



SWIFT Partners

# Alliance Access Integration – MQ Host Adaptor

## Technical Qualification Test 2011

This document lists the tests for application providers that integrate their back-office application or middleware with Alliance Access using MQ Host Adaptor and that are looking at qualifying against the SWIFTRReady label qualification.

Version 2

September 2011

# Legal Notices

## Copyright

SWIFT © 2011. All rights reserved.

You may copy this publication within your organisation. Any such copy must include these legal notices.

## Disclaimer

SWIFT supplies this publication for information purposes only. The information in this publication may change from time to time. You must always refer to the latest available version.

## Translations

The English version of SWIFT documentation is the only official version.

## Trademarks

SWIFT is the trade name of S.W.I.F.T. SCRL. The following are registered trademarks of SWIFT: SWIFT, the SWIFT logo, Sibos, SWIFT, SWIFTReady, and Accord. Other product, service, or company names in this publication are trade names, trademarks, or registered trademarks of their respective owners.

# Table of Contents

<b>1</b>	<b>Preface</b> .....	<b>3</b>
1.1	Purpose.....	3
1.2	Audience.....	3
1.3	SWIFTReady Programme.....	3
1.4	Related Documents.....	3
<b>2</b>	<b>Introduction</b> .....	<b>4</b>
2.1	Migrating from MQSA to MQHA.....	4
2.2	Testing MQHA.....	4
<b>3</b>	<b>Alliance Configuration</b> .....	<b>5</b>
3.1	Prerequisites.....	5
3.2	Message Partner configuration.....	5
3.2.1	Migrating an MQ Queue to Message Partner Profile.....	5
3.2.2	Configuring routing of messages.....	6
3.3	Exchanging messages using Message Partner.....	6
<b>4</b>	<b>Wipro Testing Service Configuration</b> .....	<b>6</b>
<b>5</b>	<b>SWIFTReady Qualification Requirements</b> .....	<b>7</b>
5.1	Supported formats.....	7
5.2	Reconciliation of Delivery Notification.....	7
5.3	Transmission Notification.....	7
5.4	Alliance Access Info.....	7
5.5	S-Block in MQ-MT Format.....	8
5.6	Testing setup.....	8
<b>6</b>	<b>System setup</b> .....	<b>9</b>
<b>7</b>	<b>Test Evidence</b> .....	<b>9</b>
<b>8</b>	<b>Test Execution and Test result</b> .....	<b>10</b>
8.1	Sending test messages to Alliance Access.....	10
8.2	Collecting the results.....	10
<b>9</b>	<b>Support</b> .....	<b>10</b>
<b>10</b>	<b>Annex</b> .....	<b>11</b>
10.1	Checklist for MQHA Connectivity Test.....	11
10.2	Sample Message Partner Configuration.....	12
10.3	Input MT Message sample in MQ-MT format.....	20
10.4	Input MT Message sample in XML v2 format.....	20
10.5	Output MT Message sample in MQ-MT format.....	21
10.6	Delivery Notification Message.....	21
10.7	ACK Message with Original Message.....	21
10.8	ACK Message without Original Message.....	22

# 1 Preface

## 1.1 Purpose

This document describes the test scenarios that a financial application has to pass to be compliant with Alliance Access using MQ Host Adaptor (MQHA).

The purpose of the MQHA qualification is to verify the capacity of an application to correctly integrate with SWIFTNet FIN and InterAct messaging services via Alliance Access MQ Host Adapter.

## 1.2 Audience

The target audience for this document is both Partners considering the certification of a product, and SWIFT Users that look after an overview of the SWIFTReady label contents. The audience should be familiar with SWIFT portfolio from a technical and a business perspective.

## 1.3 SWIFTReady Programme

The SWIFTReady label programme covers the entire financial application chain, from Trade, Treasury and Payment, to Corporate and Securities segments.

Each SWIFTReady label defines a set of criteria, which are reviewed every year to ensure that the software remains aligned with the financial market evolution and with customer needs.

These criteria are designed to reflect the capability of a financial application to provide message processing automation in a SWIFT context, and to support straight through processing (STP) in order to increase customer value, limit customisation needs and cost, and reduce time to market.

## 1.4 Related Documents

The following documents can be found in SWIFT User Handbook:

- [WebSphere MQ Interface - For Alliance Access 7.0 Migrating to the MQ Host Adapter](#)
- [Alliance Access 7.0 - System Management Guide](#).

## 2 Introduction

Alliance Access supports two interfaces for exchanging SWIFT messages with back office applications through IBM WebSphere MQ:

- WebSphere MQ Interface for Alliance Access software application (referred to as **MQSA** in this document), which is built using functions of the Alliance Developer Kit (ADK).
- Alliance Access MQ Host Adapter (referred to as **MQHA** in this document), which is embedded in the Application Interface. It does not require the installation of any ADK software.

Alliance Access comes with a licensable option that supports interactive communications between Alliance Access and IBM's WebSphere MQ. The WebSphere MQ middleware uses a central message queuing mechanism to temporarily store messages from data providers that can be picked up by a data consumer at its own speed, while introducing a transparency layer for operating system and communications protocol.

The MQHA is using the standard mechanism of message partner definition available in Alliance Access, including routing rules and profile definition. Each WebSphere MQ queue is associated to a defined message partner, and linked to a security profile. Routing rules must also be defined to integrate each MQ based message partner inside the Alliance Access routing scheme. MQHA functionality is aligned with the way all other message partner works in Alliance Access

### 2.1 Migrating from MQSA to MQHA

MQSA will be decommissioned with Alliance Release 7. Many Partner applications that are currently supporting MQSA need to migrate to the new WebSphere MQ adapter (MQHA). To facilitate the migration process, MQSA and the new MQ Host Adapter can co-exist in Alliance Access. This co-existence can start as early as Alliance Access release 6.2. **MQSA will no longer be supported after the deployment of Alliance Access 7.0.**

As there are some differences between the messages sent by MQSA and those sent by the embedded MQ Host Adapter, it is necessary to analyse the impact, and where needed change the back office systems to allow the exchange of messages with the embedded MQ Host Adapter.

This migration can be performed gradually, per MQ queue. MQSA and MQHA can run in parallel until the migration is complete. In order to exchange communication using MQSA, the hardware hosting the Alliance Access system must have a dual-core processor and 4GB RAM memory.

### 2.2 Testing MQHA

Support of MQHA is one of the qualification criteria for granting the SWIFTReady label to a financial application and EAI.

Partners are proposed two paths to qualify their applications against MQHA compliance:

1. Partners purchase SWIFT Alliance pack from the [Developer Resource Center](#), deploy it at their premises and configure MQHA to connect their applications to SWIFT Integration Testbed (ITB). The Section 3 details the required Alliance configuration.
2. Partner subscribe to Wipro Testing Services to connect to SWIFT ITB to exchange messages and files. Currently, Wipro Testing Services allow testing application connectivity with Alliance Access AFT and MQHA. The section 4, details the connection set-up required for this service.

## 3 Alliance Configuration

This section is applicable for Partners that deployed an ITB environment and connect to Alliance Access **from their premises**.

### 3.1 Prerequisites

To prepare for the tests, the Alliance Access system must be installed and configured at Partner premises. The following upgrade / migration must be performed before attempting to communicate through MQHA:

- Alliance Access must be upgraded to release 7.0
  - Additional licence **13:AI MQS Adapter** is required for MQHA
  - For exchanging XML messages **19:AI FILE XML**, license is needed
- If Migrating from MQSA to MQHA, MQSA 6.0.0. must be migrated to 6.2.0 or 6.3 and then migrate to Alliance 7.0

It is necessary to get acquainted with the Alliance Access 7.0 System Guide to further configure the interface for test purpose. The Application Interface module of Access provides all the functions necessary to manage message partner profiles. Using Application Interface application, the Partner needs to set up the connection profiles that are used by Alliance Access to connect with external message partners. The Application Interface allows exchanging messages with external back-office systems or "message partners".

### 3.2 Message Partner configuration

The MQHA communication session is set up and controlled with a dedicated Message Partner configuration in Alliance Access. Using the procedure described in the Alliance Access System Management Guide – WebSphere MQ Connection Method, create a Message Partner for the MQ queue.

- Specify the connection method as WebSphere MQ
- Specify the direction of message transfer (from Message Partner and To Message Partner)
- Configure the other parameter

Alternatively, follow the steps provided in section 3.21 and 3.2.2 to migrate MQSA MQ Queue to MQHA Message Partner.

#### 3.2.1 Migrating an MQ Queue to Message Partner Profile

To migrate from MQSA to MQHA configuration, every MQ queue must be re-configured as follows:

- create a message partner profile with the Connection Method "WebSphere MQ"
- specify the direction of message transfer and
- configure the profile.

For detailed procedure for creating message partner profile for every MQ Queue, please refer to the procedure described in the System Management Guide – WebSphere MQ Connection Method, create a message partner for the MQ queue.

The Partner must ensure that:

- The Alliance Access Server must be running and the message partner enabled
- Alliance Access must be able to connect the Queue Manager. The connection is made through WebSphere MQ environment variables (MQSERVER or MQCHLTAB and MQCHLLIB)

Please refer to System Management Guide – Alliance Access 7.0 or higher for additional information for configuring and managing Message Partner Profiles. A screenshot of sample Message Partner is provided in [section 10](#).

### 3.2.2 Configuring routing of messages

When the back-office application send message to Alliance Access, the SMQS component of the MQSA create messages in the SMQS\_From\_MQSeries routing point. This routing point is replaced by AI\_FROM\_APPLI queue in MQHA.

Similarly, the MQSA-ROUTINGCODE and APPL-ROUTINGCODE used in MQSA is replaced by an exit point assigned to a message partner. The routing rules must be recreated or changed to send the outgoing messages to these exit points.

For more information, refer to the System Management Guide – Classes of Configuration Parameters – WebSphere MQ.

### 3.3 Exchanging messages using Message Partner

- The MQHA supports both MT and MX messages. MT test messages can be exchanged using MQ-MT format and XML v2 format, while MX test messages can only be sent in XMLv2 format. The application provider will prepare and route the messages in to WebSphere MQ queue for Alliance Access to process.
- The messages being sent must be the message types supported by the partner application (See SWIFTReady criteria document)
- All messages entering Alliance through the Application Interface are queued at one single point of entry – **\_AI\_from\_APPLI** (AI Inbound Queue), before being routed onwards.

The successfully processed messages will be stored by Alliance Access in the **\_SI\_to\_SWIFT** Queue [MT messages) or **\_SI\_to\_SWIFTNet** (MX messages) Queue.

Messages should be sent to yourself (Sender BIC – Receiver BIC). SWIFT Network returns Notification messages for technical reconciliation and response messages, since the test messages were used for “self transfers”.

The application must download the Network notifications and messages sent in “Output from SWIFT” direction.

## 4 Wipro Testing Service Configuration

- You need to liaise with Wipro through Partner Management to enrol into Wipro Testing Services to exchange messages over this service.
- You will be identified as a branch of Wipro and accordingly, you will be provided with a Wipro Branch PIC (Partner ID Code).
- This PIC11 must be used in the sender and receiver block for exchanging messages over SWIFT ITB
- You need to configure the MQ Configuration parameter details provided by Wipro in your application
- The connectivity is very similar when you connect from within your environment, except for the reason that this connectivity is established outside your internal network environment and hence necessary permission need to be obtained upfront from your IT Security team for using Port 1414 (for MQ) over internet
- Once the connectivity is established, the outgoing messages can be pushed into the designated outbound MQ Queue.
- SWIFT Network returns Notification messages for technical reconciliation and response messages, since the test messages were used for “self transfers”, meaning, the sender and receiver BIC are the same.
- The application must download the Network notifications and messages sent in “Output from SWIFT” direction.
- SWIFT Network returns Notification messages for technical reconciliation and response messages, since the test messages were used for “self transfers”, meaning, the sender and receiver BIC are the same.
- The application must download the Network notifications and messages sent in “Output from SWIFT” direction.

## 5 SWIFTReady Qualification Requirements

### 5.1 Supported formats

MQHA supports the exchange of messages in the following data formats:

- MQ-MT (MT messages only)
  - MQHA supports ASCII character encoding only. The EBCDIC character encoding is not supported. Therefore, IBM WebSphere MQ must perform the conversion between EBCDIC and ASCII, if it is required. In that case, for messages being sent, the Format field of the MQ Descriptor must be set to the value MQFMT\_STRING. For messages received by the back-office application, the get message option MQGMO\_CONVERT must be used.
- XML version 2 (MX and MT messages, and files)
  - MQHA does not support XML version 1. The Partner must use either version 2.0.0 or 2.0.1 of XML version 2.

For more information about these formats, see the System Management Guide – a Message Formats. Sample messages of MT in [MQ-MT](#) format and [XML v2](#) format is provided in section 10.

**For SWIFTReady Label validation, at least one of the following file formats will be tested:**

- **MQ-MT for MT message**
- **XML v2 format for MT or MX messages**

### 5.2 Reconciliation of Delivery Notification

When the messages are sent to Alliance Access, the application can optionally request for a delivery notification. This will result in Alliance Access receiving a message about the message delivery, which can be reconciled with the original message.

In MQHA the reconciliation of delivery notification can be achieved in two ways:

- a) Delivery Notification system message contains MIR of the original message
- b) Alliance Access Traffic Reconciliation (TR\_REC)

**The Partner must demonstrate their application capability to process the Delivery Notification Message and reconcile with the original message.**

A sample Delivery Notification message is provided in [section 10](#).

### 5.3 Transmission Notification

A transmission notification is a message representing the result of transmission to SWIFT network. SWIFT performs full syntax and semantic checks before it returns an acknowledgement (ACK). Other checks, such as validity of the sender and the receiver, are also performed. These checks can cause a message to be rejected and a negative acknowledgement (NAK) is returned in response.

**The Partner must demonstrate their application capability to process the Transmission Notification Message and reconcile with the original message.**

### 5.4 Alliance Access Info

MQHA can transmit Alliance Access Info either in the MQ Message Data part or in the MQ Descriptor

When this information is transmitted in Message Data part, in MQ-MT format, the Alliance Access Info is transmitted in S-Block in XML v2 format, the Alliance Access Info is transmitted through the new Alliance Access Info element.

**The partner application must accept these new tags in S-Block and accepts the new Alliance Access Info XML block for XML v2 messages.**

## 5.5 S-Block in MQ-MT Format

MQHA always transmits the S-Block in MQ-MT format. The acknowledgement (ACK or NAK) is sent to the back office application with or without including the original message. S-Block is always placed at the end of the transmission notification.

ACK or NAK	MQHA Format
With Original Message	<ACK><MT-Blocks 1,2,3,4,5><S-block>
Without Original Message	<ACK><MT-Blocks 1,2,3,5><S-block>. (There is no Block 4)

The back office application need to capture the ACK or NAK received back from SWIFT and reconcile with the original message.

**The back office application should be able to handle the S-Block positioned at the end of the transmission notification message and the additional tags sent back by MQHA.**

A sample Acknowledgement message [with original message](#) and [without original message](#) is provided in section 10

After successful exchange of test messages, the test evidences of reconciliation mechanism (screen dump, event log, dataset extract,) should be sent by email to the Validation Service Provider.

## 5.6 Testing setup

To be qualified as MQHA compliant, Partners need to send and receive MT and/or MX messages. The application testing will be configured as follows:

1. The Partner application prepares the MT / MX message types required by the SWIFTReady Label. The "From" session is started and the test messages are created in the WebSphere MQ queue configured in Message Partner Profile in the Alliance Access server.
2. The required message types are label specific and can be found in the technical validation guide pertaining to the label at stake.
3. Partners having their own ITB environment must use their PIC in the sender and receiver fields of the message. The Partner using Wipro Testing Services for connecting to ITB, must use the PIC11 provided by Wipro in the sender and receiver field of the message.
4. If the business application supports only MT messages, the test messages can be sent in either ASCII format or XML v2 format. If the business application supports SWIFT Solutions, then the only format Alliance can read is XML v2 format.
5. The business application must include delivery notification instructions while generating the test messages. MQHA will transmit the delivery notification either through system message or through Traffic Reconciliation message. The back office application must receive the delivery notification information and reconcile with the original message sent to Alliance Access.
6. When transmitting MT messages in MQ-MT format, MQHA can be configured to return Access info either in the MQMD or in the S-Block with new tags. The business application must accept these new tags in S-Block. When using Wipro Testing Services for connecting to ITB, the message format and acknowledgement format need to be agreed in advance.
7. The acknowledgement of transmission is sent through transmission notification message. The back office application need to capture the ACK and NAK received back from SWIFT and reconciled in the back-office application. Evidences of reconciliation mechanism (screen dump, event log, dataset extract,) will be handled back to SWIFT Validation Service provider.

## 6 System setup

### For partners having ITB connectivity

- Alliance Access must have all events active (default configuration).
- The Alliance Access Server must be running and the Message Partner enabled
- The Partner needs to set up the connection profiles that are used by Alliance Access to connect with external message partners
- The Partner must ensure proper configuration of the session parameters in the message partner profile.
- The Partner must ensure that the routing and message partners are defined correctly

### For partners testing through Wipro Testing Service

- Alliance Access will be configured to exchange the message formats supported by the Partner application.
- The configuration details will be provided to the Partner before commencing the test execution
- The Partner must access the queues provided to them to send and receive messages

## 7 Test Evidence

The Partner will extract the following evidences covering the testing period and send them via email to the Validation Service provider for Technical Validation of MQHA Connectivity Test.

### For partners having ITB connectivity

- Alliance Access Event Journal report
- Message File report
- Samples of ASCII (MT) and XMLv2 files (MT and MX)
- Screenshots / Log File / Dataset extract / Reports generated from the Partner application evidencing the test execution through Partner application and the reconciliation mechanism against delivery notification and transmission notification for ACK and NAK

### For partners testing through Wipro Testing Service

- Alliance Access Event Journal report and Message File report will be generated by Wipro
- The Partner to provide samples of ASCII (MT) and XMLv2 files (MT and MX)
- Screenshots / Log File / Dataset extract / Reports generated from the Partner application evidencing the test execution through Partner application and the reconciliation mechanism against delivery notification and transmission notification for ACK and NAK

## 8 Test Execution and Test result

### 8.1 Sending test messages to Alliance Access

A “From” session is started with the transmission of the required MX and / or MT messages supported by the application.

These messages are sent from the sender PIC licensed in Alliance Access to the same PIC (sender=receiver).

### 8.2 Collecting the results

The Partner will extract the following evidences covering the test performed period and send them via email to the Validation Service provider for Technical Validation of MQHA Connectivity Test.

- Alliance Access Event Journal report
- Message File report
- Screenshots / Log File / Dataset extract / Reports generated from the Partner application evidencing the test execution through partner application.

For the Partners connecting to Wipro Testing Services, Alliance Access Event Journal and Message File will be generated by Wipro.

## 9 Support

The execution of these tests implies some intervention within the Alliance Access system itself. Should that create difficulties, the Partner is invited to contact the SWIFT validation Service provider for assistance to the setup of the system.

## 10 Annex

### 10.1 Checklist for MQHA Connectivity Test

Checklist for MQHA Configuration Test			
#	Description	Partner Response	Remarks
1	Alliance Access is upgraded to 7.0		
2	The additional license package 13:AI MQS Adapter is installed for MQHA		
3	The application exchanges MX messages, 19:AI FILE XML license package is installed		
4	The application is updated to support XML format v2		
5	Back-office application supports Alliance Access Info in the Message Data Part		
6	Handling of the messages that fail due to Bad LAU		
7	The business application handles Delivery Notification system message returned by Alliance Access. If so, please provide the screenshots of the Reconciliation with Original Message updated in the business application (using either MIR of the original message or Alliance Access traffic reconciliation – TR_REC)		
8	Provide a sample of SWIFT Acknowledgement message (FIN MT) to back-office application <b>with</b> the complete Original Message and screenshots of the reconciliation with Original Message updated in the business application		
9	Provide a sample of SWIFT Acknowledgement message (FIN MT) to back-office application <b>without</b> the complete Original Message and screenshots of the reconciliation with Original Message updated in the business application		
10	The back-office application supports XML format v2. If so, send a copy of XML format V2 message exchanged by financial application to Alliance Access direction		
11	The back-office application process transmission notification messages MT05 Quit ACK. If so, provide a sample of MT05 message and also the screen shot of how the transmission notification message was processed by the back-office application		
12	Handling of RTV Trailer varies between MQSA and MQHA. If the back-office application processes RTV Trailer information, please provide screenshot of how the RTV Trailer information is handled in the application		
Configuration Parameter for Input Message Partner			
13	Specify the Error Queue Name.		
14	Send a copy of MQ-MT format message exchanged by financial application to Alliance Access direction		
15	What is the validation Level set for Input Message Partner – Messages sent from Application to Alliance Access direction? Specify the Message Type and the validation assigned. If validation is set to <b>“None”</b> , the reason for the same may be provided		
16	Confirm if <b>“Validation Error Code”</b> is selected for MQ-MT data format		

## 10.2 Sample Message Partner Configuration

Direction: From Message Partner to Alliance Access

Format: MQ-MT

**Application Interface - Message Partner MPEXAMENMQMTIN**

Profile | Session | Authentication | Reception

Message partner:  Status:

Description:

Allowed direction:  Connection method:

Details:

Session initiation:  Data format:

Queue Manager Name:

Queue Name:

Error Queue Name:

Keep Session Open:

Transfer SAA Information:

Use MQ Descriptor:

**Application Interface - Message Partner MPEXAMENMQMTIN**

Profile | Session | Authentication | Reception

Messages:

	To partner	From partner
Sequence number:	<input type="text" value="1"/>	<input type="text" value="16"/>
Accepted:	<input type="text" value="0"/>	<input type="text" value="5"/>
Rejected:	<input type="text" value="0"/>	<input type="text" value="0"/>
Bypassed:	<input type="text" value="0"/>	
Not yet acknowledged:	<input type="text" value="0"/>	<input type="text" value="0"/>

Session:

Connection point:

Direction:

Session identifier:

Status:

Last error message:

Application Interface - Message Partner MPEXAMENMQMTIN

Profile | Session | Authentication | Reception

Local authentication required

Application Interface - Message Partner MPEXAMENMQMTIN

Profile | Session | Authentication | Reception

Parameters

Validation level: No Validation

Profile name: R6.3\_MsgPartner

Message modification allowed: Allowed

Unit to be assigned: None

UUMID included in Original Message:

Routing

Disposition: Dispose message in Ready-to-Send

Report Format

Transfer UUMID:  Original Message:  Validation Error Code:

Direction: From Alliance Access to Message Partner

Format: MQ-MT

**Application Interface - Message Partner MPEXAMNMQMTOUT**

Profile | Session | Authentication | Emission

Message partner:  Status:

Description:

Allowed direction:  Connection method:

Transfer PKI Signatures:  Increment Sequence Number across Sessions:

Always transfer MAC/PAC:

Details:

Session initiation:  Data format:

Queue Manager Name:

Queue Name:

Error Queue Name:

Keep Session Open:

Transfer SAA Information:

Run output session

Number of messages =

Or at (hh.mm):

**Application Interface - Message Partner MPEXAMNMQMTOUT**

Profile | Session | Authentication | Emission

Messages

	To partner	From partner
Sequence number:	<input type="text" value="14"/>	<input type="text" value="1"/>
Accepted:	<input type="text" value="4"/>	<input type="text" value="0"/>
Rejected:	<input type="text" value="0"/>	<input type="text" value="0"/>
Bypassed:	<input type="text" value="0"/>	<input type="text" value="0"/>
Not yet acknowledged:	<input type="text" value="0"/>	<input type="text" value="0"/>

Session

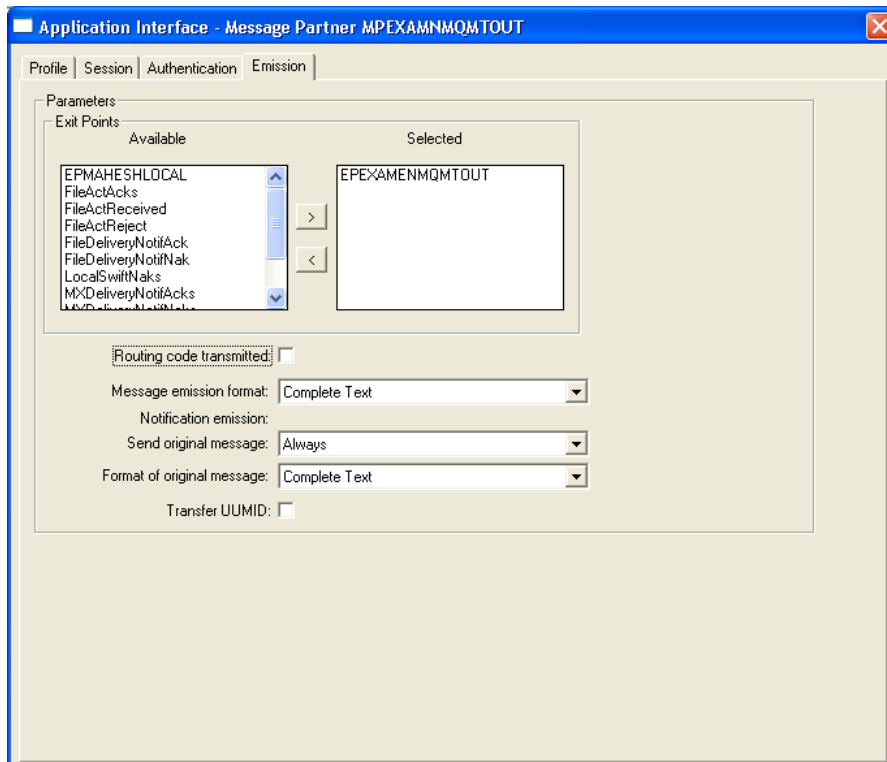
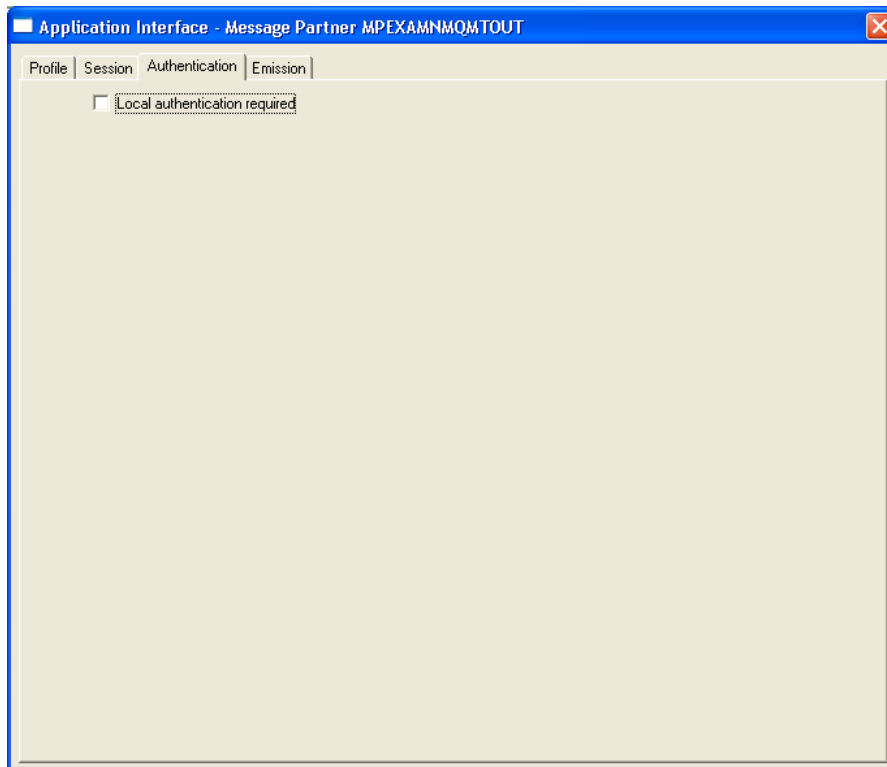
Connection point:

Direction:

Session identifier:

Status:

Last error message:



Direction: From Message Partner to Alliance Access

Format: XML v2

**Application Interface - Message Partner MPEXAMENMQMXIN**

Profile | Session | Authentication | Reception

Message partner:  Status:

Description:

Allowed direction:  Connection method:

Details:

Session initiation:  Data format:

Queue Manager Name:  Version:

Queue Name:  Revision:

Error Queue Name:

Keep Session Open:

Transfer SAA Information:

**Application Interface - Message Partner MPEXAMENMQMXIN**

Profile | Session | Authentication | Reception

Messages:

	To partner	From partner
Sequence number:	<input type="text" value="1"/>	<input type="text" value="1"/>
Accepted:	<input type="text" value="0"/>	<input type="text" value="0"/>
Rejected:	<input type="text" value="0"/>	<input type="text" value="0"/>
Bypassed:	<input type="text" value="0"/>	<input type="text" value="0"/>
Not yet acknowledged:	<input type="text" value="0"/>	<input type="text" value="0"/>

Session:

Connection point:

Direction:

Session identifier:

Status:

Last error message:

Application Interface - Message Partner MPEXAMENMQMXIN

Profile | Session | Authentication | Reception

Local authentication required

Application Interface - Message Partner MPEXAMENMQMXIN

Profile | Session | Authentication | Reception

Parameters

Validation level: No Validation

Profile name: R6.3\_MsgPartner

Message modification allowed: Allowed

Unit to be assigned: None

Routing

Disposition: Dispose message in Ready-to-Send

Direction: From Alliance Access to Message Partner

Format: XML v2

**Application Interface - Message Partner MPEXAMNMQMXOUT**

Profile | Session | Authentication | Emission

Message partner:  Status:

Description:

Allowed direction:  Connection method:

Always transfer MAC/PAC:  Increment Sequence Number across Sessions:

Details:

Session initiation:  Data format:

Queue Manager Name:  Version:

Queue Name:  Revision:

Error Queue Name:

Keep Session Open:

Transfer SAA Information:

Run output session

Number of messages =

Or at (hh.mm):

**Application Interface - Message Partner MPEXAMNMQMXOUT**

Profile | Session | Authentication | Emission

Messages:

	To partner	From partner
Sequence number:	<input type="text" value="1"/>	<input type="text" value="1"/>
Accepted:	<input type="text" value="0"/>	<input type="text" value="0"/>
Rejected:	<input type="text" value="0"/>	<input type="text" value="0"/>
Bypassed:	<input type="text" value="0"/>	<input type="text" value="0"/>
Not yet acknowledged:	<input type="text" value="0"/>	<input type="text" value="0"/>

Session:

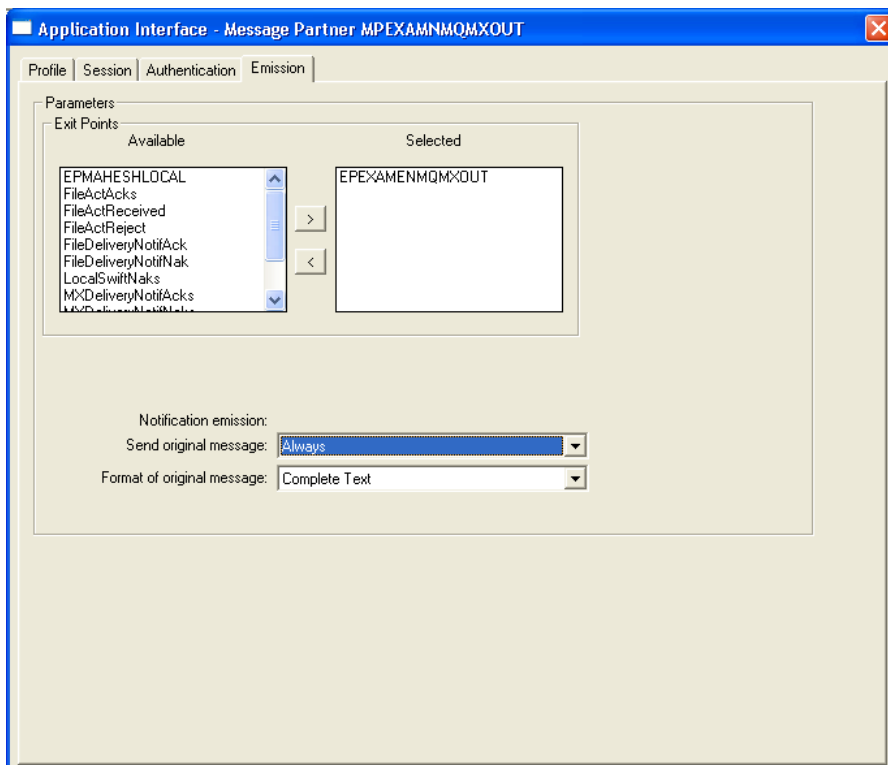
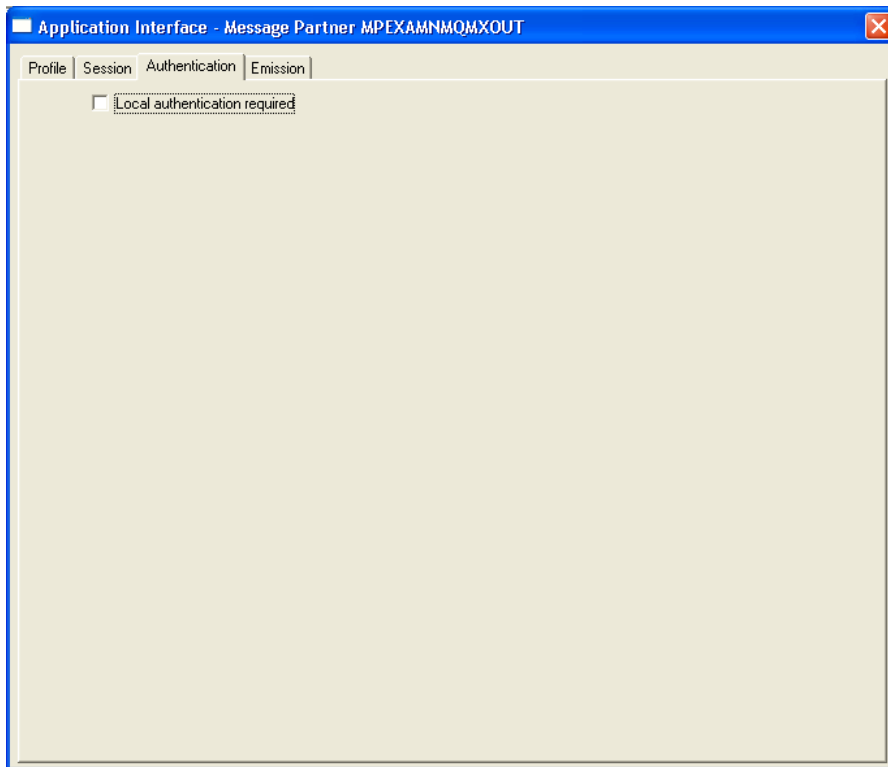
Connection point:

Direction:

Session identifier:

Status:

Last error message:





## 10.5 Output MT Message sample in MQ-MT format

```
{1:F01SPXAINJAA010029026640}{2:01031525110125SPXAINJJAXXX00290945301101251525N}{3:
{108:N2-XXX-A01}}{4:
:20:09-103-NVR-0015
:13C:/CLSTIME/0945+0100
:23B:CRED
:23E:CHQB
:26T:K90
:32A:091120USD15000,00
:33B:USD15000,00
:50A:/123456
SWHQBEBB
:52A:SWHQBEBB
:53B:/C/23456789
NEW YORK BRANCH
:54D:/87654321
RECEIVERS CORRESPONDENT
NEW YORK BRANCH
USA
:55D:/456789
THIRD REIMBURSEMENT INSTITUTION
:56A:/C/654321
SWHQBEBB
:57D:ACCOUNT WITH INSTITUTION
NEW YORK
USA
584214
:59A:SWHQBEBB
:70:/ROC/REF12365
:71A:OUR
:71G:USD60,00
:77B:/ORDERRES/US
-}{5:{MAC:00000000}{CHK:04E48C43D6D9}}{S:{SAC:}{COP:P}}
```

## 10.6 Delivery Notification Message

```
{1:F01SPXAINJJAXXX0029026635}{2:00111333110125DYDYXXXXHXXX00010716881101251433S}{4:
{175:1433}{106:110125SPXAINJJAXXX0029094525}{108:N2-XXX-
A01}{175:1433}{107:110125SPXAINJAA010029026634}}{5:{CHK:E06F1158C776}{SYS:}}{S:{CO
P:P}{INS:access/EPEXAMENMQMOUT}{UNT:None}{USR:Administrator}}
```

## 10.7 ACK Message with Original Message

```
{1:F21SPXAINJJAXXX0029094526}{4:{177:1101251508}{451:0}{108:N-XXX-
A01}}{1:F01SPXAINJJAXXX0029094526}{2:1103SPXAINJJXA01N}{3:{108:N-XXX-A01}}{4:
:20:09-103-NVR-0011
:13C:/CLSTIME/0945+0100
:23B:CRED
:23E:CHQB
:26T:K90
:32A:091120USD15000,00
:33B:USD15000,00
:50A:/123456
SWHQBEBB
:52A:SWHQBEBB
:53B:/C/23456789
NEW YORK BRANCH
:54D:/87654321
RECEIVERS CORRESPONDENT
NEW YORK BRANCH
USA
:55D:/456789
THIRD REIMBURSEMENT INSTITUTION
:56A:/C/654321
SWHQBEBB
```

:57D:ACCOUNT WITH INSTITUTION  
NEW YORK  
USA  
584214  
:59A:SWHQBEBB  
:70:/ROC/REF12365  
:71A:OUR  
:71G:USD60,00  
:77B:/ORDERRES/US  
-

} {5: {MAC:00000000} {CHK:04E4CC43D6D9}} {S: {INS:access/EPEXAMENMQMTOUT} {UNT:None} {USR:Administrator}}

## 10.8 ACK Message without Original Message

{1:F21SPXAINJJAXXX0029094528} {4: {177:1101251514} {451:0} {108:N-XXX-A01}} {1:F01SPXAINJJAXXX0029094528} {2:I103SPXAINJJXA01N} {3: {108:N-XXX-A01}} {5: {MAC:00000000} {CHK:04E4EC43D6D9}} {S: {CON:}}

**End of document**