WebSphere Message Broker
iSeries Adapter Nodes

Version 1.0.7

December 1st 2011

Joerg Wende
jwende@de.ibm.com
IBM Software Group Germany

Property of IBM

First Edition, December 01st 2011

This edition applies to Version 1.0.1 of WebSphere Message Broker – iSeries Adapter Nodes and to all subsequent releases and modifications unless otherwise indicated in new editions.

Take Note!

Before using this report be sure to read the general information under "Notices".

This page was left blank intentionally.

Table of contents

Notices

The following paragraph does not apply in any country where such provisions are inconsistent with local law.

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Any reference to an IBM licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent program that does not infringe any of the intellectual property rights may be used instead of the IBM product.

Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, New York 10594, USA.

The information contained in this document has not be submitted to any formal IBM test and is distributed AS-IS. The use of the information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Trademarks and service marks

The following terms, used in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

- IBM
- WebSphere MQ
- WebSphere Message Broker

The following terms are trademarks of other companies:

Windows NT, Visual Studio,	Microsoft Corporation

Acknowledgments

Additional author:
Jochen Benke
IBM Software Group Services for WebSphere, Germany

Summary of Amendments

Date	Changes
December 01st 2011	Initial release
August 06th 2012	Minor updates

Preface

This SupportPac supplies WebSphere Message Broker plug-in nodes that allow connection to programs and data queues on IBM I (Power Systems including AS/400, iSeries, and System I).

The supplied version had been developed and tested on Linux 86 and Microsoft Windows XP .

POSSIBLE USES

This plug-in is designed for people who:

- need to implement integration scenarios between WebSphere Message Broker and iSeries programs (ILE RPG or Cobol) and iSeries data queues.

SKILL LEVEL REQUIRED

To use the plug-in, you need to be familiar with WebSphere Message Broker V7 and IBM I.

Bibliography

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp http://publib.boulder.ibm.com/infocenter/iadthelp/v6r0/index.jsp?topic=/com.ibm.iseries.xd.doc/html/javaapps.htm	WebSphere Message Broker V7 InfoCenter Developing Java Applications
http://www-03.ibm.com/systems/i/software/toolbox/	IBM Toolbox for Java
ISBN: 193118206X	Java for RPG Programmers Authors: Coulthard P., Farr G.

Overview

Need for iSeries Adapter Nodes

Many companies run their business on packaged or home-grown applications deployed on a System i Application system. Some of these applications don't provide service interfaces to integrate them with other applications so there is still the need for some kind of adapters to enable these applications for service orientation.

WebSphere Message Broker (WMB) provides a large set of functions that enables architects and developers to build interface logic and service bus functions and run them on the same infrastructure. This results in lower cost of ownership for the middleware infrastructure and reuse of development and administration skills.

This SupportPac delivers a set of WMB extension nodes which can be used to create integration scenarios between WMB message flows and iSeries applications without extensive coding.

Installing the Plug-in node

Plug-in contents

The supplied zip file should be unzipped in a temporary directory. The following files will be created:

<temp>\Iseries Adapter Nodes User Guide.pdf	This document
<temp>\runtime\jplugin <temp>\runtime\XSL	Runtime components to be placed on the server where IBM WebSphere Message Broker is installed
<temp>\toolkit\	Build time component for the WebSphere Message Broker Toolkit
<temp>\toolkit\samples\iSeriesAdaptersamplesPI.zip	Project interchange file with sample projects that can be imported into IBM WebSphere Message Broker Toolkit
<temp>\toolkit\samples\wmbsample.savf	Savefile containing an iSeries library containing the required objects for the sample project

Prerequisites

This plug-in provides a user defined node to be used with the IBM WebSphere Message Broker version 7 and above.

Required software

- IBM Message Broker V7 and above.
  - http://www-01.ibm.com/software/integration/wbimessagebroker/requirements/#BrokersV70
- IBM Toolbox for Java
  - http://www-03.ibm.com/systems/i/software/toolbox/

Supported Platforms

This plug-in has been developed and tested in a Microsoft Windows XP and Linux environment but it should run on any platform supported by IBM WebSphere Message Broker.

Installing the plug-in node on broker system

The node implementation ‘**jar**’ file should be installed by copying or moving the appropriate file to a directory that is listed in the LIL_PATH environment variable of the WebSphere Message Broker installation.

On Windows that could be <WMB-Installation-Root>\jplugin or any other directory that you added to the LIL_PATH yourself.

On Linux and AIX the recommendation is to put the '**jar**' file into /var/mqsi/extensions, but here – as well as on Windows – you can use any directory that you added to the LIL_PATH yourself.

The node implementation has a dependency to the IBM Toolbox for Java library(jt400.jar). This library is delivered as part of **IBM i Series Access**.
JTOpen is the open source version of the IBM Toolbox for Java licensed program.

- Go to: http://jt400.sourceforge.net/
- Download the latest version of JTopen (7.4)
- Extract the compressed file to a temporary folder
- Copy jt400.jar to <Broker Workpath>\shared-classes

You must stop and restart the broker to enable it to detect the existence of the new 'jar'.

The IseriesPcmlCall node uses XML style sheets to transform messages from and to PCML format. These style sheets needs to be placed in the *<Broker Workpath>\XSL* directory. For Windows XP this path is by default:
'C:\Documents and Settings\All Users\Application Data\IBM\MQSI\XSL'
This SupportPac contains sample style sheets that can be copied from the *<temp>\runtime\XSL* folder.

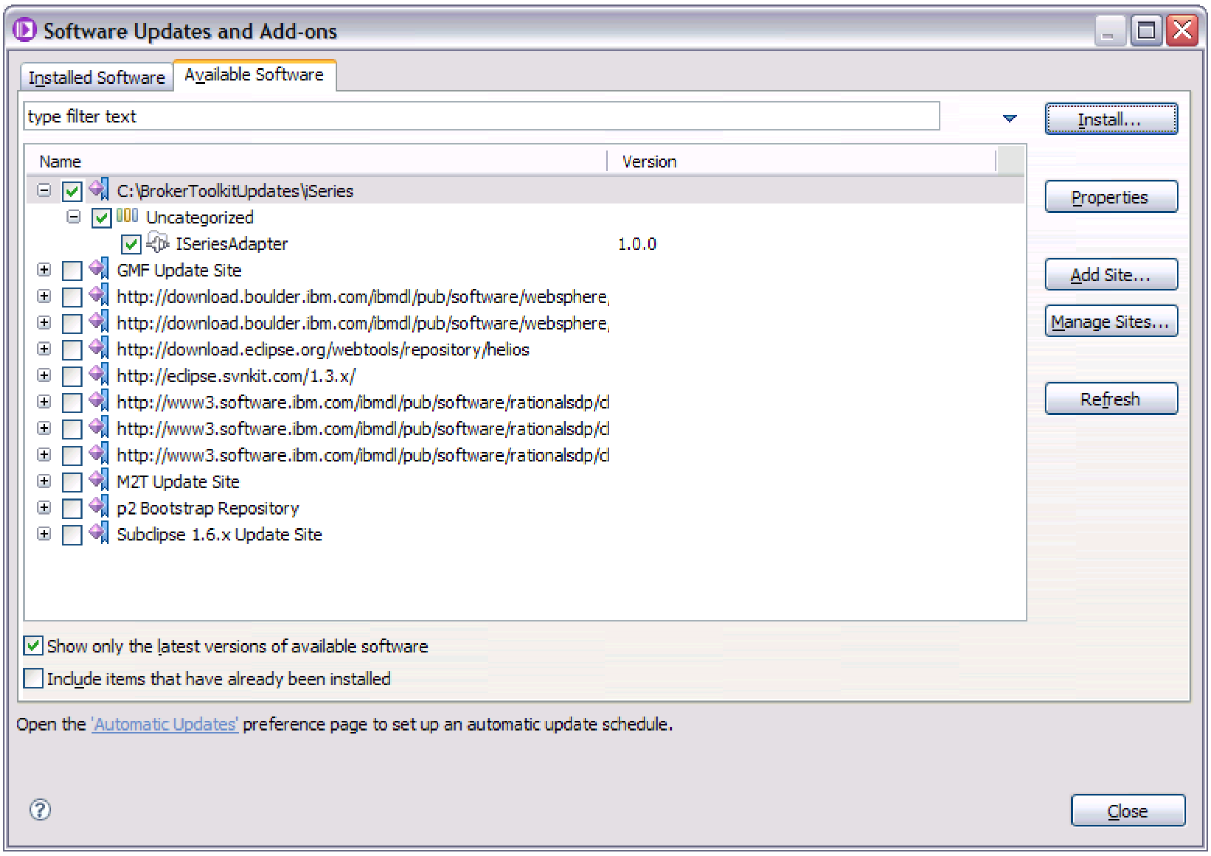
Note: The node implementation '**jar**' file and its dependencies must be installed on all broker systems where message flows containing thisplug-in node will be deployed.

Integrating the plug-in node into the WebSphere Message Broker Toolkit

The extension nodes for WebSphere Message Broker Toolkit come in two packages: as an Eclipse Feature and as a plug-in jar file.

To install a feature into the toolkit you need to:

- start the WebSphere Message Broker Toolkit
- go to **Help->Software Updates**
- switch to the **Available Software** tab
- click on **Add Site**
- click **Local** and select the toolkit root directory from location you have used to extract the SupportPac and click **OK**
  - *e.g.: <temp>\toolkit\plugins*
- select the **IseriesAdapter** feature and click on **Install**
- follow the installation wizard and restart the toolkit (if requested)



type filter text

Name	Version
⊟ ☑ C:\BrokerToolkitUpdates\iSeries	
⊟ ☑ Uncategorized	
☑ ISeriesAdapter	1.0.0
⊞ □ GMF Update Site	
⊞ □ http://download.boulder.ibm.com/ibmdl/pub/software/websphere/	
⊞ □ http://download.boulder.ibm.com/ibmdl/pub/software/websphere/	
⊞ □ http://download.eclipse.org/webtools/repository/helios	
⊞ □ http://eclipse.svnkit.com/1.3.x/	
⊞ □ http://www3.software.ibm.com/ibmdl/pub/software/rationalsdp/d	
⊞ □ http://www3.software.ibm.com/ibmdl/pub/software/rationalsdp/d	
⊞ □ http://www3.software.ibm.com/ibmdl/pub/software/rationalsdp/d	
⊞ □ M2T Update Site	
⊞ □ p2 Bootstrap Repository	
⊞ □ Subclipse 1.6.x Update Site	

☑ Show only the latest versions of available software

□ Include items that have already been installed

Open the 'Automatic Updates' preference page to set up an automatic update schedule.

Alternative installation by copying the plug-in .jar file:
Copy the .jar file that is contained in the Install/Toolkit subdirectory to the Plugins directory that is part of your WebSphere Message Broker Toolkit installation.
By default, this will be the directory: <Toolkit_RootInstallDirectory>/plugins on the Windows or Linux workstation where the WebSphere Message Broker Toolkit is installed.

Start the Toolkit.

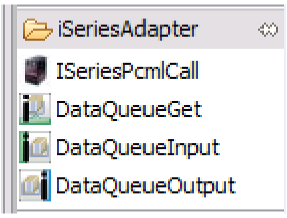
The iSeries Adapter Nodes should now be available for use as part of the Toolkit Node Palette.

Using the plug-in node

Description

This SupportPac contains four nodes:

- IseriesPcmlCall
- DataQueueGet
- DataQueueInput
- DataQueueOutput



iSeriesAdapter
ISeriesPcmlCall
DataQueueGet
DataQueueInput
DataQueueOutput

The nodes can be found in the palette of the message flow editor under the category iSeriesAdapter.

IseriesPcmlCall Node

The IseriesPcmlCall node can call iSeries ILE programs and modules written in RPG or Cobol
It makes use of PCML program call documents to describe input and output parameter structures.
The PCML file needs to be generated by the iSeries programmer and transferred to the WebSphere Message Broker Toolkit (replace the program and file names with real values):

- Generate the PCML file:
  - Compile the program and specify the PGMINFO option:
    *CRTBNDRPG PGM(LIBNAME/CUSTRPT) PGMINFO(*PCML) INFOSTMF('/home/custrpt.pcml')*
- Transfer the generated PCML file from the System i integrated file system(IFS) to your workstation. This could be done with the System i Operations Navigator for instance.
- Import the PCML file into your WebSphere Message Broker Toolkit.
- Use the provided XML style sheet (*transformPcmlToXsd.xsl*) to transform the PCML file into a XML schema.
  - Right click on the style sheet and select: **Run-As>XSL Transformation**
  - Select the PCML file as source file and specify a target file name (for instance: *custrpt.xsd*)
  - Right click on the generated schema and select: **Validate**
  - There should be no error.

Note:
The IseriesPcmlCall node supports two types of program calls (PCML and XPCML calls). Two different XML style sheets are provided to support the generation of XML schema files.

The generated XML schema can be used to create WMB message definitions by importing them into an XML enabled message set.

Note:
There are some data types not supported by PCML currently. See the restrictions at:
http://publib.boulder.ibm.com/infocenter/iadthelp/v7r0/index.jsp?topic=/com.ibm.etools.iseries.pgmgd.doc/c0925076180.htm
The generated PCML file will have additional comments if there are not supported data types.

Data Queue Nodes

The nodes can send and retrieve messages from iSeries data queues. They support keyed and non-keyed data queues. The DataQueueInput node can start a message flow.
The nodes work on BLOB messages but MRM CWF parser can be used to define data format.

For more details about iSeries data queues see the following links:

- http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp?topic=%2Frzaik%2Frzaikdqapitypuse.htm
- http://publib.boulder.ibm.com/infocenter/iadthelp/v7r5/index.jsp?topic=/com.ibm.etools.iseries.toolbox.doc/dtaq.htm

Connectivity Parameters

Connectivity parameters are externalized using a User Defined Configurable Service.
This entry could be created by:

- command line using mqsicreateconfigurableservice
  - http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r1m0/index.jsp?topic=/com.ibm.etools.mft.doc/an37200_.htm
  - Example command:
    - mqsicreateconfigurableservice MB7BROKER -c UserDefined -o myIseries -n " \"iSeriesAddress\", \"iSeriesPassword\", \"iSeriesUser\", \"maxConnections\", \"maxUseTimeout\", \"minConnections\" " -v " \"ADDRESS\", \"PASSWORD\", \"USER\", \"5\", \"-1\", \"1\" "
- WebSphere Message Broker Explorer
  - right click on **Configurable Services** and select **New->Configarable Service**
  - scroll down in the **Type** drop down box and select **UserDefined**
- Drag and drop a XML file on a running broker in WebSphere Message Broker Explorer
  - Example XML file:

    -
      ```
      <?xml version="1.0" encoding="UTF-8"?><configurableservice
      csName="myISeries" csType="UserDefined" iSeriesAddress="ADDRESS"
      iSeriesPassword="PASSWORD" iSeriesUser="USER" maxConnections="5"
      maxUseTimeout="-1" minConnections="1"/>
      ```

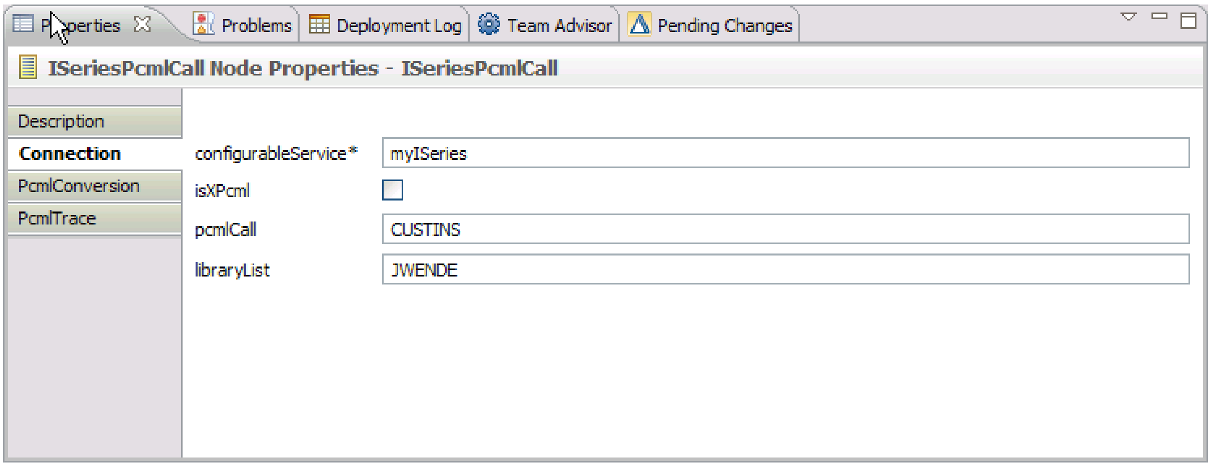
Name	Value
iSeriesAddress	Network address of the iSeries server
iSeriesUser	User name
iSeriesPassword	User password
minConnections	Minimum number of pre-established connections for the connection pool
maxConnections	Maximum number of connections
maxUseTimeout	Timeout for returning connections to the pool

Plug-in node terminals

Terminal	Description
Failure	The output terminal which the message is routed to if failure is detected during the processing.
Out	The output terminal that outputs the generated message.
In	The input terminal which accepts a message (not available for the DataQueueInput Node)

IseriesPcmlCall Node properties

These properties are displayed in the Properties view when you select one of the iSeries Adapter Node entries in the Message Flow Editor pane.



ISeriesPcmlCall Node Properties - ISeriesPcmlCall

configurableService* myISeries

isXPcml □

pcmlCall CUSTINS

libraryList JWENDE

The tab with the title “Description” is the standard tab, that allows you to change the node's name in the flow , give a short and a long description of it's purpose in the message flow.

Tab	Name	Usage
Connection	configurableService	Name of a Userdefined Configurable Service
	isXPcml	Use XPCML syntax instead of PCML syntax
	pcmlCall	Name of the module or program to be called
	libraryList	Comma separated list of additional libraries required to run the program
PcmlConversion	toPCML	XML stylesheet used to convert the inbound message to PCML format

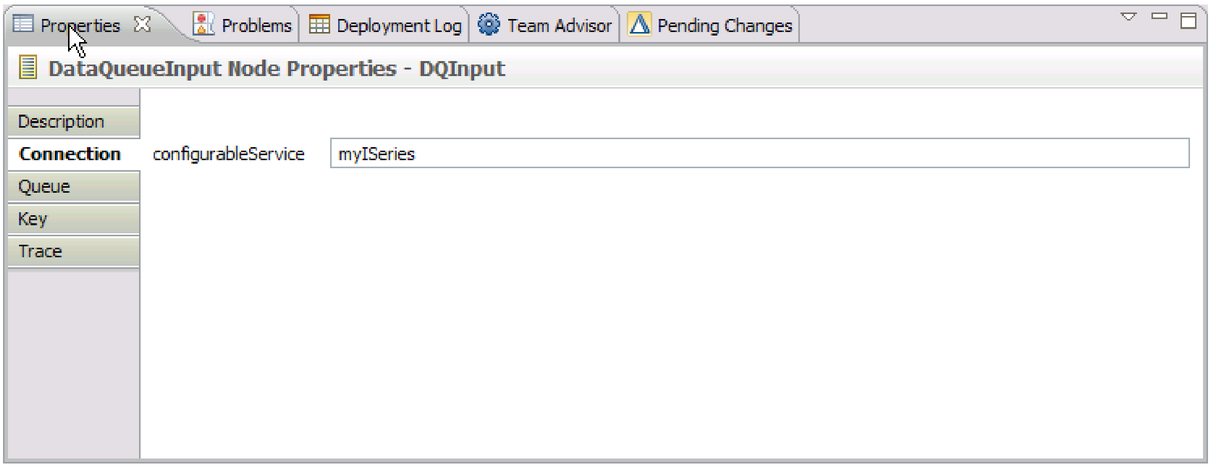
Tab	Name	Usage
	fromPCML	XML stylesheet used to convert the outbound PCML message
PcmlTrace	enable	IF selected the node will write trace data
	traceDirectory	location of the trace files

The properties for the ISeriesPcmlCall node can be changed using LocalEnvironment overrides:

ISeriesPcmlCall
- *pcmlCall*
- *libraryList*
- *toPcml*
- *toXPcml*
- *fromXPcml*

DataQueueInput Node and DataQueueGet properties

These properties are displayed in the Properties view when you select one of the iSeries Adapter Node entries in the Message Flow Editor pane.



DataQueueInput Node Properties - DQInput

configurableService: myISeries

The tab with the title “Description” is the standard tab, that allows you to change the node's name in the flow , give a short and a long description of it's purpose in the message flow.

Tab	Name	Usage
Connection	configurableService	Name of a Userdefined Configurable Service
Queue	queueName	name of the dataqueue object
	library	library object where the

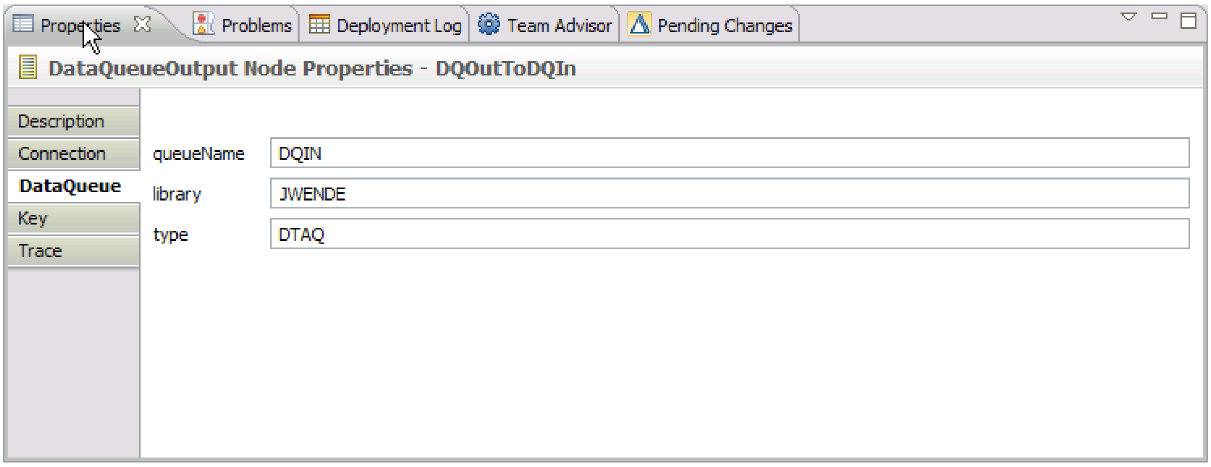
Tab	Name	Usage
		dataqueue is stored in
	waitInterval	time to wait for a message
	type	DTAQ (fixed value)
Key	keyed	IF selected the node expects a keyed dataqueue
	key	key value to search a message for
	searchType	operation to be used for comparing the key during the search operation
Trace	enabled	IF selected the node will write trace data
	directory	location of the trace files

The properties for the DataQueueGet node can be changed using LocalEnvironment overrides:

```
DataQueue
    queueName
    library
    type
    waitInterval
    keyed
    key
    searchType
```

DataQueue Ouput Node properties

These properties are displayed in the Properties view when you select one of the iSeries Adapter Node entries in the Message Flow Editor pane.



DataQueueOutput Node Properties - DQOutToDQIn

queueName: DQIN
library: JWENDE
type: DTAQ

The tab with the title “Description” is the standard tab, that allows you to change the node's name in the flow , give a short and a long description of it's purpose in the message flow.

Tab	Name	Usage
Connection	configurableService	Name of a Userdefined Configurable Service
Queue	queueName	name of the dataqueue object
	library	library object where the dataqueue is stored in
	type	DTAQ (fixed value)
Key	keyed	IF selected the node expects a keyed dataqueue
	key	key value to search a message for
	searchType	operation to be used for comparing the key during the search operation
Trace	enable	IF selected the node will write trace data
	directory	location of the trace files

The properties for the DataQueueOutput node can be changed using LocalEnvironment overrides:

```
DataQueue
    queueName
    library
    type
    keyed
    key
```

Development using the iSeriesPCMLCall Node

The following sections describes an example development flow fot the integration of a fictive iSeries application.

Generate and Import the Application Interface Structure

To support the definition of message structures and the development of mappings it is necessary to generate the PCML file and transfer it to the developer workstation.

- Generate the PCML file:
  - Compile the program and specify the PGMINFO option:
    *CRTBNDRPG PGM(*CURLIB/CUSTRPT) PGMINFO(*PCML) INFOSTMF('/home/custrpt.pcml')*
- Transfer the generated PCML file from the System i integrated file system(IFS) to your workstation. This could be done with the System i Operations Navigator for instance.
- The PCML file should like similar to this:

```
<pcml version="4.0">
   <!-- RPG program: CUSTRPT  -->
   <!-- created: 2011-03-25-08.14.53 -->
   <!-- source: xyzLib/QRPGLESRC(CUSTRPT) -->
   <!-- 5 -->
   <program name="CUSTRPT"
path="/QSYS.LIB/xyzLib.LIB/CUSTRPT.PGM">
      <data name="ID" type="int" length="4" precision="32"
usage="inputoutput" />
      <data name="BALANCE" type="packed" length="7"
precision="2" usage="inputoutput" />
   </program>
</pcml>
```

- Import the PCML file into your WebSphere Message Broker Toolkit.
- Use the provided XML style sheet (*transformPcmlToXSD.xsl)* to transform the PCML file into a XML schema.
  - Right click on the style sheet and select: **Run-As>XSL Transformation**
  - Select the PCML file as source file and specify a target file name (for instance: *custrpt.xsd*)
  - Right click on the generated schema and select: **Validate**
  - There should be no error.

Create the Message Definitions

A Message Set project is used by the toolkit to hold message definitions for a message flow. You will now create a Message Set project and define message formats.

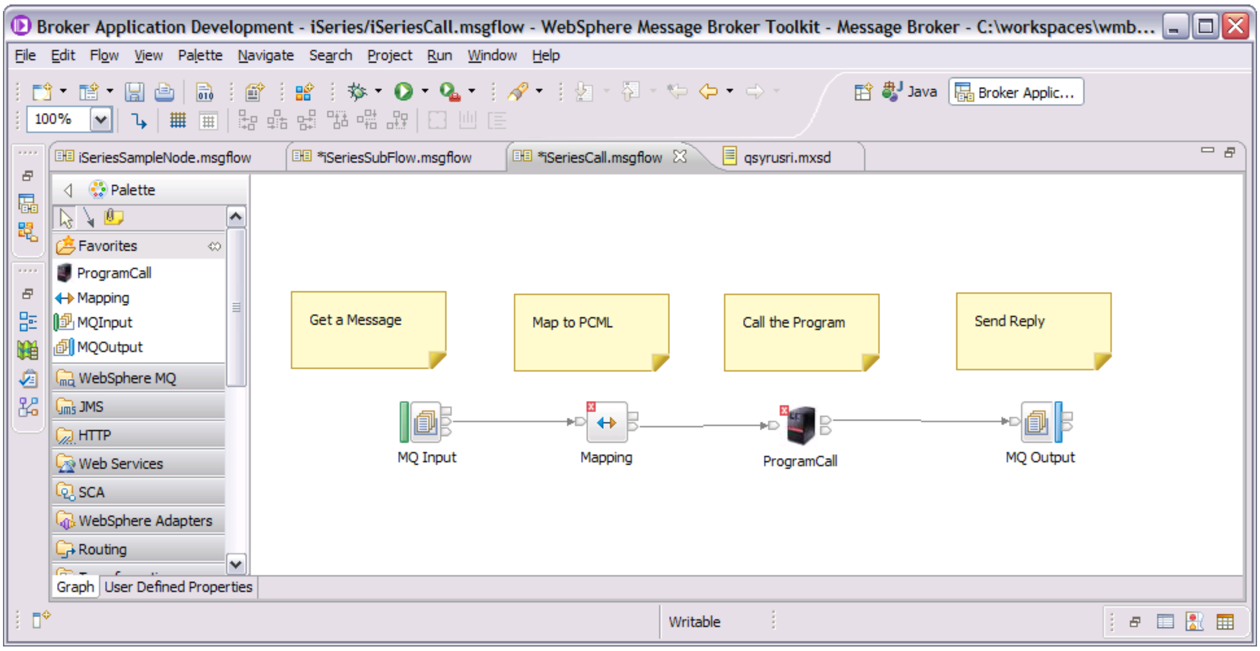
- Select: **File->New->Message Set**
  - Specify a name for the message set (for instance: *custrptMsgDef*)
  - Click: **Finish**

- Right click on the XML schema file and select: **New->Message Definition From->XML Schema File**
- Select your message set and click: **Next**
- Select *pcml* as the root element for your message definition
- Click: **Finish**

Create and Configure the Message Flow

Now you are ready to start message flow development:

- Select: **File->New->Message Flow Project**
    1. Specify a name for the message flow project (for instance: *iSeriesMsgFlows*)
    2. Click**: Next**
    3. Select *custrptMsgDef* as a dependency for the message flow project
    4. Click: **Finish**
- Click on the message flow project and select: File->New->Message Flow
    6. Specify a name for the message flow project (for instance: *custrptFlow*)
    7. Click**: Finish**
- Add the following nodes to construct the flow:
    9. MQ Input Node
        10. Specify
    11. MQ Output Node
    12. Mapping Node
    13. ProgramCall Node
- Connect the nodes:



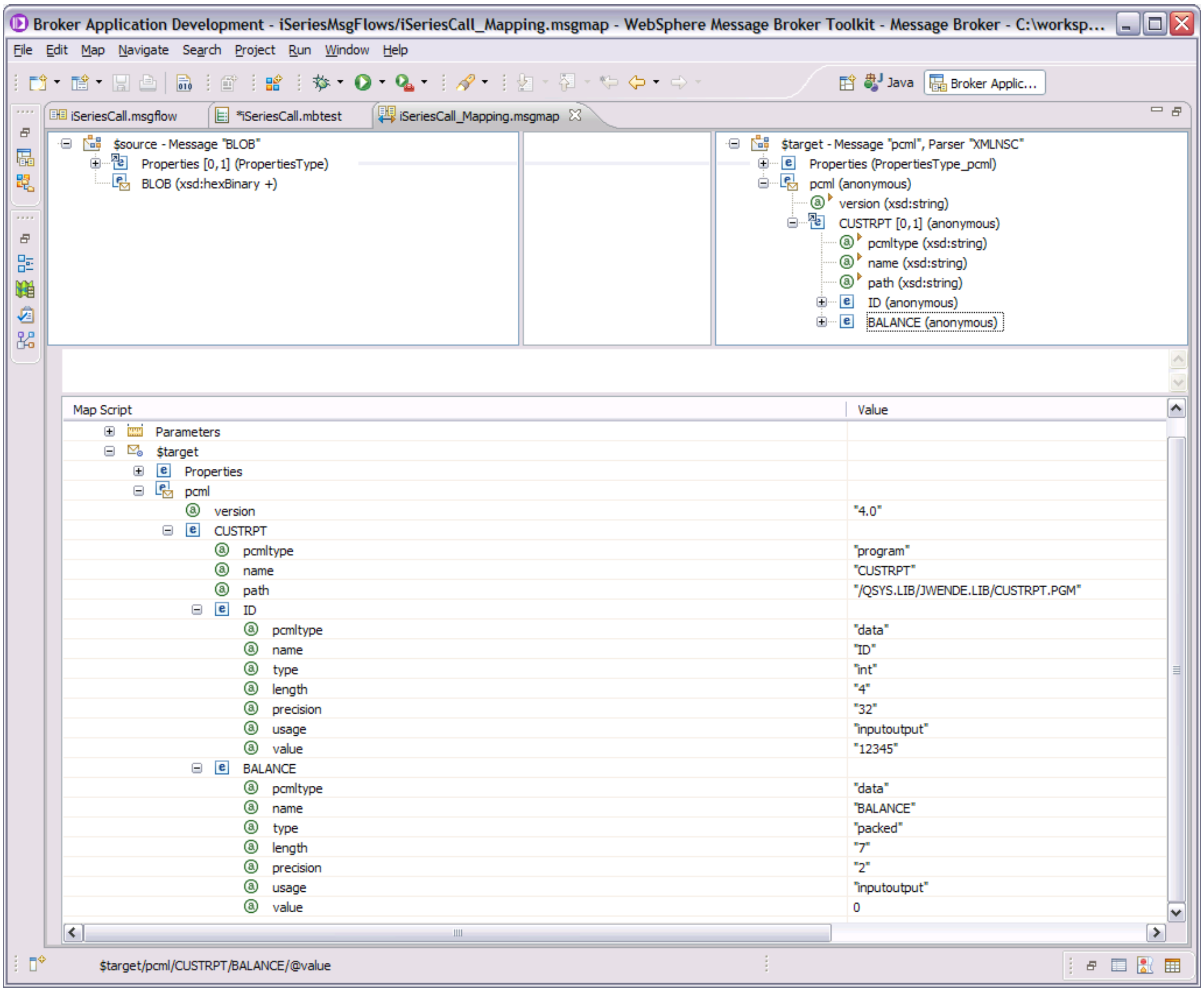
In the next required step you need to configure the processing

- Configure the MQ Input Node to read a message from *iSeries.IN*
- Configure the MQ Output Node to write a message to *iSeries.OUT*
- Configure the ProgramCall Node:

- configurableServiceName: *myISeries*
- the configurable service will be created in the deployment section
- program: *custrpt*
- libraryList: <name of the library>
- multiple libraries could specified as a comma separated list

○ Double click on the mapping node to open the map editor

- Select '*BLOB*' as your source message
- Select '*pcml*' as your target message
- Click: **OK**
- Create a target mapping like in the following picture (Note: all fields of the target structure need to be created – this can be done by clicking on the top level field and select: **Insert Children**) :


Broker Application Development - iSeriesMsgFlows/iSeriesCall_Mapping.msgmap - WebSphere Message Broker Toolkit - Message Broker - C:\worksp...
File Edit Map Navigate Search Project Run Window Help
Java
Broker Applic...
iSeriesCall.msgflow
*iSeriesCall.mbtest
iSeriesCall_Mapping.msgmap

```
$source - Message "BLOB"
    Properties [0,1] (PropertiesType)
    BLOB (xsd:hexBinary +)
```

```
$target - Message "pcml", Parser "XMLNSC"
    Properties (PropertiesType_pcml)
    pcml (anonymous)
        version (xsd:string)
        CUSTRPT [0,1] (anonymous)
            pcmltype (xsd:string)
            name (xsd:string)
            path (xsd:string)
            ID (anonymous)
            BALANCE (anonymous)
```

Map Script	Value
Parameters	
$target	
Properties	
pcml	
version	"4.0"
CUSTRPT	
pcmltype	"program"
name	"CUSTRPT"
path	"/QSYS.LIB/JWENDE.LIB/CUSTRPT.PGM"
ID	
pcmltype	"data"
name	"ID"
type	"int"
length	"4"
precision	"32"
usage	"inputoutput"
value	"12345"
BALANCE	
pcmltype	"data"
name	"BALANCE"
type	"packed"
length	"7"
precision	"2"
usage	"inputoutput"
value	0

$target/pcml/CUSTRPT/BALANCE/@value

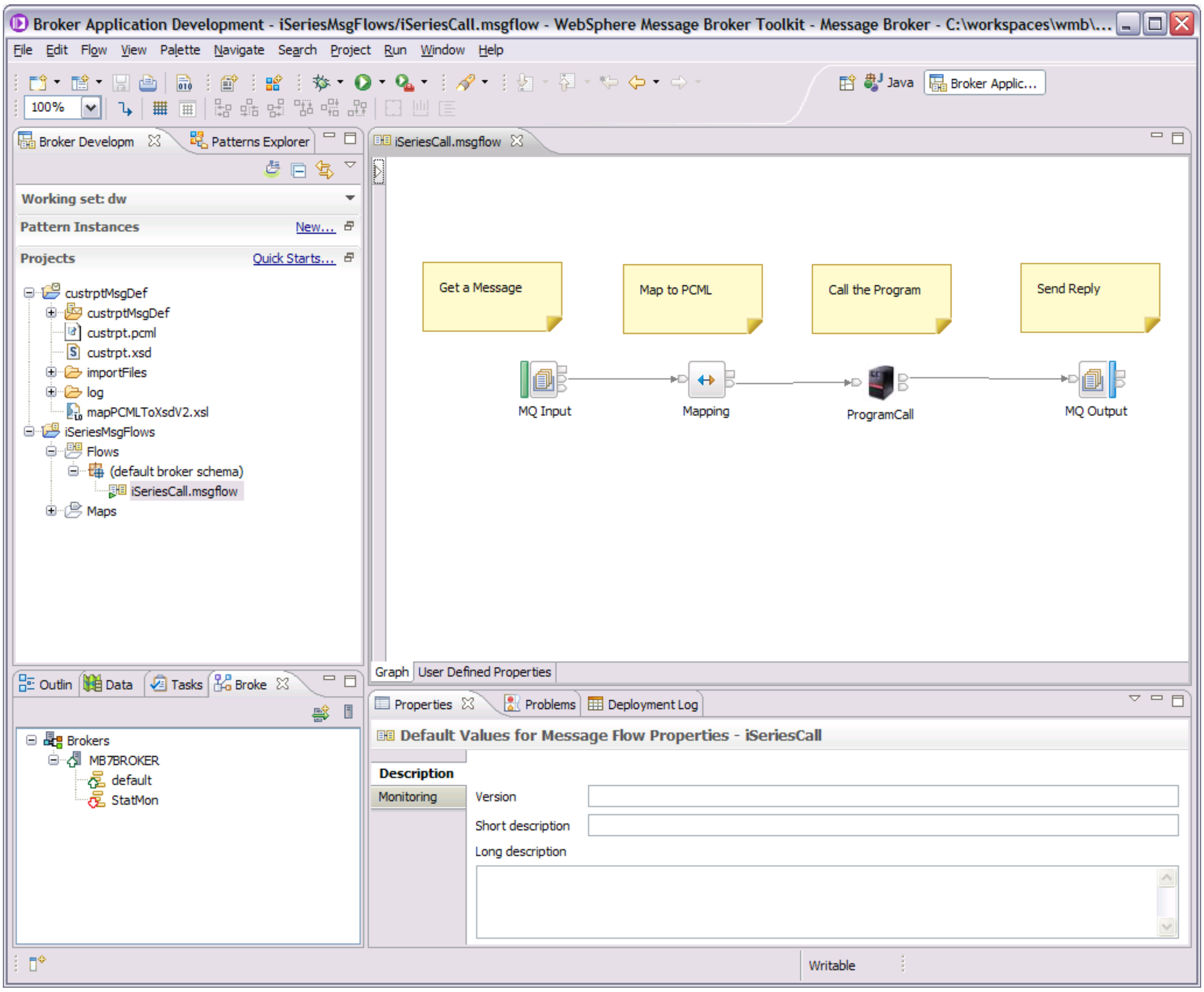
Save and close the mapping and save the flow. There should be no error marker!

Run the Message Flow

There are some steps required to run the configured flow. It is assumed that the deployment will be done on a local default configuration.

- Start the WebSphere Message Broker Explorer
- Create the two queues
  - iSeries.IN
  - iSeries.OUT
- Expand the default broker tree (MB7BROKER) and right click on: **Configurable Services**
- Select: New->**Configurable Service**
- Enter a name, for instance: myISeries (this name has to match with the value entered for the ProgramCall Node)
- Select '**UserDefined'** for the configurable service type
- Create three properties:
  - iSeriesAddress
    - this property holds the IP address of the System i
  - iSeriesPassword
    - this property holds the password for the System i
  - iSeriesUser
    - this property holds the name of the user for the System i
- Click: Finish

Switch back to the WebSphere Message Broker Toolkit.


Broker Application Development - iSeriesMsgFlows/iSeriesCall.msgflow - WebSphere Message Broker Toolkit - Message Broker - C:\workspaces\wmb\...

Working set: dw

Default Values for Message Flow Properties - iSeriesCall

Now the deployed flow could be tested. This can be done directly from the WebSphere Message Broker Toolkit.

- Right click on the message flow and select: **Test Message Flow**
- Enter any value into the input message field
- Click: **Send Message**
- After a short time the test client should show a response message:

Using the samples

Requirements

To run the samples you need a connection to an iSeries server. Part of this SupportPac is an iSeries library containing sample programs and iSeries data queues.

Installation Steps

iSeries

- Download the SupportPac and unzip it to a temporary directory. The file is named IAM7.zip.
- On iSeries, create a library containing an empty save files to hold the uploaded file
  - To create the library and save file use the commands:
    - CRTLIB LIB(WMBSAVEF)
    - CRTSAVF WMBSAVEF/WMBSAMPLE
- Start an ftp session to your iSeries and upload the save file with the commands
  - ftp (your_iSeries)
  - cd WMBSAVF
  - bin
  - put wmbsample.savf WMBSAVEF/WMBSAMPLE
- Sign on to your iSeries machine
- Enter the RSTLIB command, specifying the install device as *SAVF and naming the save file containing the sample library.
  - RSTLIB LIB(WMBSAMP) DEV(*SAVF) SAVF(WMBSAVEF/WMBSAMP)

Broker Toolkit

Start the WebSphere Message Broker Toolkit and import the provided sample project interchange file.

Broker Runtime

- Create a UserDefined configurable service for your runtime environment:
  - mqsicreateconfigurableservice MB7BROKER -c UserDefined -o myIseries -n " \"iSeriesAddress\", \"iSeriesPassword\", \"iSeriesUser\", \"maxConnections\", \"maxUseTimeout\", \"minConnections\" " -v " \"ADDRESS\", \"PASSWORD\", \"USER\", \"5\", \"-1\", \"1\" "
- Make sure that you followed the installation steps for the runtime plugin nodes
- Create the following queues:
  - def ql('TEST.PCML.IN')
  - def ql('TEST.PCML.OUT')

- def ql('TEST.XPCML.OUT')
- def ql('TEST.DQ.GET.IN')
- def ql('TEST.DQ.GET.NOMESSAGE')
- def ql('TEST.DQ.GET.OUT')
- def ql('TEST.DQ.IN.OUT')
- def ql('TEST.DQ.OUT.IN')
- def ql('TEST.KEY.DQ.GET.IN')
- def ql('TEST.KEY.DQ.GET.OUT')
- def ql('TEST.KEY.DQ.GET.NOMESSAGE')
- def ql('TEST.KEY.DQ.IN.OUT')
- def ql('TEST.KEY.DQ.OUT.IN')

Understand the samples

Please review the content of the sample flows in the IseriesAdapterTest project to understand the usage of extension nodes provided by this SupportPac.

--- End of Document