stocks/Requirements List</td>
<td>Business Objects-&gt;ERP Common Parts-&gt;Manufacturing-&gt;Display Stocks/Requirements List List-&gt;Display</td>
</tr>
<tr>
<td>Display Nomination</td>
<td>Business Objects-&gt;XXXX-&gt;Traders &amp; Scheduler Workbench-&gt;Nomination-&gt;Display Nomination</td>
</tr>
</tbody>
</table>

Note: Business Objects and operations have been reused in this example. It is also possible to create new business objects or operations and these can be used instead.

User Administration
Assign the role ‘POWL Role’ to the required users.
Integrating Business Object, Business Operation and POWL

Block Diagram
This Block Diagram gives a clear understanding of how a Business Object, BO Operations (Portal) and POWL Application (Backend/ECC) are linked and integrated together.

Note:
Boxes / Text in Blue represent development in Portal
Boxes / Text in Black represent development in ECC

1 ➔ User Logins in to Portal with User ID and access the assigned Role
2 ➔ Access the assigned POWL Application. POWL iView ➔ Page ➔ Workset ➔ Role
3 ➔ Click on the action Button on POWL
4 ➔ iView/ Page linked to BO.BO_Operation is triggered
5 ➔ ID of the BO and BO_Operation developed in Portal is provided as reference in the Feeder class of POWL Application. This provides the Object Based Navigation

If any of the parameters from the above 5 links is missed, Runtime exception will be thrown.
How to Create a Business Object in Portal, Add Business Operation to it, and finally integrate it with a POWL Application?

Steps
Step 1 – Create a folder (ex- “Contract Request”) for BO in PCD under Business Objects
Folder with the name “Contract Request” is created.

Step 2 – Right Click on the Folder > select “Import Business Objects”

Import Business Objects screen opens in the right hand side.
Step 3 –
Give the name for BO – Contract Request
Select System Alias – SAP_ECC_Manufacturing
BO ID – Contract_Request

Step 4 - Click on Add

Step 5 - Click on Save

The BO is created.
Add the operations to BO

Step 6 - Open the BO

BO Editor is opened.

Step 7 - Add the operations to BO by clicking on Add Button. Provide Operation Name and Operation ID.
Add iView / Page to BO_Operation

Step 8 - Select the Operation > Navigate to the iView created in PCD > Add iView to the Operation Create.

Step 9 - Click on Display iViews/Pages button

Step 10 – Note: This iView should be linked to the Role in Portal.
This will make the property ‘In Navigation’ = ‘Yes’
Integrating Business Object with the POWL Application

POWL screen with Action Buttons

Following is the screen shot of a POWL application Contract Request that the user would see when logged in to Portal.

Following are the steps for Integrating BO.BO_Operation with the feeder Class of POWL

Step 1 – Identify POWL Application Name

In PCD Open the Role (Contracts) and Navigate to the Page within the Workset to which the POWL Application is assigned as iView
Locate the Page in PCD

Right Click and select Edit on the page

Select the POWL iView assigned to the page and Click Open

From the Portal Catalog, select the content you want to add to the page. In the Property Editor, edit the property values of the page and those of its iViews and pages.
Get the name of the POWL Application (Application ID) from the iView property 'Application Parameters'.

Step 2 ➔ Login into ECC.
Enter Transaction POWL_TYPER
Click Position

Provide the POWL Application ID noted in the previous step.

Note the POWL Type → Z_SUPPLY_RPC_TY

Change View "View: Type - Role assignment": Overview
Step 3 → Go to Transaction POWL_TYPE

Change View "View: Type definition": Overview

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/SREP/CO_0M_CA_20_01</td>
<td>Cost Centers: Actual/Plan/Variance (Cost Elements)</td>
</tr>
<tr>
<td>/SREP/CO_0M_CA_40_01</td>
<td>Cost Centers: Actual Line Items</td>
</tr>
<tr>
<td>/SREP/CO_0M_OP_20_01</td>
<td>Orders: Actual/Plan/Variance (Cost Elements)</td>
</tr>
<tr>
<td>/SREP/CO_0M_OP_40_01</td>
<td>Orders: Actual Line Items</td>
</tr>
<tr>
<td>/SREP/CO_0M_OP_50_01</td>
<td>Orderlist: Budget/Actual/Commitments</td>
</tr>
<tr>
<td>/SREP/CO_FC_CF_10_01</td>
<td>Product Costing Analysis: Costing List</td>
</tr>
<tr>
<td>/SREP/CO_FC_CF_20_01</td>
<td>Product Costing Analysis: Cost Components</td>
</tr>
<tr>
<td>/SREP/CO_FC_CF_30_01</td>
<td>Product Costing Analysis: Multi-Level Hirarchization</td>
</tr>
<tr>
<td>/SREP/CO_FC_CI_10_01</td>
<td>Production: Order Cost Detailed Analysis</td>
</tr>
<tr>
<td>/SREP/CO_FC_CI_11_01</td>
<td>Product Drill Down</td>
</tr>
<tr>
<td>/SREP/CO_FC_CI_20_01</td>
<td>Production: Cost Analysis: Order List</td>
</tr>
<tr>
<td>/SREP/FI_AP_20_1_01</td>
<td>Vendor Balances: Period Drill Down</td>
</tr>
<tr>
<td>/SREP/FI_AP_28_2_01</td>
<td>Vendor Balances: Totals</td>
</tr>
<tr>
<td>/SREP/FI_AP_36_01</td>
<td>Vendor Line Items</td>
</tr>
<tr>
<td>/SREP/FI_AP_31_1_01</td>
<td>Vendor Due Date Analysis: Document Drill Down</td>
</tr>
<tr>
<td>/SREP/FI_AP_31_2_01</td>
<td>Vendor Due Date Analysis: Totals</td>
</tr>
<tr>
<td>/SREP/FI_AP_31_2_02</td>
<td>Vendor Due Date Analysis: Overdue Totals</td>
</tr>
<tr>
<td>/SREP/FI_AP_31_2_03</td>
<td>Vendor Due Date Analysis: Forecast Totals</td>
</tr>
<tr>
<td>/SREP/FI_AP_56_01</td>
<td>Check Register</td>
</tr>
</tbody>
</table>
Click Position ➔ Enter POWL Type

Step 4 ➔ Double Click on the required POWL TYPE
Note the Feeder Class Name ➔ ZCL_RFC_POWL

Change View "View: Type definition": Details
Step 5 → Go to Class Builder (Transaction SE24)

![Image of Class Builder: Initial Screen](image)

Give Object Type as Feeder Class Name and Click Display

Displays all the Methods of the Feeder Class

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
<th>Exceptions</th>
<th>Level</th>
<th>View</th>
<th>Mod.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF_POWL_FEEDER~GET_ACTIONS</td>
<td>Instance</td>
<td>Method</td>
<td>Public</td>
<td></td>
<td></td>
<td>define action meta data for represented object type</td>
</tr>
<tr>
<td>IF_POWL_FEEDER~GET_ACTION_CONF</td>
<td>Instance</td>
<td>Method</td>
<td>Public</td>
<td></td>
<td></td>
<td>define action confirmation message</td>
</tr>
<tr>
<td>IF_POWL_FEEDER~GET_SEL_CRITERIA</td>
<td>Instance</td>
<td>Method</td>
<td>Public</td>
<td></td>
<td></td>
<td>define selection criteria meta data</td>
</tr>
<tr>
<td>IF_POWL_FEEDER~GET_FIELD_CATALOG</td>
<td>Instance</td>
<td>Method</td>
<td>Public</td>
<td></td>
<td></td>
<td>define field catalog meta data</td>
</tr>
<tr>
<td>IF_POWL_FEEDER~GET_OBJECT_DEFINITION</td>
<td>Instance</td>
<td>Method</td>
<td>Public</td>
<td></td>
<td></td>
<td>define data structure for represented object type</td>
</tr>
<tr>
<td>IF_POWL_FEEDER~GET_OBJECTS</td>
<td>Instance</td>
<td>Method</td>
<td>Public</td>
<td></td>
<td></td>
<td>data retrieval for represented object type</td>
</tr>
<tr>
<td>IF_POWL_FEEDER~GET_DETAIL_CMP</td>
<td>Instance</td>
<td>Method</td>
<td>Public</td>
<td></td>
<td></td>
<td>object detail/OD comp., implementing IF_POWL_DETAIL</td>
</tr>
<tr>
<td>IF_POWL_FEEDER~HANDLE_ACTION</td>
<td>Instance</td>
<td>Method</td>
<td>Public</td>
<td></td>
<td></td>
<td>handle actions for object type (cf. GET_ACTIONS)</td>
</tr>
</tbody>
</table>

IF_POWL_FEEDER~HANDLE_ACTION is the Method where Action Handlers for the POWL Action Buttons are implemented.

IF_POWL_FEEDER~GET_ACTION is the Method where the Action IDs for these POWL Action Buttons are defined.
Following are the Attributes referred in the Feeder Class of POWL that refers to BO System, BO Name and BO Operation which are developed in Portal

Following is the Implementation code for BO System, BO Name and BO Operation name provided as reference in the Feeder class of POWL Application. Each BO Operation corresponds to the respective Action Button in POWL.

When the button is clicked, the iView linked to the BO Operation is triggered. This provides the Object Based Navigation.

*fill e portal actions to navigate
   is_portal_actions-bo_system = zcl_rfc_powl->mc_sap_ecc_manufacturing,
   is_portal_actions-bo_name = zcl_rfc_powl->mc_contract_request,
   is_portal_actions-bo_op_name = zcl_rfc_powl->mc_display.

CASE 1_actionid.
* Create Action
   WHEN zcl_rfc_powl->mc_actionid_create.

   is_portal_actions-bo_system = zcl_rfc_powl->mc_sap_ecc_manufacturing,
   is_portal_actions-bo_name = zcl_rfc_powl->mc_contract_request,
   is_portal_actions-bo_op_name = zcl_rfc_powl->mc_create.

*fill e portal actions to navigate
is_portal_actions-bo_system = zcl_rfc_powl->mc_sap_ecc_manufacturing,
is_portal_actions-bo_name = zcl_rfc_powl->mc_contract_request,
is_portal_actions-bo_op_name = zcl_rfc_powl->mc_edit.

*fill e portal actions to navigate
is_portal_actions-bo_system = zcl_rfc_powl->mc_sap_ecc_manufacturing,
is_portal_actions-bo_name = zcl_rfc_powl->mc_contract_request,
is_portal_actions-bo_op_name = zcl_rfc_powl->mc_printpreview.
Following is the screen shot of the POWL with the above mentioned action buttons.

Benefits to Business on Implementing Business Objects

The Integration of Business Objects into SAP provides great opportunities to serve business better.

- Business Objects are fully integrated into SAP’s proven Support Infrastructure
- Customers and Partners of Business Objects get benefits
  - from the utilization of a common support infrastructure
  - Long-term proven support applications adapted to customers’ needs
  - Intuitive and easy-to-use interfaces with guided navigation
  - Stable and well performing applications – high availability
- Business Objects Support is integrated into SAP’s Global Support Backbone as well!
- Business Objects Enterprise customers and partners use the SAP Support Portal to receive Business Objects enterprise support
- Business Objects Small and Medium Business customers use the Crystal Reports and Xcelsius Support portal to receive technical support
- All rollout information to Business Objects is directly accessible via quick link https://websmp201.sap-ag.de/bosap-support
Monitory Savings to Business when Business Objects are implemented

By partnering with SAP BusinessObjects, the business intelligence (BI) market leader, a business will gain access to an established brand and award-winning products.

The SAP’s partner program offers numerous benefits that will help a company:

- Increase revenue
  - Provide opportunity to develop market-leading and standards-based solutions
  - Attain flexible licensing options for buy versus build
  - Increase average deal size
  - Leverage the comprehensive market presence of SAP BusinessObjects
- Focus efforts on core competencies of a company
  - Accelerate time to market
  - Reduce development costs
- Provide incremental value to the customers
- Gain faster ROI by offering customers a complete solution

Following are some of the Business news headlines published in SAP Global Portal key contribution and revenue generation in the area of Business Objects

SAP Extends the Power of SAP® BusinessObjects™ Explorer to Tap into All Data

SAP® BusinessObjects™ Explorer Software Helps Companies Obtain Deeper Insight and Clarity into Their Business Challenges and Opportunities

VIENNA, Austria - October 27, 2009 - In May 2009, SAP AG (NYSE: SAP) unveiled SAP® BusinessObjects™ Explorer software, a breakthrough solution that enables business users to explore vast amounts of information and find relevant answers in seconds. In November, SAP will usher in the next wave of SAP BusinessObjects Explorer, accelerated version, which delivers business insight on any data set through an intuitive Web-based interface anyone can use. With instant visibility into all the information needed to do their jobs, users can easily tackle tough questions and identify the root causes of business issues while navigating their business data at the speed of thought. Today, companies like Consol, Kraft Foods and OraSure Technologies are looking to SAP BusinessObjects Explorer to solve key questions using a new approach. The announcement was made at SAP TechEd 2009, being held in Vienna October 27-29.

Customer and Partner Ecosystem Thrives With SAP® BusinessObjects™ Governance, Risk, and Compliance Solutions

BearingPoint, MTU Detroit Diesel, Sharp and Valero Rely on SAP® BusinessObjects™ GRC Solutions to Automate, Improve and Reduce the Costs of GRC Processes;

SAP Collaborates with Key Partners to Bring Added Value to GRC Customers

PRAGUE, Czech Republic - November 18, 2009 - SAP AG (NYSE: SAP) today announced that a growing number of companies are using SAP® BusinessObjects™ governance, risk, and compliance (GRC) solutions to improve and automate their compliance and risk management processes. Worldwide, customers like BearingPoint, MTU Detroit Diesel, Sharp and Valero are using SAP solutions to help achieve higher compliance ratings, promote substantial cost savings and help increase transparency across different lines of business to operate more efficiently in regulatory-mandated industries. Additionally, the global customer network of SAP BusinessObjects GRC solutions has been able to benefit from new services from the SAP partner community. The announcement was made at GRC 2009, being held Nov. 18-20 in Prague.

Mission of SAP and Business Objects is:

Transform the Way the World Works by Connecting People, Information, and Business
Related Content

- Business Client and OBN
- Object-Based Navigation
- Integrating POWL with WebDynpro ABAP
- POWL – Technical Information

For more information, visit the [Business Objects homepage](http://www.sap.com/).
Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.