Swift Compatible Interface
FileAct Real-time Interface Conformance Statement

Diapason-Rsg

This document lists the mandatory and optional requirements supported by the FileAct Real-time messaging interface.

January 2024

Table of Contents

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	MCC-SOFT
Author	LESCOT Gilles
Date	July 2018 (Renewal 2024)

1.2 Product Information

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

Product Name	Diapason-Rsg
Product Version Number	2.2

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 X3430 2 Go 292 Go (2x 146 Go) SAS 2.0 6Gb/s RAID
Software Platform on which product is guaranteed	CentOS Linux release 7.4.1708 (Core) RHEL 6.4

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	
Software Platform on which product was implemented	RHEL7.6 / OpenJDK Runtime Environment

1.5 Packaging Statement

The main possibilities are:

- The product is a messaging interface only if the main purpose is to exchange 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.
- 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)	Yes (only Diapason)
Communication Interface	SAG in relaxed mode
RMA Interface	No
Security Administration	Via SAG (Relaxed mode)
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	No
RAHA	SNL API - No SAG API - Yes
Other	No

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.

- 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 compliance validation.
- Column 4 is ticked “Y” or “N” to indicate support of the feature.

2.1 General Features

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

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 Enhanced errors are used by the messaging interface when the ErrorMode is set appropriately.

2.2 Protocol 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.

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.

2.3 Messaging Interface Applications 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	N
Provide messaging interface processing information	C.5	M	Y
Provide SWIFTNet processing information	C.6	O	N

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	Mand. / Optional	Support (Y/N)
Traffic logging	D.1	O	N
Unattended operations	D.2	O	N
Backup/restore of messaging information	D.3	O	N
Backup/restore of configuration data	D.4	O	N

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	Mand. / Optional	Support (Y/N)
Check authorisation-to-send	E.1	O	N
Check authorisation-to-receive	E.2	O	N
Import RMA Authorisations	E.3	O	N
RMA deployment – RMAChecked	E.4	O	N
Configure local check mode in RMA trial period	E.5	O	N
RMA deployment – reports	E.6	O	N

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

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	Mand. / Optional	Support (Y/N)
Send and receive delivery notifications	G.1	M	Y
Reconcile received delivery notifications	G.2	O	N
Send delivery notification with LogicalName	G.3	O	N

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.

G.3 The messaging interface notifies receipt with use of the logical filename.

2.8 FileAct Features – File Processing Features

Feature	Note	Mand. / Optional	Support (Y/N)
Send files – 4eyes	H.1	O	N
Send files	H.2	O	Y
Responding to a GetFileRequest		O	N
Preparing a PutFileRequest		O	Y
Support of 2 GB files		M	Y
Receive files	H.3	M	Y
Responding to a PutFile		M	Y
Preparing a GetFile		O	N
Support of 2 GB files		M	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. Support of 2GB files is mandatory with release 7.2.

H.3 Responding to a PutFile or preparing a GetFile. Support of 2GB files is mandatory with release 7.2.

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	Mand. / Optional	Support (Y/N)
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