WebSphere MQ Integrator – MQSIE header parser

Version 1.3

13th January 2003

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".

Fourth Edition, January 2003

This edition applies to Version 1.3 of *WebSphere MQ Integrator - MQSIE parser* and to all subsequent releases and modifications unless otherwise indicated in new editions.

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 MQ Integrator, WMQI
- AIX
- Exchange for WebSphere MQ

The following terms are trademarks of other companies:

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

Summary of Amendments

Date	Changes
14 September 2001	Initial release
27 September 2001	Linux (on Intel) support added Makefile included for AIX, Sun Solaris and Linux (on Intel)
19 October 2001	HP-UX support added Updated MQSeries Services URL
1 July 2002	Updated pictures Appendix A
13 January 2003	JMS clients with Exchange for WebSphere MQ samples added New brand names: WebSphere MQ, Exchange for WebSphere MQ

Preface

This SupportPac contains a parser written for WebSphere MQ Integrator Version 2.0.1. 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. It adds support for parsing of input messages and creation of output messages in the MQSIE header format.

Additional, this Support Pac will demonstrate how to use JMS clients with Exchange for WebSphere MQ. At the time of writing (January 2003), Exchange for WebSphere MQ does not support the JMS header properties. Using WebSphere MQ Integrator you can map JMS messages onto Exchange for WebSphere MQ messages with the MQSIE header.

A general understanding of JMS, WebSphere MQ Integrator V2 and Exchange for WebSphere MQ is necessary to use this SupportPac.

For more information on Exchange for WebSphere MQ:
http://edi.services.ibm.com/mqseries/spec_sheets.shtml

Possible uses

This SupportPac is designed for people who:

- Will be writing B2B applications for Exchange for WebSphere MQ
- Want to send/receive data to/from Information Exchange that includes the MQSIE header.
- Want to migrate from traditional EDI file transfer to native WebSphere MQ
- Are interested in transmitting messages between Exchange for WebSphere MQ and JMS applications

Prerequisites

- A platform supported by
    - Exchange for WebSphere MQ
    - WebSphere MQ Integrator V2

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

- *IBM MQSeries Services Administration and Application Development,* IBM Corporation. GC34-2344

Chapter 1. Installing the plug-in parser

SupportPac contents

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

- /source
  - MQSIEparser.c
  - MQSIEparser.h
  - FFMSG001parser.c
  - FFMSG001parser.h
  - node_utils.c
  - node_utils.h
- /NT
  - MQSIEparser.lil
  - FFMSG001parser.lil
- /AIX
  - MQSIEparser.lil
  - FFMSG001parser.lil
  - makefile.MQSIE
  - makefile.FFMSG001
- /SUN
  - MQSIEparser.lil
  - FFMSG001parser.lil
  - makefile.MQSIE
  - makefile.FFMSG001
- /Linux86
  - MQSIEparser.lil
  - FFMSG001parser.lil
  - makefile.MQSIE
  - makefile.FFMSG001
- /HP-UX
  - MQSIEparser.lil
  - FFMSG001parser.lil
  - makefile.MQSIE
  - makefile.FFMSG001

license2.txt

ia0d.pdf

Workspace.xml

Prerequisites

This SupportPac provides a plug-in parser 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 parser on broker system

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

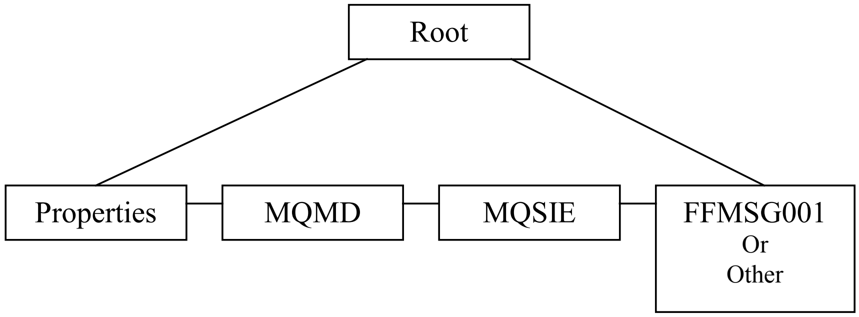
• <mqsi_root>\bin	(Windows)
• <mqsi_root>/lil	(AIX)
• <mqsi_root>/lil	(Sun Solaris)
• <mqsi_root>/lil	(Linux on Intel)
• <mqsi_root>/lil	(HP-UX)

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

Chapter 2. Using the plug-in parser

The MQSIE header parser takes input messages in valid MQSIE header format and creates MQSeries Integrator V2 logical message tree structures that can be processed by MQSeries Integrator message flows. Similarly, it will take a logical message tree created by a message flow and produce the data portion of an MQSeries message in MQSIE header format.

All logical message trees used within MQSeries Integrator V2 have a basic structure. This comprises of a single high-level element known as the root element. The user data is found in the body of the message. The body has a single high-level element that is the last child of the root element. For a MQSIE header message the name of this high-level element must be "MQSIE" to match the message domain supported by MQSIE header parser. Additional, for a FFMSG001 message the name of this high-level element must be "FFMSG001" to match the message domain supported by FFMSG001 parser.



MQSIE header data tags mapping

Tag name	MQSeries Integrator mapping
DATATYPE	CCI_VALUE_TYPE_CHAR
CLASS	CCI_VALUE_TYPE_CHAR
CONTROLNUM	CCI_VALUE_TYPE_INTEGER
MSGNAME	CCI_VALUE_TYPE_CHAR
MSGSEQNO	CCI_VALUE_TYPE_CHAR
RECLEN	CCI_VALUE_TYPE_INTEGER
RECDLM	CCI_VALUE_TYPE_CHAR
RECFM	CCI_VALUE_TYPE_CHAR
SENDER	CCI_VALUE_TYPE_CHAR
SENDQUAL	CCI_VALUE_TYPE_CHAR
RECEIVER	CCI_VALUE_TYPE_CHAR
RECVQUAL	CCI_VALUE_TYPE_CHAR
FROM	CCI_VALUE_TYPE_CHAR
UNIQUEID	CCI_VALUE_TYPE_CHAR
DATE	CCI_VALUE_TYPE_DATE
TIME	CCI_VALUE_TYPE_TIME

Chapter 3. Compiling the plug-in parser

Windows NT

```
cl /VERBOSE /LD /MT /I. MQSIEparser.c node_utils.c -link /DLL imbdfplg.lib
/OUT:MQSIEparser.lil
```

AIX

```
DEFINES        = -DAIX -qcpluscmt
CC             = xlc_r
HOMEDIR        = /home/mqmdvl
MQSIROOT       = /home/mqmdvl/mqsiv2

all:    MQSIEparser.lil

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

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

MQSIEparser.lil:       MQSIEparser.o node_utils.o
                       $(CC) -bM:SRE -bexpall -bnoentry -o MQSIEparser.lil -L
$(MQSIROOT)/lib -l imbdfplg MQSIEparser.o node_utils.o
```

Sun Solaris

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

all:    MQSIEparser.lil

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

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

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

MQSIEparser.lil:       MQSIEparser.o node_utils.o BipSampPluginUtil.o
                       $(CC) -G -o MQSIEparser.lil -L $(MQSIROOT)/lib -l imbdfplg
MQSIEparser.o node_utils.o BipSampPluginUtil.o
```

Linux (on Intel)

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

all:                  MQSIEparser.lil

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

MQSIEparser.lil:      MQSIEparser.o node_utils.o BipSampPluginUtil.o
                      ld -shared -o MQSIEparser.lil -L $(MQSIROOT)/lib -l imbdfplg
MQSIEparser.o node_utils.o BipSampPluginUtil.o


MQSIEparser.lil:      MQSIEparser.o node_utils.o BipSampPluginUtil.o
                      $(CC) -G -o MQSIEparser.lil -L $(MQSIROOT)/lib -l imbdfplg
MQSIEparser.o node_utils.o BipSampPluginUtil.o
```

HP-UX

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

all:                  MQSIEparser.lil

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

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

Chapter 4. Example using the plug-in parser

Sending an XML e-mail with an MQSIE header

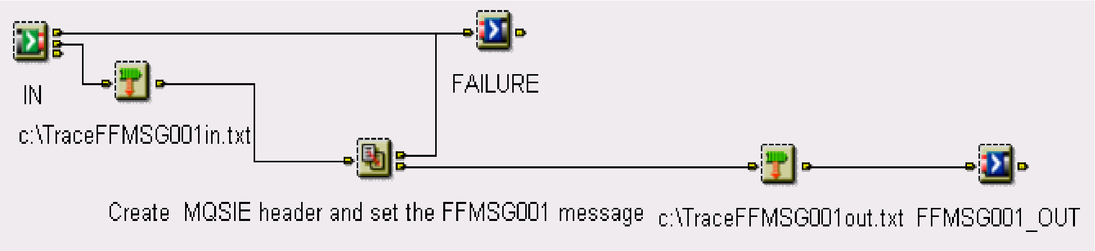
To send an e-mail message to an Information exchange user, you must format the MQSeries message as 79-byte records, set the class to FFMSG001, and copy the description data to the e-mail subject data. Any other fields may be used to qualify the message further for the reipient.

The following is an example of sending an e-mail (XML formatted) with an MQSIE header:

- Message descriptor Format field: MQSIE
- Message content (three MQSIE tags):

```
<Message>
 <Recipient>
   <Address>EUR.NLEBCC.NLEBC001</Address>
 </Recipient>
 <Header>
   <Class>FFMSG001</Class>
   <Datatype>T</Datatype>
   <Description>Tomorrow's meeting</Description>
 </Header>
 <Body>
  <Line>Bob,</Line>
  <Line>Can you give me a call before the meeting tomorrow?</Line>
  <Line>I'll be in the office until about 6 p.m.</Line>
  <Line>Alice</Line>
 </Body>
</Message>
```

MQSeries Integrator Version 2 workspace



Create MQSIE header and set the FFMSG001 message

```
DECLARE I INTEGER;
SET I = 1;
WHILE I < CARDINALITY(InputRoot.*[]) DO
       SET OutputRoot.*[I] = InputRoot.*[I];
       SET I=I+1;
END WHILE;
-- Enter SQL below this line.  SQL above this line might be regenerated, causing any
modifications to be lost.
DECLARE LOOPCOUNTER INTEGER;

SET OutputDestinationList.Destination.MQDestinationList.Defaults.queueManagerName =
'MQ31';

SET LOOPCOUNTER = 1;

WHILE LOOPCOUNTER <= CARDINALITY(InputRoot.XML.Message.Recipient.Address[]) DO
  SET
OutputDestinationList.Destination.MQDestinationList.DestinationData[LOOPCOUNTER].queue
Name =
       InputRoot.XML.Message.Recipient.Address[LOOPCOUNTER];
  SET LOOPCOUNTER = LOOPCOUNTER + 1;
END WHILE;

SET OutputRoot.MQMD.Format = 'MQSIE';

SET OutputRoot.MQSIE.DATATYPE    = InputRoot.XML.Message.Header.Datatype;
SET OutputRoot.MQSIE.CLASS       = InputRoot.XML.Message.Header.Class;
SET OutputRoot.MQSIE.DESCRIPTION = InputRoot.XML.Message.Header.Description;

SET LOOPCOUNTER = 1;

WHILE LOOPCOUNTER <= CARDINALITY(InputRoot.XML.Message.Body.Line[]) DO
  SET OutputRoot.FFMSG001.Record[LOOPCOUNTER] =
       InputRoot.XML.Message.Body.Line[LOOPCOUNTER];
  SET LOOPCOUNTER = LOOPCOUNTER + 1;
END WHILE;
```

TraceFFMSG001out.txt

```
(0x1000000)MQMD       = (
    (0x3000000)SourceQueue      = 'IN'
    (0x3000000)Transactional    = TRUE
    (0x3000000)Encoding         = 546
    (0x3000000)CodedCharSetId   = 437
    (0x3000000)Format           = 'MQSIE'
    (0x3000000)Version          = 2
    (0x3000000)Report           = 0
    (0x3000000)MsgType          = 8
    (0x3000000)Expiry           = -1
    (0x3000000)Feedback         = 0
    (0x3000000)Priority         = 0
    (0x3000000)Persistence      = 1
    (0x3000000)MsgId            = X'414d51204d51535330312020202020201705533b12100200'
    (0x3000000)CorrelId         = X'000000000000000000000000000000000000000000000000'
    (0x3000000)BackoutCount     = 0
    (0x3000000)ReplyToQ         = '                                                '
    (0x3000000)ReplyToQMgr      = 'MQSS01                                          '
    (0x3000000)AccountingToken  =
X'16010515000000a501c87f015b4d386d74d443eb03000000000000000000000b'
    (0x3000000)ApplIdentityData = '                                '
    (0x3000000)PutApplType      = 11
note: not all fields are shown
  )
(0x1000000)MQSIE      = (
  (0x3000000)DATATYPE    = 'T'
  (0x3000000)CLASS       = 'FFMSG001'
  (0x3000000)DESCRIPTION = 'Tomorrow's meeting'
)
(0x1000000)FFMSG001   = (
  (0x3000000)Record = 'Bob,'
  (0x3000000)Record = 'Can you give me a call before the meeting tomorrow?'
  (0x3000000)Record = 'I'll be in the office until about 6 p.m.'
  (0x3000000)Record = 'Alice'
)

===== Destination List =====

(
  (0x1000000)Destination = (
    (0x1000000)MQDestinationList = (
      (0x1000000)Defaults        = (
        (0x3000000)queueManagerName = 'MQ31'
      )
      (0x1000000)DestinationData = (
        (0x3000000)queueName = 'EUR.NLEBCC.NLEBC001'
      )
    )
  )
)
```

Output e-mail FFMSG001 message

List Inbound Messages

```
M3100US                      List Inbound Messages

Account ID.............   NLEBCC
User ID................   NLEBC001

Codes (multiple selection)
  D = Delete message    H = View common data header     M = Hold in mailbox
  R = Remove hold       S = See additional information  V = View text (BILLABLE)

      --------- Sender ---------  User msg  --- Submitted ----  Rcv  Arch  CDH
Code  Sys ID  Account   User ID   class     Date      Time      ind  ind   ind
  _    MQS    IBMMQS    NLMQSS01  FFMSG001  01/01/03  13:11:49              Y
```

Common Data Header for the Inbound Message

```
M3103US          Common Data Header for the Inbound Message   (Panel 1 of 2)

Account ID.............   NLEBCC
User ID................   NLEBC001

Action... _  D = Delete message  M = Hold in mailbox  R = Remove hold
             S = See additional information           V = View text  (BILLABLE)


File ID.....
File loc....
Description.  Tomorrow's meeting


Record format.          Record length...          Record delimiter.... NONE
Data type..... EBCDIC   Data format..... NOT EDI  Translate type...... IE
Sending system 62       Sending version. 1        Sending code page...
File date..... 01/01/03 File time....... 08:11:50 Unique ID........... B5307795
```

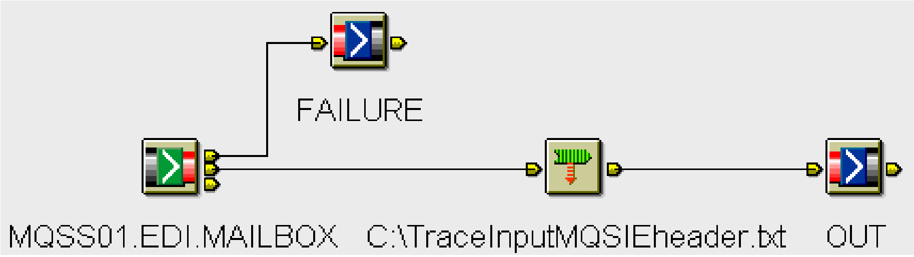
View Message Text

```
M3102US                    View Message Text          Page     000001 of 000001
                                                      Columns   00001 -   00079
Message text sent to
  Account ID / User ID      NLEBCC   / NLEBC001    This is a BILLABLE function
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----
Bob,
Can you give me a call before the meeting tomorrow?
I'll be in the office until about 6 p.m.
Alice
```

Sample MQSIE header on item from Information Exchange

For each message sent from Information Exchange to MQSeries Services, the Report option on the destination queue is checked tot determine if the recipient wants an MQSIE header to be prefaced to the message.

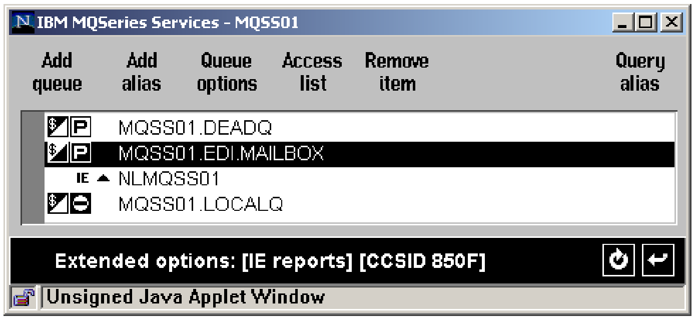
MQSeries Integrator Version 2 workspace



TraceInputMQSIEheader.txt

```
(0x1000000)MQMD       = (
    (0x3000000)SourceQueue       = 'MQSS01.EDI.MAILBOX'
    (0x3000000)Transactional     = TRUE
    (0x3000000)Encoding          = 785
    (0x3000000)CodedCharSetId    = 850
    (0x3000000)Format            = 'MQSIE   '
    (0x3000000)Version           = 2
    (0x3000000)Report            = 0
    (0x3000000)MsgType           = 8
    (0x3000000)Expiry            = GMTTIMESTAMP '2001-08-17 07:23:55.273998'
    (0x3000000)Persistence       = 1
    (0x3000000)MsgId             = X'c0762d929626b6d7f4d5d3d4d8e2e2f0f1b6269786a6a356'
    (0x3000000)CorrelId          = X'000000000000000000000000000000000000000000000000'
    (0x3000000)BackoutCount      = 0
    (0x3000000)ReplyToQ          = 'EUR.NLEBCC.NLEBC001                              '
    (0x3000000)ReplyToQMgr       = 'MQ31                                             '
note: not all fields are shown
    (0x3000000)PutDate           = DATE '2001-07-18'
    (0x3000000)PutTime           = GMTTIME '07:24:35'
  )
  (0x1000000)MQSIE       = (
    (0x3000000)DATATYPE = 'T'
    (0x3000000)CLASS    = 'FFMSG001'
    (0x3000000)RECLEN   = 79
    (0x3000000)RECDLM   = 'N'
    (0x3000000)RECFM    = 'F'
    (0x3000000)FROM     = 'EUR.NLEBCC.NLEBC001'
    (0x3000000)UNIQUEID = '72018796'
    (0x3000000)DATE     = DATE '2001-07-18'
    (0x3000000)TIME     = TIME '07:20:18'
  )
  (0x1000000)FFMSG001    = (
    (0x3000000)Record = 'This is a message from Information Exchange
'
  )
```

MQSeries Services Java-based administration tool



Extended options: [IE reports] [CCSID 850F]

Chapter 5. Exchange for WebSphere MQ – JMS messages

This section will demonstrate how to use JMS clients with Exchange for WebSphere MQ. At the time of writing (January 2003), Exchange for WebSphere MQ does not support the JMS header properties. Using WebSphere MQ Integrator you can map JMS messages onto Exchange for WebSphere MQ messages with the MQSIE header.

JMS messages are composed of the following parts: Header, Properties and Body.

Mapping JMS messages onto Exchange for WebSphere MQ messages

This section describes how the JMS message structure is mapped onto an Exchange for WebSphere MQ message. It is of interest to programmers who wish to transmit messages between JMS clients and an EDI VAN using Exchange for WebSphere MQ.

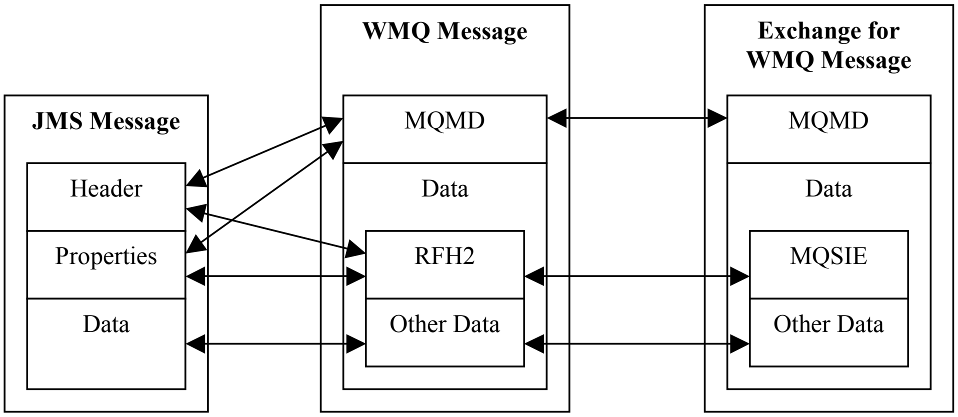
MQ JMS messages are composed of three components:

- The MQSeries Message Descriptor (MQMD)
- An MQSeries MQRFH2 header
- The message body.

Exchange for WebSphere MQ messages are composed of three components:

- The MQSeries Message Descriptor (MQMD)
- An Exchange for WebSphere MQ (MQSIE) header (optional)
- The message body.

How messages are transformed between JMS and Exchange for WebSphere MQ:



The MQRFH2 header

This section describes the MQRFH2 header, which carries the JMS-specific data and the Exchange for WebSphere MQ-specific data. This section covers only its use for Exchange for WebSphere MQ.

There are two parts of the MQRFH2 header, a fixed portion and a variable portion. The variable portion follows the fixed portion. The variable portion contains a variable number of MQRFH2 Folders. The <usr> folder is used to transport the Exchange for WebSphere MQ properties associated with the message.

JMS properties with corresponding MQSIE fields

Table below lists the JMS Exchange for WebSphere MQ-defined properties that are mapped directly to MQSIE fields.

JMS MQSS specific property	MQSIE field
IBM_IE_DATATYPE	DATATYPE
IBM_IE_CLASS	CLASS
IBM_IE_CONTROLNUM	CONTROLNUM
IBM_IE_MSGNAME	MSGNAME
IBM_IE_MSGSEQNO	MSGSEQNO
IBM_IE_RECLEN	RECLEN
IBM_IE_RECDLM	RECDLM
IBM_IE_RECFM	RECFM
IBM_IE_SENDER	SENDER
IBM_IE_SENDQUAL	SENDQUAL
IBM_IE_RECEIVER	RECEIVER
IBM_IE_RECVQUAL	RECVQUAL
IBM_IE_FROM	FROM
IBM_IE_UNIQUEID	UNIQUEID
IBM_IE_DATE	DATE
IBM_IE_TIME	TIME
IBM_IE_DATATYPE	DATATYPE
IBM_IE_RETAIN	RETAIN
IBM_IE_DESCRIPTION	DESCRIPTION

Mapping JMS fields onto MQSIE header fields (outgoing messages)

Sending EDI data

If the system defaults extracted from your EDI data headers are appropriate for your needs, and you do not wish to override them, you can send the data as is with no header information. The following JMS MQ specific properties can be supplied to override any of the default values. The IBM_IE_TO property should be used for qualifying the recipient.

MQSIE field	JMS MQSS specific property
CLASS – class	IBM_IE_CLASS
MSGNAME – msgname	IBM_IE_MSGNAME
MSGSEQNO – msgseqno	IBM_IE_MSGSEQNO
DATATYPE – datatype	IBM_IE_DATATYPE
RETAIN – retain time	IBM_IE_RETAIN
DESCRIPTION - description	IBM_IE_DESCRIPTION
RECLEN – record length	IBM_IE_RECLEN
RECDLM – record delimiter	IBM_IE_RECDLM
RECFM – record format	IBM_IE_RECFM

Sending non-EDI data

No data tags are assumed for non-EDI messages. If you need to qualify the message further for the recipient (IBM_IE_TO property), use the following JMS MQ specific properties:

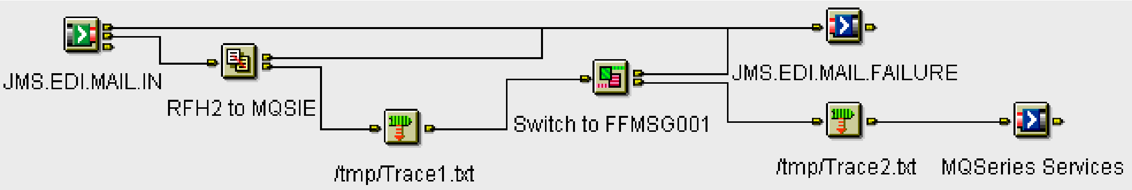
MQSIE field	JMS MQSS specific property
CLASS – class	IBM_IE_CLASS
MSGNAME – msgname	IBM_IE_MSGNAME
MSGSEQNO – msgseqno	IBM_IE_MSGSEQNO
RETAIN – retain time	IBM_IE_RETAIN
DESCRIPTION - description	IBM_IE_DESCRIPTION

Sending e-mail

To send an e-mail message to an Information Exchange user, you must format the JMS message as a 79-byte record, set the class (IBM_IE_CLASS) to FFMSG001, and copy the e-mail subject data to the description data. Any other properties may be used to qualify the message further for the recipient. The IBM_IE_TO property should be used for qualifying the recipient.

MQSIE field	JMS MQSS specific property
MSGNAME – msgname	IBM_IE_MSGNAME
MSGSEQNO – msgseqno	IBM_IE_MSGSEQNO
RETAIN – retain time	IBM_IE_RETAIN
DESCRIPTION - description	IBM_IE_DESCRIPTION

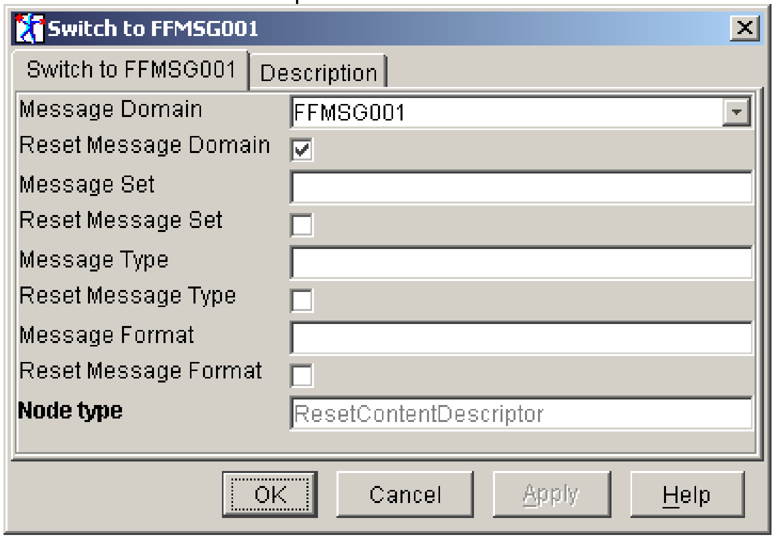
WebSphere MQ Integrator flow



Compute node “RFH2 to MQSIE”

```
SET OutputDestinationList.Destination.MQDestinationList.Defaults.queueManagerName =
'MQ31';

SET OutputDestinationList.Destination.MQDestinationList.DestinationData.queueName =
InputRoot.MQRFH2.usr."IBM_IE_TO";

SET OutputRoot.MQMD.Format = 'MQSIE';
SET OutputRoot.MQRFH2 = NULL;

SET OutputRoot.MQSIE.DATATYPE    = InputRoot.MQRFH2.usr."IBM_IE_DATATYPE";
SET OutputRoot.MQSIE.CLASS       = InputRoot.MQRFH2.usr."IBM_IE_CLASS";
SET OutputRoot.MQSIE.DESCRIPTION = InputRoot.MQRFH2.usr."IBM_IE_DESCRIPTION";
SET OutputRoot.MQSIE.MSGNAME     = InputRoot.MQRFH2.usr."IBM_IE_MSGNAME";
SET OutputRoot.MQSIE.MSGSEQNO    = InputRoot.MQRFH2.usr."IBM_IE_MSGSEQNO";
SET OutputRoot.MQSIE.RETAIN      = InputRoot.MQRFH2.usr."IBM_IE_RETAIN";

SET OutputRoot."BLOB"."BLOB" = BITSTREAM(InputBody);
```

Node ResetContentDescriptor “Switch to FFMSG001”



Message Domain	FFMSG001
Reset Message Domain	☑
Message Set	
Reset Message Set	☐
Message Type	
Reset Message Type	☐
Message Format	
Reset Message Format	☐
Node type	ResetContentDescriptor

Mapping MQSIE header fields onto JMS fields (incoming messages)

Each message sent from Information Exchange to Exchange for WebSphere MQ, the Report[1] on the destination queue is checked to determine if the recipient wants an MQSIE header to

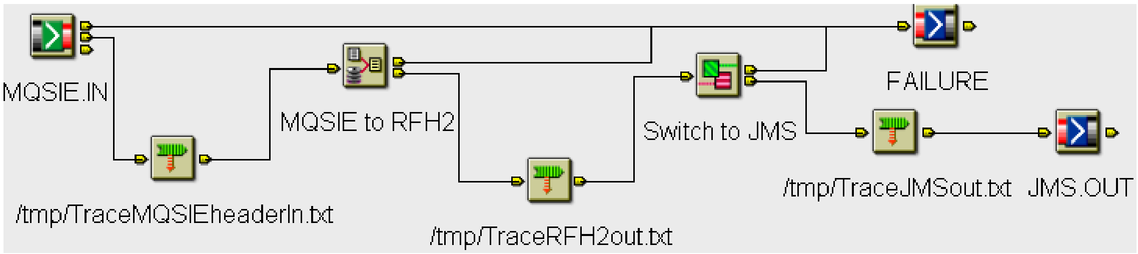
---

[1] Using the Administration Interface, you can enable the **Include report in messages from Information Exchange** option; the bridge includes an MQIE header in any message it sends to the specified queue.

be prefaced to the message. The table below shows how JMS Exchange for WebSphere MQ-defined properties that are mapped to the MQSIE fields.

MQSIE field	JMS MQSS specific property
DATATYPE – datatype	IBM_IE_DATATYPE
CLASS – class	IBM_IE_CLASS
CONTROLNUM – interchangecontrolnumber	IBM_IE_CONTROLNUM
MSGNAME – msgname	IBM_IE_MSGNAME
MSGSEQNO – msgseqno	IBM_IE_MSGSEQNO
RECLEN – record length	IBM_IE_RECLEN
RECDLM – record delimiter	IBM_IE_RECDLM
RECFM – record format	IBM_IE_RECFM
SENDER – EDI sender ID	IBM_IE_SENDER
SENDQUAL – EDI sender qualifier	IBM_IE_SENDQUAL
RECEIVER – EDI receiver ID	IBM_IE_RECEIVER
RECVQUAL – receiver EDI qualifier	IBM_IE_RECVQUAL
FROM – SYS.ACCOUNT.USERID	IBM_IE_FROM
UNIQUEID – uniqueid	IBM_IE_UNIQUEID
DATE – date	IBM_IE_DATE
TIME – time	IBM_IE_TIME

WebSphere MQ Integrator flow

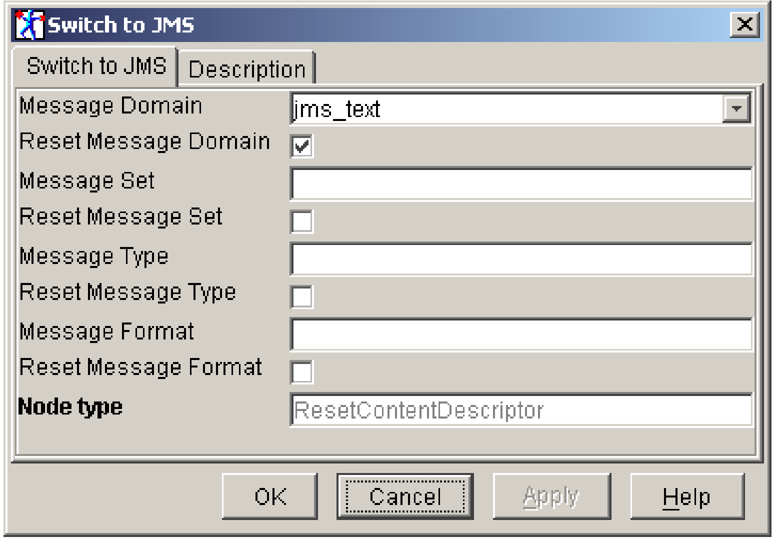


Compute node “MQSIE to RFH2”

```
SET OutputRoot.MQMD.Format = 'MQHRF2  ';
SET OutputRoot.MQRFH2.(MQRFH2.Field)Format = 'MQSTR   ';

SET OutputRoot.MQRFH2.jms.Dst = 'queue:///' || InputRoot.MQMD.SourceQueue;

SET OutputRoot.MQRFH2.usr."IBM_IE_CLASS"    = InputRoot.MQSIE."CLASS";
SET OutputRoot.MQRFH2.usr."IBM_IE_DATATYPE" = InputRoot.MQSIE."DATATYPE";
SET OutputRoot.MQRFH2.usr."IBM_IE_RECLEN"   = CAST(InputRoot.MQSIE."RECLEN" AS CHAR);
SET OutputRoot.MQRFH2.usr."IBM_IE_RECDLM"   = InputRoot.MQSIE."RECDLM";
SET OutputRoot.MQRFH2.usr."IBM_IE_RECFM"    = InputRoot.MQSIE."RECFM";
SET OutputRoot.MQRFH2.usr."IBM_IE_FROM"     = InputRoot.MQSIE."FROM";
SET OutputRoot.MQRFH2.usr."IBM_IE_UNIQUEID" = InputRoot.MQSIE."UNIQUEID";
SET OutputRoot.MQRFH2.usr."IBM_IE_DATE"     = SUBSTRING(CAST(InputRoot.MQSIE."DATE" AS
CHAR) FROM 7 FOR 10);
SET OutputRoot.MQRFH2.usr."IBM_IE_TIME"     = SUBSTRING(CAST(InputRoot.MQSIE."TIME" AS
CHAR) FROM 7 FOR 8);
```

```
SET OutputRoot.MQSIE     = UNKNOWN;
SET OutputRoot.FFMSG001 = UNKNOWN;
SET OutputRoot.XML       = UNKNOWN;

SET OutputRoot."BLOB"."BLOB" = BITSTREAM(InputBody);
```

Node ResetContentDescriptor “Switch to JMS”



Switch to JMS

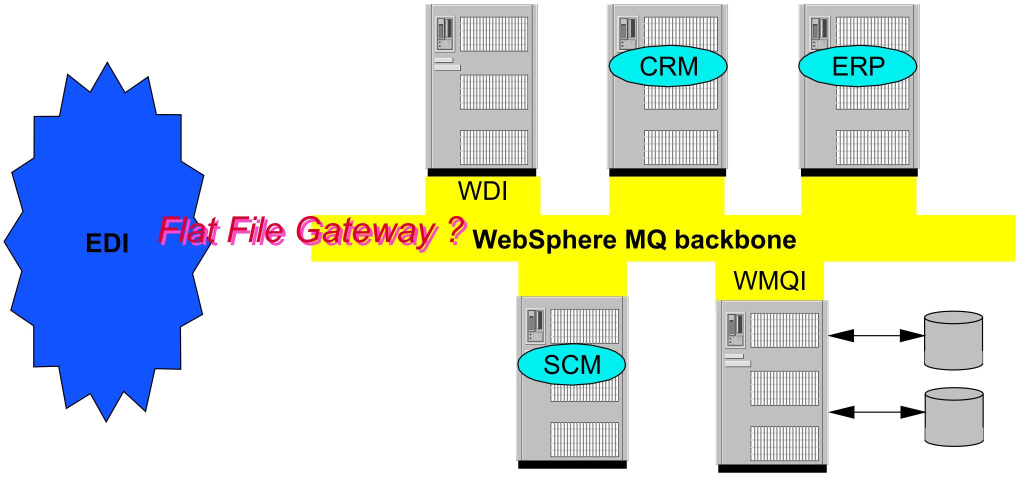
Message Domain	jms_text
Reset Message Domain	☑
Message Set	
Reset Message Set	☐
Message Type	
Reset Message Type	☐
Message Format	
Reset Message Format	☐
Node type	ResetContentDescriptor

Appendix A: Exchange for WebSphere MQ

IBM Exchange for WebSphere MQ provides the network, protocol conversion, message routing, and storage to support secured transaction processing among WebSphere MQ and EDI trading partners. A single wide area networking or remote access connection to the service eliminates the need to connect individually with trading partners, while providing a security buffer from them. Transactions can be routed immediately or saved for later retrieval. A Java-based administration tool is provided for enhanced management of queues and trading partner definitions.

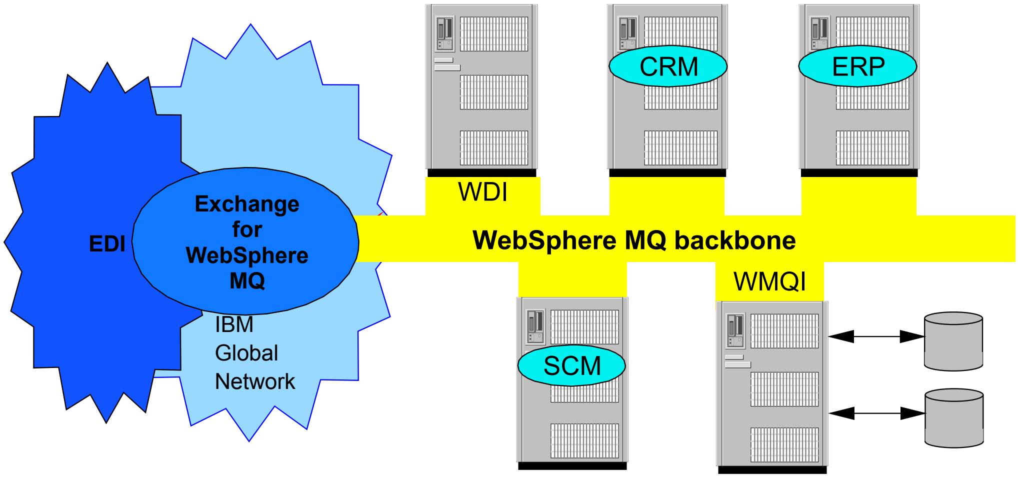
For further information: http://edi.services.ibm.com/mqseries/spec_sheets.shtml

Traditional EDI with a flat file gateway



WMQI	WebSphere MQ Integrator V2.x
WDI	WebSphere Data Interchange

Connecting to EDI using Exchange for WebSphere MQ and a WMQ backbone



WMQI	WebSphere MQ Integrator V2.x
WDI	WebSphere Data Interchange

End of Document