Swift Compatible Interface
FileAct Store-and-forward Interface Conformance Statement

CGI FASTWIRE Open

This document lists the mandatory and optional requirements supported by the FileAct Store-and-forward messaging interface.

January 2025

Table of Contents

1 General Information **3**

1.1 Supplier 3

1.2 Product Information 3

1.3 Operational Environment 3

1.4 Customer Implementation Environment 3

1.5 Packaging Statement 4

1.6 Integration Support 4

2 Conformance Requirements **5**

2.1 Messaging Interface Features 5

2.2 Store-and-forward Features 7

2.3 FileAct Features 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.

Organisation	CGI IT UK Ltd
Author	Carryl Durnell
Date	April 2014 (Renewal 2025)

1.2 Product Information

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

Product Name	CGI FASTWIRE Open
Product Version Number	FASTWIRE Open 6.13.6.1.1 and higher versions

1.3 Operational Environment

The hardware platform(s) and/or software platforms for which this product's performance is guaranteed.

Hardware Platform on which product is guaranteed	Intel® Xeon® CPU E5420, x86_64
Software Platform on which product is guaranteed	Linux – Red Hat 8.8 and higher 8.x versions Oracle 19c IBM MQ Server 9.3 IBM MQ Client 9.1 WAS 9.0.5.3 Tomcat 9.0.80 Edge 96+ / Chrome 76+ SAG 7.7 AMQ Broker 7.11 AMQ C++ Client 2.11 AMQ JMS Client 2.11

1.4 Customer Implementation Environment

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

Hardware Platform on which product was implemented	RHEL on x86
Software Platform on which product was implemented	RHEL 8.9 Oracle 19c IBM MQ Server 9.3 IBM MQ Client 9.3 WebSphere Application Server 9.0.5

1.5 Packaging Statement

The main possibilities are:

- The product is a messaging interface only if the main purpose is to exchange messages/messages between back office applications and Swift.
- The product is integrated with another if the product offers other functionality such as connectivity to other external networks or the product is also a business application that creates and processes messages and files.
- The communication interface used by the product.
- The Relationship Management Application (RMA) used by the product.
- The security administration interface used by the product. For example, the management of security endpoints and its roles by security officers can be done by the product itself and/or by using SWIFTNet Online Operations Manager or another product.

Other variations are possible. If used, these should be described below.

Product is a messaging interface only	Yes
Product is integrated with another (which)	No
Communication Interface	Alliance Gateway
RMA Interface	The RMA Interface is stand-alone and runs on its own platform
Security Administration	SWIFTNet Online Operations Manager
Other	

1.6 Integration Support

The table describes if the product uses the Message Queue Host Adapter or Remote API Host Adapter as specified by Swift, or if it uses a proprietary or other industry standard solution.

MQHA	Yes
RAHA	Yes
Other	Remote File Handler

2 Conformance Requirements

The conformance requirements for a Store-and-forward FileAct messaging interface for SWIFTNet release 7.0 are specified in the corresponding interface specifications. A 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 Store-and-forward FileAct messaging interface product may support.

- Column 1 identifies the feature.
- Column 2 contains references to note which describe the feature in more detail and where appropriate give 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 compliance validation.
- Column 4 is ticked “Y” or “N” to indicate support of the feature.

2.1 Messaging Interface Features

2.1.1 General Features

Feature	Note	Mand. / Optional	Support (Y/N)
Application identification within ProductList	A.1	M	Y
Usage of E2EControl for indication/detection of PDE	A.2	M	Y
Provide client and server functionality	A.3	M	Y
Usage of enhanced errors	A.4	O	Y

Notes

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 The enhanced error identifies more accurately the severity and if the error was on the RequestPayload or on the ResponsePayload. The ErrorMode must be set appropriately

2.1.2 Security Features

Feature	Note	Mand. / Optional	Support (Y/N)
Usage of SWIFTNet Link security contexts	B.1	M	Y
Renew rarely used SWIFTNet Link security contexts	B.2	M	Y
Signature processing	B.3	M	Y
Support of digital signature within payload	B.4	O	N

Notes

B.1 SWIFTNet Link security contexts must be useable by entitled entities only. The implementation depends on the features offered by the communication interface.

B.2 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.3 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. . The messaging interface must report on invalid received signature.

B.4 The messaging interface may offer functions to use SWIFTNet PKI certificates for digital signature within the ISO20022 Business Application Header (BAH or head.001.001.01) or within other elements such as the Xchg element (Business File Header (BFH) or head.002.001.01) used by T2S.

2.1.3 Messaging Interface Application Support

Feature	Note	Mand. / Optional	Support (Y/N)
Route incoming traffic to the correct business application	C.1	M	Y
Forward received signatures	C.2	O	N
Forward own signatures	C.3	O	N
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.

2.1.4 Operational Features

Feature	Note	Mand. / Optional	Support (Y/N)
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.

2.1.5 RMA Management

Feature	Note	Mand. / Optional	Support (Y/N)
Check authorisation-to-send	E.1	O	Y

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 checks the existence of an authorisation-to-send for the request that will be sent on the service.

E.2 The messaging interface checks 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 FileRequestControl and FileResponseControl as appropriate.

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 and a check fails, the traffic is stopped. When check mode is off and 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.

2.1.6 Application Service Profile

Feature	Note	Mand. / Optional	Support (Y/N)
Application Service Profile Package Import	F.1	O	Y
Application Service Profile Package Usage	F.2	O	Y

Notes

F.1 The messaging interface is 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.

2.2 Store-and-forward Features

2.2.1 General Features

The table below lists the general features required by messaging interfaces supporting Store-and-forward.

Feature	Note	Mand. / Optional	Support (Y/N)
Start a Session on a Queue	G.1	M	Y
Stop a Session on a Queue	G.2	M	Y
Monitor Queues	G.3	M	Y
Generic Queues for Requests and Files	G.4	M	Y
Support of System Recovery	G.5	M	Y
Send Y-Copy Authorisation or Refusal	G.6	O	N
Exchanging messages over Store-and-forward requires at least one of the following to be Supported:	G.7	M	Y
Pull mode	G.8	O	N

Push mode	G.9	O	Y
Support of Cold Start	G.10	M	Y

Notes

G.1 This is performed by opening an output channel.

G.2 This is performed by closing an output channel.

G.3 The ability of the messaging interface to monitor queues status.

G.4 This supports the single window concept, where different business services can use the same queue. Generic queues can simplify the MRR setup.

G.5 A SWIFTNet Store-and-forward system recovery is done when the active site becomes inoperable for some reason. In this case, some data previously sent to SWIFT may need to be resent. The amount of data depends on the replication status of Store-and-forward prior to the incident that caused the active site to stop.

G.6 This is mandatory for messaging interfaces used by third-party organisations that offer Y-Copy services. The authorisation itself is always created by a business application using the messaging interface.

G.7 Store-and-forward interface must choose one of the following modes:

G.8 The product may use Pull mode to fetch messages from its queues, or

G.9 The product may use Push mode to receive messages from its queues automatically.

G.10 The procedure for restarting the operations after a cold start will use the available features of the interface. Accurate documentation is essential to guide the user so that recovery from the cold start is as easy as possible. This documentation must be available at the time of cold start.

2.2.2 Sequence of Transmission/Delivery

Feature	Note	Mand. / Optional	Support (Y/N)
Support of Output Channels	H.1	M	Y
Queue Sharing	H.2	O	N

Notes

H.1 The support of output channels consists of using the primitives for output channels according to the output channel protocol and the monitoring of their state through the event handling.

H.2 Queue sharing is done by opening several output channels on the same queue.

2.2.3 System Messages

Feature	Note	Mand. / Optional	Support (Y/N)
Create System Messages	I.1	O	N
Create System Report Requests	I.2	O	N
Send System Messages	I.3	M	Y
Receive System Messages	I.4	M	Y
Process Received System Messages	I.5	O	N

Notes

I.1 Creation of system messages can be done by a business application using the messaging interface or by the messaging interface itself.

I.2 The ability of the messaging interface to create report requests.

I.3 The ability of the messaging interface to send system messages.

I.4 The ability of the messaging interface to receive system messages.

I.5 This can be done by a business application using the messaging interface or by the messaging interface itself.

2.2.4 Delivery Management

Feature	Note	Mand. / Optional	Support (Y/N)
Acknowledge Receipt	J.1	M	Y

Acknowledge Receipt with Logical Filename	J.2	O	Y
Receive Delivery Notifications	J.3	M	Y
Receive Notifications of Authorisation or Refusal	J.4	M	Y
Reconcile Undelivered Traffic Reports	J.5	O	N
Receive Delivery Notifications version 02	J.6	O	N

Notes

J.1 The messaging interface acknowledges receipt after having safe stored the data it receives. This requires persistent storage.

J.2 The messaging interface acknowledges receipt with use of the Logical Filename.

J.3 The ability of the messaging interface to receive delivery notifications.

J.4 The ability of the messaging interface to receive notifications of authorisations or refusal (for5Y-Copy services)..

J.5 The ability of the messaging interface to reconcile undelivered traffic reports.

J.6 The ability of the messaging interface to receive delivery notifications of version 02 with additional information. The messaging interface may offer functionality to process the additional information. An example of additional information is the Signature provided by T2S.

2.3 FileAct Features

2.3.1 File Processing Features

The table below lists the general features required by messaging interfaces supporting FileAct.

Feature	Note	Mand. / Optional	Support (Y/N)
Send files – 4eyes	K.1	O	Y
Send files	K.2	M	Y
Support 2GB files	K.3	M	Y
Receive files	K.4	M	Y
Support 2GB files	K.5	M	Y
Use system messages as delivery notifications	K.6	O	Y
Select 1 or more copy destinations	K.7	M	Y
Support of remote file handler	K.8	O	Y

Notes

K.1 This is a step in the flow of sending files where another operator is involved. This can be implemented by business applications.

K.2 This is using a PutFileRequest. The file is prepared by the business application and made available in some way.

K.3 Support of 2GB files is mandatory with release 7.2.

K.4 For Store-and-forward, the reception of a file through FetchFile

K.5 Support of 2GB files is mandatory with release 7.2.

K.6 Delivery notifications as system messages contain the FileRequestHeader as context information.

K.7 This is mandatory for the messaging interface used to indicate which copy destination a file is to be copied to. This selection is from a list of possible third parties for that service. (These can be taken from the application service profile, if supported). This can also include multiple destinations.

If messaging interface is deployed to corporate users only, then this functionality is optional.

K.8 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.

2.3.2 File Distribution Features

The table below lists the distribution features that messaging interfaces supporting FileAct may use.

Feature	Note	Mand. / Optional	Support (Y/N)
Support delayed file fetch	L.1	O	N
Send request to distribute files	L.2	O	N
Fetch a distributed file	L.3	M	Y

Notes

L.1 The ability of the messaging interface to perform a delayed FileFetch. This is configurable and can be based on the priority of the file message, the service and request type or the correspondent.

L.2 This is the creation of a file distribution request.

L.3 This is the same as any other fetched file, except for the handling of the RecipientList.

2.3.3 Copy Features

The following table lists the features required by FileAct messaging interfaces supporting third party services.

Feature	Note	Mand. / Optional	Support (Y/N)
Act as a copy destination for T-Copy	M.1	O	N
Act as a copy destination for Y-Copy	M.2	O	N
Act as a copy destination for Y-Copy in TCopyFallback	M.3	O	N

Notes

M.1 This is mandatory for the messaging interface used by a copy destination for processing copied file notifications. T-Copy is a service where the file is delivered to the receiver without any dependency of the copied information.

M.2 This is mandatory for the messaging interface used by a copy destination for processing copied file notifications with authorisation of delivery. When the service is Y-Copy, the file is copied to the third party. The third party must authorise or refuse the file to be delivered to the receiver. The third party may refuse the delivery of the file to the receiver, in which case, the sender is informed via a refusal notification message.

M.3 This is mandatory for the messaging interface used by a third party for processing copied file notifications with CopyState=TCopyFallBack.

2.3.4 Event Handling

Feature	Note	Mand. / Optional	Support (Y/N)
Process File Events	N.1	M	Y

Notes

N.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. File events are processed according to the applicable protocol (sending files or receiving files).

Legal Notices

Copyright

Swift ©2025. All rights reserved.

Restricted Distribution

Do not distribute this publication outside your organisation unless your subscription or order expressly grants you that right, in which case ensure you comply with any other applicable conditions.

Disclaimer

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 and binding version.

Trademarks

Swift is the trade name of S.W.I.F.T. SC. The following are registered trademarks of Swift: Swift, the Swift logo, 3SKey, Innotribe, MyStandards, Sibos, SWIFTNet, SWIFT Institute, the Standards Forum logo, SWIFT gpi with logo, the SWIFT gpi logo, and UETR. Other product, service, or company names in this publication are trade names, trademarks, or registered trademarks of their respective owners.