

# webMethods SWIFT Module 7.1 Enhancements and Fixes

October 2009

This file provides the enhancements and fixes contained in webMethods SWIFT Module 7.1. To view the readme file for this release, see the [Software AG Documentation Web site](#).

## Contents

[1.0 Fixes and Service Packs Included in This Release](#)

[2.0 Other Resolved Issues](#)

[3.0 Enhancements](#)

[4.0 Documentation Changes](#)

[5.0 Terminology Changes](#)

[6.0 Deprecated Items](#)

[7.0 Added, Removed, or Changed Built-in Services](#)

[8.0 Added, Removed, or Changed APIs](#)

## 1.0 Fixes and Service Packs Included in This Release

webMethods SWIFT Module 7.1 includes the following fixes and service packs made available since the last release. See the product documentation pages on the [Software AG Documentation Web site](#) for more information about service packs, and see the [webMethods Knowledge Base](#) for more information about fixes.

- SWIFTNET\_6-0-1\_SP1\_Fix8
- SWIFTNET\_6-0-1\_SP1\_Fix7
- SWIFTNET\_6-0-1\_SP1\_Fix6
- SWIFTNET\_6-0-1\_SP1\_Fix5
- SWIFTNET\_6-0-1\_SP1\_Fix4
- SWIFTNET\_6-0-1\_SP1\_Fix2

- SWIFTNET\_6-0-1\_SP1\_Fix1
- SWIFTNET\_6-0-1\_SP1
- SWIFTFIN\_6-1\_SP2\_Fix16
- SWIFTFIN\_6-1\_SP2\_Fix15
- SWIFTFIN\_6-1\_SP2\_Fix14
- SWIFTFIN\_6-1\_SP2\_Fix13
- SWIFTFIN\_6-1\_SP2\_Fix12
- SWIFTFIN\_6-1\_SP2\_Fix11
- SWIFTFIN\_6-1\_SP2\_Fix10
- SWIFTFIN\_6-1\_SP2\_Fix9
- SWIFTFIN\_6-1\_SP2\_Fix8
- SWIFTFIN\_6-1\_SP2\_Fix7
- SWIFTFIN\_6-1\_SP2\_Fix6
- SWIFTFIN\_6-1\_SP2\_Fix5
- SWIFTFIN\_6-1\_SP2\_Fix4
- SWIFTFIN\_6-1\_SP2\_Fix3
- SWIFTFIN\_6-1\_SP2\_Fix2
- SWIFTFIN\_6-1\_SP2\_Fix1
- SWIFTFIN\_6-1\_SP2

## **2.0 Other Resolved Issues**

This section lists other issues that were resolved in this release.

- None

## 3.0 Enhancements

- Consolidated SWIFT solution:  
webMethods SWIFT Module 7.1 combines SWIFT FIN (MT) and SWIFTNet support into a single product enabling unified connectivity for both SAA (SWIFT Alliance Access) and SAG (SWIFT Alliance Gateway).
- User interface for configuration and administration:  
webMethods SWIFT 7.1 Module now provides a graphical user interface to configure connectivity with SAG.
- Support the exchange and validation of MX and MT Messages in XMLv2 format:  
webMethods SWIFT Module 7.1 supports the exchange of MX and MT messages over the SWIFT Network using SWIFT Alliance Access. SWIFT Module sends and receives MT and MX messages to SAA wrapped in XML v2 data format. Structural and semantic validation of all MX and MT message types, including the latest 2009 MT message types, is provided.
- Support for FpML message exchange and validation over SWIFTNet:  
webMethods SWIFT Module 7.1 transfers FpML-compliant messages in XML format over SWIFTNet using the SWIFTNet InterAct store-and-forward messaging service. Message validation, as well as built-in support for schema validation and semantic validation of all XML messages is provided.
- Support for BICPlusIBAN directory:  
webMethods SWIFT Module 7.1 enables lookup and validation of MT and MX messages against the BICPlusIBAN directory. The module also validates IBAN-BIC combinations, enables translation of BICs into national bank/clearing codes, and derives the BIC from the IBAN code when the BIC is missing.
- Support for reconciliation of MT and MX messages:  
webMethods SWIFT Module 7.1 now provides out of the box support for reconciling all levels of notifications, including ACK and NAK acknowledgments with the original SWIFT messages that are exchanged over SAA in XML v2 format.
- Support for MQHA Transport for FIN MT Messages, XMLv2 (MT, MX and FpML), SWIFTNet InterAct and File Act Services:  
MQSA has been replaced by MQHA from SWIFT. webMethods SWIFT Module 7.1 supports plain text MT messages and XML v2 messages. For transporting XML v2 messages to SAA over MQHA, SWIFT Module 7.1 generates the required header block for XML v2 messages. webMethods SWIFT 7.1 Module also supports SWIFTNet InterAct and FileAct messaging services in Real Time and Store and Forward modes using MQHA for SAG.

## 4.0 Documentation Changes

This section describes significant changes to the documentation, such as the addition, relocation, or removal of product guides, online help, chapters, or other major content.

- The *webMethods SWIFT Module Installation and User's Guide* is a new guide. It contains concept information, and installation and upgrade instructions. It also explains how to send and receive SWIFT FIN messages over SWIFT Alliance Access (SAA), how to send and receive FileAct and InterAct messages over SWIFT Alliance Gateway (SAG) using the RAHA or MQHA transport, and how to use the SWIFT File Transfer Adapter (FTA) to transfer files over SWIFTNet.
- Samples are available during product installation, and they are documented in the *webMethods SWIFT Module Installation and User's Guide*.
- Product documentation is available on the [Software AG Documentation Web site](#).

## 5.0 Terminology Changes

### Old Term

BIC or BIC Plus

### New Term

BICPlusIBAN

## 6.0 Deprecated Items

### Deprecated Item

WmUNIFI Package

### Description

The WmUNIFI package in webMethods SWIFT FIN Module contains the services that handle the MX messages. The package contains the following services:

- `wm.unifi.convertXMLtoIData`
- `wm.unifi.transportToSAA`
- `wm.unifi.utils.validateRules`

## 7.0 Added, Removed, or Changed Built-in Services

### Added Service

`wm.fin.format:convertISMTDocToFINFormat`

### Description

Creates a FIN formatted block 4 of an MT message from the input backend MT IS doc type.

`wm.fin.format:convertFINBlock4ToISDoc`

Maps the contents of a FIN formatted block 4 of an MT message from the backend MT IS document type.

`wm.fin.sepa:checkOperationalReadiness`

Validates a BIC's operational readiness to ensure that a BIC is ready to receive SEPA payment instructions for a particular scheme.

## Added Service

wm.fin.sepa:getAvailablePaymentChannels

wm.fin.sepa:getOtherPaymentChannel

wm.fin.sepa:getPreferredPaymentChannel

wm.fin.sepa:validateAdherenceStatus

wm.unifi.validation:validateBEI

wm.unifi.validation:validateBIC

wm.unifi.validation:validateCountryCode

wm.unifi.validation:validateCurrencyCode

wm.unifi.validation:validateIBAN

wm.unifi.validation:validateMXMsg

wm.xmlv2.dev:createSWIFTItems

wm.xmlv2.notifications:handleDeliveryNotifications

wm.xmlv2.process:createSAADoc

wm.xmlv2.process:getInboundMessageType

wm.xmlv2.process:outbound

## Description

Identifies the available payment channels for a BIC code.

Identifies other payment channels available through an intermediary institution that the financial institution with the input BIC code can use for a payment.

Determines whether the counterpart financial institution has specified a preferred payment channel for receiving payment instructions.

Validates a BIC's adherence status to confirm that an institution has signed an adherence agreement for a particular scheme, and is published in the EPC Register of Participants (that is, the adherence database).

Checks if the BEI code exists in the webMethods SWIFT database.

Checks if the BIC code exists in the webMethods SWIFT database.

Checks if the Country code exists in the webMethods SWIFT database.

Checks if the Currency code exists in the webMethods SWIFT database.

Checks if the IBAN is valid.

Performs different validations of the XML v2 MX message.

Creates TN items for a particular MT or MX message type.

Processes an incoming document as follows:

- Extracts the MIR from the delivery notification, and searches the documents in the Trading Networks database for the same Sender Reference as the MIR extracted from the notification. When the search is successful, the service relates the delivery notification to the search results message as a Delivery Notification.

- Changes the status of the delivery notifications in Trading Networks to Reconciled.

A utility service that converts a DataPDU XML element into an IS DataPDU.

Takes a Data PDU in XML format as input and processes it to determine the type of the incoming documents from SAA.

Processes an outbound TN bizdoc object.

## Added Service

wm.xmlv2.process:processInbound

wm.xmlv2.process:reconcileInboundDocuments

wm.xmlv2.transport:submitDataPDU

wm.xmlv2.utils:encodeBlock4

wm.xmlv2.utils:encodeFinMessage

wm.xmlv2.utils:formatXMLV2

wm.xmlv2.utils:getDataPDUsFromFile

wm.xmlv2.utils:putInBatchFile

com.wm.common.CacheHandler.getContextForMessagePartner

com.wm.common.CacheHandler.saveContextForMessagePartner

com.wm.common.services.createTNdocForMQResponse

com.wm.common.services.getEnvAndXMLReqFromMQResponse

com.wm.common.services.getSagEnv

com.wm.common.services.getSagReqEnvAsString

## Description

Processes all inbound documents from SAA to SWIFT Module. This service does the preliminary processing of the inbound Data PDU and submits it to Trading Networks for further processing.

Reconciles all incoming notifications from SAA. This service identifies the category of the notification and relates the notification to the original document based on the Sender Reference of the document attribute. It also updates the status of the original document, depending on the incoming notification.

Takes a DataPDU in XML format as input and after persisting the DataPDU in Trading Networks, it routes the resultant bizdoc for further processing.

Encodes block 4 of an MT message to a base64 string.

Encodes block 4 of a FIN message to a base64 string. It extracts the block 4 contents of the FIN message from the input FIN message and encodes it to a base64 string.

Formats the XML contents of a Data PDU to a proper XML v2 format.

Takes a batch file containing the Data PDUs as input and extracts all the Data PDUs from the file. Also, generates a string array of the Data PDUs.

Creates a batch file of the Data PDUs that are submitted to SAA for processing.

Retrieves the security context for the specified message partner from the shared cache.

Saves the security context for the message partner in the cache after the initialization process is complete.

Creates a TN document type for the response received from MQHA.

Breaks down the MQ response into a SAG envelope and a response message.

Retrieves the SAG envelope from the bizdoc.

Creates a string representation of the SAG response envelope.

## Added Service

Added Service	Description
com.wm.common.services.getXMLData	Retrieves the SWIFT request or response XML data from the TN BizDocEnvelope. The input bizdoc provided to this service must have <i>xmldata</i> as one of the content parts.
com.wm.common.services.handleContextResponse	Saves the security context for the message partner obtained in the context response from SWIFT, provided such a context has been successfully created.
com.wm.common.services.submitContextResponse	Routes the MQ response to Trading Networks.
om.wm.common.services.submitMQResponseToTN	Submits the MQ response to Trading Networks. This service adds <i>sagenv</i> , <i>correlationId</i> and <i>msgId</i> as content parts of the bizdoc created for the XML data.
com.wm.common.services.submitRequestToTN	Submits any outgoing request to SWIFT to Trading Networks.
com.wm.common.Util.createSagReqEnv	Creates a SAG request envelope for the message partner and the security context.
com.wm.common.Util.invokeMQService	Invokes the MQ service identified by the service name.
com.wm.common.Util.resolveNameSpaceAndEntity	Resolves any namespace prefixes for the XML request.
wm.swiftnet.client.mq:processRequest	Sends the XML request to SWIFT over the MQ transport. This service creates the SAG envelope to be submitted to SWIFT.
wm.swiftnet.client.mq:sendToMQ	Gets the SAG envelope and XML Data from the Trading Networks BizDocEnvelope and creates an MQ request to be sent to SWIFT over the MQ transport.
wm.swiftnet.client.transport.FTA:generateCompanionFile	Generates a companion .par file with the provided inputs.
wm.swiftnet.client.transport.FTA:scanForReports	Scans the input directory for the report files and submits them to Trading Networks.
wm.swiftnet.client.transport.FTA:submitToTN	Submits the file to Trading Networks.
wm.swiftnet.server.mq.inbound.getInfoFromNotificationDoc	Fetches the MQ message body and message ID for the document.
wm.swiftnet.server.mq.inbound.handleSWIFTRrequest	Retrieves the message body and message ID from the incoming document. The service must be configured for the notification document triggered for an inbound request.
wm.swiftnet.server.mq.trp.respond	Sends the server responses back to SWIFT through the MQ transport.

**Added Service**

wm.swiftnet.server.mq.util.sendToMQ

**Description**

Sends the server requests to SAG through the MQ transport.

**Removed Service**

None

**Description****Changed Service**

None

**Description**

## 8.0 Added, Removed, or Changed APIs

- None

ESTD-SWIFT-E&F-71-20091005