MQSeries Integrator – Sendmail Plug-In Version 1.5

5th November 2001

Arjan van Vught
e-business Hosting Services
Watsonweg 2
1423 ND Uithoorn
The Netherlands

avv@nl.ibm.com

Take Note!

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

Sixth Edition, November 2001

This edition applies to Version 1.5 of *MQSeries Integrator – Sendmail Plug-In* and to all subsequent releases and modifications unless otherwise indicated in new editions.

Table of Contents

MQSeries Integrator – Sendmail Plug-In i
Table of Contents iii
Notices v
Acknowledgments vi
Summary of Amendments vii
Preface viii
Possible uses viii
Bibliography ix
Chapter 1. Installing the plug-in node 1
SupportPac contents 1
Prerequisites 3
Supported Platforms 3
Installing the plug-in node on broker system 3
Integrating the plug-in node into the Windows Control Center 3
Defining the node to the configuration repository 4
Updating the node to the configuration repository 4
Chapter 2. Using the plug-in node 6
Description 6
Plug-in node terminals 6
Plug-in node properties 6
Chapter 3. Compiling the plug-in node 8
Windows NT (smtp version) 8
AIX 8
Smtp version 8
Sendmail version 8
Sun Solaris 9
Smtp version 9
Sendmail version 9

Linux (on Intel) 10
Smtp version 10
Sendmail version 10
HP-UX 11
Smtp version 11
Sendmail version 11
Chapter 4. Example using the plug-in node 13
Simple flow sending e-mail 13
Send FAX by e-mail 13
Chapter 5. Mapping fields to e-mail message header 15
Chapter 6. The XML message 16
Input message 16
Output error message 16
Error strings SMTP (TCP/IP) 17
Error strings SMTP (SMTP server) 17
Error strings sendmail 17
Chapter 7. Sendmail support implementation 18

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:

- AIX
- IBM
- MQSeries
- MQSeries Integrator
- MQSI

The following terms are trademarks of other companies:

- Windows NT Microsoft Corporation
- Sun Solaris Sun Corporation
- HP-UX Hewlett-Packard Company
- Linux Developed under the GNU General Public License

Acknowledgments

The author would like to acknowledge the help that was received from a number of individuals. First, Peter Lambros and Malcolm Ayres from the IBM Hursley Laboratories provided invaluable and tremendously useful information many times. Second, Paul Smit from IBM Global Services was very helpful at numerous times, for his experience on both MQSeries Integrator and the C-language development environment. Third, Andy Stanford-Clark from IBM Hursley MQSeries Technology was very helpful at numerous times for his input on MQSeries Integrator V2.0.2. Fourth, Gary Willoughby from Software Group Services at Hursley Park for his Sun Solaris support. Fifth, Andrew Ford from IBM Hursley Laboratories for his Sun Solaris support. Finally, the author wishes to thank the many people who also helped but who the author has unintentionally omitted from his brief mention.

Summary of Amendments

Date	Changes
6 March 2001	Initial release
17 May 2001	▪ New properties added : • Basic • Bcc • Priority • Set Priority • SMTP • Port number • Full name of the sender • Name of the 'from' person ▪ Bug fixes: • Sending large messages • Empty body (with Outlook) ▪ Updated for MQSeries Integrator V2.0.2
13 July 2001	▪ Enhanced/improved SMTP return messages, e.g. the messages returned by the SMTP server ▪ Additional tags for the "to" and "from" fields in the input message so they aren't reserved words, e.g. MailTo and MailFrom ▪ Enhanced the Body tag with Line tags, every <Line></Line> tag is a new line in the mail body ▪ Bug fixes: • MQMD.MsgId not set
23 July 2001	▪ Sun Solaris support
26 September 2001	▪ Linux (on Intel) support added ▪ Makefile included for AIX, Sun Solaris and Linux (on Intel)
19 October 2001	▪ HP-UX support added

Preface

This SupportPac supplies a MQSeries Integrator Version 2 plug-in node that constructs an e-mail message from a XML input message. Included is a simple SMTP client. On AIX, Sun Solaris, Linux (on Intel) and HP-UX the *sendmail* [1] command is supported. Versions are supplied for use in the Microsoft Windows NT, AIX, Sun Solaris, Linux (on Intel) and HP-UX environments along with source code and documentation.

Possible uses

This SupportPac is designed for people who:

- Want to send an e-mail message based on a XML message
- Are interested in the design of extension nodes for MQSeries Integrator Version 2.

[1] See the man pages on AIX, Sun Solaris, Linux (on Intel) and HP-UX for more information about the *sendmail* command.

Bibliography

- *IBM MQSeries Integrator for Windows NT Version 2 Installation Guide,* IBM Corporation. SC34-5600.
- *IBM MQSeries Integrator for Sun Solaris Version 2 Installation Guide,* IBM Corporation. SC34-5842
- *IBM MQSeries Integrator for AIX Version 2 Installation Guide,* IBM Corporation. SC34-5841
- *IBM MQSeries Integrator for Linux Version 2 Installation Guide,* IBM Corporation. AA00-0000
- *IBM MQSeries Integrator for HP-UX Version 2 Installation Guide,* IBM Corporation. SC34-5907
- *IBM MQSeries Integrator Version 2 Using the Control Center,* IBM Corporation. SC34-5602
- *IBM MQSeries Integrator Version 2 Programming Guide,* IBM Corporation. SC34-5603

Chapter 1. Installing the plug-in node

SupportPac contents

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

/source

SendMailPlugIn.c
SendMailPlugIn.h
SendMessage.c
SendMessage.h
Node_utils.c
Node_utils.h

/smtp

smtp.c
smtp.h

/sendmail

sendmailSendMessage.c
sendmailSendMessage.h

/spawn

spawn.c
spawn.h

/NT

SendMailPlugIn
SendMailPlugIn.gif
SendMailPlugIn.lil
SendMailPlugIn.properties
SendMailPlugIn.wdp
SendMailPlugIn30.gif
SendMailPlugIn42.gif
SendMailPlugIn58.gif
SendMailPlugIn84.gif

/AIX

makefile.smtp

makefile.sendmail

/smtp

SendMailPlugIn.lil

/sendmail

SendMailPlugIn.lil

/SUN

makefile.smtp

makefile.sendmail

/smtp

SendMailPlugIn.lil

/sendmail

SendMailPlugIn.lil

/Linux86

makefile.smtp

makefile.sendmail

/smtp

SendMailPlugIn.lil

/sendmail

SendMailPlugIn.lil

/HPUX

makefile.smtp

makefile.sendmail

/smtp

SendMailPlugIn.lil

/sendmail

SendMailPlugIn.lil

license2.txt

ia07.pdf

Prerequisites

This SupportPac provides a plug-in node to be used with the IBM MQSeries Integrator Version 2.0.1 and above. For normal use, there are no other prerequisite products other than those required by IBM MQSeries Integrator Version 2.0.1 itself. If any changes are to be made to the plug-in node, an appropriate C++ compiler is required.

Supported Platforms

This SupportPac has been developed and tested in a Microsoft Windows NT, AIX, Sun Solaris, Linux (on Intel) and HP-UX environment.

Installing the plug-in node on broker system

The plug-in 'lil' file can be installed by copying or moving the appropriate file to the following directory:

- <mqsi_root>\bin (Windows)
- <mqsi_root>/lil (AIX)
- <mqsi_root>/lil (Sun Solaris)
- <mqsi_root>/lil (Linux on Intel)
- <mqsi_root>/lil (HP-UX)

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

Integrating the plug-in node into the Windows Control Center

The necessary files for integrating the plug-in into the Windows Control Center are provided in the /NT directory.

Use the following table to copy the files to their correct location. These locations should already exist providing you have deployed at least one message flow. Append your <MQSI V2 root install path> to the **Copy to location** value.

Use the following to replace the placeholders:

<hostname> - TCP/IP hostname

<CM QMName> - Configuration Manager's queue manager name

Filename	Copy to location
SendmailPlugIn	\Tool\repository\private\<hostname>\<CM QMName>\MessageProcessingNodeType
SendmailPlugIn.wdp	\Tool\repository\private\<hostname>\<CM QMName>\MessageProcessingNodeType
SendmailPlugIn.gif	\Tool\images
SendmailPlugIn30.gif	\Tool\images
SendmailPlugIn42.gif	\Tool\images

SendmailPlugIn58.gif	\Tool\images
SendmailPlugIn84.gif	\Tool\images
SendmailPlugIn.properties	\Tool\com\ibm\ivm\mqitool\extensions

Defining the node to the configuration repository

When you have installed the files in the appropriate directories, as described in the previous section, you must make these definitions available to the Control Center.

1. Start the Control Center. The user ID you are using must be a member of the MQSeries Integrator group ***mqbrdevt***. You are recommended to use the superuser ***IBMMQSI2*** to complete this task[2]. This causes your new node to be locked under the same user ID as all the supplied IBM primitive nodes. If you do not use this user ID, the definition files in the configuration repository might be accidentally locked, and therefore open to unauthorized update.

2. Select the Message Flows view.

3. Select an existing Message Flow Category, or create a new one.

4. Right-click the selected category, and select *Add->Message Flow*.

   A list box is displayed showing all existing IBM-supplied primitive nodes and any defined message flows you have installed following the instructions provided.

5. Select the message flow (the node).

   This node now appears within the message flow category you selected in the tree view in the left-hand pane.

6. Select your new node, and right-click. Select *Check In*.

7. Right-click again, and select Lock. Then right-click again and select Check In for a second time. After this check, the interface and ****.wdp*** definition files disappear from the local directory and go into the shared repository, where they are available to all users of the Control Center. However, user can only use this new node if they have installed the additional files (icons, properties files, and so on) on their own system.

Updating the node to the configuration repository

When you have installed a previous release of this node, then you must make the new definitions available to the Control Center. You must follow these steps:

1. Start the Control Center. If you locked the node using the MQSeries Integrator superuser ID ***IBMMQSI2*** , you must be logged on with this user ID to make any changes[2] .If not you can use any user ID that is a member of the MQSeries Integrator group **mqbrdevt**.

2. Select the Message Flows view.

3. Select the node that you want to update.

---

[2] You must take care if you change logon IDs to complete this task. Changing logon IDs can effect the operation of the Configuration Manager's queue manager if it is on this system, but not running as a Windows NT service. See the *MQSeries Integrator Administration Guide* for more information about queue manager operation (Chapter 2) and the superuser *IBMMQSI2* (Chapter 4).

4. Check out, unlock and delete the selected node.

5. End your Control Center Session.

6. Install the updated files for this node into the appropriate directories (described in "Integrating the plug-in node into the Windows Control Center" on page 3).

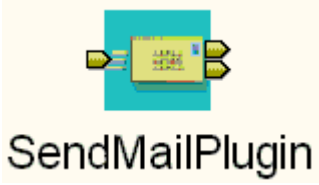
7. Add the node back into the workspace following the instructions given in the section "Defining the node to the configuration repository" on page 4.

8. Deploy all the Message Flows using the plug-in node

Chapter 2. Using the plug-in node

Description

This sendmail plug-in node builds an e-mail message from a XML input message.



Plug-in node terminals

Terminal	Description
in	The input terminal that accepts a message for processing by the node
out	The output terminal that outputs the original message
failure	The output terminal which the message is routed if failure is detected during sending the e-mail. A xml tag with the error message is added to the original message.

Plug-in node properties

These properties are displayed when you right click a SendmailPlugIn node entry in the Message Flow Types pane, and click Properties. The values displayed are the default properties for this instance of the node. They cannot be edited when displayed from the Message Flow Types pane.

Bcc

Sets the name of the Bcc.

Priority

Identifies the priority of the message that will be sent. Valid values are:

Lowest, **Low**, **Normal** (default), **High**, **Highest**

Set Priority

If this value is selected, the Priority field is set.

By default, this value is not selected and all message are sent with Normal priority.

Full name of the sender

Sets the full name of the sender

Name of the 'from' person

Sets the name of the sender of the mail.

SMTP server (SMTP version only)

Specifies the SMTP server

Port number (SMTP version only)

Specifies the SMTP server port number (default port is: 25)

Alternate configuration file (sendmail version only)

Starts the sendmail command using an alternate configuration file specified.

Hop count (sendmail version only)

The hop count is the number of times that the message has been processed by an SMTP router (not just the local copy of the sendmail command). The mail router increments the hop count every time the message is processed. When it reaches a limit, the message is returned with an error message in order to prevent infinite looping.

Log all traffic (sendmail version only)

Logs all traffic in and out of sendmail in the specified file for debugging mailer problems. Use this flag sparingly, since it produces a lot of data very quickly.

Chapter 3. Compiling the plug-in node

Windows NT (smtp version)

```
cl /VERBOSE /LD /MT /I. SendMailPlugIn.c SendMessage.c node_utils.c smtp.c -link /DLL
imbdfplg.lib wsock32.lib /OUT:SendMailPlugIn.lil
```

AIX

Smtp version

makefile:

```
DEFINES        = -DAIX -qcpluscmt -DSMTP
CC             = xlc_r
MQSIROOT       = /home/mqmdvl/mqsiv2
SMTPDIR        = /home/mqmdvl/SimpleSMTP

all:    SendMailPlugIn.lil

SendMailPlugIn.lil:    SendMailPlugIn.o smtp.o SendMessage.o node_utils.o
                       $(CC) -bM:SRE -bexpall -bnoentry -o smtp/SendMailPlugIn.lil -L
$(MQSIROOT)/lib -l imbdfplg $(DEFINES) SendMailPlugIn.o SendMessage.o smtp.o
node_utils.o

SendMailPlugIn.o:      SendMailPlugIn.c SendMailPlugIn.h
                       $(CC) -I $(MQSIROOT)/include -I $(MQSIROOT)/include/plugin -c
SendMailPlugIn.c $(DEFINES)

smtp.o:                $(SMTPDIR)/smtp.c $(SMTPDIR)/smtp.h
                       $(CC) -c $(SMTPDIR)/smtp.c $(DEFINES)

SendMessage.o:         SendMessage.c SendMessage.h
                       $(CC) -c SendMessage.c $(DEFINES)

Node_utils.o:          node_utils.c node_utils.h
                       $(CC) -I $(MQSIROOT)/include -I $(MQSIROOT)/include/plugin -c
node_utils.c $(DEFINES)
```

Sendmail version

makefile:

```
DEFINES        = -DAIX -qcpluscmt -DSENDMAIL
CC             = xlc_r
MQSIROOT       = /home/mqmdvl/mqsiv2
SENDMAILDIR    = /home/mqmdvl/sendmail

all:    SendMailPlugIn.lil

SendMailPlugIn.lil:    SendMailPlugIn.o sendmailSendMessage.o spawn.o SendMessage.o
node_utils.o
                       $(CC) -bM:SRE -bexpall -bnoentry -o sendmail/SendMailPlugIn.lil
-L $(MQSIROOT)/lib -l imbdfplg $(DEFINES) SendMailPlugIn.o SendMessage.o
sendmailSendMessage.o spawn.o node_utils.o

SendMailPlugIn.o:      SendMailPlugIn.c SendMailPlugIn.h
                       $(CC) -I $(MQSIROOT)/include -I $(MQSIROOT)/include/plugin -c
SendMailPlugIn.c $(DEFINES)

sendmailSendMessage.o: $(SENDMAILDIR)/sendmailSendMessage.c
$(SENDMAILDIR)/sendmailSendMessage.h
                       $(CC) -c $(SENDMAILDIR)/sendmailSendMessage.c $(DEFINES)

spawn.o:               $(SENDMAILDIR)/spawn.c $(SENDMAILDIR)/spawn.h
                       $(CC) -c $(SENDMAILDIR)/spawn.c $(DEFINES)

SendMessage.o:         SendMessage.c SendMessage.h
                       $(CC) -c SendMessage.c $(DEFINES)

node_utils.o:          node_utils.c node_utils.h
                       $(CC) -I $(MQSIROOT)/include -I $(MQSIROOT)/include/plugin -c
node_utils.c $(DEFINES)
```

Sun Solaris

Smtp version

makefile:

```
DEFINES        = -DSUN -xCC  -DSMTP
CC             = cc
MQSIROOT       = /opt/mqsi
MQSISAMPLE     = $(MQSIROOT)/sample/plugin
MQSIINCLUDE    = -I$(MQSIROOT)/include -I$(MQSIROOT)/include/plugin -I$(MQSISAMPLE)
SMTPDIR        = SimpleSMTP

all:    SendMailPlugIn.lil

SendMailPlugIn.o:      SendMailPlugIn.c SendMailPlugIn.h
                       $(CC) -mt -I. $(MQSIINCLUDE) -c SendMailPlugIn.c $(DEFINES)

node_utils.o:          node_utils.c node_utils.h
                       $(CC) -mt -I. $(MQSIINCLUDE) -c node_utils.c $(DEFINES)

SendMessage.o:         SendMessage.c SendMessage.h
                       $(CC) -mt -I. -c SendMessage.c $(DEFINES)

smtp.o:                $(SMTPDIR)/smtp.c $(SMTPDIR)/smtp.h
                       $(CC) -mt -I. -c $(SMTPDIR)/smtp.c $(DEFINES)

BipSampPluginUtil.o:   $(MQSISAMPLE)/BipSampPluginUtil.c
$(MQSISAMPLE)/BipSampPluginUtil.h
                       $(CC) -mt -I. $(MQSIINCLUDE) -c
$(MQSISAMPLE)/BipSampPluginUtil.c $(DEFINES)

SendMailPlugIn.lil:    SendMailPlugIn.o smtp.o SendMessage.o node_utils.o
BipSampPluginUtil.o
                       cc -G -o smtp/SendMailPlugIn.lil -L $(MQSIROOT)/lib -l imbdfplg
SendMailPlugIn.o SendMessage.o smtp.o node_utils.o BipSampPluginUtil.o -lsocket –lnsl
```

Sendmail version

makefile:

```
DEFINES        = -DSUN -xCC  -DSENDMAIL
CC             = cc
MQSIROOT       = /opt/mqsi
MQSISAMPLE     = $(MQSIROOT)/sample/plugin
MQSIINCLUDE    = -I$(MQSIROOT)/include -I$(MQSIROOT)/include/plugin -I$(MQSISAMPLE)
SENDMAILDIR    = SimpleSendmail

all:    SendMailPlugIn.lil

SendMailPlugIn.o:      SendMailPlugIn.c SendMailPlugIn.h
                       $(CC) -mt -I. $(MQSIINCLUDE) -c SendMailPlugIn.c $(DEFINES)

node_utils.o:  node_utils.c node_utils.h
                       $(CC) -mt -I. $(MQSIINCLUDE) -c node_utils.c $(DEFINES)

sendmailSendMessage.o: $(SENDMAILDIR)/sendmailSendMessage.c
$(SENDMAILDIR)/sendmailSendMessage.h
                       $(CC) -mt -I. -c $(SENDMAILDIR)/sendmailSendMessage.c $(DEFINES)

spawn.o:               $(SENDMAILDIR)/spawn.c $(SENDMAILDIR)/spawn.h
                       $(CC) -mt -I. -c $(SENDMAILDIR)/spawn.c $(DEFINES)

SendMessage.o:         SendMessage.c SendMessage.h
                       $(CC) -mt -I. -c SendMessage.c $(DEFINES)

BipSampPluginUtil.o:   $(MQSISAMPLE)/BipSampPluginUtil.c
$(MQSISAMPLE)/BipSampPluginUtil.h
                       $(CC) -mt -I. $(MQSIINCLUDE) -c
$(MQSISAMPLE)/BipSampPluginUtil.c $(DEFINES)

SendMailPlugIn.lil:    SendMailPlugIn.o sendmailSendMessage.o spawn.o SendMessage.o
node_utils.o BipSampPluginUtil.o
                       cc -G -o sendmail/SendMailPlugIn.lil -L$(MQSIROOT)/lib -l
imbdfplg SendMailPlugIn.o SendMessage.o sendmailSendMessage.o spawn.o node_utils.o
BipSampPluginUtil.o
```

Linux (on Intel)

Smtp version

makefile:

```
DEFINES        = -DLINUX -DSMTP
CC             = gcc
MQSIROOT       = /opt/mqsi
MQSISAMPLE     = $(MQSIROOT)/sample/plugin
MQSIINCLUDE    = -I$(MQSIROOT)/include -I$(MQSIROOT)/include/plugin -I$(MQSISAMPLE)
SMTPDIR        = /home/mqmdvl/SimpleSMTP

all:    SendMailPlugIn.lil

SendMailPlugIn.o:      SendMailPlugIn.c SendMailPlugIn.h
                       $(CC) -I. $(MQSIINCLUDE) -c SendMailPlugIn.c $(DEFINES)

SendMessage.o:         SendMessage.c SendMessage.h
                       $(CC) -I. -c SendMessage.c $(DEFINES)

node_utils.o:          node_utils.c node_utils.h
                       $(CC) -I. $(MQSIINCLUDE) -c node_utils.c $(DEFINES)

smtp.o:                $(SMTPDIR)/smtp.c $(SMTPDIR)/smtp.h
                       $(CC) -I. -c $(SMTPDIR)/smtp.c $(DEFINES)

BipSampPluginUtil.o:   $(MQSISAMPLE)/BipSampPluginUtil.c
$(MQSISAMPLE)/BipSampPluginUtil.h
                       $(CC) -I. $(MQSIINCLUDE) -c $(MQSISAMPLE)/BipSampPluginUtil.c
$(DEFINES)

SendMailPlugIn.lil:    SendMailPlugIn.o smtp.o SendMessage.o node_utils.o
BipSampPluginUtil.o
                       ld -shared -o smtp/SendMailPlugIn.lil -L $(MQSIROOT)/lib -l
imbdfplg SendMailPlugIn.o SendMessage.o smtp.o node_utils.o BipSampPluginUtil.o
```

Sendmail version

makefile:

```
DEFINES        = -DLINUX -DSENDMAIL
CC             = gcc
MQSIROOT       = /opt/mqsi
MQSISAMPLE     = $(MQSIROOT)/sample/plugin
MQSIINCLUDE    = -I$(MQSIROOT)/include -I$(MQSIROOT)/include/plugin -I$(MQSISAMPLE)
SENDMAILDIR     = /home/mqmdvl/SimpleSendmail

all:    SendMailPlugIn.lil

SendMailPlugIn.o:      SendMailPlugIn.c SendMailPlugIn.h
                       $(CC) -I. $(MQSIINCLUDE) -c SendMailPlugIn.c $(DEFINES)

sendmailSendMessage.o: $(SENDMAILDIR)/sendmailSendMessage.c
$(SENDMAILDIR)/sendmailSendMessage.h
                       $(CC) -I. -c $(SENDMAILDIR)/sendmailSendMessage.c $(DEFINES)

spawn.o:               $(SENDMAILDIR)/spawn.c $(SENDMAILDIR)/spawn.h
                       $(CC) -I. -c $(SENDMAILDIR)/spawn.c $(DEFINES)

SendMessage.o:         SendMessage.c SendMessage.h
                       $(CC) -I. -c SendMessage.c $(DEFINES)

node_utils.o:          node_utils.c node_utils.h
                       $(CC) -I. $(MQSIINCLUDE) -c node_utils.c $(DEFINES)

BipSampPluginUtil.o:   $(MQSISAMPLE)/BipSampPluginUtil.c
$(MQSISAMPLE)/BipSampPluginUtil.h
                       $(CC) -I. $(MQSIINCLUDE) -c $(MQSISAMPLE)/BipSampPluginUtil.c
$(DEFINES)

SendMailPlugIn.lil:    SendMailPlugIn.o sendmailSendMessage.o spawn.o SendMessage.o
node_utils.o BipSampPluginUtil.o
                       ld -shared -o sendmail/SendMailPlugIn.lil -L$(MQSIROOT)/lib -l
imbdfplg SendMailPlugIn.o SendMessage.o sendmailSendMessage.o spawn.o node_utils.o
BipSampPluginUtil.o
```

HP-UX

Smtp version

makefile:

```
DEFINES        = -DHPUX -DSMTP +z
CC             = cc
MQSIROOT       = /opt/mqsi
MQSISAMPLE     = $(MQSIROOT)/sample/plugin
MQSIINCLUDE    = -I$(MQSIROOT)/include -I$(MQSIROOT)/include/plugin -I$(MQSISAMPLE)
SMTPDIR         = SimpleSMTP

all:    SendMailPlugIn.lil

SendMailPlugIn.o:      SendMailPlugIn.c SendMailPlugIn.h
                       $(CC) -I. -I$(MQSIINCLUDE) -c SendMailPlugIn.c $(DEFINES)

SendMessage.o: SendMessage.c SendMessage.h
               $(CC) -I. -c SendMessage.c $(DEFINES)

node_utils.o:  node_utils.c node_utils.h
               $(CC) -I. -I$(MQSIROOT)/include -I$(MQSIROOT)/include/plugin -c
node_utils.c $(DEFINES)

smtp.o:        $(SMTPDIR)/smtp.c $(SMTPDIR)/smtp.h
               $(CC) -I. -c $(SMTPDIR)/smtp.c $(DEFINES)

BipSampPluginUtil.o:   $(MQSISAMPLE)/BipSampPluginUtil.c
$(MQSISAMPLE)/BipSampPluginUtil.h
                       $(CC) -I. $(MQSIINCLUDE) -c $(MQSISAMPLE)/BipSampPluginUtil.c
$(DEFINES)

SendMailPlugIn.lil:    SendMailPlugIn.o smtp.o SendMessage.o node_utils.o
BipSampPluginUtil.o
                       ld -b -o smtp/SendMailPlugIn.lil -L $(MQSIROOT)/lib -l imbdfplg
SendMailPlugIn.o SendMessage.o smtp.o node_utils.o BipSampPluginUtil.o
```

Sendmail version

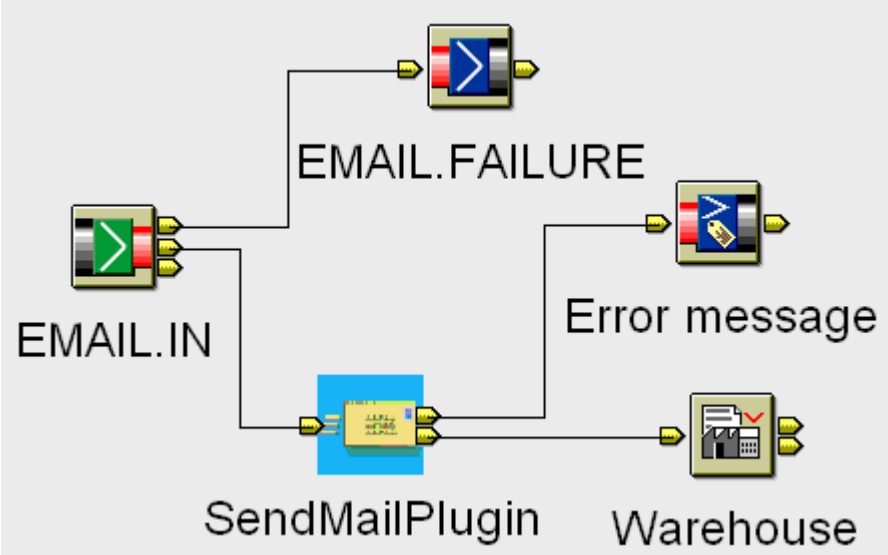
makefile:

```
DEFINES        = -DHPUX -DSENDMAIL +z
CC             = cc
MQSIROOT       = /opt/mqsi
MQSISAMPLE     = $(MQSIROOT)/sample/plugin
MQSIINCLUDE    = -I$(MQSIROOT)/include -I$(MQSIROOT)/include/plugin -I$(MQSISAMPLE)
SENDMAILDIR     = SimpleSendmail

all:    SendMailPlugIn.lil

SendMailPlugIn.o:      SendMailPlugIn.c SendMailPlugIn.h
                       $(CC) -I. -I$(MQSIINCLUDE) -c SendMailPlugIn.c $(DEFINES)

sendmailSendMessage.o: $(SENDMAILDIR)/sendmailSendMessage.c
$(SENDMAILDIR)/sendmailSendMessage.h
                       $(CC) -I. -c $(SENDMAILDIR)/sendmailSendMessage.c $(DEFINES)

spawn.o:       $(SENDMAILDIR)/spawn.c $(SENDMAILDIR)/spawn.h
               $(CC) -I. -c $(SENDMAILDIR)/spawn.c $(DEFINES)

SendMessage.o: SendMessage.c SendMessage.h
               $(CC) -I. -c SendMessage.c $(DEFINES)

node_utils.o:  node_utils.c node_utils.h
               $(CC) -I. -I$(MQSIROOT)/include -I$(MQSIROOT)/include/plugin -c
node_utils.c $(DEFINES)

BipSampPluginUtil.o:   $(MQSISAMPLE)/BipSampPluginUtil.c
$(MQSISAMPLE)/BipSampPluginUtil.h
                       $(CC) -I. $(MQSIINCLUDE) -c $(MQSISAMPLE)/BipSampPluginUtil.c
$(DEFINES)

SendMailPlugIn.lil:    SendMailPlugIn.o sendmailSendMessage.o spawn.o SendMessage.o
node_utils.o BipSampPluginUtil.o
```

```
                        ld -b -o sendmail/debug/SendMailPlugIn.lil -L$(MQSIROOT)/lib -l
imbdfplg SendMailPlugIn.o SendMessage.o sendmailSendMessage.o spawn.o node_utils.o
BipSampPluginUtil.o
```

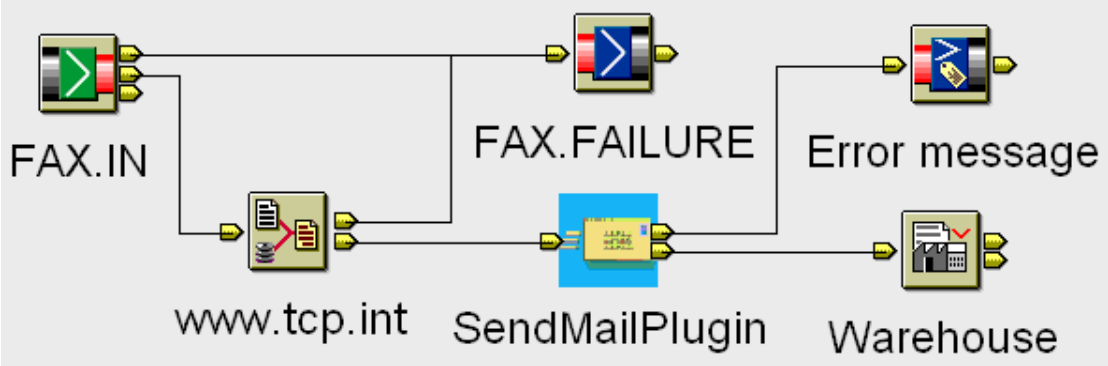
Chapter 4. Example using the plug-in node

Simple flow sending e-mail



Send FAX by e-mail

This message flow uses the free fax Internet service from: www.tcp.int/faxbymail.html



Input XML message

```
<Message>
  <From>avv@nl.ibm.com</From>
  <Recipient>Arjan van Vught</Recipient>
  <FaxNumber>31791239876</FaxNumber>
  <Subject>Subject for the FAX by e-mail</Subject>
  <Body>This is a fax from MQSeries Integrator V2 using the e-mail
plug-in node</Body>
</Message>
```

ESQL Compute node “www.tcp.int”

```
SET OutputRoot.XML.Message."To" = InputRoot.XML.Message.Recipient;

SET I = POSITION(' ' IN OutputRoot.XML.Message."To");

WHILE (I <> 0) DO
   SET OutputRoot.XML.Message."To" =
OVERLAY(OutputRoot.XML.Message."To" PLACING '_' FROM I);
   SET I = POSITION(' ' IN OutputRoot.XML.Message."To");
END WHILE;

SET OutputRoot.XML.Message."From"   = InputRoot.XML.Message."From";
SET OutputRoot.XML.Message."To"     = 'remote-printer.' ||
                                      OutputRoot.XML.Message."To" ||
                                      '@' ||
                                      InputRoot.XML.Message.FaxNumber
||
                                      '.iddd.tpc.int';

SET OutputRoot.XML.Message.Subject  = InputRoot.XML.Message.Subject;
SET OutputRoot.XML.Message.Body     = InputRoot.XML.Message.Body;
```

Chapter 5. Mapping fields to e-mail message header

The following MQSeries MD fields are mapped onto e-mail header fields:

MQMD field	Header field	Description
MsgId	Message-ID:	Message id in e-mail header
PutDate	Date:	Date in e-mail header
PutTime	Date:	Time in e-mail header

The Priority field is mapped onto the following e-mail header fields:

Priority	Header field	
	Priority	X-Priority
Lowest	Low	5 (Lowest)
Low	Low	4 (Low)
Normal	Normal	3 (Normal)
High	High	2 (High)
Highest	High	1 (Highest)

The following e-mail header fields are hard-coded:

```
Importance: Normal
X-mailer: IBM MQSeries Integrator V2 Plug-in node
```

Example e-mail header fields:

```
Date: 01 May 2001 09:54:27 GMT
Message-ID: 414D51204E4C52473131392020202020205037543A13800200
Priority: Normal
Importance: Normal
X-Priority: 3 (Normal)
X-mailer: IBM MQSeries Integrator V2 Plug-in node
```

Chapter 6. The XML message

Input message

The input message must adhere the following structure:

```
<Message>
  <From> </From>
  <To> </To>                    (optional repeated)
  <Cc> </Cc>                    (optional repeated)
  <Bcc> </Bcc>                  (optional repeated)
  <Subject> </Subject>
  <Body> </Body>
</Message>
```

or

```
<Message>
  <From> </From>
  <To> </To>                    (optional repeated)
  <Cc> </Cc>                    (optional repeated)
  <Bcc> </Bcc>                  (optional repeated)
  <Subject> </Subject>
  <Body>
    <Line> </Line>              (optional repeated)
  </Body>
</Message>
```

From release V1.2 and above, all tags can be prefixed with “Mail” e.g. MailTo and MailFrom

Output error message

The output message will have the following structure:

```
<Message>
  <From> </From>                (if included in input message)
  <To> </To>                    (if included in input message)
  <Cc> </Cc>                    (if included in input message)
  <Bcc> </Bcc>                  (if included in input message)
  <Subject> </Subject>          (if included in input message)
  <Body> </Body>                (if included in input message)
  <SendMailError> </SendMailError>
</Message>
```

Error strings SMTP (TCP/IP)

- TCP/IP error
- TCP/IP bad port
- TCP/IP host unknown
- TCP/IP no more sockets
- TCP/IP connection failed
- TCP/IP memory allocation error
- Unknown TCP/IP error

Error strings SMTP (SMTP server)

The messages returned by the SMTP server.

Error strings sendmail

- The sendmail command cannot create a file that the user specified.
- An error was found in the format of the configuration file.
- An error occurred while trying to gain access to a database.
- An error occurred during I/O.
- The sendmail command could not recognize the specified host name.
- The user does not have permission to perform the requested operation.
- The sendmail command could not recognize a specified user ID.
- A temporary operating system error occurred. An example of such an error is a failure to create a new process.
- A system file error occurred. For example, a system file (such as /etc/passwd) does not exist, cannot be opened, or has another type of error preventing it from being used.
- The remote system returned something that was incorrect during a protocol exchange.
- An internal software error occurred (including bad arguments).
- The sendmail command could not create a connection to a remote system. Try the request again later.
- A service or resource that the sendmail command needed was not available.
- The command syntax was not correct.
- Unknown sendmail error.

Chapter 7. Sendmail support implementation

On AIX and Sun Solaris support is available for the *sendmail* command. The sendmail command is always called with the `–t` parameter. The temporary input file is created in the default `/tmp` directory and the content is based on the XML tag´s from the input message. Based on the plug-in properties, addition parameters are added when calling the *sendmail* command.

Code fragment:

```
if (SendmailOptParms[0] != '\0')
   sprintf(systemBuffer,"sendmail %s -t < %s", SendmailOptParms, tempFileName);
else
   sprintf(systemBuffer,"sendmail -t < %s", tempFileName);

rc = spawnl(_P_WAIT, "/usr/bin/sh", "sh", "-c", systemBuffer, NULL);
```

The function `spawnl`[3] is a user-defined function. This function creates and executes a new process. The return value is the exit status of the new process. The exit status is 0 if the process terminated normally. A return value of –1 indicates an error (the new process is not started). In this case, **`errno`** is set.

```
int spawnl(int mode, const char *path, const char *argv0, const char *argv1, const
char *argv2)
{
   int status  = -1;
   int err     = -1;
   int retstat = -1;
   int rd      = -1;
   int pfd[2]  = { -1, -1 };

   pid_t pid = -1;
   pid_t w   = -1;

   void(*istat)(), (*qstat)();

   fflush(stdout);
   fflush(stderr);

   if (pipe(pfd) == -1) {
#ifdef _DEBUG
       perror("pipe");
#endif
       return -1;
   }

   fcntl(pfd[0], F_SETFD, 1);
   fcntl(pfd[1], F_SETFD, 1);

   istat = signal(SIGINT, SIG_IGN);
   qstat = signal(SIGQUIT, SIG_IGN);

   pid = fork();

   if (pid == -1) {
#ifdef _DEBUG
     perror("fork");
#endif
   }
   else if (pid == 0) {

      execl(path, argv0, argv1, argv2, NULL);

      err = errno;
```

[3] This function is compatible with the spawn functions provided on the Windows NT platform.

```
#ifdef _DEBUG
      perror(path);
#endif
      write(pfd[1], (char *)&err, sizeof(err));
      _exit(127);
   }
   else {
      close(pfd[1]);
      rd = read(pfd[0], &err, sizeof(err));
      close(pfd[0]);

      while ((w = wait(&status)) != pid)
         if (w < 0 && errno != EINTR)
            break;

      if (rd > 0) {
          /* exec failed - errno was read from pipe */
          retstat = -1;
      } else if (w < 0) {
         /* something went wrong in wait */
         err = errno;
         retstat = -1;
      } else if (!WIFEXITED(status)) {
          /* program probably was killed or signalled */
          err = EINTR;
          retstat = -1;
      } else {
         /* program terminated */
        err = 0;
        retstat = WEXITSTATUS(status);
      }
   }

   (void) signal(SIGINT, istat);
   (void) signal(SIGQUIT, qstat);

   return retstat;
}
```

End of Document