

IA9R: WebSphere Message Broker – Inbound POP3 Email Support

December, 2006

Wang Chao

wchao@cn.ibm.com

SWG Lab Services, CSDL

IBM China

Property of IBM

Abstract

This SupportPac provides a sample of how to poll a POP3 server periodically and retrieve the email message bodies and attachments as input messages for WebSphere Message Broker V6 message flows.

When the SupportPac pulls an email message, it serializes the BodyPart into a MQ message. If ProcessAttachments is specified, other Parts are also serialized as individual MQ messages. These Parts' bodies are then parsed with specified WMB parsers and populated to downstream nodes. Other information of the part, such as subjects, senders, receivers, is passed along using LocalEnvironment.

Please note that due to the limitation of POP3 protocol, there will be no once-and-only-once guarantee.

Prerequisites

The following are prerequisites of this SupportPac:

- WebSphere Message Broker Runtime v6.0.0.0 or above
- WebSphere Message Broker Toolkit v6.0.0.0 or above
- JavaMail v1.4 or above
- JavaBeans Activation Framework (JAF) v1.1 or above

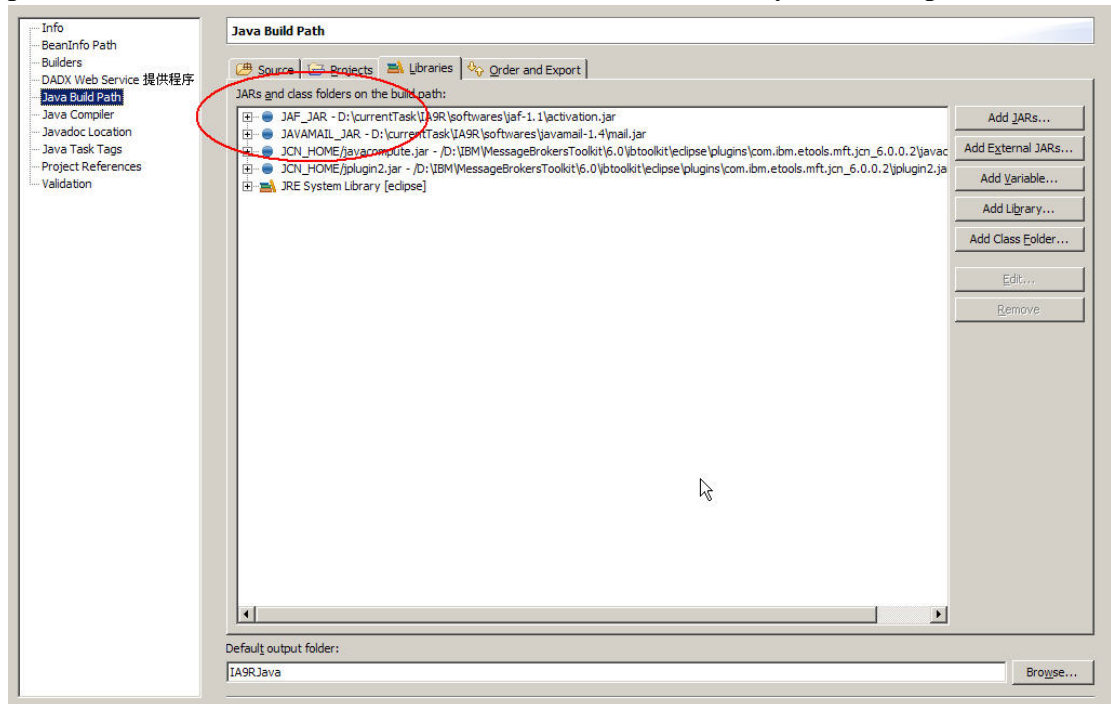
JavaMail and JAF can be obtained from Sun's web site (<http://java.sun.com/products/javamail/> and <http://java.sun.com/products/javabeans/jaf/downloads/index.html>) or other appropriate sources.

Installation

To install this SupportPac into your Message Broker Toolkit workspace:

1. Download IA9R.zip.
2. Unzip to any location.
3. Import the ia9r_PI.zip as Project Interchange into your workspace.
4. Add mail.jar (from JavaMail 1.4) and activation.jar (from JAF 1.1) into IA9RJava project's Java Build Path. You may want to do this by configuring the two

predefined variables, JAF_JAR and JAVAMAIL_JAR with your actual paths:



Development/Customization

After you import the SupportPac into your workspace, you will see two new projects:

- EmailPOP3 which is a message flow project
- EmailPOP3Java which is a Java project referenced by EmailPOP3

There are two flows in EmailPOP3:

Drag and drop the flow named POP3Input under the “com.ibm.cn.wchao” schema onto your message flow canvas. The sub-flow will appear as a node.

See the flow named TestEmailPOP3 under “sample” schema as an example:



Right click the POP3Input node, click “Properties”, and fill in the following information.

Property	Property	Purpose	Default
----------	----------	---------	---------

Category			Value
Email	POP3Host	Hostname or IP of the POP3 server	
	POP3User	Valid user name of the POP3 account	
	POP3Password	Valid password for the POP3 account	
	POP3Folder	The POP3 folder to be polled	INBOX
Basic	CheckMailInterval	Interval by seconds that the POP3 account is polled	30
	TempStorageQueue	The SupportPac places newly arrived emails into a temporary WebSphere MQ queue, before it is populated to subsequent nodes. You will need to create this queue on the Broker's queue manager and fill in the queue name here. If there are multiple POP3Input nodes running in the same broker, you may want to separate these queues.	Q.TEMP
	NodeID	To allow multiple POP3Input nodes running in the same broker, you will need to give each of them a unique name by overriding the default value of this parameter.	POP3Input
Output	ParserDomain	Parser domain of the parser for the incoming messages such as MRM, XMLNS	
	MessageType	If MRM parser is used to interpret the incoming messages, specify the Message Type	
	MessageFormat	If MRM parser is used to interpret the incoming messages, specify the Message Format	
	MessageSet	If MRM parser is used to interpret the incoming messages, specify the Message Set	
	ProcessAttachments	Check if you want the attachments are processed as input message	Unchecked
	ProcessBody	Check if you want the email body is processed as input message	Checked

The most important parameters that will affect your application flow design are those in Output group.

Message Parsing

POP3Input will use the parser, message type, message set and message format that you specify in this group to transform the email body or attachment into the output message tree.

For example, if you choose to use XML parser:

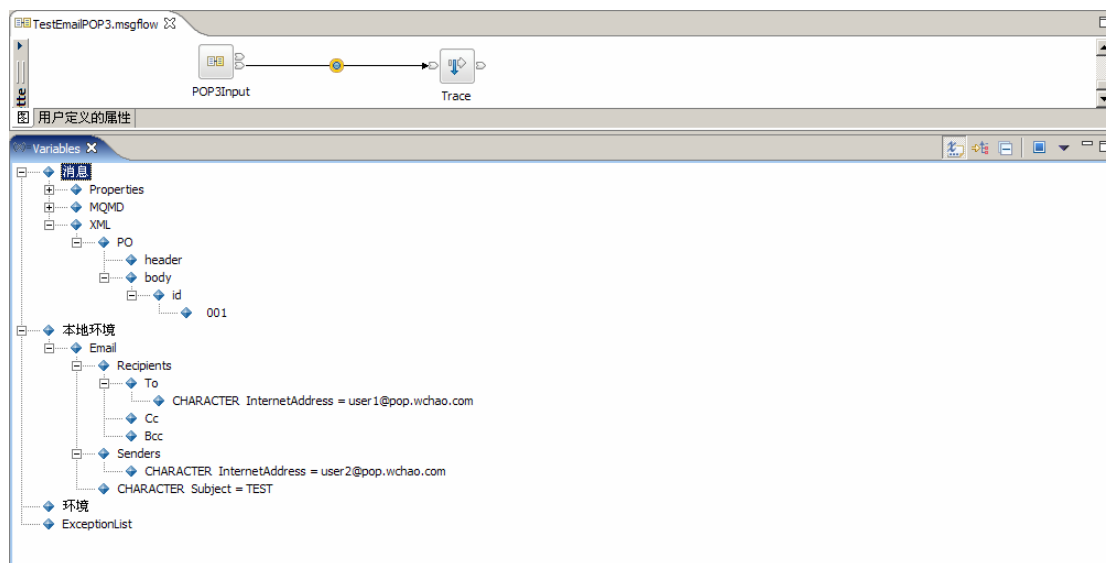
The screenshot shows a configuration window with a tree view on the left containing 'Email', 'Output', 'Basic', and '描述'. The 'Output' tab is active. The 'ParserDomain' field contains 'XML'. The 'MessageFormat', 'MessageSet', and 'MessageType' fields are empty. The 'ProcessAttachments' checkbox is unchecked, and the 'ProcessBody' checkbox is checked. At the bottom right, there are three buttons: 'OK', 'Cancel', and '应用(A)'.

And for the following email body:

```
From - Wed Dec 27 16:09:25 2006
X-Mozilla-Status: 0001
X-Mozilla-Status2: 00800000
Message-ID: <45922A34.2000403@pop.wchao.com>
Date: Wed, 27 Dec 2006 16:09:24 +0800
From: User2 <user2@pop.wchao.com>
User-Agent: Thunderbird 1.5.0.8 (Windows/20061025)
MIME-Version: 1.0
To: user1 <user1@pop.wchao.com>
Subject: TEST
Content-Type: text/plain; charset=GB2312
Content-Transfer-Encoding: 7bit

<P0><header></header><body><id>001</id></body></P0>
```

The output of POP3Input would be:



Attachments

If you choose to process attachments, each attachment will be populated as a message, with the same LocalEnvironment.Email tree structure and the additional Filename field.

For example, for the following email with two attachments:

From - Wed Dec 27 16:56:31 2006
X-Mozilla-Status: 0001
X-Mozilla-Status2: 00800000
Message-ID: <4592353E.3020100@pop.wchao.com>
Date: Wed, 27 Dec 2006 16:56:30 +0800
From: User2 <user2@pop.wchao.com>
User-Agent: Thunderbird 1.5.0.8 (Windows/20061025)
MIME-Version: 1.0
To: user1 <user1@pop.wchao.com>
Subject: TEST
Content-Type: multipart/mixed;
boundary="-----060505090107050800090500"

This is a multi-part message in MIME format.

-----060505090107050800090500
Content-Type: text/plain; charset=GB2312
Content-Transfer-Encoding: 7bit

Hello World!

-----060505090107050800090500
Content-Type: text/plain;
name="attachment2.txt"
Content-Transfer-Encoding: base64
Content-Disposition: inline;
filename="attachment2.txt"

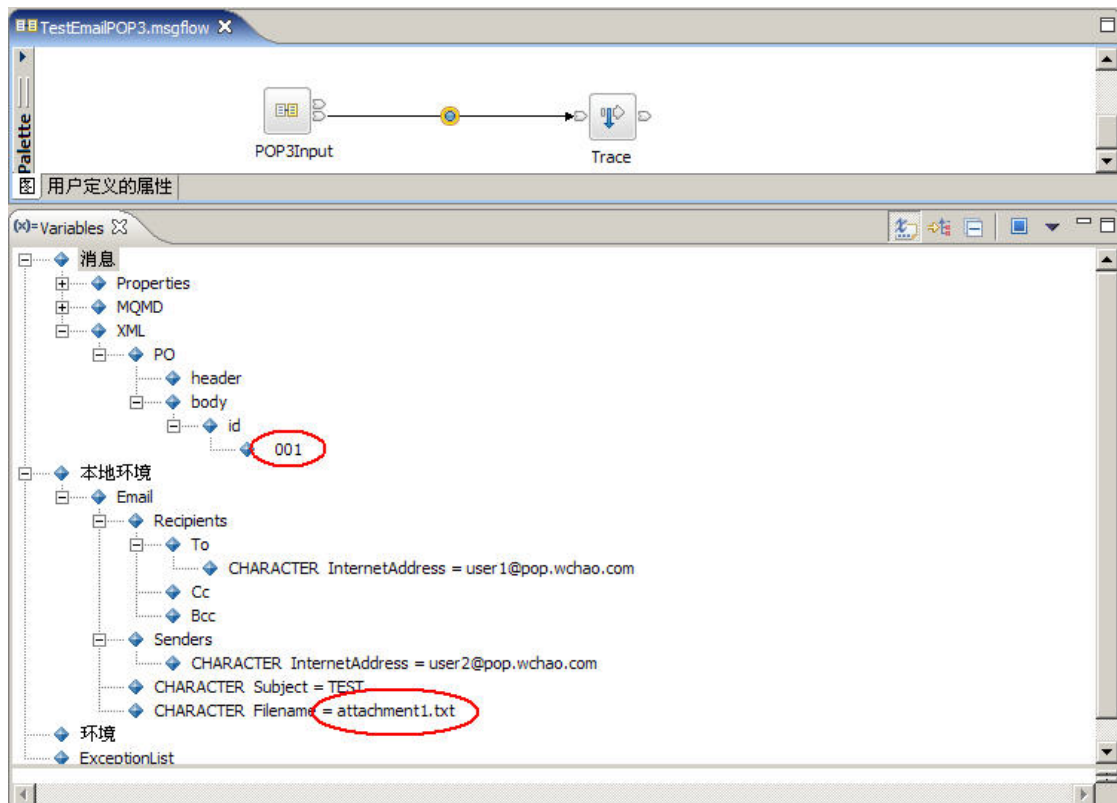
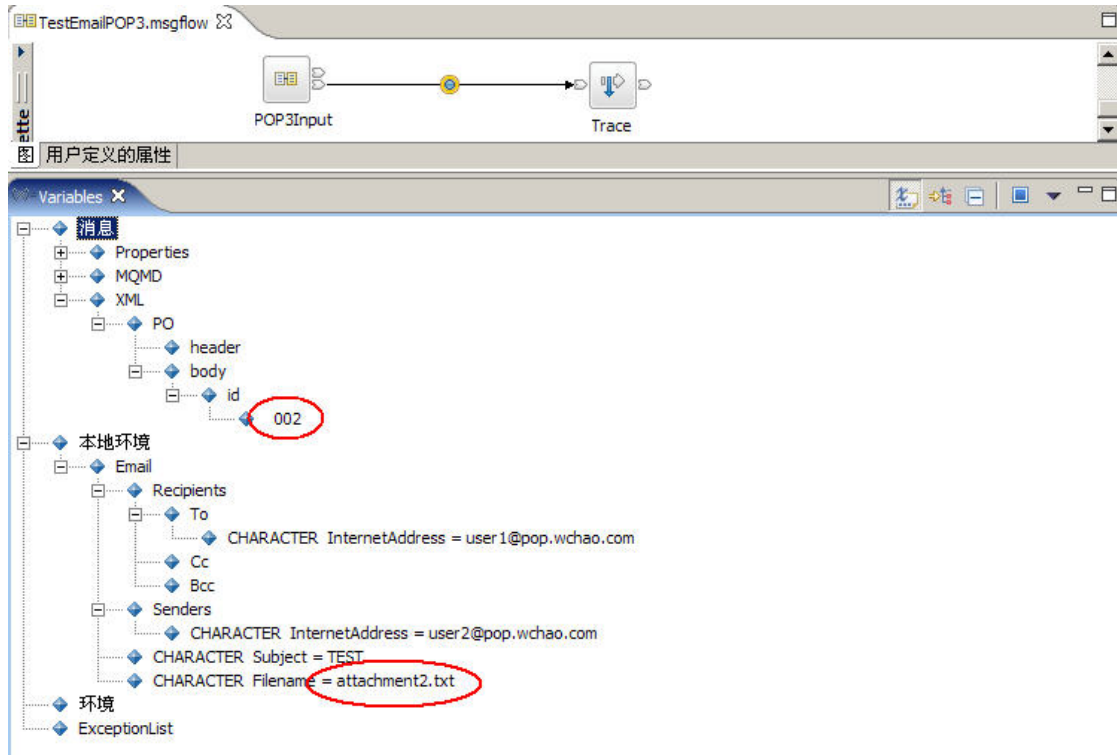
PFBPPjxoZWFkZXI+PC9oZWFkZXI+PGJvZGk+PG1kPjAwMjwvaWQ+PC9ib2R5PjwvUE8+

-----060505090107050800090500
Content-Type: text/plain;
name="attachment1.txt"
Content-Transfer-Encoding: base64
Content-Disposition: inline;
filename="attachment1.txt"

PFBPPjxoZWFkZXI+PC9oZWFkZXI+PGJvZGk+PG1kPjAwMTwvaWQ+PC9ib2R5PjwvUE8+

-----060505090107050800090500--

The populated messages will be like:



Note that there is no specific order for the attachments to be populated.

Deployment

To deploy this SupportPac into your Message Broker runtime:

1. Copy the following files into the shared-classes path under the broker's workpath.
 - mail.jar (from JavaMail)
 - activation.jar (from JAF)

Message Broker's work path can be determined by issuing the following command:

```
$> echo ${MQSI_WORKPATH}      (UNIX/Linux)
echo %MQSI_WORKPATH%         (Windows)
```

Typically this will be “/var/mqsi/shared-classes” on UNIX/Linux, or “C:\Documents and Settings\All Users\Application Data\IBM\MQSI\shared-classes\” on Windows

2. Deploy the message flows that incorporate this SupportPac and other resources using Toolkit or mqsideploy command.

Uninstallation

To uninstall the SupportPac from the Message Broker runtime:

1. remove your flows that use this SupportPac from the Broker
2. delete mail.jar and activation.jar file which you deployed in previous step