Swift Compatible Applications

Funds

Technical Validation Guide 2026

Version 1

February 2026

Public

Table of Contents

1 Preface 3

1.1 Introduction 3

1.2 Purpose and Scope 3

1.3 Target Audience 3

1.4 Related Documents 3

2 Technical Validation Process 4

2.1 Integration with Alliance Interfaces 4

2.1.1 Direct Connectivity 5

2.1.2 Confirmation of Test Execution & Evidence Documents 6

2.1.3 Verification of the Test Results 6

2.1.4 Qualification Criteria Verified 6

2.2 Standards MX support – Incoming Messages 7

2.2.1 Confirmation of Test Execution & Evidence Documents 8

2.2.2 Verification of the Test Results 8

2.2.3 Qualification Criteria Verified 8

3 Summary of Technical Validation 9

4 FAQ 10

Public

1 Preface

1.1 Introduction

Swift initiated the Swift Compatible Application programme to help application vendors into offering products that are compliant with the business and technical requirements of the financial industry.

Swift validates selected third-party business applications offered by Swift registered providers to ensure that they are aligned with well-defined requirements that are relevant to Swift standards, messaging, and connectivity.

Swift has engaged with Wipro (referred here after as the “Validation Service provider”) for performing the technical validation of the products applying for a Swift Compatible Application.

1.2 Purpose and Scope

The compatibility of the Swift Compatible Application Funds label is based on a set of pre-defined qualification criteria, which will be validated by means of a technical, functional and customer validation process.

The set of pre-defined qualification criteria is defined in the Swift Compatible Application Funds label criteria 2026.

This document focuses on the approach that a vendor application must follow to complete the technical validation against Swift Compatible Application Funds criteria.

In this document, a distinction is made between a **New Application** (vendors who apply for the first time for a specific product release) and an **Application Renewal** (for product releases that already received the Swift Compatible Application label in the past).

1.3 Target Audience

The target audience for this document is application vendors considering the compliance validation of their middleware suite / business application for the Swift Compatible Application Funds label. The audience must be familiar with Swift from a technical and a business perspective.

1.4 Related Documents

1) The Swift Compatible Application Programme overview provides a synopsis of the Swift Compatible Application programme, including the benefits to join for application vendors. It also explains the Swift Compatible Application validation process, including the technical, functional and customer validation.
2) The Swift Compatible Application Funds label criteria provide an overview of the criteria that the middleware suite / Funds business application must comply with to be granted the Swift Compatible Application.
3) FUNDS Service Description this service description provides an overview of the Funds solution including the investment funds distribution, the alternative investments, and the funds administration areas. This information includes the market background, an overview of the proposed solution, the message supported, a description of the key components, and a rulebook. This document is for all Swift users and Swift Partners that participate or plan to participate in Funds.

Public

2 Technical Validation Process

In this document, a distinction is made between new Swift Compatible Application applications and label renewal applications in terms of number of criteria verified and tests executed by the vendor. The Technical validation focuses on the message validation, standards support, connectivity to Alliance Interfaces and Reference Data Directory integration. The remaining label criteria are subjected to validation during the functional validation.

The following matrix explains the tests that will be performed by the vendor application.

Label Type	Depth of Testing	Message Validation	Standards Support	Integration with Alliance Interfaces	Reference Data
New Label	Comprehensive	✓	✓	✓	✓
Renewal	Delta only	✓	✓	✓	X

New Applicants will go through a complete technical validation against the criteria laid down in the Swift Compatible Application Funds criteria document.

The criteria that are verified include:

- Integration with Alliance interfaces
- Support of messaging services
- Support of Swift Standards
- Reference Data

Validation Test Bed

The vendor will need to set up and maintain 'a Swift test lab' to develop the required adaptors needed for validation and to perform the qualification tests. The Swift lab will include the Alliance Access or Alliance Cloud Interface as the direct connectivity to the Integration Test bed (ITB) (including SWIFTNet Link, VPN Box, RMA security and HSM box) and the subscription to the InterAct messaging services.

The installation and on-going maintenance of this Swift lab using a direct ITB connectivity is a pre-requirement for connectivity testing.

2.1 Integration with Alliance Interfaces

Requirement: The vendor will demonstrate the capability of the product to integrate with Swift Alliance Interfaces. When integrating with Alliance Access or Alliance Cloud, support for Release 7.7 or higher is mandated for the Swift Compatible Application in 2026.

Note: New label applicant vendors and vendors renewing their label application must exchange test messages using AFT or MQHA or SOAP.
Swift will only publish information for which evidences have been provided during the technical validation. In case the vendor application supports several of the above adapters, the vendor is required to provide the appropriate evidences for all of them.

Public

2.1.1 Direct Connectivity

Alliance Access 7.7 or higher or Alliance Cloud is the mandatory choice for connectivity. The table below specifies the adaptors and formats. The vendor is required to perform the connectivity testing with any one of the adaptors mentioned below:

Label Type	Alliance Access 7.7 or higher	
	Adaptor	Format
New and Renewal	AFT or MQHA or SOAP	XML v2

Label Type	Alliance Cloud	
	Adaptor	Format
New and Renewal	Swift Messaging API or AFT or MQHA	XML v2/RJE

The vendor needs to successfully connect to and exchange test messages with the Integration Test Bed (ITB).

The vendor must demonstrate the capability of their product to support MX Messaging Standards and its associated features (example: message validation).

2.1.1.1 Alliance Access Integration

- Testing for connectivity to Alliance Access Interface will be verified on the Swift Integration Test Bed (ITB) using Alliance Access Release 7.7 or higher.
- The vendor will demonstrate the capability of the product to integrate with the Alliance Access. The support for the one of the following adaptors will be demonstrated:
  - Automated File Transfer mode (AFT)
  - WebSphere MQ Host Adaptor (MQHA)
  - SOAP Host Adaptor (SOAPHA)

The vendor must connect to Swift ITB and receive Swift ACK / NAK notifications and delivery notifications

The Technical Validation documents for the AFT, MQHA and SOAPHA adaptors are available separately on Swift.com (Partner section).

2.1.1.2 Alliance Cloud Integration

The application provider must develop and test the Swift application integration using Alliance Cloud.
The direct connection from the business application to Alliance Cloud can be achieved using:
- The Swift Messaging API
- File connectivity through the Alliance Cloud Connectors on the Swift Micro gateway
- MQ connectivity through the Alliance Cloud Connectors on the Swift Micro gateway

For more information, see the Alliance Cloud Service Description.

Note: For Alliance Cloud connectivity, access to the Enterprise Testing Solution (ETS) is required to validate Swift messaging services within a controlled test environment.
For more information, see section Enterprise Testing Solution (ETS) of the Swift Compatible Application Programme Overview document

Public

Notes for vendors having ITB connectivity:

- The vendor must inform Swift and the Validation Service provider before starting the test execution through ITB
- The testing on ITB can start any time before the validation window allocated to the vendor. However, the entire testing on the ITB must be completed within the time window allotted to the vendor.
- The vendor application should generate a minimum of ten outbound test messages comprising of Subscription Order, Request for Order Status and Redemption Order. The list of mandatory support is provided in section 2.2 here below.
- The vendor application must exchange the Swift messages as per the XML v2 format.
- The sender destination used in the messages is the PIC (Partner Identifier Code) that was used by the application provider to install and license Alliance Access or Cloud. The receiver destination of messages must be the same PIC. Or simply stated messages should be sent to own vendor PIC
- The vendor must connect to Swift ITB, send MX messages, receive Swift ACK/NAK, Delivery Notification and properly reconcile them by updating the status of sent messages
- The test messages must be compliant to Funds version 5.6
- The vendor must inform Swift and the Validation Service provider about the completion of the test execution and provide evidence of testing though application event logs, transmitted messages and ACK / NAK received messages

2.1.2 Confirmation of Test Execution & Evidence Documents

After successful exchange of the MX test messages, the vendor should send the following evidences by email to the Validation Service provider:

- A copy of the XML v2 format files generated by the business application
- Application log / Screenshots evidencing the
  - processing of Swift messages
  - reconciliation of delivery notifications and Acknowledgements
- Alliance Access Event Journal Report and Message File spanning the test execution window
- Message Partner Configuration details

2.1.3 Verification of the Test Results

In order to issue the scorecard and necessary recommendation, the Validation Service provider will review the log files, event journal, the screenshots produced by the vendor to ascertain that:

- All messages are positively acknowledged by the Swift Network by reviewing the log files
- Test messages have been exchanged by the vendor over ITB
- Test messages adhere to the Swift format requirement (XML v2 formats).
- Application is able to reconcile technical messages

2.1.4 Qualification Criteria Verified

Sl. No	Swift Compatible Application Qualification Criteria		Pass / Fail Status
	Section Ref Number	Label Requirement	
1.	4.4.1	Alliance Access Integration – AFT / MQHA/SOAPHA	
2.		Alliance Access Integration Support – Release 7.7 or Higher	
3.		Alliance Access Integration – XML v2 Format	
4.		Alliance Cloud Integration- AFT / MQHA/Swift Messaging API	
5.	4.4.3	Standards MX Support for Outgoing Messages	
6.	4.5.1	Standards Release	
7.	4.6	Funds Rulebook	

Public

2.2 Standards MX support – Incoming Messages

Requirement: Depending on whether the vendor application is a business application or a middleware product, for Swift Compatible Application Qualification, the application must support the following categories of MX messages for SwiftNet Funds.

Note: Testing for message validation and standards support is applicable for new and renewal label applicant vendors.

Sr. No.	Message Category	Business Application	Middleware Product
1	Subscription orders / cancellations / confirmation	Mandatory (*)	Mandatory
2	Order status	Mandatory (*)	Mandatory
3	Redemption orders / cancellations / confirmations	Mandatory (*) (*)	Mandatory
4	Switches	Mandatory (*) (*)	Mandatory
5	Price reports	Mandatory (*) (*)	Mandatory
6	Statements	Mandatory (*) (*)	Mandatory
7	Switches	Mandatory (*)	Mandatory
8	Transfers	Optional	Optional
9	Account Management	Optional	Optional
10	Generic	Mandatory	Mandatory
11	Fund Reference Data	Optional	Optional
12	Portfolio Transfer	Optional	Optional

(*) Support for some of the messages in this category is mandatory.

- The Validation Service provider will send test messages for in scope MX messages for business application, the test messages will be supplied for the messages defined as mandatory in the criteria document.
- For middleware application, test messages will cover all the message categories defined in the above table.
- All test messages will be compliant to the MX validation and Funds Rule book in the “inward to the application” direction.
- The test messages will have generic test data for Accounts, Dates and BIC. The vendor can change the values / customise to their application needs. For ease of customisation, the test messages will be sent in a spread sheet format, with a facility to convert the output into a XML file for every test message.

File Naming Convention

- The files will be named Funds **nn**_MXValidation.xls, where “**nn**” will represent the Funds Version that will be tested. For example, for a file containing test messages for Funds, the file name will be **“Funds _5.8_MXValidation.xls”**
- The Validation Service provider will also send an MX Test Result Summary file in excel spread sheet format for the vendor to capture the test results into. The file name will be **xxxx**_Funds**nn**_MX_Validation_Test_Result.xls, where “**xxxx**” represents the vendor name and “**nn**” represents the Funds version.

Public

Processing the provided Swift Message Types

The vendor must input the above-mentioned files into the application and perform the business validations.

2.2.1 Confirmation of Test Execution & Evidence Documents

The vendor should send the following test evidences by e-mail to the Validation Service provider:

- Sample evidence demonstrating that the application has processed the test messages. This will be done by sending screenshots / log file / application generated reports.
- The MX Test Result Summary file, updated with the test results (Error Code and Error Line Number)

A sample of the spread sheet is provided here below.

Sl. No.	Message ID (MUR in Block 3)	Business Validation Results	Error Line Number	Error Description	Expected Error Code	Expected Error Line Number	Pass / Fail Status
1	setr.010.001.04	Pass	-				
2	setr.004.001.03	Error	11				

2.2.2 Verification of the Test Results

The Validation Service provider will analyse the log files, the screenshots produced by the vendor.

2.2.3 Qualification Criteria Verified

Sl. No	Swift Compatible Application Qualification Criteria		Pass / Fail Status
	Section Ref Number	Label Requirement	
1.	4.4.3	Standards – Support for Incoming Messages	
2.	4.5	Message Validation (Syntax and Semantic)	
3.	4.5.1	Standards Release – Funds version	

Public

3 Summary of Technical Validation

<table>
<tr><th colspan="2">Validation Activity</th><th>Label New</th><th>Label Renewal</th></tr>
<tr><td rowspan="2">Message Validation [Business Workflow]</td><td rowspan="2">Outgoing & Incoming</td><td>Business products – Validate Mandatory MX messages from
Subscription Orders /Cancellations /Confirmations
Order Status
Redemption Orders / Cancellations /Confirmations
Price Reports
Statements
Switches
Generic</td><td>Subscription Orders /Cancellations /Confirmations
Redemption Orders / Cancellations /Confirmations
Order Status
Price Reports
Switches</td></tr>
<tr><td>Middleware products – Validate Mandatory MX messages from
Subscription Orders /Cancellations /Confirmations
Order Status
Redemption Orders / Cancellations /Confirmations
Price Reports
Statements
Switches
Generic</td><td>Subscription Orders /Cancellations /Confirmations
Redemption Orders / Cancellations /Confirmations
Order Status
Price Reports
Switches</td></tr>
<tr><td rowspan="2">Standards</td><td>Standards Release</td><td colspan="2">SR 2026</td></tr>
<tr><td>Version
Rule Book Ref</td><td colspan="2">Funds 5.8</td></tr>
<tr><td rowspan="4">Connectivity</td><td>Alliance Access
7.7 or higher</td><td colspan="2">AFT or MQHA or SOAPHA</td></tr>
<tr><td>Alliance Cloud</td><td colspan="2">Swift Messaging API or AFT or MQ</td></tr>
<tr><td>Message Format</td><td colspan="2">XML v2</td></tr>
<tr><td>Local Authentication (LAU)</td><td>LAU is mandatory for 2026</td><td>NA</td></tr>
</table>

Public

4 FAQ

1. Would you (the Validation Service Provider) need to verify the detailed test scenarios, which we have created for the compliance validation test?

   You have to provide the following scope definitions before commencing the test window:
   - Number and Name of MX Messages supported by the application
   - List of scenarios identified for test execution
   - Test execution start date and end date

2. The Technical Validation Guide for Swift Compatible Application Funds Label talks about successful message exchange scenarios. Can you please confirm if additional Negative scenarios (i.e. invalid messages rejected within our application) need to be tested?

   You do not need to prepare negative [invalid] test messages. The purpose of informing you to generate MX messages is to verify if the application is able to generate and integrate with Alliance Access interface. The Event Journal File and Message File from Alliance Access covering the test execution window must be provided as test evidence.

3. We are already connected to SwiftNet via ITB. Can you please share the details of any configuration parameters/certificates/application etc. required to subscribe to the relevant service/CUG to exchange the messages for Funds compliance validation?

   You need to subscribe to SwiftNet Funds Services for Funds in ITB. The service name that must be used for SwiftNet funds messaging in ITB is Swift.if.ia!x. For more information about SwiftNet messaging environments and SwiftNet services, you may refer to the SwiftNet Service Description document.

4. Will Swift / Validation Service provider provide sample test messages?

   Test messages / scenarios will be sent to you before the start of the technical validation window.

Legal Notices

Copyright

Swift © 2026. 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. SC. The following are registered trademarks of Swift: Swift, the Swift logo, Sibos, Swift Net and Accord. Other product, service, or company names in this publication are trade names, trademarks, or registered trademarks of their respective owners.

***** End of document *****