
Swift Compatible Interface

InterAct Store-and-forward Interface

Conformance Statement

Integrated Financial Gateway (IFG)

This document lists the mandatory and optional requirements supported by the InterAct 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	3
1.6 Integration Support	4
2 Conformance Requirements	5
2.1 Messaging Interface Features	5
2.2 SWIFNet generic features	6
2.3 SWIFNet protocol features	7
2.4 Store-and-forward features	8
2.5 InterAct features	11
2.6 Swift Platform-Transaction manager features	12
2.7 Copy destination features	12

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	Incentage AG
Author	Jeroen van Hal
Date	November 2021 (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	Integrated Financial Gateway (IFG)
Product Version Number	2025.1.0

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	Any Hardware Platform that supports and is supported by the listed Software Platforms
Software Platform on which product is guaranteed	Red Hat Enterprise Linux 7 and 8

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	Intel(R) Core (TM) i7-3770 CPU @ 3.40GHz / 24.0 GB RAM / 64-bit Operating System, x64-based processor
Software Platform on which product was implemented	Red Hat Enterprise Linux 7

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	No
Product is integrated with another (which)	Incentage Acceleration Platform (IAP)
Communication Interface	Alliance Gateway (RAHA)
RMA Interface	Incentage RMA Module (IRM)
Security Administration	Alliance Web Platform with SWIFT Web Access SWIFTNet Online Operations Manager
Other	IAP includes back-office integration

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.

Products	SNL API	SAG API
MQHA	No	No
RAHA	No	Yes
Other	Integration with back-office applications and source systems is fully supported by the Incentage acceleration platform (IAP) which is equipped with all industry standard adapters such as File, SFTP, MQ, JMS, Oracle, DB2, Postgres, REST, Web Services etc.	

2 Conformance Requirements

The conformance requirements for a Store-and-forward InterAct messaging interface for SWIFTNet release 7.6 are specified in the corresponding interface specifications. A Store-and-forward InterAct messaging interface for SWIFTNet release 7.6 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 InterAct 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 (M/O).
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

This section lists the features that are applicable to the vendor messaging interfaces as an application.

2.1.1 Security features

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

Notes

- A.1 Access to the vendor 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.
- A.2 The audit log allows analysis of who changed what when. It also logs attempts to login by a user.
- A.3 Special users that can access and update security related information. Examples of security related updates are Adding, deleting, modifying users Resetting passwords
- A.4 Two different users have to agree with a change before it is applied.
- A.5 This applies to the message protocol between communication interface and vendor messaging interface. The Local Authentication Requirement Specification provides more information. If the vendor messaging interface manages SWIFTNet PKI security contexts, LAU is optional. If it relies on the communication interface to open security contexts, LAU is mandatory
- A.6 This applies to the message protocol between vendor messaging interface and back-office applications.
- A.7 SWIFTNet Link security contexts must be useable by entitled entities only. The implementation depends on the features offered by the communication interface.
- A.8 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.

2.1.2 Applications support

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

Notes

- B.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.
- B.2 The signatures on data received can be made available to business applications requiring them.
- B.3 Ability to request a return of signatures on data sent and making them available to business applications requiring them.
- B.4 This feature allows business applications to send and receive messages even if the vendor messaging interface is not connected via the communication interface to Swift. The vendor messaging interface is a kind of hub between the business application and Swift.
- B.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 based on the verification result of the signature.
- B.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.

2.1.3 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 sent or received is available for event analysis.
- D.2 The ability to use a vendor messaging interface with minimal operator intervention.
- D.3 The ability to backup and restore messaging data (messages).
- 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.

2.2 SWIFNet generic features

This section list the features related to the ability of the vendor messaging interface to connect and support SWIFTNet as a network.

2.2.1 Application service profile (ASP)

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

Notes

- E.1 The vendor messaging interface is be able to import the package and apply the definitions of the application service profiles.
- E.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 vendor messaging interfaces and applications to correctly send and receive traffic for that service.

2.2.2 Product information

Feature	Note	M/O	Sup
Vendor messaging interface identification within <code>ProductList</code>	F.1	M	Y
Application identification within <code>ProductList</code>	F.2	M	Y

Notes

- F.1 The vendor messaging interface identifies itself in the `ProductList`.
- F.2 It also provides the ability for registered applications to use the `ProductList` within messages created by those applications or the vendor messaging interface adds the `ProductList` when it identifies that an application is connected to the vendor messaging interface.

2.2.3 Relationship management

Feature	Note	M/O	Sup
Check RMA authorisation to send	G.1	M	Y
Check RMA authorisation to receive	G.2	M	Y
RMA deployment – RMAChecked message indication	G.3	M	Y
Local check mode in RMA trial period configuration	G.4	M	Y
RMA deployment reporting	G.5	M	Y
RMA distribution file import	G.6	M	Y

Notes

- G.1 The vendor messaging interface checks the RMA authorisation before sending InterAct messages. Traffic that fails is made available for manual investigation by an entitled user.
- G.2 The vendor messaging interface checks the RMA authorisation after receiving InterAct messages. Traffic that fails is made available for manual investigation by an entitled user.
- G.3 The vendor messaging interface indicates the usage of the authorisation-to-send in the RequestControl as appropriate.
- G.4 This 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.
- G.5 The vendor 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.
- G.6 The vendor messaging interface import partial and/or complete RMA distribution files from other vendor messaging interfaces, RMA interfaces or RMA portal.

2.3 SWIFNet protocol features

This section list the features related to the ability of the vendor messaging interface to process specific features applicable to InterAct store-and-forward protocol.

2.3.1 PKI signature

Feature	Note	M/O	Sup
SWIFNet Signature processing	H.1	M	Y
User Digital Signature processing	H.2	O	N

Notes

- H.1 The vendor 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). The what is signed depends on the service and request type.
- H.2 The vendor 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.

2.3.2 Non-repudiation

Feature	Note	M/O	Sup
Non-repudiation support	I.1	M	Y

Notes

- I.1 Non-repudiation can be mandatory or optional for a service.

2.3.3 Duplicate messages

Feature	Note	M/O	Sup
Using E2EControl for PDE	J.1	M	Y
PDE management with input channel	J.2	M	Y

Notes

- J.1 E2EControl must be used to identify the message for which a possible duplicate information is to be provided. The vendor messaging interface must add an E2EControl if not provided by the business application generating the message.
- J.2 When input channels are used, SWIFTNet provides automatic duplicate message filtering. This changes the vendor messaging interface behaviour in the re-sending process.

2.3.4 Other

Feature	Note	M/O	Sup
Provide client and server functionality	L.1	M	Y
Usage of enhanced errors	L.2	O	N

Notes

- K.1 The client and server primitives from the communication interface must be used so that the vendor 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 as explained in SWIFTNet Service Design Guide, and depends on the features offered by the communication interface.
- K.2 Enhanced errors are used by the vendor messaging interface when the *Sw:ErrorCodeMode* is set appropriately.

2.4 Store-and-forward features

This section list the features related to the ability of the vendor messaging interface to support SWIFTNet store-and-forward flows and protocols.

2.4.1 Store-and-forward sessions

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

Feature	Note	M/O	Sup
Input channel support	M.1	M	Y
Output channel support	M.2	M	Y

Swift Compatible Interface Programme

Starting and stopping a session on a queue	M.3	M	Y
Queue access mode (Push or Pull). Push mode support.	M.4	O	Y
Generic queues for message	M.5	M	Y
Queues monitoring	M.6	M	Y
Queues sharing	M.7	O	Y

Notes

- M.1 The vendor messaging interface is able to create, delete, open and close store-and-forward input channels as well as resolving input message gaps.
- M.2 Output channels are a pre-requisite for sharing of queues. The vendor messaging interface is able to create, delete, open and close output channels.
- M.3 The usage of `OpenOutputChannelSnFRequest` is mandatory. The usage of `AcquireSnFRequest` is optional. The usage of `CloseOutputChannelSnFRequest` is mandatory. The usage of `ReleaseSnFRequest` is optional.
- M.4 Store-and-forward interface must choose one of the following modes:
 -1-Pull mode to fetch messages from its queues, or
 -2-Push mode to receive messages from its queues automatically.
 Push mode is the preferred mode.
- M.5 This supports the single window concept, where different business services can use the same queue. Generic queues can simplify the Message Reception Registry (MRR) setup.
- M.6 The vendor messaging interface monitors the state of output channels through the event handling.
- M.7 Queue sharing is done by opening several output channels on the same queue.

2.4.2 Delivery monitoring

These are the features supporting the monitory of message deliveries to counterparties.

Feature	Note	M/O	Sup
Ask for Delivery Notifications	N.1	M	Y
Receive Delivery Notifications	N.2	M	Y
Reconcile Undelivered Traffic Reports	N.3	O	N
Use system messages as delivery notifications	N.4	O	N
Undelivered traffic report	N.5	O	N

Notes

- N.1 The vendor messaging interface is able to request a positive delivery notification and an overdue warning.
- N.2 The ability of the vendor messaging interface to receive positive and negative delivery notifications as well as overdue warnings.
- N.3 This is a system message reporting on the delivery status of messages sent. The reconciliation can be done by the business application.
- N.4 Delivery notifications as system messages contain the `RequestHeader` as context information.
- N.5 The ability for the vendor messaging interface to send an undelivered traffic report request and to process the receive report.

2.4.3 Message distribution

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

Complete this section if your vendor messaging interface supports these distribution features.

Feature	Note	M/O	Sup
Send a message distribution request	O.1	O	N
Receive a distributed message	O.2	M	Y
Delivery monitoring of distributed messages	O.3	O	N

Notes

- O.1 This is the creation of a message distribution request.
- O.2 This is the same as any other messages, except for the handling of the RecipientList.
- O.3 A delivery notification can be requested on distributed message. There will be as many delivery notifications as they are recipients.

2.4.4 Message retrieval

Feature	Note	M/O	Sup
Send a retrieval request	P.1	O	N
Send a bulk retrieval request	P.2	O	N
Receive retrieved messages	P.3	O	N
Receive a retrieval report	P.4	O	N

Notes

- P.1 The vendor messaging interface is able to send a retrieval request system message for input and output messages. This can be achieved by SWIFTNet Online Operation Manager (O2M) or via a system message.
The vendor messaging interface must open its output channel in protocol level 2 in order to receive the retrieved input and output messages.
- P.2 The vendor messaging interface is able to send a bulk retrieval request system message for input and output messages. This can be achieved by SWIFTNet Online Operation Manager (O2M). Retrieved messages are grouped in a file received over FileAct instead of individual messages over InterAct.
- P.3 The vendor messaging interface received retrieved input and output messages as requested. The messaging interface must open its output channel in protocol level 2 in order to receive the retrieved input and output messages.
- P.4 The retrieval report summarises what has been retrieved.

2.4.5 System messages

Feature	Note	M/O	Sup
Create System Messages	Q.1	O	Y
Create System Report Requests	Q.2	O	N
Send System Messages	Q.3	M	Y
Receive System Messages	Q.4	M	Y
Process Received System Messages	Q.5	O	Y

Notes

- Q.1 Creation of system messages can be done by a business application using the vendor messaging interface or by the vendor messaging interface itself.
- Q.2 The ability of the vendor messaging interface to create report requests.
- Q.3 The ability of the vendor messaging interface to send system messages.
- Q.4 The ability of the vendor messaging interface to receive system messages.
- Q.5 This can be done by a business application using the vendor messaging interface or by the vendor messaging interface itself.

2.4.6 Participant of copy services

Feature	Note	M/O	Sup
Participant message exchange in a T-Copy service	R.1	M	Y
Participant message exchange in a Y-Copy service	R.2	M	Y
Receive Notifications of Authorisation or Refusal	R.3	M	Y
Receive Notifications of Authorisation or Refusal as business message	R.4	O	Y

Notes

- R.1 The vendor messaging interface send and receive messages as a participant to a T-Copy service. The copy can be partial or full.
- R.2 The vendor messaging interface send and receive messages as a participant to a Y-Copy service. The copy can be partial or full.
- R.3 These are system messages used to report on the processing result of messages part of a Y-copy service.
- R.4 The approval or rejection notification can be provided as a business message instead of a system message. For example the pacs.002 is received instead of the xsys.002.

2.4.7 Store-and-forward recovery

Feature	Note	M/O	Sup
Support of SnF System Recovery	S.1	M	Y
Support for cold-start	S.2	M	Y

Notes

- S.1 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.
- S.2 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.

2.5 InterAct features

This section list the features related to the ability of the vendor messaging interface to support SWIFTNet InterAct and message management functions.

2.5.1 Message processing features

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

Feature	Note	M/O	Sup
Manual messages entry	T.1	O	Y
Manual message correction	T.2	O	Y
Send messages	T.3	M	Y
Send message with 4-eyes approval	T.4	O	Y
Receive messages	T.5	M	Y
Multi-format message processing	T.6	M	Y

Notes

- T.1 Allow manual message entry. This can be done by business applications.
- T.2 Allow message correction. This can be done by business applications. The vendor messaging interface must then be able to route messages that failed to be sent to a business application for correction, or to provide the necessary information to identify the failed messages.
- T.3 The client part of the communication interface primitives for sending InterAct is invoked.
- T.4 This is a step in the flow of sending messages where another operator is involved and has to approve the sending. This can be implemented by business applications.
- T.5 The messages are received from the server part of the communication interface.
- T.6 A multi-format message contains an ISO 20022 message and its equivalent FIN message. The multi-format message is used as part of the ISO 20022 migration project. The vendor messaging interface has to validate the "Translated MT" DigestValue and process the translation result code.

2.6 Swift Platform-Transaction manager features

Over time Transaction manager will process more and more messages. Transaction manager introduces changes in the InterAct protocol and message structure.

Feature	Note	M/O	Sup
Processing of the Transaction Data InterAct tag Sw:TRD	U.1	M	Y
Verification of the Transaction manager signature	U.2	M	Y

Notes

- U.1 The vendor messaging interface must process the Sw:TRD field.
 U.2 The vendor messaging interface must verify the received Transaction manager's signature.

2.7 Copy destination features

The following table lists the features required by InterAct vendor messaging interfaces supporting third party services. The "S" indication in the "M/O" column specifies that this is service specific. If the service is supported, the feature must also be supported.

Feature	Note	M/O	Sup
Process received messages as a T-Copy destination	V.1	S	N
Process received messages as a Y-Copy destination	V.2	S	N
Process received messages as a Y-Copy destination in TCopyFallback mode	V.3	S	N
Send Y-Copy Authorisation or Refusal	V.4	S	N
Send Y-Copy Authorisation or Refusal with a business extension (i.e. pacs.002)	V.5	S	N
Send a retrieval request for copied messages	V.6	O	N

Notes

- V.1 This is mandatory for the vendor messaging interface used by a copy destination for processing copied message. T-Copy is a service where the message is delivered to the receiver without any dependency of the copied information.
 V.2 This is mandatory for the vendor messaging interface used by a copy destination for processing copied message. When the service is Y-Copy, the message is copied to the third party. The third party must authorise or refuse the message to be delivered to the receiver. The third party may refuse the delivery of the message to the receiver, in which case, the sender is informed via a refusal notification message.
 V.3 This is mandatory for the vendor messaging interface used by a copy destination for processing copied messages with CopyState=TCopyFallBack.
 V.4 This is mandatory for messaging interfaces used by third-party organisations running in a Y-copy mode. The authorisation itself is always created by a business application using the messaging interface.
 V.5 The approval or rejection notification contains a business message that will be provided to the participant instead of the system message. This is typically a pacs.002 message.
 V.6 The vendor messaging interface sends a retrieval request for sent and received traffic as copy-destination. This can be achieved by SWIFTNet Online Operation Manager (O2M). It processes input and output message.
 The vendor messaging interface must open its output channel in protocol level 2 in order to receive the retrieved input and output message.

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.