

Web Server Integration with IBM WebSphere® MQ Integrator

Revision : 1.0
Date: 25 March 2002

Author : Dirk Grosskopf
New Era Of Networks
A Sybase Company

1 Abstract

This Document is a "how to" for using IBM WebSphere MQ Integrator and IBM WebSphere MQ capabilities for a web server integration without any coding. The first part of this document explains the fundamental architecture, followed by the second part, a detailed description of the configuration. The last part explains alternatives and compares them with this solution.

2 Introduction

Very often, companies need a simple integration of existing application, such as SAP, PeopleSoft, Baan, Siebel, .. to a simple web front-end. Existing solutions, provided by the application providers, are often too large or need additional maintenance. Compared to the benefit, these costs are often too high.

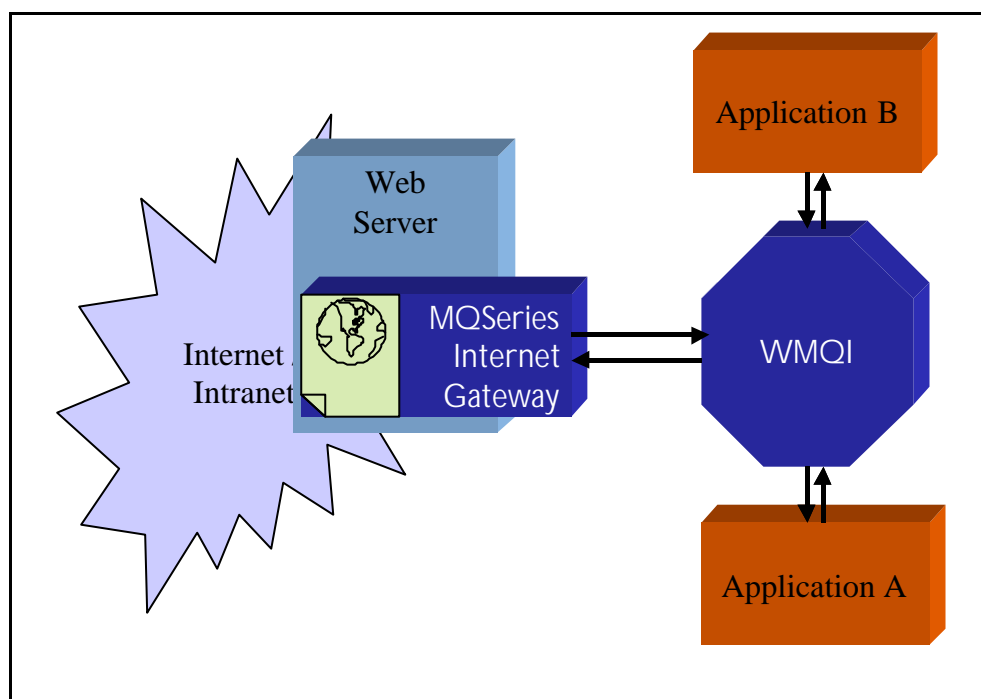
An alternative, used very often by companies, are self-written integration. The initial cost for this integration is low, but there's a potential risk of missed timelines and fundamental code changes are needed, if changes are done to the existing environment or web front-end.

Using an integration server, such as WebSphere MQ Integrator, reduces the amount of maintenance to a minimum. For a detailed description on how to determine the ROI see [1]. One major outcome of this document is, "Application development is in a long term cost intensive". This document will show you how to do a simple web integration without the long term investment of application development.

3 The Architecture

The MQSeries Internet Gateway is part WebSphere MQ on all major platforms. It is an optional package, so very often it is not installed. The gateway is a CGI plug-in and can be used in every HTML form. Once the "Send" or "Submit" button in the HTML form is pressed, the internet server starts the gateway and passes all information entered in the form, to the gateway.

The MQSeries Internet Gateway put this Information onto a queue to the WebSphere MQ Integrator and waits for a reply message. This reply message is an HTML page that is presented by the web server after its arrival.



A WebSphere MQ Integrator broker picks up the message from the gateway and reformats it into any number of multiple suitable application formats. The WebSphere MQ Integrator broker also sends a reply message to the waiting MQSeries Internet Gateway. Either this message is generated when the reformatted message is sent to the application (confirmation on delivery) or the message is generated when the application sends a confirmation back to the broker (confirmation on process).

This solution only uses existing components of WebSphere MQ Integrator and WebSphere MQ and needs no extra coding. It is suitable for low volume, because the web server will spawn a process for each CGI call.

4 Step By Step Example

The example simply picks the MQSeries Internet Gateway message and generates a reply message, using information entered in the form. It also logs all entered information into a trace file.

4.1 Pre Requisites

Install the MQSeries Internet Gateway on the web server machine. Configure and check the gateway as described in the manual. For initial tests, use the provided examples.

Install and configure WebSphere MQ Integrator. Create at least the configuration manager and one broker. Create the following queues on the broker queuemanager :

Queue name	Description
CUSTOMER.Request	The gateway puts the messages on this queue
DMQGATEWAY	Reply queue for the gateway
MQSI.FAILURE	The broker will store messages that could not be processed on this queue

4.2 A Simple HTML Form

The screenshot shows a web browser window titled "New Era Of Networks Sales System - Microsoft Internet Explorer". The address bar shows "http://drlg-test2/wmqsnip/SiebelFront.html". The page content includes the SYBASE logo and the title "New Era Of Networks Sales System". Below the title, a message states: "This Form allows you to create or delete a customer in the Siebel and SAP system. After creation of a new customer, the SAP customer number will be returned." The main section is titled "Customer View" and contains a form with the following fields and controls:

- Action: ☒ Insert Record ☐ Delete Record
- Company name:
- Location:
- Status: ☒ Active ☐ Passive
- Address:
- City:
- US-State or country:
- Zip:
- Phone number:
- Home Page:
- Submit Cancel

The HTML form allows a user to enter customer information and send it to a pseudo sales system. There are two possible actions : create a new record or delete an existing record. Action is controlled by a radio button. Here is the source code for the web page.

```

<HEAD>
<TITLE>New Era Of Networks Sales System</TITLE>
</HEAD>
<body bgcolor="#ffffff" background="dmqdg08.gif">

<A NAME=Top_Of_Page>&nbsp;</A>
<H2> New Era
Of Networks Sales System</H2>
<p>This Form allows you to create or delete a customer in the Siebel and SAP
system. After creation of a new customer, the SAP customer number will be returned.
</p>
<HR>
<center><h4>Customer View</h4></center>

<FORM ACTION="/dmq-bin/dmqcgi" METHOD="POST"
<!-- TRNOTE: Please translate the following comment. -->
<!-- If you are using a Domino or an Apache server, the ACTION must be changed to:
"/dmq-bin/dmqcgi.exe" otherwise use "/dmq-bin/dmqsample.mpf" -->

<INPUT NAME="MQIGwQueue" TYPE="hidden" VALUE="CUSTOMER.REQUEST">
<INPUT NAME="MQIGwWaitInterval" TYPE="hidden" VALUE="10000">
<font size=+1><CENTER>
<TABLE BORDER=1 CELLPADDING=1>

<TR><TD><P>Action:</TD><TD>
<Input type="radio" Name="Action" Value="InsertRecord" checked="checked">Insert
Record
<Input type="radio" Name="Action" Value="DeleteRecord">Delete Record<BR></TD></TR>

<TR><TD><P>Company name:</TD><TD>
<Input type="text" Name="CustomerName" Size="40"><BR></TD></TR>

<TR><TD><P>Location:</TD><TD>
<Input type="text" Name="Location" Size="40"><BR></TD></TR>

<TR><TD><P>Status:</TD><TD>
<Input type="radio" Name="Status" Value="Active" checked="checked" >Active
<Input type="radio" Name="Status" Value="Passive">Passive<BR></TD></TR>

<TR><TD><P>Address:</TD><TD>
<textarea name="Address" rows="3" cols="40"></textarea>
</right><br></TD></TR>
<TR><TD><P>City:</TD><TD>
<Input type="text" Name="City" Size="40"><BR></TD></TR>

<TR><TD><P>US-State or contry:</TD><TD>
<Input type="text" Name="State" Size="16"><BR></TD></TR>

<TR><TD><P>Zip:</TD><TD>
<Input type="text" Name="Zip" Size="10"><BR></TD></TR>

<TR><TD><P>Phone number:</TD><TD>
<Input type="text" Name="Phone" Size="40"><BR></TD></TR>

<TR><TD><P>Home Page:</TD><TD>
<Input type="text" Name="Home-Page" Size="40"><BR></TD></TR>
<TR><TD Align=RIGHT><p><RIGHT><INPUT TYPE="submit" VALUE="Submit"></RIGHT></TD>
<TD><INPUT TYPE="reset" VALUE="Cancel"></td><TR>

</TABLE></CENTER>
</font>
</P>
</FORM>
<hr>
</BODY>
</HTML>

```

If you are using a web sever other than Apache, Domino or WebSphere, change the post method in html form as described in the form.

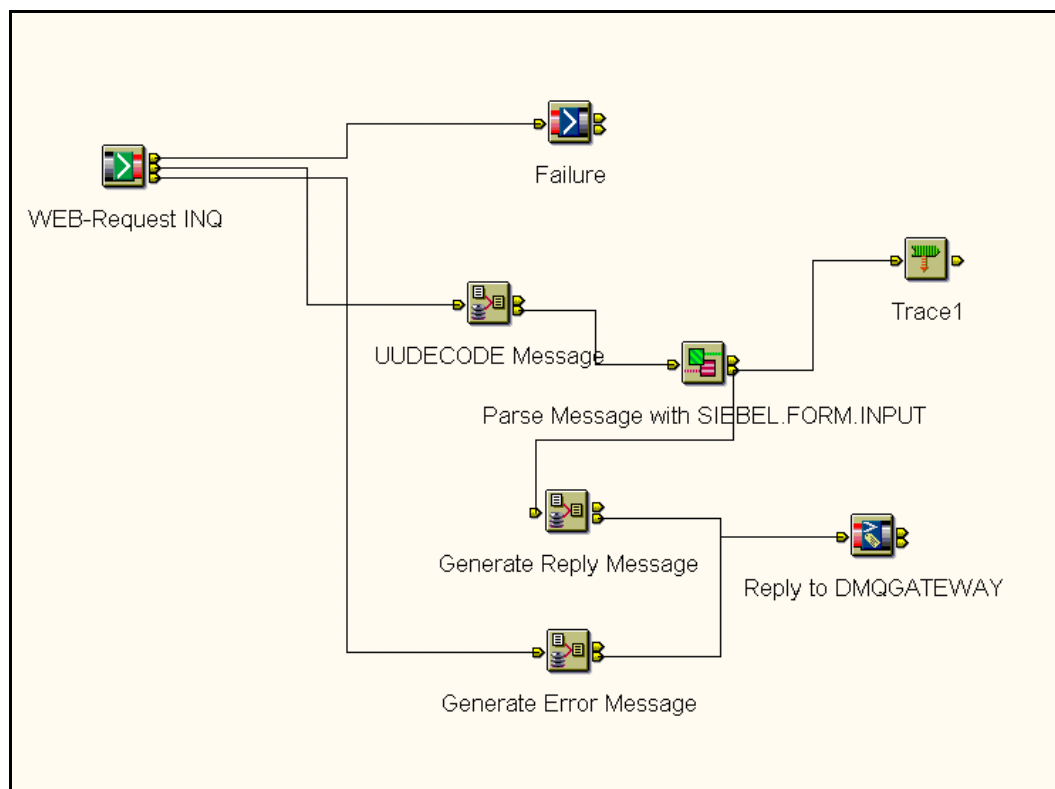
Also change the configuration of the Internet Gateway Server (dmq.ini) and set MQIGWContext to none.

Make sure, that the gateway sends the message to the broker queue manager. If not, change the client channel definition table (AMQCLCHL.TAB).

4.3 The WebSphere MQ Integrator Message Flow

In WebSphere MQ Integrator create a new message flow. Use the following nodes

Name	Type	Properties	Output wired to
WEB-Request INQ	MQInput node	Queue Name = CUSTOMER.REQUEST Message Domain = BLOB	Out to UUDECODE Message Catch to Generate Error Message Failure to Failure
Failure	MQOutput node	Queue Name = MQSI.FAILURE	
Reply to DMQGATEWAY	MQReply node		
UUDECODE Message	Compute node	See the ESQL scripts	Out to Parse Message with SIEBEL.FORM.INPUT
Parse Message with SIEBEL.FORM.INPUT	ResetContentDescriptor node	Message Domain = NEONMSG Message Type = SIEBEL.Form.Input	Out to trace 1 Out to Generate Reply Message
Generate Reply Message	Compute node	See the ESQL scripts	Out to Reply to DMQGATEWAY
Generate Error Message	Compute node	See the ESQL scripts	Out to Reply to DMQGATEWAY
Trace 1	Trace node	Description = File Pattern = \${Root}	



The input node "Web-Request INQ" reads messages generated by the MQSeries Internet Gateway.

These messages comes in a format that is not easy manageable by the WebSphere MQ Integrator parse, so the "UUDECODE Message" transforms the content into a suitable format.

The "Parse Message with SIEBEL.FORM.INPUT" invokes the New Era Of Networks Parser to parse the message. The content is traced afterwards and used in the "Generate Reply Message" node.

The "Generate Reply Message" node generates a confirmation message for the gateway in form of an HTML message and passes this message to the "Reply to DMOGATEWAY" node. This node sends a reply message to the gateway.

If any exception occurs, the catch terminal of the "Web-Request INQ" node passes the original message to the "Generate Error Message" node.

The "Generate Error Message" discards the whole message and generates an error reply message for the gateway.

4.3.1 ESQL scripts

UUDECODE Message

```
DECLARE I INTEGER;
SET I = 1;
WHILE I < CARDINALITY(InputRoot.*[]) DO
    SET OutputRoot.*[I] = InputRoot.*[I];
    SET I=I+1;
END WHILE;
-- Geben Sie den SQL-Ausdruck unterhalb dieser Linie ein. SQL-Ausdrücke oberhalb
dieser Linie werden unter Umständen erneut generiert und führen zu einem Verlust
eventuell vorgenommener Änderungen.
DECLARE tmp CHARACTER;
DECLARE msgbuffer CHARACTER;
DECLARE chr CHARACTER;
DECLARE t CHARACTER;
DECLARE J INTEGER;

SET I=3;                                     -- Do not start with the first
Character, because this is not needed
SET msgbuffer = CAST (InputRoot."BLOB"."BLOB" as CHARACTER);      -- copy msg as Hex
SET OutputRoot."BLOB"."BLOB" = '';          -- Start with a empty
message

WHILE I < LENGTH (msgbuffer) DO
    SET tmp = SUBSTRING( msgbuffer FROM I for 2);      -- Read one char as Hex
    SET tmp =                                          -- Convert special chars
        CASE
            WHEN tmp = '26' THEN '7C'                -- transform & to |
            WHEN tmp = '2B' THEN '20'                -- transform + to space
            WHEN tmp = '2b' THEN '20'                -- transform + to space
            WHEN tmp = '00' THEN '7C'                -- end of msg is a 00
        else tmp
    END;
    IF tmp='25' THEN
```

```

        SET chr=SUBSTRING(msgbuffer FROM I+2 for 4);
        SET I = I + 4;
        SET J = 1;
        SET tmp= '' ;

        WHILE J<5 DO
            SET t = SUBSTRING( chr from J for 2);
            SET tmp = tmp || CASE
                WHEN t = '30' THEN '0'
                WHEN t = '31' THEN '1'
                WHEN t = '32' THEN '2'
                WHEN t = '33' THEN '3'
                WHEN t = '34' THEN '4'
                WHEN t = '35' THEN '5'
                WHEN t = '36' THEN '6'
                WHEN t = '37' THEN '7'
                WHEN t = '38' THEN '8'
                WHEN t = '39' THEN '9'
                WHEN t = '41' THEN 'A'
                WHEN t = '42' THEN 'B'
                WHEN t = '43' THEN 'C'
                WHEN t = '44' THEN 'D'
                WHEN t = '45' THEN 'E'
                ELSE 'F'
            END;
            SET J = J+2;
        END WHILE;
    END IF;
    SET I = I+2;
    IF I = 5 THEN
        SET OutputRoot."BLOB"."BLOB" = CAST(tmp as BLOB); -- construct message
    ELSE
        SET OutputRoot."BLOB"."BLOB" = OutputRoot."BLOB"."BLOB" || CAST(tmp as
BLOB); -- construct message
    END IF;
END WHILE;

-- finally we have to change the codeset of the message, because it's 819 (Latin 1
ISO 8859-1)
SET OutputRoot.MQMD.CodedCharSetId = 819;

-- that's all !

```

Generate Reply Message

```

DECLARE I INTEGER;
SET I = 1;
WHILE I < CARDINALITY(InputRoot.*[]) DO
    SET OutputRoot.*[I] = InputRoot.*[I];
    SET I=I+1;
END WHILE;
--SET "OutputRoot"."NEONMSG"."STRING" = "InputBody"."STRING";
-- Geben Sie den SQL-Ausdruck unterhalb dieser Linie ein. SQL-Ausdrücke oberhalb
dieser Linie werden unter Umständen erneut generiert und führen zu einem Verlust
eventuell vorgenommener Änderungen.
DECLARE HTML CHARACTER;
SET HTML = '<p>Your request ';

IF (InputRoot."NEONMSG".Action = 'InsertRecord') THEN
    SET HTML = HTML || ' <strong>Insert New Customer</strong> ';
ELSE
    SET HTML = HTML || ' <strong>Delete Customer</strong> ';
END IF;

SET HTML = HTML || 'has been sent to the sales system</p>';
SET HTML = HTML || ' <p>Customer name : <strong>' || InputRoot."NEONMSG".CName ||
'</strong></p>';

```

```

SET OutputRoot."NEONMSG"."STRING"= HTML;
SET OutputRoot.Properties.MessageDomain = 'NEONMSG';
SET OutputRoot.Properties.MessageType = 'HTML.OutMsg';
SET OutputRoot.MQMD.MsgType = 2;

```

Generate Error Message

```

DECLARE I INTEGER;
SET I = 1;
WHILE I < CARDINALITY(InputRoot.*[]) DO
    SET OutputRoot.*[I] = InputRoot.*[I];
    SET I=I+1;
END WHILE;
--SET "OutputRoot"."NEONMSG"."STRING" = "InputBody"."STRING";
-- Geben Sie den SQL-Ausdruck unterhalb dieser Linie ein. SQL-Ausdrücke oberhalb
dieser Linie werden unter Umständen erneut generiert und führen zu einem Verlust
eventuell vorgenommener Änderungen.
DECLARE HTML CHARACTER;
SET HTML = 'An error was dedected while processing your request. Your request was
not sent to the sales system. Please check your entered information and try it
again.';
SET OutputRoot."NEONMSG"."STRING"= HTML;
SET OutputRoot.Properties.MessageDomain = 'NEONMSG';
SET OutputRoot.Properties.MessageType = 'HTML.OutMsg';
SET OutputRoot.MQMD.MsgType = 2;

```

4.4 New Era Of Networks Formats

Due to limitations in MRM and BLOB, it is much easier to use several simple New Era formats for message parsing & transformation. Start the New Era Of Networks formatter GUI to define the formats.

Step one

Create the following literals. Replace '~' with the Hex Value '0a'.

Name	Value
HTML.Line1	Content-type:text/html~~
HTML.Line2	<html>~~
HTML.Line3	<head>~~
HTML.Line4	<title>New Era Of Networks Sales System </title>~~
HTML.Line5	<head>~~<body>~~
HTML.Line6	<H2>your request has been received by the sales system</H2>~~
HTML.Line7	<hr>~~
HTML.Line9	</body>~~
HTML.LineA	</html>~~
SIEBEL.Form.Delimeter	
SIEBEL.Form.Tag.Address	Address=
SIEBEL.Form.Tag.City	City=
SIEBEL.Form.Tag.COName	CustomerName=
SIEBEL.Form.Tag.HomePage	Home-Page=
SIEBEL.Form.Tag.Location	Location=
SIEBEL.Form.Tag.Mode	Action=
SIEBEL.Form.Tag.Phone	Phone=
SIEBEL.Form.Tag.State	State=
SIEBEL.Form.Tag.Status	Status=
SIEBEL.Form.Tag.Zip	Zip=

Step two

Create the following Fields

STRING, Address, City, COname, HomePage, Location, Mode, Phone, State, Status, Zip

Step three

Create the following Output Controls :

HTML.Linux with x=1,2,3,4,5,6,7,9,A

Set the following properties of the output controls :

Property	Value
Output Control Name:	HTML.Linux
Output Control Type	Literal
Output Operation	NONE
Data Type	String
Length Type	Not Applicable
Tag Type	Not Applicable
Field Value	HTML.Linux

Step four

Create the following Input Controls :

SIEBEL.Form.Address, SIEBEL.Form.City, SIEBEL.Form.COname, SIEBEL.Form.HomePage, SIEBEL.Form.Location, SIEBEL.Form.Mode, SIEBEL.Form.Phone, SIEBEL.Form.Status, SIEBEL.Form.State, SIEBEL.Form.Zip. Use for each of these Input Controls the following properties :

Property	Value
Input Control Name:	SIEBEL.Form.x
Control Type	Tag and Data
Data Type	String
Data Termination	Delimiter
Data Delimiter	SIEBEL.Form.Delimiter
Tag Type	String
Tag Value	SIEBEL.Form.Tag.x
Tag Termination	Not Applicable

Step five:

Create the input format "SIEBEL.Form.Input". This is a flat input format.

Add all fields, except the "STRING" Field.

Check "Random Field Order" in the "Properties" tab.

Associate with each field the input control with the equivalent name, e.g.: Field name = "Address", input control name= "SIEBEL.Form.Address"

Step six:

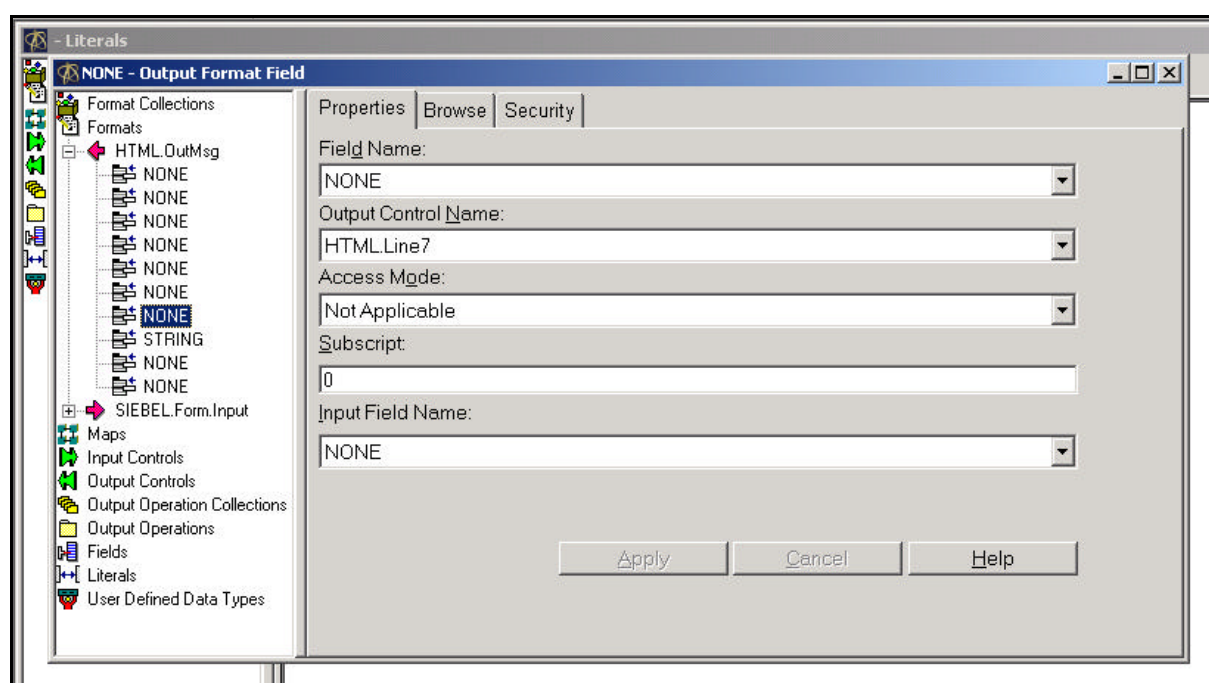
Create the output format "HTML.OutMsg". This is also a flat input format.

Add the field "STRING" 10 times.

Under the properties tab of each field in the message, change the properties of all STRING occurrences, except the 8th to the following values:

Property	Value
Field Name	NONE
Output Control Name	HTML.Line7 with x=1..9,A
Access Mode	Not Applicable
Subscript	0
Input Field Name	NONE

After this step, the output format should look like this :



Now you have finished the configuration. Restart the broker after making changes to the New Era Of Network formats.

5 Other Methods For Integration

Other methods for web integration are Java Server Pages or Java applets with JMS in HTML pages. These methods always require Java coding. Also changes in the integration logic will require code changes, so more maintenance is required.

The given solution with MQSeries Internet Gateway only allows interaction via simple forms and is suitable for low volume integration because of use of the CGI-capability.

6 References

- [1] Gartner Research: Determining Return on Investment for Integration Brokers, 1 June 2001, R. Altman
- [2] IBM WebSphere MQ Integrator Version 2.1 : Installation Guide
- [3] IBM WebSphere MQ Integrator Version 2.1 : Using The Control Center
- [4] IBM WebSphere MQ Integrator Version 2.1 : ESQL Reference
- [5] IBM WebSphere MQ Integrator Version 2.1 : New Era Of Networks User's Guide
- [6] IBM WebSphere MQ Integrator Version 2.1 : New Era Of Networks System Management Guide

7 Trademarks

IBM WebSphere MQ and IBM WebSphere are trademarks of International Business Machines Corporation.

New Era of Networks is a trademark of New Era of Networks, Inc., a Sybase company.

Sybase and the SYBASE (logo) are trademarks of Sybase, Inc.

SAP is a trademark of SAP AG Walldorf

Siebel is a trademark of Siebel Systems, Inc.

All other product names are copyright and/or registered trademarks/tradenames of their respective owners