

# Demo for Mass Approval of Tasks



## Applies to:

SAP NetWeaver Composition Environment. For more information, visit the [Business Process Modeling homepage](#).

## Summary

This article provides guidelines on how to utilize the API provided for approving more than one task at a time using help of a demo.

**Author:** Santosh Giri Govind Marthi

**Company:** SAP LABS India Pvt. Ltd.

**Created on:** 29 Oct 2010

## Author Bio



Santosh has been part of the NW BRM team for about 4 ½ years and has been involved in topics relating to NW BRM component like creating of customer POCs, Development of Rules Manager (Web based business user tool) and Testing

## Table of Contents

1. Introduction .....	3
Business Requirement .....	3
Technical Solution .....	3
Value Proposition .....	3
2. Process Model .....	3
3. Technical Implementation.....	4
Overview .....	4
Approving multiple tasks at a time .....	5
4. Configuring and Running Demo. ....	7
Related Content.....	11
Copyright.....	12

## 1. Introduction

Approval or rejection of a task is to be done one by one and will become a problem if the user has many tasks in his work list. And with the existing infrastructure, one can approve/reject only one task at a time. So this has prompted for a solution using which a responsible can approve/reject many tasks at once (Mass approval/rejection of tasks).

### Business Requirement

Let us assume that there are many tasks (say 10 tasks) in the worklist of a responsible and he has to approve/reject them. With the existing approach, he/she has to go to each task, open it and then approve/reject accordingly. Doing this for all the 10 tasks will be a time consuming and tedious process. So if there is a way for the responsible to select more than one task and approve/reject collectively then it will be a huge value add to the person and in-turn a value add to the product.

### Technical Solution

Technical implementation for this solution is as follows:

- a. Use the public API for task to approve/reject more than one task at a time.
- b. The current integration of SAP NetWeaver BPM with UWL does not provide multi-selection of tasks. So we have to create separate UI for UWL which will show all the tasks that are assigned to a particular user and will allow multi selection and Accept/Reject.
- c. To demonstrate this we have used the IAP scenario, where the first task is of approve or reject. In this scenario we will use the new UI for UWL which will show all the tasks that are assigned to users with group "Approvers".

In the new UI for UWL, we provide Approve and Reject button along with the "Approvers Note" field where the approver can write his comments.

### Value Proposition

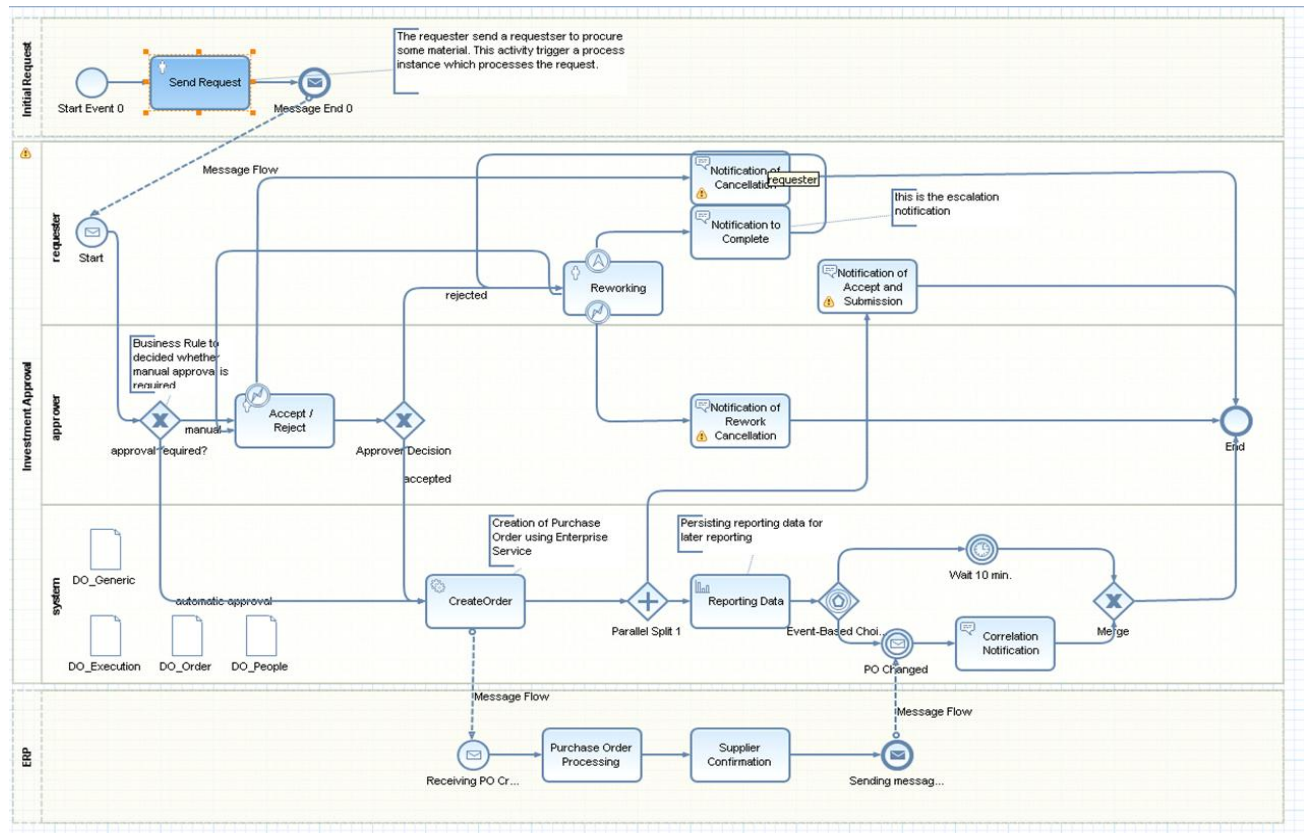
Customers can now use this API to approve/reject more than one task at a time, instead of doing it individually for all the tasks. This significantly reduces the time taken for approving or rejecting tasks, in-turn increasing the productivity of the users.

## 2. Process Model

Model of the BPM process that has been used for this demo (Investment Approval Process (IAP) scenario) is shown below. In this scenario, there will be initial request for a material from the "Requesters". And if the total amount for the requested material exceeds particular limit for certain regions, then there should be an approval from the "Approvers". Once a task is "Approved" there will be a notification mail sent to the corresponding person whose details are provided in the input to the process. And if the task is "Rejected" then the process goes to the "Reworking" step and continues accordingly.

The "Approve/Reject" step in the process in the "approver" lane of the "Investment Approval" pool is the human activity which provides the UI for Approve/Reject of the task. And this demo showcases the use of the newly created API for mass approval of tasks at this step. In this demo we use a separate UI for UWL instead of the UWL UI available with the CE server.

The new UI contains a list of tasks that are of "Approve/Reject" type and a field for the approver to provide approval/rejection comment and "Approve" and "Reject" buttons for respective actions. In this new UI, the user can select more than one task from the list and "Approve" or "Reject" at a time instead of opening each and every task individually for "Approve" or "Reject". It also provides the link to open up the UI assigned for individual task and use that UI for "Approve" or "Reject".



### 3. Technical Implementation

#### Overview

The tasks filtering, UWL UI and Approve/Reject of tasks were implemented using the new task API. The display of the "Approve/Reject" task is done as a list in a table along with the link to open the task UI separately. Multi selection of the tasks is enabled in the new UWL UI that this demo uses and there is a provision for "Approve"-ing or "Reject"-ing the selected tasks simultaneously along with giving the reason for the action taken.

You can find the details of how to use the API for 'Building a Task Worklist using BPM API' here:

[http://help.sap.com/saphelp\\_nw73/helpdata/en/79/d505324ef8429489979e43b76b56a9/frameset.htm](http://help.sap.com/saphelp_nw73/helpdata/en/79/d505324ef8429489979e43b76b56a9/frameset.htm)

## Approving multiple tasks at a time

1. Filling the task list in the UI with the tasks is achieved with the below code:

```

public void fillTaskList() {
    //@@begin fillTaskList()
    wdContext.nodeTask().invalidate();

    try {
        TaskInstanceManager taskInstanceManager = BPMFactory.getTaskInstanceManager();

        HashSet<Status> statusSet = new HashSet<Status>();
        statusSet.add(Status.READY);
        statusSet.add(Status.RESERVED);
        statusSet.add(Status.IN_PROGRESS);
        statusSet.add(Status.CREATED);

        //get user's tasks
        Set<TaskAbstract> myTasks = taskInstanceManager.getMyTaskAbstracts(statusSet);
        TaskAbstract myTask;
        IPublicTaskList.ITaskElement taskElem;

        Iterator<TaskAbstract> taskAbstIt = myTasks.iterator();

        while (taskAbstIt.hasNext()){
            myTask = taskAbstIt.next();
            if(taskList.contains(myTask.getName())){
                taskElem = wdContext.createAndAddTaskElement();
                java.util.Date actDate = myTask.getActivationTime();
                if(actDate!=null){
                    java.sql.Timestamp sqlActDate = new java.sql.Timestamp(actDate.getTime());
                    taskElem.setActivationTime(sqlActDate);
                }
                java.util.Date createDate = myTask.getCreatedTime();
                if(createDate!=null){
                    java.sql.Timestamp sqlCreateDate = new java.sql.Timestamp(createDate.getTime());
                    taskElem.setCreationTime(sqlCreateDate);
                }
                String displayName=myTask.getPresentationName();
                TaskDetail taskDetail=taskInstanceManager.getTaskDetail(myTask.getId());
                if(taskDetail!=null) {
                    DataObject do_Start = taskDetail.getInputDataObject().getDataObject("start");
                    if(do_Start != null){
                        DataObject dO_Order=dO_Start.getDataObject("DO_Order");
                        taskElem.setMaterialId(dO_Order.getString("MaterialNumber"));
                        taskElem.setQuantity(dO_Order.getString("Quantity"));
                        taskElem.setRegion(dO_Order.getString("Region"));
                        taskElem.setPrice(dO_Order.getString("Price"));
                    }
                }
                taskElem.setDisplayName(/*myTask.getPresentationName()*/displayName);
                taskElem.setDisplaySubject(myTask.getPresentationSubject());
                taskElem.setId(myTask.getId());
                taskElem.setIsEscalated(myTask.isEscalated());
                if(myTask.isEscalated()==true){
                    taskElem.setImageEscalated("~sapicons/s_b_alet.gif");
                } else {
                    taskElem.setImageEscalated(null);
                }
                taskElem.setName(myTask.getName());
                String prio;
                taskElem.setPrio(/*prio*/myTask.getPriority().name());/**
                taskElem.setStatus(myTask.getStatus().name());/**
                taskElem.setTaskType(myTask.getTaskType());
                URL taskExecutionURL = taskInstanceManager.generateTaskExecutionUrl(myTask.getId());
                taskElem.setLinkTEx(taskExecutionURL);
                taskElem.setLinkAsString(taskExecutionURL.toString());
            }
        }
    } catch (BPMEException e) {
        mm.reportWarning("Could not retrieve Data Object");
    }
    //@@end

```

In this portion of code, we are getting all the tasks that are with the provided status as their state

After retrieving the tasks assigned to a user, here we can apply filter on the tasks that comply with a certain naming, so that they can be

In this portion of code, we are setting the details of each task in the UI

In this portion of code, we are getting the URL link to the task, if the user wants to Approve/Reject the task by going to the specific UI assigned for the task.

## 2. Achieving the Mass Approval of tasks:

```

private void approveReject (boolean approve, String reason)
{
    if (wdContext.currentContextElement().getComment() != null &&
wdContext.currentContextElement().getComment().length() > 0)
    {
        reason = wdContext.currentContextElement().getComment();
    }
    try {
        TaskInstanceManager taskInstanceManager = BPMFactory.getTaskInstanceManager();
        IPublicTaskList.ITaskNode taskNode = wdContext.nodeTask();
        IPublicTaskList.ITaskElement taskElem;
        Iterator<Integer> multiIt = taskNode.getMultiSelection().iterateIndexes();

        while (multiIt.hasNext()) {
            taskElem = taskNode.getTaskElementAt(multiIt.next());
            if (taskElem.getStatus().equalsIgnoreCase(Status.READY.name())) {
                taskInstanceManager.claim(taskElem.getId());
            }
        }

        DataObject
output = taskInstanceManager.getTaskDetail(taskElem.getId()).getOutputDataObject();
        DataObject
input = taskInstanceManager.getTaskDetail(taskElem.getId()).getInputDataObject().
getDataObject("start");
        DataObject taskOutput = output.createDataObject(0);
        taskOutput.setBoolean("ApprovedFlag", approve);
        taskOutput.set("ApprovalNote", reason);
        taskOutput.set("RequesterMail", input.get("RequesterMail"));
        taskOutput.set("RequesterNote", input.get("RequesterNote"));
        taskOutput.set("RequesterID", input.get("RequesterID"));

        DataObject output_Order = taskOutput.createDataObject("DO_Order");
        DataObject input_Order = input.getDataObject("DO_Order");
        output_Order.set("Quantity", input_Order.get("Quantity"));
        output_Order.set("Region", input_Order.get("Region"));
        output_Order.set("MaterialText", input_Order.get("MaterialText"));
        output_Order.set("MaterialNumber", input_Order.get("MaterialNumber"));
        output_Order.set("Price", input_Order.get("Price"));
        output_Order.set("totalPrice", input_Order.get("totalPrice"));

        taskOutput.set("DO_Order", output_Order);
        output.set(0, taskOutput);
        taskInstanceManager.complete(taskElem.getId(), output);
    }
    refresh();
} catch (BPMEException e) {
    mm.reportWarning("Could not complete task");
}
}

```

We need to claim the tasks for completing them

In this code snippet we take the input of the task, and process it and set the output of the task. This is being achieved using the SDO API. The DO used for this demo is specific to the IAP scenario.

This code snippet completes the task with the specified output

## 4. Configuring and Running Demo.

Please follow the steps below to run the demo.

1. Import the attached .SCA into NWDS. And deploy all the DCs within the SCA through NWDS the AS Java server



MyComponen00\_0\_allDCs.sca

2. Please follow the instructions given in the following document in the .zip file to configure the AS Java server for IAP process to run.



IAP\_Configuration.zip

3. Set the following properties for getting the Task URL correctly:  
Navigate to "Configuration -> Infrastructure -> Java System Properties" and select "Applications" tab in the "Details" table. Search for "tc~bpem~base~ear" and modify the following property in the "Extended Details" table to relevant hostname.
  - a. http.baseUrl : http://<host\_ipaddress>:50000
4. Set the following details in HTTP Proxy tab navigating to "SOA -> Technical Configuration -> SOA Middleware Global Settings" to:
  - a. Host: proxy
  - b. Port: 8080
  - c. Username:
  - d. Password:
  - e. Exclude List: <add CE server IP Address to the existing list>
  - f. Bypass Proxy Server for Local Addresses: checked
5. Configuring Services Registry service interface:
  - a. Go to SOA -> Application and Scenario Communication -> Single Service Administration
  - b. Select Consumer Proxies tab
  - c. Search for Find: "ServicesRegistrySi" Search by: "WSDL Port Type Name" State: "(All)" and click "Go"
  - d. In the search results select "ServicesRegistrySi", a set of tabbed views will be shown below the results table. Now select the "Configuration" tab shown below. There will be another set of tabbed views shown below the current set of tabbed views. In those tabs, select General and click "Edit" button and provide the Web Service Endpoint URL as <http://sr.esworkplace.sap.com:80/ServicesRegistrySiService/ServicesRegistrySiPort>.
  - e. Now select the "Security" tab and select "HTTP Authentication" for Authentication. And select "User ID/Password (Basic)" and click "Details" button.
  - f. Now provide the following "User ID" and "Password" for logging in to ES Workplace.  
UserID: **sruser**  
Password: **eswork123**
  - g. Save the changes
6. After deploying all the DCs contained in the .SCA file onto the AS Java server, login to the SAP NWA as an administrator and do the following steps to run the demo:
  - a. In SAP NetWeaver Administrator, go to Configuration -> Process and Tasks and select "Process Repository"

- b. Search for “demo\_720” in the “List of components” table and select “demo\_720” from the displayed results.
- c. In the “Details of Component demo\_720” table displayed at the bottom of the screen, select the “demo720\_IAP\_solution” process from the “Processes and Tasks” tab. This should enable the “Process Flow” and “Start Porcess...” buttons. Click on “Start Process...” button.

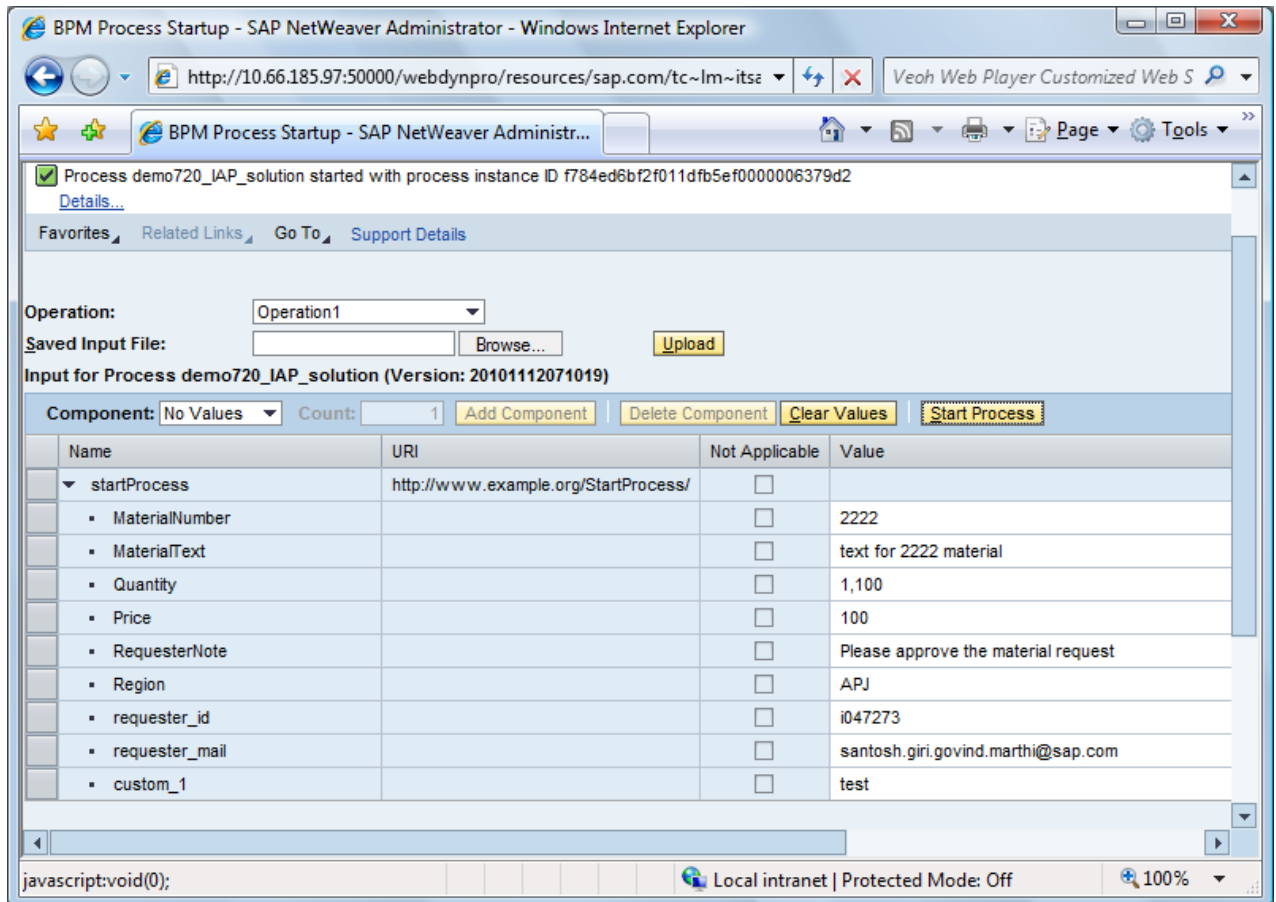
The screenshot displays the SAP NetWeaver interface for component management. It is divided into three main sections:

- List of components:** A table listing various components. The component "demo\_720 (demo.sap.com)" is highlighted in yellow. Other components include "mytest", "gartner\_demo", "assessmentday-pm", "assessmentday-bl-rules", and several "tc-bpem-content" and "test-glxsimpletypesdc" components.
- List of Versions of Component demo\_720 (demo.sap.com):** A table showing different versions of the selected component. The version "20101112071019 (localDevelopment)" is selected with a checkmark. Other versions include "20101112065527", "20101112064820", "20101112063929", and "2010111141004".
- Details of Component demo\_720 (demo.sap.com) and Version 20101112071019 (localDevelopment):** This section has two tabs: "Processes and Tasks" and "Resources". The "Processes and Tasks" tab is active, showing a table of processes. The process "demo720\_IAP\_solution" is highlighted in yellow. Other processes include "ApproveRejectIAP\_Sol", "correlation\_test", "correlation\_test\_2", and "rework\_iap\_solution".

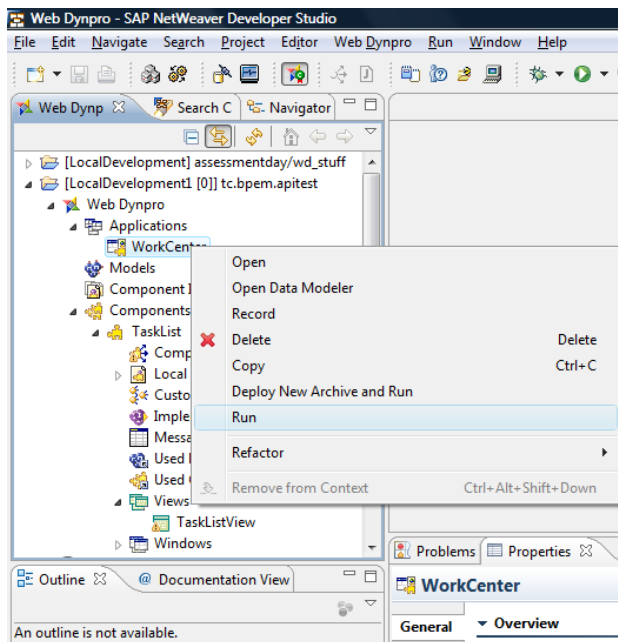
At the bottom of the interface, there are buttons for "Process Flow" and "Start Process...". The status bar at the bottom indicates "Local intranet | Protected Mode: Off" and "100%".

- d. Please provide the input while starting the process. Sample screenshot for the process start page is as follows:





- e. Now open the "tc.bpem.apitest" WebDynpro project in NWDS and go to the "WorkCenter" node under the Web Dynpro -> Applications. And select the "Run" option from the context menu.



- f. The above action will open up a new browser showing the latest UWL UI where you can select multiple tasks at a time and either “Accept” or “Reject” providing a “Remark”

**Task List** ☰

Remark: Approving two tasks at a time
Approve

Reject

<input type="checkbox"/>	Task	Material ID	Quantity	Price	Priority	Status	Created on
<input type="checkbox"/>	<a href="#">Please Approve Request from santosh.qiri.govind.marthi@sap.com</a>	1111	10000	99	MEDIUM	READY	2010-11-18T9:50:37.950
<input type="checkbox"/>	<a href="#">Please Approve Request from santosh.qiri.govind.marthi@sap.com</a>	2222	1100	100	MEDIUM	READY	2010-11-18T9:51:01.817
<input type="checkbox"/>	<a href="#">Please Approve Request from santosh.qiri.govind.marthi@sap.com</a>	3333	1000	200	MEDIUM	READY	2010-11-18T9:52:15.343
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							

- g. Select more than one task and provide the “Remark” for the approval, and click on “Approve” button.
- h. There should be a mail sent to the e-mail address provided in the input screen (Step d)
- i. In the Step g. instead of “Approve”-ing the selected tasks you can even “Reject” them. Then the process will continue to the next step accordingly.

## Related Content

For more information, visit the [Business Process Modeling homepage](#)

## Copyright

© Copyright 2010 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 Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., 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.