

How to Create a JCO BAPI/RFC Itemset to xMII Rowset Conversion Utility

Applies to:

xApp Manufacturing Integration & Intelligence version 11.5.x

Summary

The purpose of this document is to provide the user with step by step instructions on how to create a utility transaction that will allow for the quick conversion of JCO response XML to standard xMII document format.

Author(s): Salvatore Castro

Company: SAP Labs, LLC

Created on: 2 October 2006

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 Engineering and Field Enablement Services group of xMII under Mo Ghanem.

Table of Contents

Purpose.....	3
Creating the Utility	4
Creating a Conversion Testing Utility	10
Copyright.....	14

Purpose

The purpose of this document is to address the need for consistent ERP integration techniques between the xMII Business Logic environment and the ERP system. This document will explain in detail how to create and use a transaction utility to generically convert any itemized response set from an xMII JCO (Could be modified for WAS but JCO will be shown) to the standard xMII document format shown in the XML below:

ERP Itemized Response XML	xMII Document Format
<pre> <?xml version="1.0" encoding="utf-8"?> <RSAQ_REMOTE_QUERY_CALL_CATALOG> <INPUT> <GENERIC_FUNCAREA>*</GENERIC_FUNCAREA> <GENERIC_QUERYNAME>*</GENERIC_QUERYNAME> <GENERIC_USERGROUP>SYSTQV000003</GENERIC_US ERGROUP> <WITH_SYSTEM_OBJECTS>X</WITH_SYSTEM_OBJECTS > <WORKSPACE /> </INPUT> <TABLES> <QUERYCATALOG> <item> <NUM>SYSTQV000003</NUM> <QUERY>ZTEST_SAM</QUERY> <VARIANT>ZTEST_VAR</VARIANT> <DIACT>X</DIACT> <QTEXT>Sam, xMII, test query for a customer</QTEXT> <VTEXT>Filter Date, last 10 days</VTEXT> </item> <item> <NUM>SYSTQV000003</NUM> <QUERY>ZTEST_SAM</QUERY> <VARIANT>Z_SAM_VAR</VARIANT> <DIACT>X</DIACT> <QTEXT>Sam, xMII, test query for a customer</QTEXT> <VTEXT>Sam test variant InfoSet Query</VTEXT> </item> </QUERYCATALOG> </TABLES> </RSAQ_REMOTE_QUERY_CALL_CATALOG> </pre>	<pre> <?xml version="1.0" encoding="utf-8"?> <Rowsets DateCreated="2006-08-18T13:16:14" EndDate="2006-08-18T13:16:14" StartDate="2006-08- 18T13:16:14" Version="11.5.1"> <Rowset> <Columns> <Column Description="This column represents a value from an Itemized response JCO doc" MaxRange="1" MinRange="0" Name="VARIANT" SQLDataType="1" SourceColumn="VARIANT"> <Column Description="This column represents a value from an Itemized response JCO doc" MaxRange="1" MinRange="0" Name="VTEXT" SQLDataType="1" SourceColumn="VTEXT"> </Columns> <Row> <VARIANT>ZTEST_VAR</VARIANT> <VTEXT> Filter Date, last 10 days</VTEXT> </Row> <Row> <VARIANT>Z_SAM_VAR</VARIANT> <VTEXT>Sam test variant InfoSet Query</VTEXT> </Row> </Rowset> </Rowsets> </pre>

Table 1: Side by side comparison of XML before and after conversion

Creating the Utility

In order to create the conversion utility configure a transaction to match the one in the image below:

Save the transaction as: JCO_Item_To_Rowset_Converter

Create the following transaction properties:

Transaction Properties

Name: JCO_Input_XML

Description: This is the input XML from JCO

Value: <<XML>>

Name: JCO_Item_XPath

Description: This is the XPath location for the Parent Node of the Item

Value: /RSAQ_REMOTE_QUERY_CALL_CATALOG/TABLES/QUERYCATALOG

Name: NodeList

Description: Use this list, comma delimited, to specify the names of the nodes you wish to have returned.

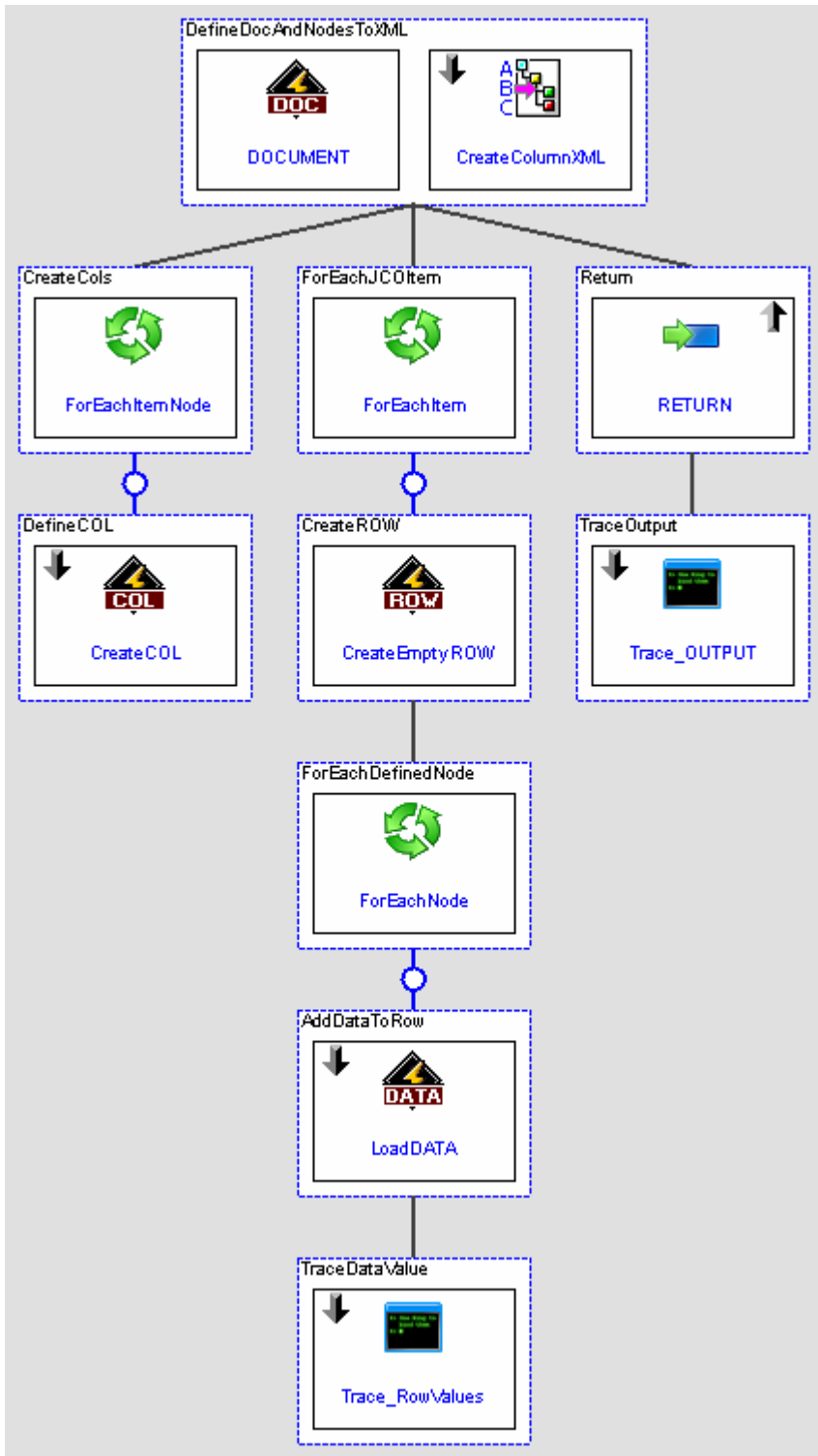
Value: NUM,QUERY,VARIANT,QTEXT,VTEXT

Name: OUTPUT

Description: This is the xMII formatted return XML

Value: <<XML>>

Next place each of the action block in the appropriate Sequences to match the image below:



Finally configure each of the action blocks to match the configurations below:



DOCUMENT - IlluminatorDocument

Description: Create an empty document structure that will serve as a container for the xMII output document.

Inputs:

ColumnSpecifications	
StartDate	2006-08-14T11:21:50
EndDate	2006-08-14T11:21:50
Output	<<XML>>



CreateColumnXML - StringListToXml

Description: Convert the comma separated list of node names to XML format for easy reference.

Incoming Links:

Type: Assign
From: Transaction.NodeList
To: CreateColumnXML.Input

Inputs:

TrimWhitespace	false
StripQuotes	true
Input	
Output	<<XML>>



ForEachItemNode - Repeater

Description: For each item value in the CreateColumnXML action block create a column.

Inputs:

Break	false
CurrentItem	0
ItemCount	0
Source	CreateColumnXML.Output{/Rowsets/Rowset/Row}
Output	<<XML>>



CreateCOL - IlluminatorColumn

Description: Define a new column in the document. The column name comes from the name of the user specified node. Since the datatype value is not specified in the BAPI/RFC call every column is defined as a string.

Incoming Links:

Type: Assign

From: ForEachItemNode.Output{/Row/Item}

To: CreateCOL.Name

Inputs:

MinRange	0
MaxRange	100
IlluminatorDocument	DOCUMENT.Output
SQLDataType	1
Name	JCOutil
Description	This column represents a value from an Itemized response JCO doc



ForEachItem - Repeater

Description: Repeat for each of the items returned in the JCO table.

Inputs:

Break	false
CurrentItem	0
ItemCount	0
Source	Transaction.JCO_Input_XML{#Transaction.JCO_Item_XPath#/item}
Output	<<XML>>



CreateEmptyROW - IlluminatorRow

Description: Create a place holder ROW in the DOCUMENT so that DATA values can be dynamically added.

Inputs:

IlluminatorDocument	DOCUMENT.Output
---------------------	-----------------



ForEachNode - Repeater

Description: Repeat on each of the user defined node values within each of the items.

Inputs:

Break	false
CurrentItem	0
ItemCount	0
Source	CreateColumnXML.Output{/Rowsets/Rowset/Row}
Output	<<XML>>



LoadDATA - IlluminatorDataItem

Description: Load data under the dynamically specified column in the current row.

Incoming Links:

Type: Assign

From: ForEachNode.Output{/Row/Item}

To: LoadDATA.Name

Type: Assign

From: Transaction.JCO_Input_XML{#Transaction.JCO_Item_XPath#/item[#ForEachItem.CurrentItem#]/#ForEachNode.Output{/Row/Item#}}

To: LoadDATA.Value

Inputs:

IlluminatorDocument	DOCUMENT.Output
Name	Test
Value	Test



Trace_RowValues - Tracer

Description: Trace the data value loaded into the row to the execution logger for the user to debug.

Incoming Links:

Type: Assign

From: "Column: " & ForEachNode.Output{/Row/Item} & " Data: " &

Transaction.JCO_Input_XML{#Transaction.JCO_Item_XPath#/item[#ForEachItem.CurrentItem#]/#ForEachNode.Output{/Row/Item#}}

To: Trace_RowValues.Message

Inputs:

Message	
Level	INFO



RETURN - Assignment

Description: Return the xMII formatted XML back to the calling transaction.

Outgoing Links:

Type: AssignXml
From: DOCUMENT.Output
To: Transaction.OUTPUT



Trace_OUTPUT - Tracer

Description: Log the output XML of the transaction to the execution logger for the user to debug the operation.

Incoming Links:

Type: Assign
From: "Item to Rowset Output: " & Transaction.OUTPUT
To: Trace_OUTPUT.Message

Inputs:

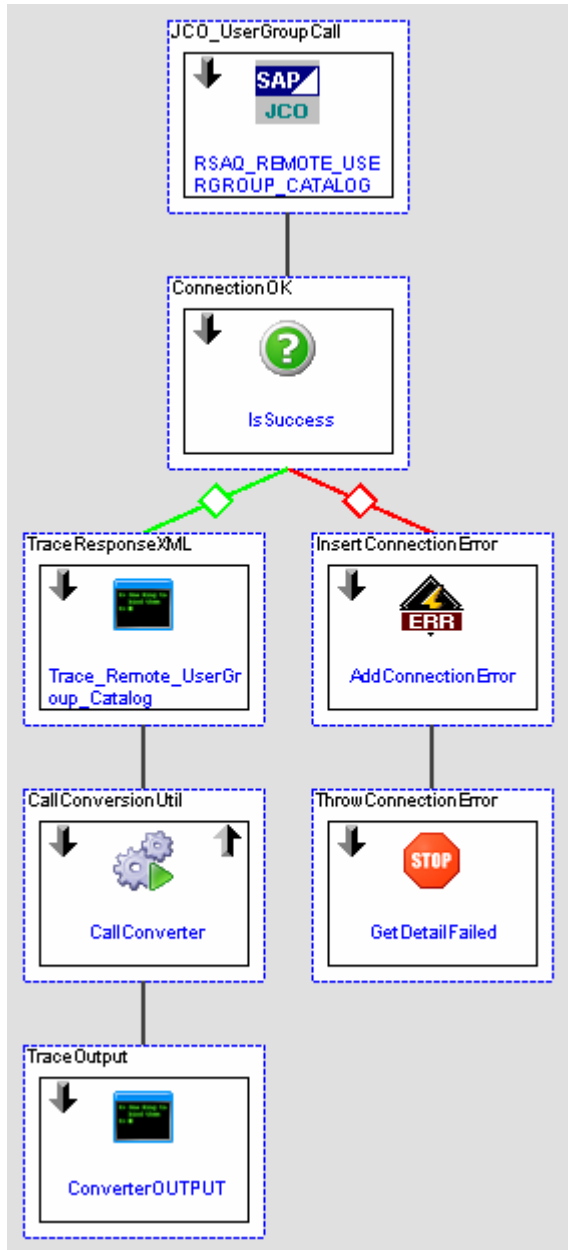
Message	
Level	INFO

Creating a Conversion Testing Utility

Once the utility has been created create the following transaction in order to test its operation:

Save the transaction as: JCO_Item_To_Rowset_Testbench

Create the following transaction:





RSAQ_REMOTE_USERGROUP_CATALOG - SAPJCOInterface

Description: Call the UserGroup Catalog BAPI/RFC using the JCO interface.

Incoming Links:

Type: Assign

From: ""

To: RSAQ_REMOTE_USERGROUP_CATALOG.Request{/RSAQ_REMOTE_USERGROUP_CATALOG/INPUT/GENERIC_USERGROUP}

Type: Assign

From: "X"

To: RSAQ_REMOTE_USERGROUP_CATALOG.Request{/RSAQ_REMOTE_USERGROUP_CATALOG/INPUT/WITH_SYSTEM_OBJECTS}

Inputs:

Request	<<XML>>
Response	<<XML>>
AutoCommit	false
SAPRFC	RSAQ_REMOTE_USERGROUP_CATALOG
ExecuteFunction	true
Language	EN
SAPSystemAlias	
SAPServerName	
SAPClient	
SAPUserName	
SAPPassword	
SAPSSO2Ticket	
SAPSystemNumber	



IsSuccess - Conditional

Description: If the JCO Success flag is set to true then the JCO API successfully connected to the ERP instance.

Incoming Links:

Type: Assign

From: RSAQ_REMOTE_USERGROUP_CATALOG.Success

To: IsSuccess.Input1

Inputs:

Input1	false
Output	false
InputCount	1
LogicalAnd	false



Trace_Remote_UserGroup_Catalog - Tracer

Description: Trace out to the execution logger the JCO Response XML

Incoming Links:

Type: Assign

From: "UserGroupCat: " & RSAQ_REMOTE_USERGROUP_CATALOG.Response

To: Trace_Remote_UserGroup_Catalog.Message

Inputs:

Message

Level INFO



CallConverter - Transaction

Description: Call the conversion utility transaction, that will convert the items at the specified XPath in the input XML to xMII Rowsets format.

Incoming Links:

Type: AssignXml

From: RSAQ_REMOTE_USERGROUP_CATALOG.Response

To: CallConverter.JCO_Input_XML

Type: Assign

From: "NUM,UTEXT"

To: CallConverter.NodeList

Type: Assign

From: "/RSAQ_REMOTE_USERGROUP_CATALOG/TABLES/USERGROUPCATALOG"

To: CallConverter.JCO_Item_XPath

Inputs:

ResetState	false
JCO_Input_XML	<<XML>>
JCO_Item_XPath	/RSAQ_REMOTE_QUERY_CALL_CATALOG/TABLES/QUERYCATALOG
NodeList	NUM,QUERY,VARIANT,QTEXT,VTEXT



ConverterOUTPUT - Tracer

Description: Log out to the execution logger the converted XML structure.

Incoming Links:

Type: Assign

From: CallConverter.OUTPUT

To: ConverterOUTPUT.Message

Inputs:

Message

Level INFO



AddConnectionError - IlluminatorFatalError

Description: This will store the returned error message from the JCO API error in the Output XML of the transaction.

Incoming Links:

Type: Assign

From: RSAQ_REMOTE_USERGROUP_CATALOG.LastErrorMessage

To: AddConnectionError.Message

Inputs:

IlluminatorDocument
Message

Transaction.OutputXML



GetDetailFailed - TerminateTransaction

Description: This will set the JCO API error message as the error message for the transaction.

Incoming Links:

Type: Assign

From: RSAQ_REMOTE_USERGROUP_CATALOG.LastErrorMessage

To: GetDetailFailed.TerminationMessage

Inputs:

TerminationMessage
TerminateWithError

Transaction Execution Terminated Upon Request
true

Copyright

© Copyright 2006 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, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

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.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, 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 in several other countries all over the world. 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.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.