

Web Service Fault Handling in SAP Netweaver Business Process Management



Applies To:

SAP Netweaver Composition Environment 7.2, 7.3. For more information, visit the [Business Process Modeling homepage](#).

Summary

This document describes how to handle faults from web services in SAP Netweaver BPM by using Enterprise Java Bean (EJB) as a service composition layer.

Author: Chembrakalathil Venugopal

Company: SAP AG

Created on: 13 December 2010

Author Bio

Chembrakalathil Venugopal has experience in various SAP Netweaver CE technologies and working closely with SAP Netweaver CE customers.

Table of Contents

Introduction	3
Overview	3
Scenario Description.....	4
Value Proposition.....	5
Technical Solution Description	5
How to Model the Scenario.....	6
Transport Binding	9
Web Service Retry in BPM.	11
Related Content.....	12
Copyright.....	13

Introduction

This document describes how webservice faults can be captured and propagated in SAP Netweaver Composition Environment (CE) 7.2. It also describes calling services without using http protocol where the end point is in the same SAP Netweaver CE system.

SAP Netweaver Business Process Management (BPM) consumes web services for automated activities. The end point / provider of the webservice could be the same SAP Netweaver CE system or any other backend or third party system. It is very important to handle the faults from these webservice calls to have a smooth process execution.

Overview

An automated activity can fail due to many reasons and this will change the process state into a failed state. A failed process instance will stop executing the process steps until it has been recovered by the administrator or a system restart has occurred. To avoid the delay in responding to such failures, the system needs to capture the webservice failures and notify the concerned users.

An automated activity can fail either due to a technical failure or an application related failure. In both cases the process instance will be in a failed state. In certain scenarios the webservice runtime may not throw an exception even if there is an application related failure. For example while executing an Enterprise Service the response message will either contain a successful message or a set of logs with the cause of failure. If the response is a fault message due to an application fault, normally the process should go to an error state. However, there is no fault raised by the service interface and the process will not be in a failed state.

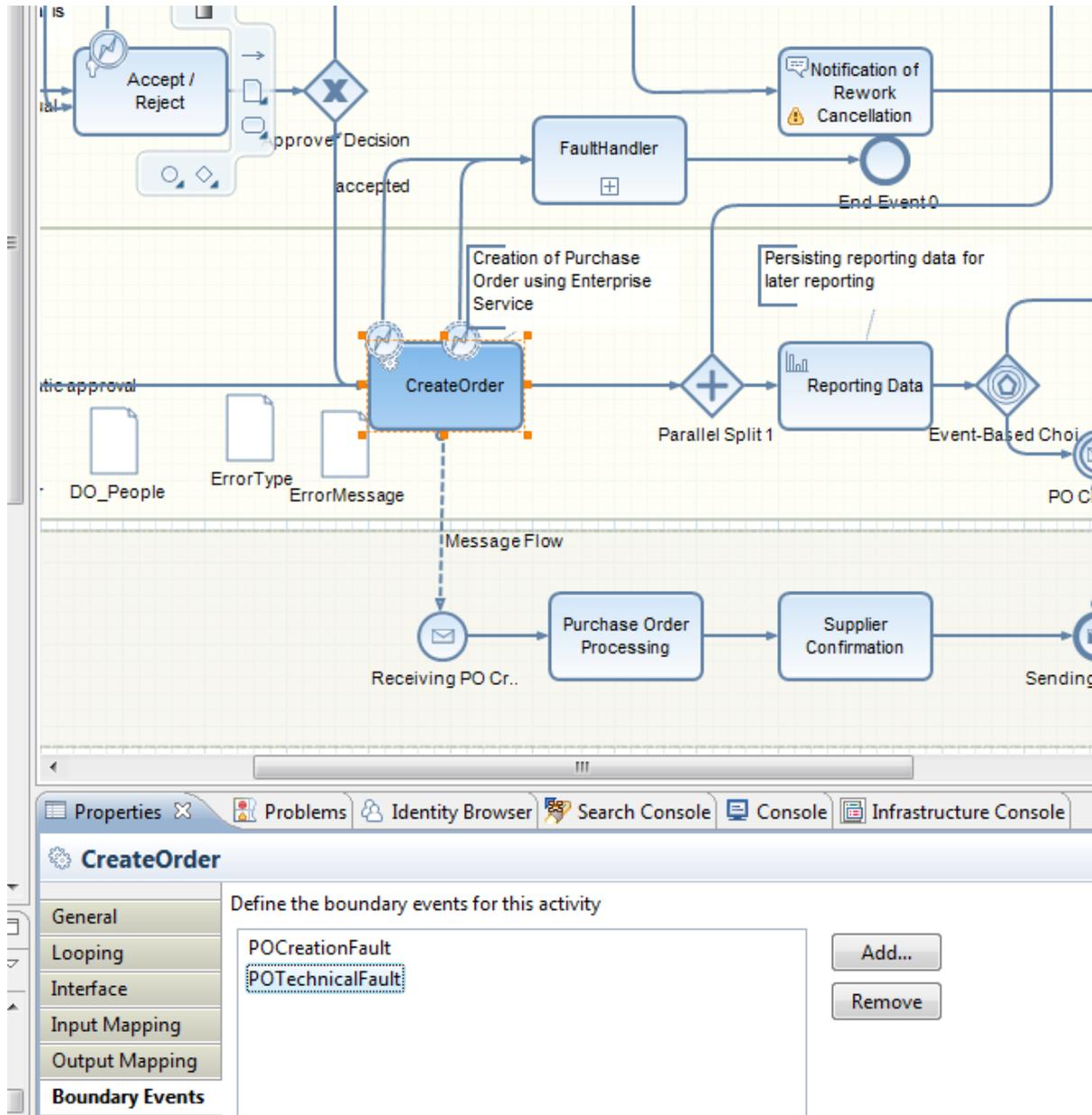
SAP Netweaver BPM design time offers a possibility to define boundary events for an automated activity. The boundary events are defined using the SOAP faults modelled in a webservice. These are the potential application related faults which are expected by the Service provider. At the same time web services can fail due to technical reasons, too. For example: time out, authentication, network related issues, other http related issues etc. These technical faults cannot be modeled in the service interface. But these faults too will change the process state into a failed status. In order to quickly react to such failures, the process model needs to capture all the potential faults and act on it.

For more details about boundary events refer the following SAP Help Portal documentation (related to SAP Netweaver CE 7.2).

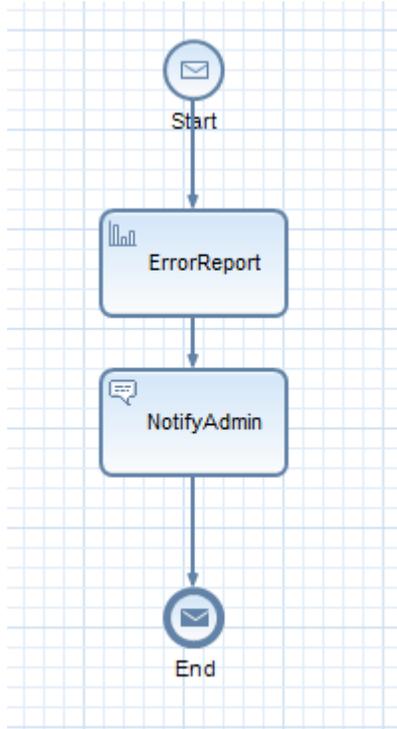
http://help.sap.com/saphelp_nwce72/helpdata/en/54/bf98c82cd84614a85cfda25d70b175/content.htm

Scenario Description

In this example we have implemented how a purchase order request process is consuming an Enterprise Service to create a purchase order in the ERP system (please find the service definition in [ESWorkplace](#)). The service interface exposes the message faults (both technical and application fault) and they are handled in the process model.



In the screen shot above an automated activity(CreateOrder) is consuming the POCreation serviceinterface and this interface is exposing two message faults. The PO Creation fault will address any application related faults. The POTechnical fault will address technical exceptions thrown by the webservice runtime. Both message faults are handled through a subprocess.



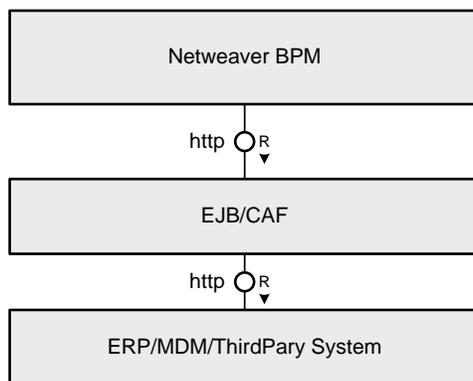
The subprocess consumes the errors and notify the administrator and persists the message faults in a reporting activity for analytical purposes.

Value Proposition

The technical fault handling will be part of the process model and the administrator will get notification for such events. This will reduce the turnaround time to resume the process instance and effectively increase the efficiency of the process. These failures can also be analyzed using lean analytics later to do root cause analysis of the webservice failures.

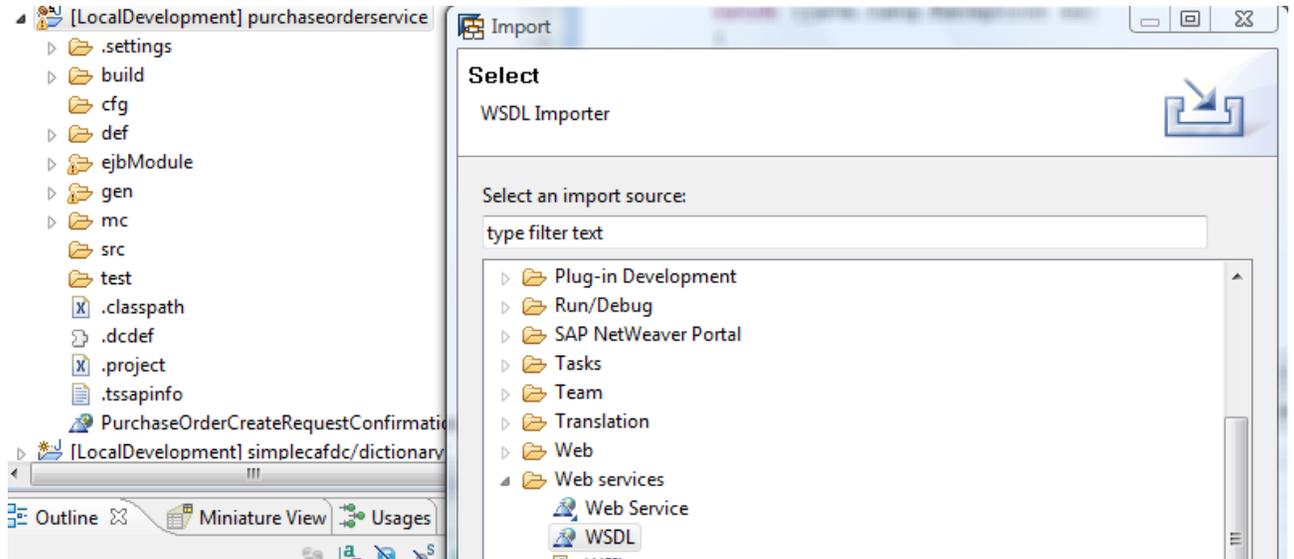
Technical Solution Description

An EJB DC is created and the purchase Order Creation wsdl is imported into this DC. The client proxies are generated and a service reference is created. The client proxies are invoked using a session bean. The session bean handles the errors and throws the faults appropriately. The session bean will be exposed as a webservice and will be consumed by SAP Netweaver BPM.

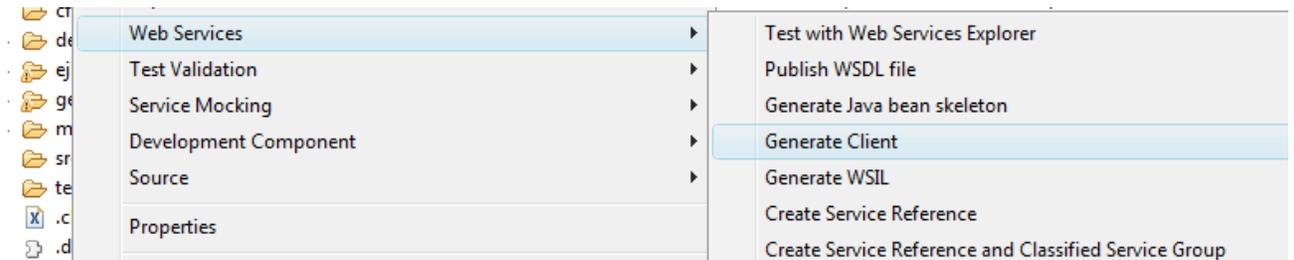


How to Model the Scenario

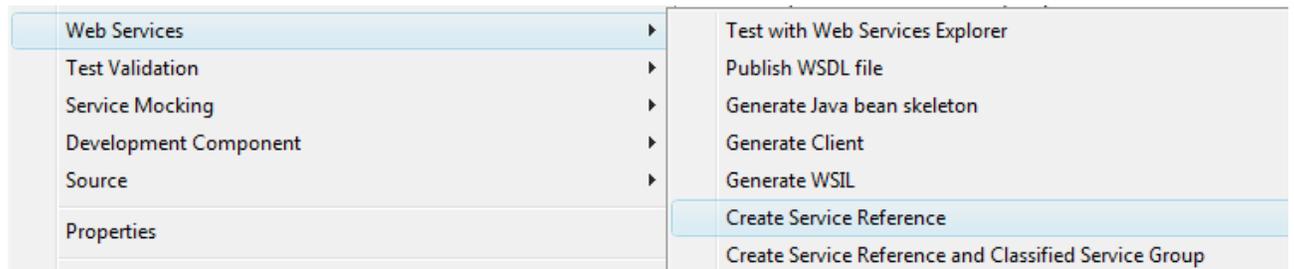
1. Import the purchase order wsdl to an EJB DC.



2. Create the client proxies for the wsdl.



3. Create a service reference for the SOA configuration.



4. Create a session bean and define an operation
5. Define the consumption of the service interface using `WebServiceRef` annotation.
6. Consume the client proxies to invoke the service operation.
7. Define a simple Java class and extend it from `java.lang.Exception`
8. Define a second Java class and extend it from `java.lang.Exception`
9. Define the exceptions at the interface level.

10. Add the below code

```

@javax.xml.ws.WebServiceRef(name="PurchaseOrderCreateRequestConfirmation_In")
protected PurchaseOrderCreateRequestConfirmationInService poClientProxy;

public PurchaseOrderCreateConfirmationMessageSync
createPurchaseOrder(PurchaseOrderCreateRequestMessageSync
purchaseOrderCreateRequest_sync) throws POCreationFault, POTechnicalFault {
    PurchaseOrderCreateConfirmationMessageSync result = null;

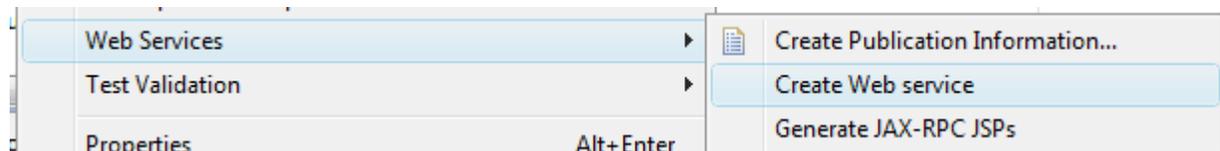
    try {

com.sap.xi.appl.se.global.PurchaseOrderCreateRequestConfirmationIn poPort =
poClientProxy.getPort(com.sap.xi.appl.se.global.PurchaseOrderCreateRequestConfirmatio
nIn.class);
result =
poPort.purchaseOrderCreateRequestConfirmationIn(purchaseOrderCreateRequest_sync);
        if(result.getPurchaseOrder()==null)
        {
            if(result.getLog().getItem().get(0) != null)
throw new POCreationFault(result.getLog().getItem().get(0).getNote());
        }

        }
        catch (java.lang.Exception ex)
        {
            if(ex instanceof POCreationFault)
                throw (POCreationFault)ex;
            if(ex instanceof StandardMessageFault)
                throw new POCreationFault(ex);
            if((ex instanceof WebServiceException))
            {
                throw new POTechnicalFault(ex);
            }
        }
        return result;
    }
}

```

11. Expose the session bean as a webservice .



12. Deploy the EJB DC and copy the wsdl of the newly generated webservice from the SAP Netweaver Administrator ->SOA Management->Application and Scenario Communication->SingleService Administration->ServiceDefinitions

Single Service Administration: Service Definitions

Service Definitions | Consumer Proxies

Search | Browse

Find: *POCo* Search by: WSDL Port Type Name State: (All) Go

Found Service Definitions: 1

WSDL Port Type Name	Namespace
POComposedServiceLocal	http://sap.demo.com/

Details about service definition "POComposedServiceLocal"

General | WSDLs | Configuration | Logs and Traces | Classifications

ZIP Download Test

WSDL

http://127.0.0.1:50000/POComposedServiceService/POComposedService?wsdl&mode=ws_policy

13. Import the webservice to the BPM DC and set the service interface to an automated activity.
14. Define boundary events by selecting the faults
15. Handle the boundary events accordingly.

Transport Binding

If the service end point / provider system of a webservice is the same as the local SAP Netweaver CE system, it is possible to define the service invocation as a local call. This means that the webservice invocation will not be via http protocol and the native binding will be used. The endpoint of the service will be invoked in the same thread of the consumer proxy .This will give a single stack trace of the end to end service call. In addition this will avoid consumption of http threads.

System Connections: Communication Profiles

Provider Systems
Communication Profiles
Required Systems

New
Edit
Delete
Activate
Deactivate
Import
Export

Name	Version	Connectivity Type	Status
SAP_DEFAULT_PROFILE	1	WS	Inactive
	4	WS	Active
	2	WS	Inactive
	3	WS	Inactive
SAP_DEFAULT_RFC_PROFILE	1	RFC	Active

Details About Communication Profile "SAP_DEFAULT_PROFILE" v.4

Authentication

Authentication Method:

- None
- User Name/Password (Basic)
- X.509 Client Certificate
- SAP Logon Ticket
- User Name/Password Doc.Auth.
- SAML Assertion
- X.509 Certificate Doc.Auth.

Transport Security

Allow Only Secure Communication:

Reliable Messaging

Lifetime in milliseconds of a sequence:

Retransmission interval in milliseconds:

Confirmation Interval:

Inactivity Timeout in milliseconds:

Exponential Backoff:

Transport Binding

Use 'Connection: KeepAlive' http header:

Support gzip responses:

Max wait-time for http response (in milliseconds):

Issue chunked requests:

Transport:

- local
- http
- local or http

For example: If the SAP_DEFAULT_PROFILE profile has been assigned to the provided service and the provider system the consumer configuration will have the following webservice end point URL.

The screenshot displays the SAP Netweaver Business Process Management console interface. At the top, there are tabs for 'Consumed Services' and 'Provided Services'. Below these are buttons for 'Assign Provider System', 'Deassign Provider System', 'Auto Assign', and 'Show Log'. The main area is titled 'Service Interface' and contains a table with columns 'Name' and 'Namespace'. The table lists two service groups: 'Service group: "ejbconsumer"; Assigned provider: "C72 on local"' and 'Service group: "www_example_org"; Assigned provider: N/A'. The 'ejbconsumer' service is highlighted in yellow, showing its namespace as 'http://client.demo/'. Below the table, there is a section titled 'Details about "ejbconsumer" reference' with tabs for 'General', 'Configuration', and 'Classifications'. Under 'Configuration', there are buttons for 'Configure' and 'Remove Configuration'. Further down, there are tabs for 'General', 'Security', 'Messaging', 'Web Service Addressing', and 'Transport Settings'. The 'Web Service Addressing' tab is active, showing the 'Logical Port Name' as '{http://www.sap.com}myprocess_demo.sap.com~myprocess~BPMcomponent_00' and the 'Web Service End Point URL' as 'http://localhost:711/sapws/demo.sap.com/ejbconsumerear_demo.sap.com%7E'.

In order to set the binding as local refer to the SAP Help Portal documentation(related to SAP Netweaver CE 7.2).

http://help.sap.com/saphelp_nwce72/helpdata/en/af/9101f1c0774a1781fbbafd5dfd3c7/content.htm

Web Service Retry in BPM.

If there is a failure while invoking a webservice, the invocation will be retried two more times by default by the BPM runtime. However, depending on the customer requirements this can be reconfigured. For example: the back end system may be busy for a specific time period or the service endpoint may not be able to process anymore http requests etc . **A higher value of this retry parameter with a higher websevice timeout will cause shortage of http threads. Therefore it is advisable to set this parameter carefully according to the requirements.**

The time interval between the retries can be reconfigured. If the endpoint is busy due to some reason the successive calls will be able to execute the webservice.

1. **transition.backoff.max.retries**
Maximum numbers of retries of a webservice call.
2. **transition.backoff.max.secs**
Maximum time gap between two retries.
3. **transition.backoff.mode**
The retry pattern will be based on this property value.
It could be "lin" or "exp". The retries will be done in a linear fashion or exponential fashion

Java System Properties: Overview Personalize | Back Forward

Templates

Name
Usage_Type_All_in_One

Details about Usage_Type_All_in_One

Kernel | **Services** | Applications | VM Environment | Memory VM Parameters | System VM Parameters | Additional VM Parameters | Filters | Shared Tables

Name	Service Component Name
Galaxy	
Galaxy Core Service	com.sap.glx.core.svc
Galaxy Core Service Facade	com.sap.glx.core.svc.facade
Galaxy Repository Service	com.sap.glx.repository.svc
Galaxy RR Handler Manager Service	com.sap.glx.rhandlermanager.svc
Galaxy Base Service	tc~bpem~base~svc

Extended Details

Properties

Save As | Restore Default | Modify | Show Details

Name	Default Value	Modifiable Online	Custom Value
service.user	SAP_BPM_Service	<input type="checkbox"/>	
transition.backoff.basis.secs	1	<input type="checkbox"/>	
transition.backoff.max.retries	3	<input type="checkbox"/>	
transition.backoff.max.secs	3	<input type="checkbox"/>	
transition.backoff.mode	lin	<input type="checkbox"/>	

Related Content

[The New SOA Configuration Approach](#)

[How to Use SOA Configuration to Call RFCs and Web Services from within SAP NetWeaver BPM](#)

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.