



# Interface Qualification for a Real-time FileAct Messaging Interface

**BOX For SWIFTNet Conformance  
Statement**

**Table of Contents**

<b>1</b>	<b>General Information .....</b>	<b>3</b>
1.1	Supplier .....	3
1.2	Product Information.....	3
1.3	Conformance Testing Environment .....	3
1.4	Operational Environment .....	3
1.5	Customer Implementation Environment .....	4
1.6	Packaging Statement.....	4
1.7	Integration Support .....	5
<b>2</b>	<b>Conformance Requirements .....</b>	<b>5</b>
2.1	General Features .....	5
2.2	Security Features .....	6
2.3	Messaging Interface Applications Support .....	7
2.4	Operational Features .....	7
2.5	RMA Management .....	7
2.6	Application Service Profile .....	8
2.7	Real-time Features .....	8
2.8	FileAct Features – File Processing Features.....	9
2.9	FileAct Features – Event Handling .....	9

## 1 General Information

### 1.1 Supplier

Full name of the organisation that has registered this interface product and the name of the author of this Conformance Statement.

<b>Organisation</b>	<b>INTERCOPE GmbH (Legal name: Interscope International Communications Products Engineering GmbH)</b>
<b>Author</b>	Jens Huser
<b>Date</b>	January 2011

### 1.2 Product Information

The name and version numbers of the interface product to which this qualification and conformance claim applies.

<b>Product Name</b>	<b>BOX For SWIFTNet</b>
<b>Product Version Number</b>	<b>V3R11</b>

### 1.3 Conformance Testing Environment

The hardware platform and software environment in which this interface product's conformance is validated.

<b>Hardware Platform on which product was tested</b>	Dell PC Intel Core 2, 3 GHz, 4 GB
<b>Software Platform on which product was tested</b>	Linux SLES 10 Database: DB2 v9 Other: MQ/Series v7, Java 1.5

### 1.4 Operational Environment

If the environment for which you want to claim and guarantee conformance is not identical to the environment in which conformance was validated, please specify the hardware platform(s) and/or software platforms for which this product's performance is guaranteed.

<b>Hardware Platform on which product is guaranteed</b>	IBM S/390 SUN IBM p-series PC
<b>Software Platform on which product is guaranteed</b>	Solaris AIX Windows Linux z/Linux z/OS-USS

**1.5 Customer Implementation Environment**

*The hardware platform and software environment in which this interface product's customer implementation is defined (as required to achieve full qualification after an interim qualification).*

<p><b>Hardware Platform on which product was implemented</b></p>	<p>System Model: IBM,9119-FHA</p> <p>Machine Serial Number: 65180BC</p> <p>Processor Type: PowerPC_POWER6</p> <p>Processor Implementation Mode: POWER 6</p> <p>Processor Version: PV_6_Compat</p> <p>Number Of Processors: 1</p> <p>Processor Clock Speed: 4208 MHz</p> <p>CPU Type: 64-bit</p> <p>Kernel Type: 64-bit</p> <p>LPAR Info: 21 B4SAQL1</p> <p>Memory Size: 6144 MB</p> <p>Good Memory Size: 6144 MB</p> <p>Platform Firmware level: EH350_049</p> <p>Firmware Version: IBM,EH350_049</p>
<p><b>Software Platform on which product was implemented</b></p>	<p>AIX: 6100-04-05-1015/64Bit</p> <p>DB2: DB2Client V9.7</p> <p>WAS: V7.0.0.11/64bit</p> <p>MQ: V7</p>

**1.6 Packaging Statement**

*Explains how this product is packaged.*

*The main possibilities are:*

- *The RMA Interface is stand-alone and runs on its own platform.*
- *The RMA Interface is integrated on the same platform as a SWIFTNet Messaging Interface and a SWIFTNet Communication Interface*
- *The RMA Interface is integrated on the same platform as a SWIFTNet Messaging Interface but requires a separately packaged SWIFTNet Communication Interface to access SWIFTNet.*
- *If other variations are used these should be described below.*

<p><b>Product is a messaging interface only</b></p>	<p>No</p>
<p><b>Product is integrated with another (specify)</b></p>	<p>Messaging Interface is part of BOX For SWIFTNet</p>

<b>Communication Interface</b>	Alliance Gateway
<b>RMA Interface</b>	BOX For SWIFTNet
<b>Security Administration</b>	Alliance WebStation
<b>Other</b>	BOX For SWIFTNet includes Backoffice integration capabilities

### 1.7 Integration Support

If not integrated how does this product link to user client or server products? Does it use the Message Queue Host Adapter or Remote API Host Adapter as specified by SWIFT? Does it use a proprietary or other industry standard solution?

MQHA	To SAG
RAHA	-
Other	Internal Interface/Protocol to BOX For SWIFTNet

## 2 Conformance Requirements

The conformance requirements for a real-time FileAct messaging interface for SWIFTNet release 7.0 are specified in the corresponding interface specifications. A real-time FileAct messaging interface for SWIFTNet release 7.0 must support the mandatory items referred to in the messaging interface specifications and any of the additional optional items.

The tables below identify the mandatory and optional elements that a real-time FileAct messaging interface product may support. Please indicate for each whether your application supports and/or requires the elements, by entering the appropriate response in each of the columns on the right.

- Column 1 identifies the feature.
- Column 2 contains references to notes which describe the feature in more detail and where appropriate gives reference to the specification source.
- Column 3 describes whether the feature is Mandatory or Optional.
  - A Mandatory feature must be available for all users of the product.
  - An Optional feature, if implemented, is also subject to qualification.
- Column 4 is ticked "Y" or "N" to indicate support of the feature.

### 2.1 General Features

Feature	Note	M/O	Sup
Application identification within ProductList	A.1	M	Y
Usage of E2EControl for indication of PDE	A.2	M	Y
Provide client and server functionality	A.3	M	Y
Usage of enhanced errors	A.4	O	Y

Notes

## Interface Qualification Programme

- A.1 The messaging interface identifies itself in the ProductList. It also provides the ability for registered applications to use the ProductList within messages created by those applications or the messaging interface adds the ProductList when it identifies that an application is connected to the messaging interface.
- A.2 E2EControl must be used to identify the message for which possible duplicate information is to be provided.
- A.3 The client and server primitives from the communication interface must be used so that the messaging interface can play the role of client and of server in an efficient way.
- This requires following the order of primitives to be sent and depends on the features offered by the communication interface.
- A.4 Enhanced errors are used by the messaging interface when the ErrorMode is set appropriately.

## 2.2 Security Features

Feature	Note	M/O	Sup
Access control	B.1	M	Y
Audit log	B.2	M	Y
Security officer	B.3	O	N
Access control using 4-eyes principle	B.4	O	Y
Usage of SWIFTNet Link security contexts	B.5	M	Y
Renew rarely used SWIFTNet Link security contexts	B.6	M	Y
Local authentication (LAU) between communication interface and messaging interface if on different hosts for messages	B.7	O	Y
LAU between communication interface and messaging interface if on different hosts for files	B.8	O	Y
Signature processing	B.9	M	Y

### Notes

- B.1 Access to the messaging interface must be controlled. Updates in configuration information must be done by a user identified in a login process by providing a user name and password or equivalent methods.
- B.2 The audit log allows analysis of who changed what when. It also logs attempts to login by a user.
- B.3 Special users that can access and update security related information. Examples of security related updates are
- Adding, deleting, modifying users
  - Resetting passwords
- B.4 Two different users have to agree with a change before it is applied.
- B.5 SWIFTNet Link security contexts must be useable by entitled entities only. The implementation depends on the features offered by the communication interface.
- B.6 If a certificate is not used regularly, there is a risk that it will become invalid or expire. Once invalid or expired, the certificate will no longer be able to be renewed and must be recovered.
- B.7 This applies to the message protocol between communication interface and messaging interface.
- B.8 This applies to files sent from/to the communication interface.
- B.9 The messaging interface must properly sign traffic it sends to SWIFT.
- Properly signing means to select the signature format (Crypto or SignatureList), and to select what is signed (what DigestRef to add). What is signed depends on the service and request type.

### 2.3 Messaging Interface Applications Support

Feature	Note	M/O	Sup
Route incoming traffic to the correct business application	C.1	M	Y
Forward received signatures	C.2	O	Y
Forward own signatures	C.3	O	Y
Availability of messaging interface without connectivity to SWIFT	C.4	O	Y
Provide messaging interface processing information	C.5	M	Y
Provide SWIFTNet processing information	C.6	O	Y

#### Notes

- C.1 This routing can be based on various parameters taken from the received data. At a minimum, routing must be possible on the Service and RequestType taken from the RequestHeader or FileRequestHeader information.
- C.2 The signatures on data received can be made available to business applications requiring them.
- C.3 Ability to request a return of signatures on data sent and making them available to business applications requiring them.
- C.4 This feature allows business applications to send and receive messages/files even if the messaging interface is not connected via the communication interface to SWIFT. The messaging interface is a kind of hub between the business application and SWIFT.
- C.5 The most important processing information that can be passed consists of the verification result of the signatures. The minimum requirement is to allow routing the message/file based on the verification result of the signature.
- C.6 The processing information is related to non-repudiation, references added by SWIFT, routing information, copy related information such as the copy status. The business application can receive all information or a configured subset of processing information to be received.

### 2.4 Operational Features

Feature	Note	M/O	Sup
Traffic logging	D.1	O	Y
Unattended operations	D.2	O	Y
Backup/restore of messaging information	D.3	O	Y
Backup/restore of configuration data	D.4	O	Y

#### Notes

- D.1 Separate log from the actual messages/files sent or received is available for event analysis.
- D.2 The ability to use a messaging interface with minimal operator intervention.
- D.3 The ability to backup and restore messaging data (messages or files).
- D.4 The ability to backup configuration data. Depending on the design it can be several types of backup/restore related to a coherent set of data of one or more subsystems.

### 2.5 RMA Management

Feature	Note	M/O	Sup
Check authorisation-to-send	E.1	O	Y

## Interface Qualification Programme

Check authorisation-to-receive	E.2	O	Y
Import RMA Authorisations	E.3	O	Y
RMA deployment – RMAChecked	E.4	O	Y
Configure local check mode in RMA trial period	E.5	O	Y
RMA deployment – reports	E.6	O	Y

### Notes

- E.1 The messaging interface must check the existence of an authorisation-to-send for the request that will be sent on the service.
- E.2 The messaging interface must check the existence of the authorisation-to-receive for the request that is received on the service.
- E.3 The messaging interface can import RMA authorisations.
- E.4 The messaging interface indicates the usage of the authorisation-to-send in the RequestControl, FileRequestControl and FileResponseControl as appropriate. This option is mandatory for messaging interfaces that support the send file functionality (option H.2).
- E.5 The local configuration changes the behaviour of the checking of RMA authorisations-to-send and authorisations-to-receive during the trial period.  
When check mode is on, then when a check fails, the traffic is stopped. When check mode is off, then when a check fails, the traffic is not stopped.
- E.6 The messaging interface provides information about the usage of authorisations for traffic sent and received.

The report can be integrated within traffic investigation reports or can be integrated within audit log reports.

## 2.6 Application Service Profile

Feature	Note	M/O	Sup
Application Service Profile Package Import	F.1	O	Y
Application Service Profile Package Usage	F.2	O	Y

### Notes

- F.1 The messaging interface must be able to import the package and apply the definitions of the application service profiles.
- F.2 An application service profile contains a set of parameters as decided by the Service Administrator during the definition of the service. The application service profile is used by messaging interfaces and applications to correctly send and receive traffic for that service.

## 2.7 Real-time Features

Feature	Note	M/O	Sup
Send and receive delivery notifications	G.1	M	Y
Reconcile received delivery notifications	G.2	O	Y

### Notes

- G.1 The sender of a file can request a delivery notification and must then specify the Responder and RequestType. The messaging interface of the receiver must be able to send the FileAct primitive to acknowledge the receipt of the file once the file is safe stored. This delivery notification must be related to the file transfer. Once this delivery notification is accepted by the sender of the file, the delivery notification cannot be retried. The messaging interface must be able to receive the delivery notification and make it available to the business application.

- G.2 This can be done by the business application. When appropriate, considering resilience requirements, the messaging interface may rely on the file status offered by the communication interface.

## 2.8 FileAct Features – File Processing Features

Feature	Note	M/O	Sup
Send files – 4eyes	H.1	O	Y
Send files	H.2	O	Y
Responding to a GetFileRequest	-	O	Y
Preparing a PutFileRequest	-	O	Y
Receive files	H.3	M	Y
Responding to a PutFile	-	M	Y
Preparing a GetFile	-	O	Y
Support of remote file handler	H.4	O	Y

### Notes

- H.1 This is a step in the flow of sending files where another operator is involved. This can be implemented by business applications.
- H.2 Responding to a GetFile or preparing a PutFile.
- H.3 Responding to a PutFile or preparing a GetFile.
- H.4 The file handler is responsible for managing the access to physical files (emission, reception) on a different remote system, which does not contain an SNL instance, enabling a file to be transferred through FileAct.

## 2.9 FileAct Features – Event Handling

Feature	Note	M/O	Sup
Process File Events	I.1	M	Y

### Notes

- I.1 This feature is required for keeping track of the file transfer status. It can be implemented either by using the file event subscription offered by the communication interface or by the file status commands. The choice depends on the resilience requirements. For example, highly resilient systems must not rely on the file status commands.

## **Legal Notices**

### **Copyright**

SWIFT © 2011. All rights reserved.

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

### **Confidentiality**

This publication contains SWIFT or third-party confidential information. Do not disclose this publication outside your organisation without the prior written consent of SWIFT.

### **Disclaimer**

The information in this publication may change from time to time. You must always refer to the latest available version on [www.swift.com](http://www.swift.com).

### **Translations**

The English version of SWIFT documentation is the only official and binding 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, SWIFTNet, SWIFTReady, and Accord. Other product, service, or company names in this publication are trade names, trademarks, or registered trademarks of their respective owners.