

# webMethods SWIFT Module 7.1 SP1 Readme

## July 2011

This file contains important information you must read before using webMethods SWIFT Module 7.1 SP1. You can find additional information about SWIFT Module 7.1 SP1 on the [Software AG Documentation Web site](#). In addition, the following suite-related information is also available at that location:

Product Documentation (includes installation and upgrade information)  
System Requirements Documentation

You can find the following information in the Products section of the [Empower Web site](#):

Security Information  
Globalization Information

## Contents

[1.0 Critical Information](#)

[2.0 Known Issues](#)

[3.0 Usage Notes](#)

[4.0 Fixes and Enhancements](#)

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

[4.2 Resolved Issues](#)

[4.3 Enhancements](#)

[4.4 Documentation Changes](#)

[4.5 Terminology Changes](#)

[4.6 Deprecated Items](#)

[4.7 Added, Removed, Deprecated, or Changed Built-In Services](#)

[5.0 Copyright Information](#)

[6.0 Support](#)

## 1.0 Critical Information

This section lists the critical issues that were known when this readme was published. For critical information

found later, go to the Knowledge Center portion of the [Empower Product Support Web site](#).

None.

## 2.0 Known Issues

This section lists issues that were known when this readme was published. For known issues found later, go to the Knowledge Center portion of the [Empower Product Support Web site](#).

None.

## 3.0 Usage Notes

This section highlights product limitations and changes to product behavior introduced in this release.

- For webMethods SWIFT Module 7.1 SP1 to operate fully, you must install certain versions of accompanying webMethods products and required fixes for those products. For details, see the *eStandards Modules System Requirements* document on the [Software AG Documentation web site](#).
- To use the SWIFT 2011 messages, use the nov11 version of the message types.

## 4.0 Fixes and Enhancements

This section describes the fixes and enhancements included in webMethods SWIFT Module 7.1 SP1.

### 4.1 Fixes and Service Packs Included in This Release

This section lists the fixes and service packs for previous releases that have been included into the current release. If only some issues in a fix or service pack have been included, that partially included fix or service pack is not listed here, and those issues are listed separately in [Section 4.2 Resolved Issues](#). See the Knowledge Center section of the [Empower Product Support Web site](#) for more information about fixes and service packs.

- 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\_SP4\_Fix2
- SWIFTFIN\_6-1\_SP4\_Fix1
- SWIFTFIN\_6-1\_SP4
- EFM\_7.1\_Fix9
- EFM\_7.1\_Fix8
- EFM\_7.1\_Fix7
- EFM\_7.1\_Fix6
- EFM\_7.1\_Fix5
- EFM\_7.1\_Fix4
- EFM\_7.1\_Fix3
- EFM\_7.1\_Fix2
- EFM\_7.1\_Fix1

## **4.2 Resolved Issues**

This section lists issues that were resolved in this release and that are not included in the fixes and service packs listed in [Section 4.1 Fixes and Service Packs Included in This Release](#). These issues might have been resolved in a fix or service pack that was not included in its entirety into this release, or they might have been resolved since the last fix or service pack was released.

- EFM-345  
SWIFT Module returns a C9 rule violation if the field 71G is not provided in message type 103STP. This error occurs because during network validation of 103STP message types, the network validation service, `wm.fin.doc.nov10.cat1:MT103STPNetworkValidationRules`, applies the C9 validation rule on field 71G, even if the field is not provided. Field 71G is an optional field, and therefore, the service should skip C9 rule validation on this field when it is not provided in the message.  
This issue is resolved. The validation service, `wm.fin.doc.nov10.cat1:MT103STPNetworkValidationRules`, now correctly applies C9 validation on field 71G only when it is included in the message.
- EFM-344

When converting MT360 FIN messages into IData, SWIFT Module generates the word "ERROR" preceding fields 14F and 22C. This error occurs because the fields 14F and 22C are not correctly defined in the DFD MT definition file (dfdmt000.xml) as "24x" and "4!a2!c4!n4!a2!c," respectively.

This issue is resolved. The fields 14F and 22C are now correctly defined in the DFD MT definition file to conform to the SWIFT specification.

- EFM-343

When validating MT360 messages, SWIFT Module generates the IS document with field 14J defined as "UNKNOWN." This error occurs because the MT360 DFD file definition (dfdmt000.xml) that SWIFT Module uses to generate the IS document incorrectly defines field 14J as unknown, instead of as type 5a.

This issue is resolved. The MT360 DFD file definition (dfdmt000.xml) now correctly defines field 14J as type 5a in the IS document, in accordance with the SWIFT specification.

- EFM-342

When performing network validation on MT535 messages, SWIFT Module returns a C3 validation error even if the field 22F::STTY//ACCT is not provided in the message. This error occurs because the network validation service, wm.fin.doc.nov10.cat5:MT535NetworkValidationRules, incorrectly applies a C3 rule validation on the optional field 22F::STTY//ACCT.

This issue is resolved. The validation service wm.fin.doc.nov10.cat5:MT535NetworkValidationRules now correctly applies C3 rule validation only when a MT535 message contains the field 22F::STTY//ACCT.

- EFM-341

When validating MT502 messages, SWIFT Module returns a validation error if field 19A is populated with the "ISDI" tag. This error occurs because the SWIFT MT definition file for message type 502 (swiftmt502.xml) omitted the definition for the ISDI tag in field 19A.

This issue is resolved. The SWIFT MT definition file for message type 502 (swiftmt502.xml) now includes the ISDI tag for field 19A.

- EFM-340

When validating MT537 messages, SWIFT Module returns a C8 network validation error for valid messages. This error occurs because the validation service, wm.fin.doc.nov10.cat5:MT537NetworkValidationRules, ignores a precondition in rule C8 that states that the conditional rule does not apply if field 25:D contains the data source scheme.

This issue is resolved. wm.fin.doc.nov10.cat5:MT537NetworkValidationRules now correctly applies rule C8 only when field 25:D does not contain the data source scheme.

- EFM-339

No search results returned from the "Search BIC Info" link.

The "Search BIC Info" link in the SWIFT Module user interface, does not return any search results when searching for BIC information. This error occurs because the "Search BIC Info" link calls a service that searches the BICPlusIBAN table instead of the BIC table.

This issue is resolved. The "Search BIC Info" link in the SWIFT Module user interface now invokes the service that searches the BIC table.

- EFM-313

In the IS schema for message type 564 (MT564), version nov10, the fields 98A:LOTO, 98B:LOTO, and 98C:LOTO are missing from the CADETL sequence in the SWIFT MT definition file (swiftmt564.xml).

This issue is resolved. The SWIFT MT definition file now includes the tags 98A:LOTO, 98B:LOTO, and

98C:LOTO.

- EFM-287  
Data parsing errors for SWIFT MT082 system messages.  
The `wm.fin.format:convertFINToIData` service throws the following error when attempting to convert a MT082 message,  
"Could not run 'convertFINToIData'. com.wm.app.b2b.server.ServiceException:  
com.wm.app.b2b.server.ServiceException: Error in parsing the object. Mandatory block SBA not found or could not be added!"  
This error occurs because the block SBA was not correctly defined for MT082 in the SWIFT MT definition file (`swiftmt082.xml`).  
This issue is resolved. The SBA block definition in the SWIFT MT definition file (`swiftmt082.xml`) is corrected and now conforms to the SWIFT specification.
- EFM-293  
When validating network rules for a MT543 message type, SWIFT Module throws an E91 validation error when the message violates network rule C8.  
The `wm.fin.doc.nov09.cat5:MT543NetworkValidationRules` service validates network rules in MT543 messages. When this service encounters a C8 rule violation, it throws the wrong error code, returning E91 instead of E93.  
This issue is resolved. The network validation rule service `wm.fin.doc.nov09.cat5:MT543NetworkValidationRules` now returns error code E93 when network rule C8 is violated.

## 4.3 Enhancements

This section lists the enhancements introduced in this release.

- Support for SWIFT 2011 messages:  
SWIFT Module now supports SWIFT 2011 messages. To use the SWIFT 2011 messages, use the `nov11` version of the message types.
- Support for SWIFT SDK:  
SWIFT Module now supports SWIFT SDK by providing MT XML schema definitions (XSDs) and Java services for performing MT message conversion from flat file into XML format or from XML into flat file format. The module also validates the XML against the XML schema definition (XSDs) using Java services.
- Consolidated SWIFT solution:  
SWIFT Module now 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:  
SWIFT 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:  
SWIFT Module supports the exchange of MX and MT messages over the SWIFT Network using SWIFT Alliance Access (SAA). SWIFT Module sends and receives MT and MX messages to SAA wrapped in XML

v2 data format. SWIFT Module also provides structural and semantic validation of all MX and MT message types, including the latest 2011 MT message types.

- Support for FpML message exchange and validation over SWIFTNet:  
SWIFT Module now transfers FpML-compliant messages in XML format over SWIFTNet using the SWIFTNet InterAct store-and-forward messaging service. SWIFT Module supports message validation, as well as built-in support for schema validation and semantic validation of all XML messages.
- Support for BICPlusIBAN directory:  
SWIFT Module 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:  
SWIFT Module 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:  
SWIFT has replaced MQSA with MQHA. SWIFT Module now supports plain text MT messages and XML v2 messages. For transporting XML v2 messages to SAA over MQHA, SWIFT Module generates the required header block for XML v2 messages. SWIFT Module also supports SWIFTNet InterAct and FileAct messaging services in real time and store-and-forward modes using MQHA for SAG.

## 4.4 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.

- *webMethods SWIFT Module Installation and User's Guide* 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. This guide also contains information about the support for SWIFT SDK and provides details for using the MT and MX schemas.
- *webMethods SWIFT Module Sample Package User's Guide* describes the sample services that you can use as models for creating your own services. The samples package and user's guide is available for download at Developer Community for webMethods, in the Code Samples area, at <http://communities.softwared.com/ecosystem/communities/public/Developer/webmethods/products/esb/>.

## 4.5 Terminology Changes

This section lists terminology that has changed since the last full release of the product.

**Old Term**

BIC or BIC Plus

**New Term**

BICPlusIBAN

**4.6 Deprecated Items**

This section lists product functionality that has been deprecated since the last full release of the product. Deprecated Built-In services and APIs are listed in the relevant sections that follow this one.

**Deprecated Item**

WmUNIFI Package

**Description**

The WmUNIFI package in SWIFT Module contains the services to handle MX messages. The package contains the following services:

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

**4.7 Added, Removed, Deprecated, or Changed Built-In Services**

This section lists Built-In services that have been added, removed, deprecated, or changed since the last full release of the product.

**Added Service**

wm.fin.map:mapApplicationBlockHeader

**Description**

This service maps the input variables into a default FIN application header.

wm.fin.map:mapBasicBlockHeader

This service maps the input variables into a default FIN application header.

wm.fin.map:mapUserBlockHeader

This service maps the input variables into a default FIN application header.

wm.fin.transport.AFT:processIncomingFile

This service processes the SWIFT FIN flat file message by breaking the batch input file into the individual SWIFT FIN messages. It then invokes wm.fin.trp:receiveMessage to process each of the SWIFT FIN messages in the inbound batch file.

wm.fin.transport.AFT:processOutgoingFile

This service generates a unique file name and writes the outbound SWIFT message to the file in the folder specified in the TPA.

## Added Service

wm.fin.transport.MQSeries:getMQSeriesListenerService

## Description

This service retrieves SWIFT FIN messages from a specified MQSeries queue, removes extraneous information in the SWIFT message, and publishes the actual SWIFT message for further processing by the service, wm.fin.trp:receive.

wm.fin.transport.MQSeries:putMessage

This service invokes the “put” message handler service (the user-created service specified in the corresponding message TPA), and puts the outbound SWIFT message in a MQ Series queue.

wm.fin.trp:receiveMessage

The service, wm.fin.trp:FINInboundMessageTrigger triggers this service to run. This service receives an incoming FINInboundMessage IData, parses it into a record, and sends it to Trading Networks for further processing.

wm.fin.trp:sendMessage

This service formats IData into a SWIFT FIN message, persists the message in Trading Networks, and validates it. Once validated, the service sends the message to SAA, using the transport protocol configured in the TPA for the corresponding message type.

wm.fin.map.mapOutboundMessage.

This service maps the input variables into *finIData*.

com.wm.common.CacheHandler.getContextForMessagePartner

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

com.wm.common.CacheHandler.saveContextForMessagePartner

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

com.wm.common.services.createTNDocForMQResponse

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

com.wm.common.services.getEnvAndXMLReqFromMQResponse

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

com.wm.common.services.getSagEnv

Retrieves the SAG envelope from the bizdoc.

com.wm.common.services.getSagReqEnvAsString

Creates a string representation of the SAG response envelope.

com.wm.common.services.getXMLData

Retrieves the SWIFT request or response XML data from the Trading Networks BizDocEnvelope. The input bizdoc provided to this service must have *xmldata* 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.



Added Service	Description
com.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:migrateServices	Replaces old service names with the corresponding new service names using the service map maintained by SWIFT Module.
com.wm.common.Util.resolveNameSpaceAndEntity	Resolves any namespace prefixes for the XML request.
wm.fin.bic:deriveBICfromIBAN	Using IBAN as input, provides a valid BIC code based on the logic specified by SWIFT.
wm.fin.bic:generateIBAN	Using the input parameters, generates the IBAN based on the logic specified by SWIFT.
wm.fin.bic:insertISList	Imports IS lists into the database.
wm.fin.bic:insertSRLList	Imports SEPA Routing list into the database.
wm.fin.bic:validateBankID	Validates the National Code for a financial institution.
wm.fin.bic:validateBICCode	Validates the BIC code for a financial institution.
wm.fin.bic:validateBICIBAN	Validates the BIC code and IBAN combination for a financial institution.
wm.fin.format:convertFINBlock4ToISDoc	Maps the contents of a FIN block 4 MT message from the back-end MT IS document type.
wm.fin.format.convertISMtDocToFINFormat	Creates block 4 of a FIN message in MT format using the back-end MT IS document type.
wm.fin.map:mapUACK	Maps the SWIFT acknowledgement to finIData.
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.
wm.fin.sepa:getAvailablePaymentChannels	Identifies the available payment channels for a BIC code.
wm.fin.sepa:getOtherPaymentChannel	Identifies the available payment channels for the financial institution to use for a payment.
wm.fin.sepa:getPreferredPaymentChannel	Determines whether the counterparty financial institution has specified a preferred payment channel for receiving payment instructions.

## Added Service

Added Service	Description
wm.fin.sepa.validateAdherenceStatus	Validates a BIC's adherence status, confirming that an institution has signed an adherence agreement and has published it in the EPC Register of Participants (that is, the adherence database).
wm.fin.utils.generateUniqueIdentifier	This service generates a unique identifier, which SWIFT Module uses to reconcile notifications from SAA.
wm.sdk.docgenerator.createMTISDocFromSchema	This service creates IS document types and IS schema for the corresponding MT schemas.
wm.sdk.docgenerator.createMXISDocFromSchema	This service creates IS document types and IS schema for the corresponding MX schemas.
wm.sdk.fin.converter.convertMTBlock4ToMTXML	Converts block 4 of the MT flat file into XML format.
wm.sdk.fin.converter.convertMTFlatFileToMTXML	Converts the entire flat file MT message into a MT XML message.
wm.sdk.fin.converter.convertMTXMLblock4ToMTFlatFile	Converts block 4 of the MT XML message into flat file format.
wm.sdk.fin.converter.convertMTXMLToMTFlatFile	Converts the entire MT XML message into a MT flat file message.
wm.sdk.fin.validator.validateMTXML	Validates any MT XML message against the SWIFT SDK MT schema.
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.property.getProperty	Retrieves the value of the specified property from the <i>IntegrationServer_directory\packages\WmSWIFTNetClient\config\snl.cnf</i> file.
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.handleSWIFTRequest	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.

## Added Service

Added Service	Description
wm.swiftnet.server.mq.trp.respond	Sends the server responses back to SWIFT through the MQ transport.
wm.swiftnet.server.mq.util.sendToMQ	Sends the server requests to SAG through the MQ transport.
wm.swiftnet.server.property:setProperty	Sets the property specified in the input.
wm.unifi.validation:validateBEI	Verifies that the BEI code exists in the SWIFT Module database.
wm.unifi.validation:validateBIC	Verifies that the BIC code exists in the SWIFT Module database.
wm.unifi.validation:validateCountryCode	Verifies that the Country code exists in the SWIFT Module database.
wm.unifi.validation:validateCurrencyCode	Verifies that the Currency code exists in the SWIFT Module database.
wm.unifi.validation:validateIBAN	Verifies that the IBAN is valid.
wm.unifi.validation:validateMXMsg	Performs validations on the XML v2 MX message.
wm.xmlv2.dev:createSWIFTItems	Creates Trading Networks items for a particular MT or MX message type.
wm.xmlv2.notifications:handleDeliveryNotifications	Processes an incoming document as follows: <ul style="list-style-type: none"><li data-bbox="808 1003 1471 1249">- 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.</li><li data-bbox="808 1262 1422 1333">- Changes the status of the delivery notifications in Trading Networks to "Reconciled."</li></ul>
wm.xmlv2.process:createSAADoc	A utility service that converts a Data PDU XML element into an IS Data PDU.
wm.xmlv2.process:getInboundMessageType	Processes an incoming Data PDU in XML format, provided as input from SAA, and determines the document type.
wm.xmlv2.process:outbound	Processes an outbound Trading Networks bizdoc object.
wm.xmlv2.process:processInbound	Processes all inbound documents from SAA to SWIFT Module. This service performs the preliminary processing of the inbound Data PDU and submits it to Trading Networks for further processing.
wm.xmlv2.process:reconcileInboundDocuments	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.

## Added Service

wm.xmlv2.transport:submitDataPDU

wm.xmlv2.utils:encodeBlock4

wm.xmlv2.utils:encodeFinMessage

wm.xmlv2.utils:formatXMLV2

wm.xmlv2.utils:getDataPDUsFromFile

wm.xmlv2.utils:putInBatchFile

## Description

Processes a Data PDU in XML format as input: persisting the Data PDU in Trading Networks, and routing 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 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.

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

## Removed Service

None

## Replacement Service (if applicable)

## Deprecated Service

wm.unifi.convertXMLtoIData

This service is deprecated. It converts an XML message into IData and validates it against schema and the generic rule book.

wm.unifi.transportSAA

This service is deprecated. It sends an MX/MT message to SAA in XML v2 format.

wm.unifi.utils.validateRules

This service is deprecated. It validates a MX message for schema conformance and proper rule definition as outlined in the SWIFT generic rule book.

wm.fin.transport.AFT:AFTOutboundTrigger

This trigger is deprecated. WmSWIFTSamples provide new samples to use FIN messages without this trigger.

wm.fin.transport.MQSeries:MQSeriesPutTrigger

This trigger is deprecated. WmSWIFTSamples provide new samples to use FIN messages without this trigger.

wm.fin.transport.Test:FINSampleInboundMessage

This document is deprecated. The new FIN samples do not require this document.

wm.fin.transport.Test:FINSampleInboundMessageTrigger

This trigger is deprecated. The new FIN samples do not require this test service.

## Deprecated Service

wm.fin.transport.Test:FINSampleOutboundMessageTrigger

wm.fin.transport.Test:processFinMsg

wm.fin.trp:FINInboundMessageTrigger

wm.fin.map:mapApplicationHeader

wm.fin.map:mapBasicHeader

wm.fin.map:mapUserHeader

wm.fin.transport.AFT:processInboundFile

wm.fin.transport.AFT:processOutboundFile

wm.fin.transport.MQSeries:getListenerService

wm.fin.transport.MQSeries:put

wm.fin.trp:receive

wm.fin.trp:send

wm.fin.map.mapOutbound

## Replacement Service (if applicable)

This trigger has been deprecated. The new FIN samples provided do not require this test service.

This trigger is deprecated. The new FIN samples do not require this test service.

This trigger has been deprecated. The new FIN samples provided do not require this test service.

wm.fin.map:mapApplicationBlockHeader

wm.fin.map:mapBasicBlockHeader

wm.fin.map:mapUserBlockHeader

wm.fin.transport.AFT:processIncomingFile

wm.fin.transport.AFT:processOutgoingFile

wm.fin.transport.MQSeries:getMQSeriesListenerService

wm.fin.transport.MQSeries:putMessage

wm.fin.trp:receiveMessage

wm.fin.trp:sendMessage

wm.fin.map.mapOutboundMessage

## Changed Service

None

## Description

## 5.0 Copyright Information

Copyright © 2011 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://documentation.softwareag.com/legal/>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices and license terms,

please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products." This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

## 6.0 Support

To obtain support for webMethods Product Suite components, refer to the following resources:

### [Empower Product Support Web site](#)

Visit the Empower Web site to learn about support policies and critical alerts, read technical articles and papers, download products, fixes, and service packs, submit feature/enhancement requests, and more.

### [Software AG Developer Community for webMethods](#)

Visit the Software AG Developer Community to access additional articles, demos, and tutorials, technical information, samples, useful resources, online discussion forums, and more.

ESTD-SWIFT-RM-71SP1-20110617