

Advanced Flat File to MII XML Document Custom Action Blocks (v12.0)



Applies to:

SAP Manufacturing Integration & Intelligence (MII) - v12.0.x, Flat Files (Any delimiter or EOL character)

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

Summary

The following document will review how to install and use the Advanced Flat File to XML Document Conversion Custom Action Blocks in order to parse a flat file into an XML document. The action supports any flat file with any value delimiter and end of line delimiter and will build an MII formatted XML document as its output.

Author: Salvatore Castro

Company: SAP Labs, LLC

Created on: 29 October 2009

Author Bio

Salvatore Castro of SAP Labs has a Bachelors Degree in Computer Engineering and is currently working on completing his Masters Degree in Computer Science both through the Rochester Institute of Technology. He is a member of the MII Product Management group of MII under John Schaefer.

Table of Contents

Business Scenario for Using Flat File Parsing	3
Parameter Definitions.....	3
Input Fields.....	3
Column Definition.....	3
Operation Details	3
Output Fields.....	4
Uploading and Deploying the Custom Actions	4
Using the Flat File to MII XML Document Action.....	5
Building a Transaction that Uses the Flat File to MII XML Document Converter.....	5
Configuring the Sample Transaction for Auto Expansion.....	5
Transaction Output Document	8
Configuring the Sample Transaction with Static Columns	9
Transaction Output Document	11
Related Content.....	13
Copyright	14

Business Scenario for Using Flat File Parsing

When integrating various systems in a manufacturing environment a need can arise to deal with legacy systems whose only output is a flat file. Since there are many variations of flat files with any number of "End of Line" and value delineation characters, "Delimiters", this can be a somewhat challenging task given the current toolset. As a result the need for the Flat File to XML Document conversion action to eliminate any guesswork or integration issues, along with simplifying your integration strategy.

~~When integrating various systems in a manufacturing environment, a need can arise to write various messages in an asynchronous manner to that they can be processed by another system. Additionally, you may also wish to receive messages, or work, in an asynchronous manner inside the xMII environment as well. In order to accommodate this need the use of a Messaging Service, in this case the NetWeaver Java Messaging Service (JMS) can be used; also understood as a queuing service. Integration to this Queuing Service also allows for support of integration to various systems via a means of message passing which abstracts the xMII system from having to communicate directly with it. This will benefit the overall xMII system performance if there is a slow or unreliable network connection between the two systems.~~

Formatted: Heading 1

Parameter Definitions

The Flat File conversion action has various features in order to accommodate various flat file formats and variations. It is important to note that your columns names and data may be XML encoded in order to comply with the XML document output standard. The action input values affect the actions operation for data manipulation as follows:

Input Fields

Column Definition

Columns: When the columns and their data types are known ahead of time they can be statically specified here. This will disable the "Has Columns" checkbox so that the first row from the input will always be treated as data values. The dialog window works the same as the SAP xMII Output XML -> Document action.

Has Columns: If this input field is checked then the action will use the first row of input data to define the XML document column names. If this is unchecked then the columns names will be automatically generated as Column1...ColumnN. All columns will be of "String" data type in order to handle all data generically; SQL Data Type in the column XML definition is set to a value of 1.

Auto Expand: This input checkbox when check will cause the document to expand to handle rows with extra data values in them to prevent the loss of data during conversion. For example when the first row of flat data has 4 values and a subsequent row has more than 4 values a column will be added to the document to handle the extra value. If this value is unchecked then no additional columns will be added and if the above situation does occur, then the Boolean output value of "TruncatedData" will be set to true, otherwise it will be false.

Operation Details

End of Line Character(s): If this input is left blank then the end of line character is based on the Java Buffered Reader -> readLine() method as defined in the v1.4.2 Java Documentation ([http://java.sun.com/j2se/1.4.2/docs/api/java/io/BufferedReader.html#readLine\(\)](http://java.sun.com/j2se/1.4.2/docs/api/java/io/BufferedReader.html#readLine())). This states that a line is defined as a set of characters terminated by any one of a line feed ('\n'), a carriage return ('\r'), or a carriage

return followed immediately by a linefeed. If you require a special character or set of characters to indicate a new line enter them here.

Delimiter: If this input is left blank then the action will automatically use a comma as the delimiter which is also its default value. The delimiter is used to break up the various data values from each flat file row and load them into the proper XML document items.

Trace: This input checkbox if selected will print out a trace of the internal actions operation for design time debug and action operation validation.

Input Text: The input text area here is for configuration of input data for conversion to MII XML format.

Output Fields

Success: This Boolean flag is set to true if the parse operation was successful.

LastErrorMessage: This output string will contain any error messages received during execution.

OutputXML: The contents of the flat file parsed into MII XML Document format.

TruncatedData: If there were extra values in any of the rows and "AutoExpand" was disabled then this will be true to indicate that some of the data in the parsed rows was lost during the parse.

Uploading and Deploying the Custom Actions

The deployment of custom actions is documented in the xMII v12.0 Help documentation which can be found online (<http://help.sap.com> -> Select SAP xApps -> xApp for Manufacturing Integration & Intelligence).

The specifics for installing these actions are simply to navigate to the following in xMII:

System Management -> Custom Actions

~~Click the~~ **Select** Upload Button

In the Assembly File Upload window

For Assembly file browse to the [FlatFileUtilities.jar](#) (**Click** to download the file).

~~Click~~ **Press** the Save button

~~Click~~ **Press** the Deploy Button

Your actions will now appear in the xMII Workbench the next time you open up the logic editor under "File Manipulation".

Formatted: Font color: Auto

Formatted: Font color: Auto

Formatted: Font color: Auto

Using the Flat File to MII XML Document Action

This section will demonstrate how the Flat File to XML conversion action works and will demonstrate how to build a sample transaction around the action. Enabling the Trace parameter will give you more visibility into what's going on behind the scenes with the action and will help you to understand and debug its operation. Once this action is working properly it is recommended that you uncheck the Trace parameter.

Building a Transaction that Uses the Flat File to MII XML Document Converter

The **FlatFileToDoc** action block will create an MII XML formatted document based on the flat file input to the action.

Configuring the Sample Transaction for Auto Expansion

Create an input property of type String with the following value:

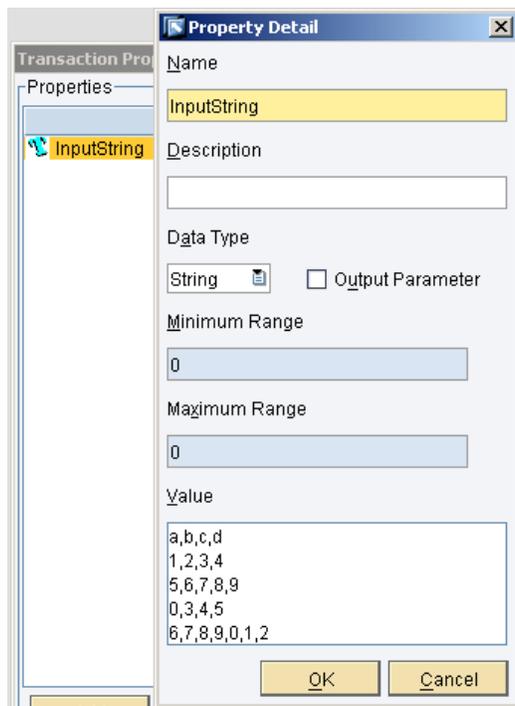
a,b,c,d

1,2,3,4

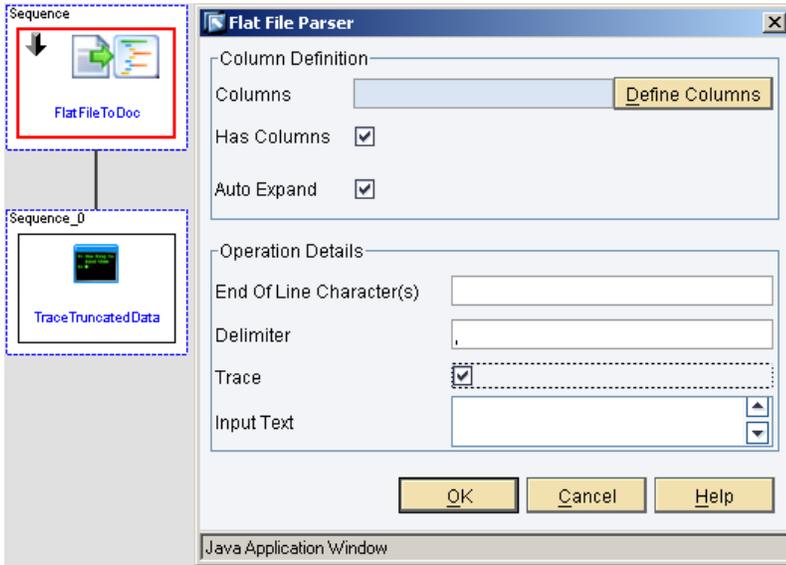
5,6,7,8,9

0,3,4,5

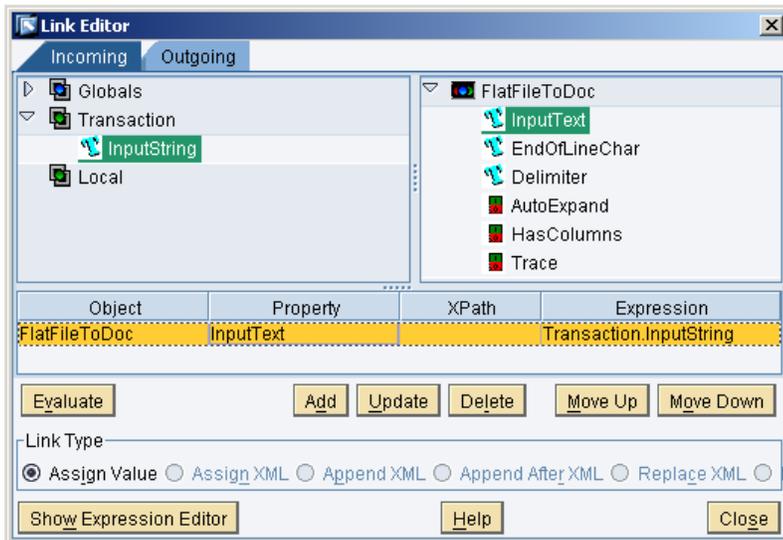
6,7,8,9,0,1,2



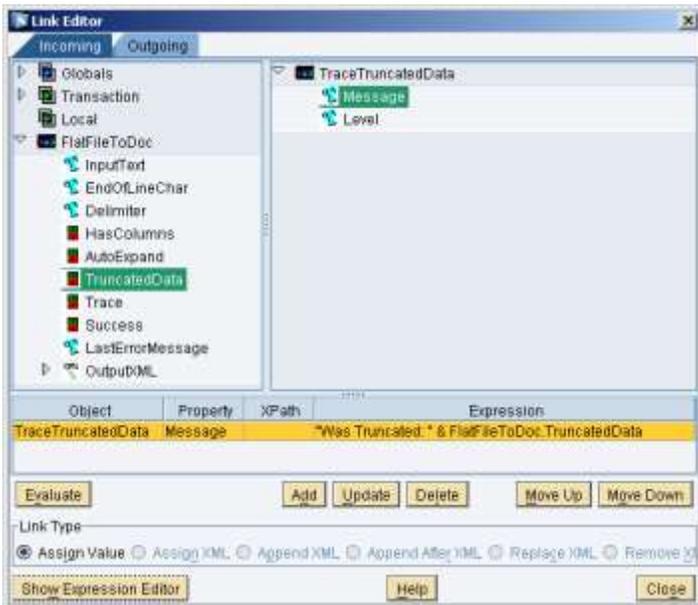
Next create a transaction that looks like the one below and configure the FlatFileToDoc action as shown below:



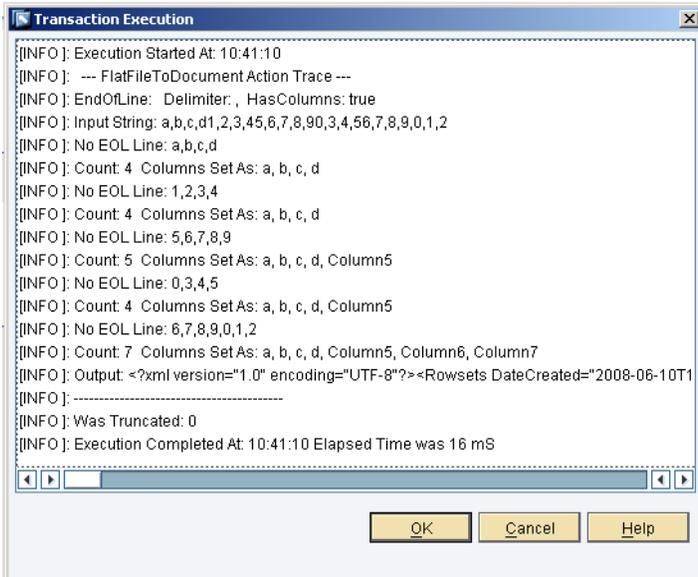
Next link the Transaction.InputString property to the FlatFileToDoc.InputText field as shown below:



Finally, link the FlatFileToDoc.Truncated data to the TraceTruncatedData.Message as shown below:
 "Was Truncated: " & FlatFileToDoc.TruncatedData



Run the transaction without step trace (Ctrl + F6), the expected results are:



It is important to note that the SQL Data Type values are defaulted to Strings and that the rows containing extra data values caused the creation of columns dynamically.

Transaction Output Document

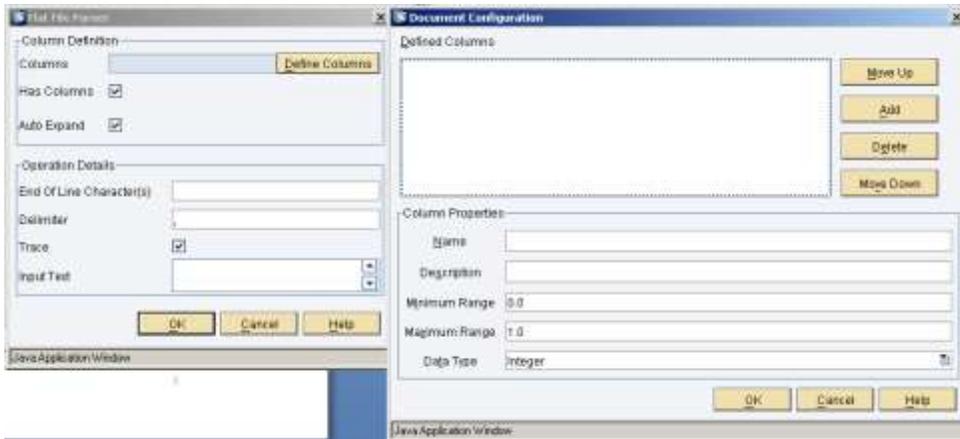
```
<?xml version="1.0" encoding="utf-8"?>
<Rowsets DateCreated="2008-05-16T08:49:07" EndDate="2008-05-16T08:49:07"
StartDate="2008-05-16T08:49:07" Version="12.0.3 Build(106)">
  <Rowset>
    <Columns>
      <Column Description="a" MaxRange="1" MinRange="0" Name="a"
SQLDataType="1" SourceColumn="a" />
      <Column Description="b" MaxRange="1" MinRange="0" Name="b"
SQLDataType="1" SourceColumn="b" />
      <Column Description="c" MaxRange="1" MinRange="0" Name="c"
SQLDataType="1" SourceColumn="c" />
      <Column Description="d" MaxRange="1" MinRange="0" Name="d"
SQLDataType="1" SourceColumn="d" />
      <Column Description="Column5" MaxRange="1" MinRange="0" Name="Column5"
SQLDataType="1" SourceColumn="Column5" />
      <Column Description="Column6" MaxRange="1" MinRange="0" Name="Column6"
SQLDataType="1" SourceColumn="Column6" />
      <Column Description="Column7" MaxRange="1" MinRange="0" Name="Column7"
SQLDataType="1" SourceColumn="Column7" />
    </Columns>
    <Row>
      <a>1</a>
      <b>2</b>
      <c>3</c>
      <d>4</d>
    </Row>
    <Row>
      <a>5</a>
      <b>6</b>
      <c>7</c>
      <d>8</d>
      <Column5>9</Column5>
    </Row>
    <Row>
      <a>0</a>
      <b>3</b>
      <c>4</c>
      <d>5</d>
    </Row>
    <Row>
      <a>6</a>
      <b>7</b>
      <c>8</c>
      <d>9</d>
      <Column5>0</Column5>
      <Column6>1</Column6>
      <Column7>2</Column7>
    </Row>
  </Rowset>
</Rowsets>
```

* It is important to note the names of the dynamically added columns and the data type is set to 1 for String.

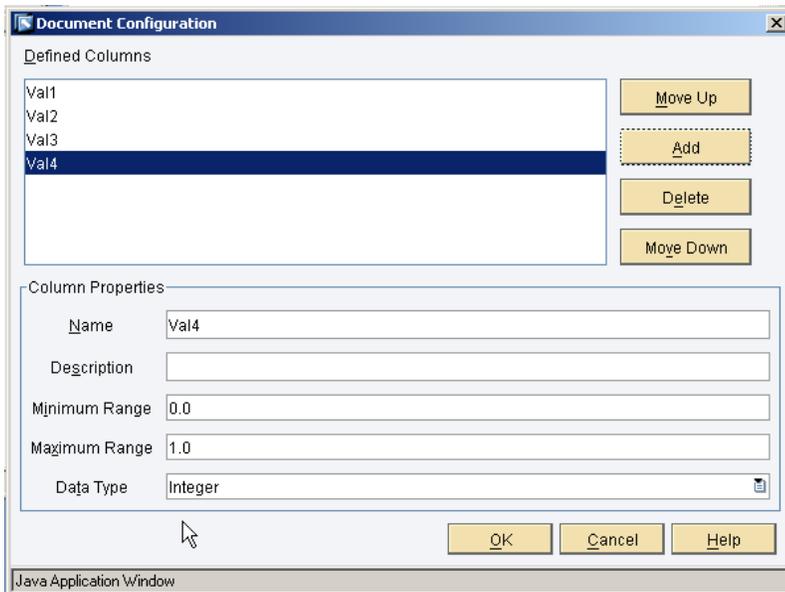
Configuring the Sample Transaction with Static Columns

Using the same transaction as defined in the previous section; change the action configuration so that the document columns are defined ahead of time.

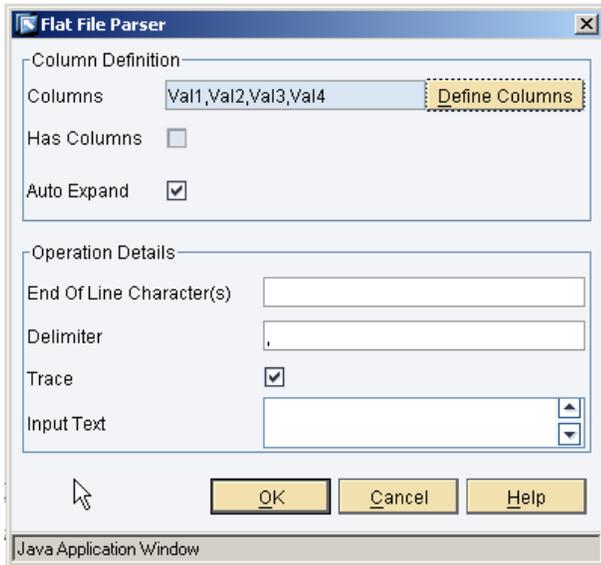
In the configuration dialog press the “Define Columns” button



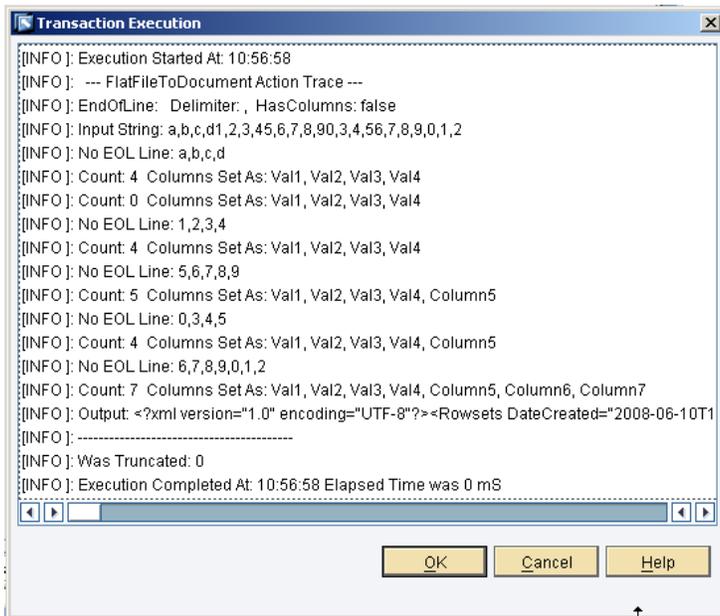
Define four columns called Val1, Val2, Val3, Val4 and set the Data Type of each to Integer and press the “OK” button:



The action configuration should now look like this:



Notice that the “Has Columns” checkbox has been grayed out, Press the “OK” button. Run the transaction without step trace (Ctrl + F6), the expected results are:



Transaction Output Document

```

<?xml version="1.0" encoding="utf-8"?>
<Rowsets DateCreated="2008-06-10T13:55:09" EndDate="2008-06-10T13:52:37"
StartDate="2008-06-10T13:52:37" Version="12.0.4 Build(117)">
  <Rowset>
    <Columns>
      <Column Description="" MaxRange="1" MinRange="0" Name="Va11"
SQLDataType="4" SourceColumn="Va11" />
      <Column Description="" MaxRange="1" MinRange="0" Name="Va12"
SQLDataType="4" SourceColumn="Va12" />
      <Column Description="" MaxRange="1" MinRange="0" Name="Va13"
SQLDataType="4" SourceColumn="Va13" />
      <Column Description="" MaxRange="1" MinRange="0" Name="Va14"
SQLDataType="4" SourceColumn="Va14" />
      <Column Description="Column5" MaxRange="1" MinRange="0" Name="Column5"
SQLDataType="1" SourceColumn="Column5" />
      <Column Description="Column6" MaxRange="1" MinRange="0" Name="Column6"
SQLDataType="1" SourceColumn="Column6" />
      <Column Description="Column7" MaxRange="1" MinRange="0" Name="Column7"
SQLDataType="1" SourceColumn="Column7" />
    </Columns>
    <Row>
      <Va11>a</Va11>
      <Va12>b</Va12>
      <Va13>c</Va13>
      <Va14>d</Va14>
    </Row>
    <Row>
      <Va11>1</Va11>
      <Va12>2</Va12>
      <Va13>3</Va13>
      <Va14>4</Va14>
    </Row>
    <Row>
      <Va11>5</Va11>
      <Va12>6</Va12>
      <Va13>7</Va13>
      <Va14>8</Va14>
      <Column5>9</Column5>
    </Row>
    <Row>
      <Va11>0</Va11>
      <Va12>3</Va12>
      <Va13>4</Va13>
      <Va14>5</Va14>
    </Row>
    <Row>
      <Va11>6</Va11>
      <Va12>7</Va12>
      <Va13>8</Va13>
      <Va14>9</Va14>
      <Column5>0</Column5>
      <Column6>1</Column6>
      <Column7>2</Column7>
    </Row>
  </Rowset>

```

```
</Rowset>  
</Rowsets>
```

* It is important to note that the SQL Data Type for the statically defined columns is set to 4 for Integer.

Related Content

Please include at least three references to SDN documents or web pages.

The SAP SDN xMII Wiki

<https://wiki.sdn.sap.com/wiki/display/xMII/Main>

The SAP SDN xMII Forum

<https://forums.sdn.sap.com/forum.jspx?forumID=237>

The SAP xMII Help Documentation

http://help.sap.com/content/documentation/xapps/docu_xapps_mii.htm

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

Copyright

© Copyright 2009 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+, POWERS, 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.