Swift

Swift Compatible Applications

Payments

Technical Validation Guide 2023

Version 1

January 2023

Table of Contents

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 hereinafter as the “Validation Service Provider”) for performing the Technical Validation of the products applying for a Swift Compatible Application.

1.2 Purpose and Scope

Swift Compatible Application Payments 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 Payments is defined in the Swift Compatible Application Payments Label Criteria 2023

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

In this document, a distinction is made between a **New Application** (vendors who apply for Compliance validation 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 compatibility of their business application for the Swift Compatible Application Payments label. The audience must be familiar with the Swift portfolio from a technical and a business perspective.

1.4 Related Documents

1. Swift Compatible Application Programme Overview provides a synopsis of 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. Swift Compatible Application Payments label criteria provides an overview of the criteria that a Payments application must comply with to obtain the Swift Compatible Application

2 Technical Validation Process

In this document, a distinction is made between new Swift Compatible 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 have to be performed by the vendor application in 2023:

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

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

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 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 FIN 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, support for Release 7.6 or higher is mandated for the Swift Compatible Application in 2023.

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.

2.1.1 Direct Connectivity

Alliance Access 7.6 or higher is the mandatory 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 7.6 or higher	
	Adaptor	Format
New and Renewal	AFT	RJE or XML v2

	MQHA	RJE or XML v2
	SOAP	XML v2

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 FIN and Interact (Finplus) protocol 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 7.6 or higher
- The vendor should demonstrate the capability of the product to integrate with the Alliance Access with one of the following adaptors:
  - Automated File Transfer mode (AFT)
  - Web Sphere MQ Host Adaptor (MQHA)
  - SOAP Host Adaptor (SOAPHA)

The vendor must connect to the Swift ITB and receive Swift network 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).

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 outbound test messages comprising a mix of MT1xx, MT2xx and MT9xx and ISO 20022(MX),Finplus, Interact messages in Payments Criteria document
- The test messages must be compliant to Standards Release 2023.
- The vendor must request for delivery notification.
- The vendor application must exchange the Swift messages using Alliance Access in RJE or 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. 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 MT messages and MX(recommended), receive Swift ACK/NAK, Delivery Notification and properly reconcile them by updating the status of sent messages

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 message

2.1.2 Confirmation of Test Execution and Evidence Documents

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

- A copy of the MT\MX test messages in RJE / XML v2 format 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 the ITB
- Test messages adhere to the Swift format (RJE and /or XML v2 formats)
- Application is able to reconcile technical messages

2.1.4 Criteria Verified

Sl. No	Swift Compatible Application Qualification Criteria		Pass / Fail Status
	Section Ref Number	Label Requirement	
1.	3.4	Alliance Access Integration Support – Release 7.6 or higher	
2.		Alliance Access Integration – AFT / MQHA /SOAPHA Support	
3.		Alliance Access Integration – RJE / XML v2 Format	
4.	3.5	Standards Support	
5.	3.7	Message Validation Standards Release 2023	
6.		Network Validation Rules (MFVR)	

2.2 Message Validation and Standards Support

The vendor must demonstrate the application's capabilities to support SR2023, the Message Format Validation Rules (MFVR), MT Usage Guidelines and STP Guidelines

- Payments label requires demonstrable capability to support CBPR+ exchanges over FINplus, as well as generic support for ISO20022 MX messages (pacs, camt, pain, and head) over InterAct. Hence MX compatibility is required for 2023
- The Validation Service provider will send a set of valid inbound MT and MX test messages that need to be uploaded and processed
- The test messages will include the message types flagged as mandatory in appendix A and B (recommended) of the Swift Compatible Application Payments label criteria 2023 document
- The application must perform the business validations while parsing the incoming message
- User Header Block (Block 3) will contain a unique reference number in the form of a Message User Reference (MUR) for each test message. The MUR will consist of the MT numerical identification followed by test message sequence number.
- 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 single RJE formatted file for all the test messages or individual RJE formatted files for every test message

File Naming Convention

- The files will be named SR**yy**_PaymentsMTValidation.xls, where "**yy**" will represent the Year of the Standards Release. For example, for a file containing MT103 and MT103+ for Standards Release 2023, the file name will be **"SR23_PaymentsMTValidation.xls"**
- The Validation Service provider will provide an MT Test Result Summary file in excel spread sheet format that the vendor should use to capture test results. The file name will be

xxxx_SR**nn**_PaymentsMTValidation_Test_Result.xls, where “**xxxx**” represents the vendor name and “**nn**” represents the Standards Release.

Processing the provided Swift Message Types

The vendor must input the above mentioned files into the application and perform the business validations. For example, the application can reject a payment message, if the value date is less than current date or greater than 1 month from today’s date. Another example could be that the account is not serviced by the application.

The error listing provided by the application must be easily understandable by business users.

2.2.1 Confirmation of Test Execution and Evidence Documents

The vendor must send the following test evidences by email 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 MT and ISO 20022(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	10310000001	Pass	-				
2	10310000002	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 to ascertain that all messages are processed by the application and analyse the test result to provide scorecard and recommendation.

2.2.3 Testing of Outgoing Messages

The application must perform the following validations before forwarding the message to Alliance Access:

- MFVR (Character Set, Syntax, Code word, Semantic, MUG)
- MT Usage Rules listed in SR 2023
- STP Guidelines listed in SR 2023
- Cross-Border Payments and Reporting Plus (CBPR+) Usage Guidelines Generating Swift Messages
- The New vendor must generate at least one test message for each of the message types flagged as mandatory in appendix A and B of the Swift Compatible Application Payments criteria 2023 document. The vendor must generate these messages through the business application as outbound (“application to Alliance Access” direction) messages
- Test messages must be compliant to SR 2023
- The vendor application must wrap the Swift messages using RJE or XML v2 format

2.2.4 Confirmation of Test Execution and Evidence Documents

After successful exchange of the test Messages the vendor must send by email the following test evidence 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 reports
- A copy of the MT and MX(recommended) test messages in RJE / XML v2 format generated by the business application

2.2.5 Verification of the Test Results

The Validation Service provider will review the log files, the screenshots produced by the vendor to ascertain that all the messages are processed by the application and analyse the test result to build the scorecard and recommendation.

2.2.6 Qualification Criteria Verified:

Sl. No	SWIFT Compatible Application Qualification Criteria		Pass / Fail Status
	Section Ref Number	Label Requirement	
7.	3.5	Standards (Support for Incoming Message)	
8.	3.5	Standards (Support for Outgoing Message)	
9.	3.7	Message Validation (FIN and MX)	
10.		Standards Release 2023	
11.		Network Validated Rules	
12.		MT Usage Rules	
13.		STP Guidelines	
14.		Cross-Border Payments and Reporting Plus (CBPR+) Usage Guidelines	

2.2.7 Testing of Reference Data

Note: New label applicants need to test BIC, Bank Directory Plus and IBAN Plus directories. Reference data validation is optional for the renewal vendors who had successfully demonstrated this requirement during 2022 Compatible validation

Requirement: The vendor must demonstrate the application's capability to validate messages against the BIC, Bank Directory Plus and IBAN Plus directories. The vendor must use the sample BIC Directory, Bank Directory Plus and IBAN Plus available on http://Swiftref.Swift.com/resource-category/products

Testing for BIC, Bank Directory Plus and IBAN Plus Validation

The test scenario for testing the BIC, Bank Directory Plus and IBAN Plus are provided in the Swiftref Test scenario document.

- The test scenarios to be executed in the vendor application will cover:
  - BIC Validation
  - IBAN Structure validation
  - Deriving BIC / Clearing code

The test data and sample directory for testing the BIC, Bank Directory Plus and IBAN Plus table look-up and validation will be provided to the application vendor before the start of the technical validation window

The application vendor must input these transactions into their application and perform the reference data validation using the sample directories

Reference Data Validation

Based on the outcome of the validation with the reference data, the output of the test execution must be captured as listed below:

- For the search resulting in positive result, Swift messages must be generated in RJE format / XML v2 format
- For the search resulting in negative result, the screenshot displaying the warning / error notification

2.2.8 Confirmation of Test Execution and Evidence Documents

After successful execution of the test scenario for BIC, Bank Directory Plus and IBAN Plus reference data validation, the vendor must send the following test evidences to the Validation Service provider by email:

- Sample evidence demonstrating that the application has processed the BIC, Bank Directory Plus and IBAN Plus reference data validation. This will be done by sending screenshots or log file.
- A copy of the MT test messages in RJE / XML v2 format generated by the business application.

2.3 Verification of the Test Results

The Validation Service provider will validate the vendor output against the expected results and analyse the test result to build the scorecard and recommendation

2.4 Qualification Criteria Verified

Sl. No	Swift Compatible Application Qualification Criteria		Pass / Fail Status
	Section Ref Number	Label Requirement	
15.	4.1	BIC Directory	
16.	4.2	Bank Directory Plus	
17.	4.3	IBAN Plus	

3 Summary of Technical Validation

Validation Activity		Label NEW	Label RENEWAL
Message Validation	Outgoing	All mandatory MTs as per Appendix A of Criteria document	NA*
	Incoming	All mandatory MTs as per Appendix A of Criteria document	NA*
	ISO20022 payments (CBPR+ 2.1 and v2023 messages)	Support for following MX messages- as per Appendix B of Criteria document is mandatory: camt.029.001.XX,camt.052.001.XX,camt.053.001.XX, camt.054.001.XX,camt.055.001.XX,camt.056.001.XX, camt.058.001.XX,pacs.002.001.XX,pacs.003.001.XX, pacs.004.001.XX,pacs.008.001.XX,pacs.009.001.XX, pacs.010.001.XX,pain.001.001.XX,pain.002.001.XX, pain.008.001.XX All these CBPR+ 2.1 and v2023 Messages should be sent over Finplus and should be validated against the CBPR+ guidelines.	
Standards	Standards Release	SR 2023	
	Market Practice	NA	
Connectivity	Alliance Access 7.6 or higher	FIN\MX →AFT or MQHA or SOAPHA	
	Message Format	RJE and/or XML v2	
		Support for XML namespace prefix for safe processing of ISO 20022 XML messages	
	LAU (Local Authentication)	LAU support is mandatory	NA
Reference Data Directory	BIC, Bank Directory Plus and IBAN Plus	Scenario Based Testing	NA
	Integration	Screenshot Verification	NA

4 FAQ

1. Is it mandatory to provide error code and line number? Our application only gives textual description of the error encountered.

   It is good if you can populate the appropriate error code [which is a standardised way of reporting an error]. However, taking into account the limitation of the business applications, Swift can still accept the textual description of the encountered error coupled with the erroneous file impacted during the message validation.

2. In the test messages supplied to us for test execution, can we change the sender and receiver BICs (in header) so that we won’t need to change this setup in our system?

   The test messages are provided in an excel file. You can change / customise the values according to your requirement, before processing through the application.

3. How should the application perform validation on the “Copy of fields” in n9x messages?

   Swift does not validate the relationship between the copied field(s) and the original message. Even if not defined for the referenced message, any valid field except 77F, 77G or 77T (error code(s): T13) is accepted as the "Copy of fields".

   Swift only validates the syntax of a BIC used in the text of the appended message. A Test and Training destination may not be referenced by a LIVE user (error code(s): T27 T46).

   The values furnished in the Copy of field[s] must be a “valid” field. Since the relationship with the original message [furnished in Field 11S] and the copied fields[s] are not checked, the validation is performed individually for the “copy of field[s].

4. What is the use of the Test data directory containing BIC and Currency directories?

   All BIC and Currency data’s provided in the test data directory should be considered as good values and be updated in your system. While performing validation of the input messages, the data’s related to BIC and Currency should be validated against the list of data’s provided in the test data directory. If the value is not present in the directory then the message should be reported as FAIL.

5. What if my application only supports a subset of the mandatory message types mentioned in the Criteria document?

   The evaluation report will be based on the messages provided to us as test evidence. The list of Message Types for which the evidences have not been provided will be reported in the evaluation report enabling Swift to take a final decision.

6. Is it mandatory to provide the MT messages in RJE files?

   Yes, all MT Messages should be supplied in RJE file format only.

7. For generating the outgoing messages, is there any restriction that we must use only the BICs that are supplied to us for the incoming messages or can we use our own data?

   There is no requirement that you should use only these BIC for generating outgoing messages. Instead, you must use your PIC as sender and receiver of the Message, which makes it easier.

Legal notices

Copyright

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

***** End of Document *****