Gentran Integration Suite™

Using SWIFTNet with Gentran Integration Suite

Version 4.2

Build 4207 or Higher

Sterling Commerce
An IBM Company

© Copyright 2007 Sterling Commerce, Inc. All rights reserved.
Additional copyright information is located on the Gentran Integration Suite Documentation Library:
http://www.sterlingcommerce.com/Documentation/GIS42/homepage.htm

Contents

SWIFTNet with Gentran Integration Suite 6
Overview 6
Prerequisite Knowledge 11
Using SWIFTNet with Gentran Integration Suite 12
Using InterAct with Gentran Integration Suite 12
Using FileAct with Gentran Integration Suite 12
Downloading and Installing the SWIFTNet Standards Data Dictionary 12
SWIFTNet Envelopes 14
Overview 14
Enabling and Disabling Address Verification 15
Configuring Inbound Address Verification 15
Configuring Outbound Address Verification 16
Maintaining the External Code Lists 16
SWIFT_Addresses Code List 16
SWIFT_BaseAddresses Code List 17
SWIFT_Currencies Code List 18
SWIFT_Countries Code List 19
SWIFT_ClearingCodes Code List 20
Creating Envelopes 22
Using Base Envelopes 23
Inbound SWIFT envelope 23
Outbound SWIFT envelope 37
Configuring the EDI Encoder Service for SWIFTNet Outbound Messages 53
SWIFT Business Processes 55
Overview 55
SWIFTDeenvelope Business Process 56
Before Using the SWIFTDeenvelope Business Process 59
SWIFTEnvelope Business Process 59
Before Using the SWIFTEnvelope Business Process 63
SWIFTNetClient Business Process 63
Before Using the SWIFTNetClient Business Process 69
SWIFTNetClientFA Business Process 69
Before Using the SWIFTNetClientFA Business Process 75
handleSWIFTNetServerRequest Business Process 75
Before Using the handleSWIFTNetServerRequest Business Process 81
handleSWIFTNetServerSnFRequest Business Process 81
Before Using the handleSWIFTNetServerSnFRequest Business Process 86
handleSWIFTNetServerFARequest Business Process 86
Before Using the handleSWIFTNetServerFARequest Business Process 90

handleSWIFTNetServerFASnFRequest Business Process 91
Before Using the handleSWIFTNetServerFASnFRequest Business Process 95
handleSWIFTNetServerFAEvent Business Process. 96
Before Using the handleSWIFTNetServerFAEvent Business Process 101
Creating SWIFTNet Maps 102
Overview 102
How SWIFTNet Terminology Correlates with the Map Editor 103
SWIFTNet Components in the Map Editor 103
Downloading and Installing the SWIFT Standards Database 105
Creating a File Layout from an MT or Market Practice 106
Creating a Custom Market Practice or Fund Template. 107
Creating a SWIFTSolutions Map using SWIFTNet FUNDS, SWIFTNet Trade Services, SWIFTNet Exceptions and Investigations, and SWIFTNet Cash Reporting. 109
SWIFTNet Syntax Validation. 111
Creating Extended Rules for SWIFTNet Maps 111
Extended Rules Used with SWIFTNet Maps 112
Using Autolink 112
Using Extended Rule Libraries with SWIFTNet 112
Calling a Rule from an Extended Rule Library in a Map. 113
Searching for SWIFTNet Correlations 115
Searching for SWIFT Messages Using Correlations. 115
Overview of SWIFTNet Transport 117
InterAct and FileAct Protocol. 117
Additional FileAct Protocol Options. 118
SWIFTNet MEFG Server. 119
Overview 119
Client Application 120
Synchronous Message Exchange 120
Asynchronous Message Exchange 121
Configuring the Client Application 121
Server Application. 121
Configuring the Server Application. 121
SWIFTNet MEFG Server Installation 121
Prerequisites. 122
Configuring SAG/SNL. 122
. 124
Installing the SWIFTNet Remote API (RA). 124
Installing the SWIFTNet MEFG Server. 125
Configuring Fail-over Processing Using the SWIFTNet MEFG Server 126
Starting the Command Line 2 Client. 127
Monitoring the Status of the SWIFTNet MEFG Server 127
Starting and Stopping the SWIFTNet MEFG Server. 127
Document Tracking for SWIFTNet Transport 129
Overview 129
Monitoring the SWIFTNet Data Flow 129
Monitoring the SWIFTNet Communication Session Records 131
SWIFT Editor. 133
Editor Tasks 134
Reviewer Tasks. 136
SWIFTNet Routing Rule 139
Creating a SWIFTNet Routing Rule and Associating it with a Business Process. 139

Searching for a SWIFTNet Routing Rule 140
Searching for a Routing Rule by Name 140
Searching for a Routing Rule from a List 141
Exporting and Importing a SWIFTNet Routing Rule 141
Configuring the WebSphere MQ Adapter/Suite to Communicate with SWIFT 142
Configuring Gentran Integration Suite to Retrieve Messages 142
Configuring Gentran Integration Suite to Send Messages 143
Configuring the UMID and Block S Options 144
SWIFTNet Error Codes 145
Overview 145
Text Validation Error Codes 145
Specific Error Codes for MUG-textval Rules 158
Special Error Codes for Value-Added Service Messages 159
Message Syntax and Semantic Rule Codes 160
Rules 000-099 160
Rules 100-199 163
Rules 200-299 166
Knn: Code Word Validation in Generic Fields 170
Header Validation Error Codes 173

SWIFTNet with Gentran Integration Suite

- *Overview* on page 6
- *Prerequisite Knowledge* on page 11
- *Using SWIFTNet with Gentran Integration Suite* on page 12
- *Using InterAct with Gentran Integration Suite* on page 12
- *Downloading and Installing the SWIFTNet Standards Data Dictionary* on page 12

Overview

Gentran Integration Suite supports the use of Society for Worldwide Interbank Financial Telecommunications (SWIFTNet), a standard for the financial industry from SWIFT™ that enables store-and-forward financial messaging through the InterAct file protocol.

The SWIFTNet standards data dictionary is optionally installed from the **Deployment > Maps** page. It supports all SWIFT Standards Release messages loaded in the standards database. The SWIFTNet standards data dictionary also contains the special exception and code word validations, and the codes words and qualifiers necessary for the validation of the ISO 15022 messages (500 series). This information is used to automatically generate the translator_swift.properties.in file, which is used by the translator to perform the validations.

Note: You can create a map for all SWIFTNet Standards Release messages through the Map Editor. All messages are validated by Gentran Integration Suite for syntax (that is, field types, field lengths, and so forth). All SWIFT message are validated for syntax *and* semantics.

The following messages are supported with both inbound and outbound syntax validation and semantic validation (that is, validating the message rules) of the messages between Gentran Integration Suite and SWIFTNet:

- All SWIFTNet 2006 message types
- All SWIFTNet 2007 message types
- SWIFTNet FUNDS
- SWIFTNet Cash Reporting
- SWIFTNet Exceptions & Investigations
- SWIFTNet Trade Services

Additionally, Gentran Integration Suite allows you to create maps to support the translation of market practices, and provides the following two Market Practices:

- Germany: MT515: Trade Confirmation (Broker to Asset Manager)
- United States: MT536: ISITC-IOA: Statement of Transactions

For Market Practices, the SWIFT standard validations are performed unless the Market Practice has customized a particular validation (for example, restricting a code word validation list). Market Practice code word and qualifier validations are maintained in property files separate from the SWIFT standard property files. By keeping the Market Practices you create separate, we can deliver updates to the SWIFT

standard without overwriting your custom-implemented Market Practices. Similarly, semantic validation rules for Market Practices are stored in an extended rule library separate from the library that implements the SWIFT standard semantic validation rules. Please note that the standard property files and semantic validation rules are used when the implemented Market Practic does not override them.

In addition, support for SWIFTNet requires you to create inbound and outbound SWIFTNet envelopes. See *SWIFTNet Envelopes* on page 14. You must also create a business process or processes to order the flow of Gentran Integration Suite activities so you can accomplish your business objectives with SWIFTNet.

For more information on the SWIFTNet standard, access this web site:

http://www.swift.com

This table describes how Gentran Integration Suite was enhanced to support SWIFTNet:

Functionality	Enhancements	Action You Need to Take to Implement SWIFTNet
SWIFTNet Standards Data dictionary	◆ Provides all validation for field tags and structures for SWIFT messages. ◆ References a SWIFT extended rule library that is automatically checked in so you can use it with your maps. ◆ SWIFT_FINMessageTypes code list, supplied as part of the install, which contains a list of valid FIN message types. ◆ If you purchase a Financial Services license, you have access to XML standards once you download them.	◆ Populate four external codes lists: SWIFT_Addresses, SWIFT_BaseAddresses (only needs to be populated when address verification is enabled), SWIFT_Currencies, and SWIFT_Countries. ◆ If you want to use BIC+ Validation, you must also populate the SWIFT_ClearingCodes code list. See *SWIFTNet Envelopes* on page 14. ◆ Install the SWIFTNet standards data dictionary. See *Downloading and Installing the SWIFTNet Standards Data Dictionary* on page 12. **Note:** The data dictionary is optionally installed from the Deployment > Maps page. The SWIFT extended rules library is automatically installed when you download the Map Editor.

Functionality	Enhancements	Action You Need to Take to Implement SWIFTNet
SWIFTNet Market Practices	The following two market practices are available in Gentran Integration Suite: ◆ Germany: MT515: Trade Confirmation (Broker to Asset Manager) ◆ United States: MT536: ISITC-IOA: Statement of Transactions You can create a new message type by customizing a standard SWIFT message type map according to the specifications of the Market Practice. The customized map can be imported into the SWIFTNet Market Practices data dictionary through the SWIFTRuleImporter utility.	◆ Install the SWIFTNet standards data dictionary. The data dictionary is optionally installed from the Deployment > Maps page. The SWIFT extended rules library is automatically installed when you download the Map Editor. ◆ You can create a new message type by implementing the additional restrictions related to the desired Market Practice.
Services	◆ EDI Deenvelope service accepts SWIFT messages. ◆ SWIFTNet Server Adapter. ◆ SWIFTNet Client Service. ◆ An application, SWIFTNet MEFG Server, supports FileAct and InterAct processing, as well as failover processing.	◆ Install the Gentran Integration Suite SWIFTNet MEFG Server on a machine running either the Sun Solaris 9.0, Windows 9, or AIX operating systems. ◆ Configure the SWIFTNet Client Service. ◆ Configure the SWIFTNet Server adapter. See *SWIFTNet Server Adapter*, *SWIFTNet Client Service*, and *SWIFTNet MEFG Server* on page 119.
Envelopes	◆ Inbound SWIFT envelope wizard to implement inbound SWIFTNet. ◆ Outbound SWIFT envelope wizard to implement outbound SWIFTNet. ◆ The SWIFT_FINMessageTypes code list is automatically installed with Gentran Integration Suite. This code list contains a list of valid FIN message types.	◆ Create the appropriate SWIFT envelopes for each message type you are sending and receiving. ◆ You need to populate four code lists to use in conjunction with the SWIFT_FINMessageTypes code list to perform SWIFTNet validations. ◆ Configure the EDI Encoder service for use with the outbound SWIFT messages. See *SWIFTNet Envelopes* on page 14.

Functionality	Enhancements	Action You Need to Take to Implement SWIFTNet
Predefined business processes	◆ SWIFT Deenvelope business process ◆ SWIFT Envelope business process ◆ SWIFTNetClient business process ◆ SWIFTNetClientFA business process ◆ handleSWIFTNetClientFASnFServer Requestbusiness process ◆ HandleSWIFTNetServerFARequest business process ◆ handleSWIFTNetServerRequest business process ◆ handleSWIFTNetServerSnfRequest business process ◆ handleSWIFTNetSnFServerRequest ◆ handleSWIFTNetInboundCorrelation ◆ handleSWIFTNetOutboundCorrelation ◆ handleSWIFTNetServerFADelNotif ◆ handleSWIFTNetServerFAEvent ◆ handleSWIFTNetServerFARequest ◆ handleSWIFTNetServerFASnFDelNotif ◆ handleSWIFTNetServerFASnFRequest ◆ handleSWIFTNetServerRequest ◆ handleSWIFTNetServerSnFDelNotif ◆ handleSWIFTNetServerFASnFEvent ◆ handleSWIFTNetSnFInboundCorrelation ◆ handleSWIFTNetSnFOutboundCorrelation.	The business processes that are related to the SWIFTNet workflow must have the Document Tracking option enabled when you check in or edit the business process. Additionally, you need to configure other parameters in the SWIFTNetClient business process to support SWIFTNet. See *SWIFT Business Processes* on page 55

Functionality	Enhancements	Action You Need to Take to Implement SWIFTNet
Map Editor	◆ Map Editor wizard enables you to generate a file layout for you using the MT that you select. Included in the SWIFTNet map are groups, records, composites, and fields that are defined by SWIFT. ◆ Properties dialog boxes that enable you to define and modify SWIFTNet map components. ◆ Autolink function automatically creates links between input and output fields that have the same name or business name. This function can be used with any data format. ◆ Extended Rules Library function (used with SWIFTNet and any other data format) contains a list of rules in a separate file outside of the Map Editor source. Map Editor stores the name of the library in its source file, so when you load a map the library is also loaded and compiled. This enables you to create a library of extended rules and then add it to any other map, so you do not have to recreate those extended rules after the first time. ◆ Extended rules are used to validate SWIFTNet messages. ◆ Extended Rules Library function (used with SWIFTNet and any other data format) contains a list of rules in a separate file outside of the Map Editor source. Map Editor stores the name of the library in its source file, so when you load a map the library is also loaded and compiled. This enables you to create a library of extended rules and then add it to any other map, so you do not have to recreate those extended rules after the first time. ◆ Extended rules are used to validate SWIFTNet messages.	◆ Download Map Editor. ◆ Create a map or maps to translate your SWIFTNet messages. ◆ Place custom extended rule code inside a validation block. See *Creating SWIFTNet Maps* on page 102, *Extended Rules Used with SWIFTNet Maps* on page 112, and *Creating Extended Rules for SWIFTNet Maps* on page 111.
SWIFTNet Correlations	◆ Enables you to search for SWIFTNet messages using specific criteria.	◆ Run inbound and outbound data and then use the search interface to search for messages. See *Searching for SWIFTNet Correlations* on page 115.

Functionality	Enhancements	Action You Need to Take to Implement SWIFTNet
Document Tracking	◆ Support for monitoring SWIFTNet data flows. ◆ Support for tracking SWIFTNet messages within Gentran Integration Suite. The Gentran Integration Suite tracking and correlation functionality automatically tracks the SWIFTNet messages the same manner as the other standards are tracked. ◆ Support for document repair and resend.	◆ Use the data flow interface to track inbound and outbound SWIFTNet processes. ◆ All the business processes that are related to the SWIFTNet workflow (SWIFTNetClient, handleSWIFTNetServerRequest, and handleSWIFTNetServerSnfRequest) must have the Document Tracking option enabled when you check in or edit the business processes. ◆ Access messages with Ready To Edit status, repair, and resend them. See *Document Tracking for SWIFTNet Transport* on page 129 and *SWIFT Business Processes* on page 55.
SWIFTNet Routing Rule	Interface to support creation, modification, and deletion of SWIFTNet routing rules.	◆ Create a business process or modify the SWIFTNetClient business process. ◆ Create a SWIFTNet routing rule that is linked to the business process you created. See *SWIFT Business Processes* on page 55. See *Creating a SWIFTNet Routing Rule and Associating it with a Business Process* on page 139.
SWIFT Message Editor	Interface to enable a user to correct a SWIFTNet message that was returned due to an error either in transmission or translation.	◆ Create permissions for two different roles: the Editor and the Reviewer. ◆ Correct appropriate messages and notify the appropriate user that activity is required.

Prerequisite Knowledge

The audience using this software should be familiar with Gentran Integration Suite, the SWIFTNet standard, and using InterAct and/or FileAct.

Note: This documentation is not intended to explain the SWIFTNet standard.

Using SWIFTNet with Gentran Integration Suite

To use SWIFTNet with Gentran Integration Suite, you must complete the following tasks:

1. Download the SWIFTNet standards data dictionary. See *Downloading and Installing the SWIFTNet Standards Data Dictionary* on page 12.
2. Create inbound and outbound SWIFTNet envelopes. See *SWIFTNet Envelopes* on page 14.
3. Create the necessary code lists and maintain them as needed. See *Maintaining the External Code Lists* on page 16.
4. Enable the document tracking option for all the business processes that are that are related to the SWIFTNet workflow. See *Document Tracking for SWIFTNet Transport* on page 129 and *SWIFT Business Processes* on page 55.
5. Configure the EDI Encoder service for use with outbound SWIFTNet messages. See *Configuring the EDI Encoder Service for SWIFTNet Outbound Messages* on page 53.
6. Create the appropriate SWIFTNet maps. See *Creating SWIFTNet Maps* on page 102.
7. Use SWIFT Alliance Access (SAA) for FIN support.
8. Use the WebSphere MQ adapter to communicate with SAA.

Using InterAct with Gentran Integration Suite

To use InterAct with Gentran Integration Suite, you must complete the following tasks:

1. Configure the appropriate service parameters, including failover processing configuration. See *SWIFTNet Server Adapter* and *SWIFTNet Client Service*.
2. Configure the Gentran Integration Suite SWIFTNet MEFG Server for InterAct processing. See *SWIFTNet MEFG Server* on page 119.

Using FileAct with Gentran Integration Suite

To use FileAct with Gentran Integration Suite, you must complete the following tasks:

1. Configure the appropriate service parameters. See *SWIFTNet Server Adapter* and *SWIFTNet Client Service*.
2. Configure the Gentran Integration Suite SWIFTNet MEFG Server for FileAct processing. See *SWIFTNet MEFG Server* on page 119.

Downloading and Installing the SWIFTNet Standards Data Dictionary

Before you install the SWIFTNet standards data dictionary (standards database) on your desktop, consider these guidelines:

- ✦ Download the Map Editor.
- ✦ For the most current version of the SWIFTNet standards, contact Sterling Commerce Customer Support.

- ✦ Be sure your desktop meets the Windows Client requirements listed in the *Gentran Integration Suite System Requirements*.

To download and install the standards database:

1. From the Gentran Integration Suite **Deployment** menu, select **Maps**.
2. In the Download and Install section next to Download SWIFT Standards, click **Go!**
3. In the **File Download** dialog box, select a download option, then click **OK**.
   - ◆ If you choose to run the file click **Run** and the operating system downloads the files immediately.
   - ◆ If you choose to save the file, the operating system prompts you to save the file. Browse to the location where you want to download the file and click **OK**. If you want to continue installing, run the file you just saved from the location you specified.
4. In the Security Warning page, select **Always trust content from Sterling Commerce (Mid America), Inc.** if you do not want to see similar security messages in the future when you download software from Sterling Commerce. Click **Yes**.
5. In the Welcome window, click **Next**.
6. In the Choose Destination Location window, select where you want to install the standards database:
   - ◆ If you accept the default location, click **Next**.
   - ◆ If you want to specify a different location, click **Browse**, specify the path to the folder, click **OK**, and click **Next**.

   If you specify a folder name that does not exist, Gentran Integration Suite displays a message asking if you want to create that folder.
7. In the Select Components window, verify that **SWIFTStandardDatabase** is selected (if you want to use the preloaded SWIFT market practices, also ensure that **SWIFTMarketPractice** is selected) and click **Next**.

   The download wizard installs the standards database.
8. In the Setup Complete window, click **Finish**.

SWIFTNet Envelopes

- *Overview* on page 14
- *Enabling and Disabling Address Verification* on page 15
- *Maintaining the External Code Lists* on page 16
- *Creating Envelopes* on page 22
- *Inbound SWIFT envelope* on page 23
- *Outbound SWIFT envelope* on page 37
- *Configuring the EDI Encoder Service for SWIFTNet Outbound Messages* on page 53

Overview

A *document envelope* consists of control information that enables organizations to effectively exchange messages. This information is added in headers and trailers to messages. Document envelopes are specific to the message protocol used. Creating document envelopes is necessary to use SWIFTNet with your trading partners.

SWIFTNet has only one level of envelope, which you must modify appropriately to reflect your information and your trading partner's information. Envelopes specify whether the message is inbound or outbound:

- The **Inbound SWIFT envelope** identifies messages that are received by Gentran Integration Suite so they can be properly routed. Inbound envelopes also give you the option to translate messages when you choose to check messages for compliance. By choosing to translate messages from within the envelope, you can reduce message processing time because you do not need to specify a separate Translation service step in the business process. You need to create a Inbound SWIFT envelope to configure deenveloping information. See *Inbound SWIFT envelope* on page 23.
- The **Outbound SWIFT envelope** identifies messages so that they can be sent to and received by trading partners. You need to configure a Outbound SWIFT envelope to configure enveloping information. See *Outbound SWIFT envelope* on page 37.

When you envelope an outbound SWIFTNet message, the SWIFTNet header and trailer are created. For an inbound message, the envelope contains the header information (the trailer information is a summary appended to the SWIFTNet data).

As part of SWIFTNet enveloping, Gentran Integration Suite uses code lists to validate the data. Gentran Integration Suite uses code pairs in code lists to identify items in transactions between two or more trading partners. A trading partner code list consists of one or many pairs of code values containing a sender code and a receiver code. Each code pair has one description and up to four additional codes relating to the pair. Code lists are dynamic and are stored in a database.

The SWIFT_FINMessageTypes code list is automatically installed with Gentran Integration Suite. This code list contains a list of valid FIN message types. The three-digit message number is entered for sender code and receiver code, and the description is set to **SWIFT Message Type**. However, you need to populate four additional code lists to perform SWIFTNet validations:

- **SWIFT_Addresses**—used to check the sender and receiver IDs within the message. This code list is shared with message authentication. The code list used is the same code list that is used for message

validations. You will type the address in the Sender Code, Receiver Code, and Description parameters, and use the Text1 parameter to indicate the subtype of the address. The SWIFT_Addresses and SWIFT_BaseAddresses code lists are used to differentiate between bad base addresses and bad branch codes when necessary. The SWIFT_Addresses code list is also used for verification of those addresses that are contained within the body of a message.

- **SWIFT_BaseAddresses**—this is a list of the 8-character address (the BIC minus the branch code) that are valid as part of a sender address when generating a message. You will type the eight-digit code in the Sender Code, Receiver Code, and Description parameters. The SWIFT_Addresses and SWIFT_BaseAddresses code lists are used to differentiate between bad base addresses and bad branch codes when necessary.

  Note: You only need to populate the SWIFT_BaseAddresses code list when you have enabled address verification. See *Enabling and Disabling Address Verification* on page 15.

- **SWIFT_Currencies**—this is a list of the valid currencies that can be used in a SWIFT message. You will use the Text1 parameter to indicate the maximum number of digits after the decimal point that the currency supports.
- **SWIFT_Countries**—this is a list of the valid countries that can be used as part of the address when generating a SWIFT message.
- **SWIFT_ClearingCodes**—this is a list of the clearing codes used when validating a message. You must populate this list if you want to use BIC+ validation.

See *Maintaining the External Code Lists* on page 16.

The validation of the SWIFT special functions <CUR>, <SWIFTBIC>, and <NON-SWIFTBIC> use these code lists. You must update and maintain these codes lists, as necessary.

For outbound SWIFTNet messages, you also need to configure the EDI Encoder service to include the proper values for the following parameters:

- AccepterLookupAlias
- ReceiverID
- SenderID
- ReceiverIDQual
- SenderIDQual

See *Configuring the EDI Encoder Service for SWIFTNet Outbound Messages* on page 53.

Enabling and Disabling Address Verification

Gentran Integration Suite allows you to enable or disable address verification. Address verification is performed using the SWIFT_Addresses and SWIFT_BaseAddresses code lists. See *Maintaining the External Code Lists* on page 16 for more information on creating these code lists.

Configuring Inbound Address Verification

The **enveloping.verify_addresses_while_deenveloping.SWIFT_FIN_INBOUND** property enables and disables inbound address verification.

To enable *inbound* address verification, complete these steps:

1. Access the ***install_dir*/properties/enveloping.properties.in** file, and change the line **enveloping.verify_addresses_while_deenveloping.SWIFT_FIN_INBOUND** to **TRUE**, as noted below:

```
enveloping.verify_addresses_while_deenveloping.SWIFT_FIN_INBOUND=True
```

Note: If you want to then disable inbound address verification, you can do so by accessing the ***install_dir*/properties/envelope.properties.in** file, and change the line **enveloping.verify_addresses_while_deenveloping.SWIFT_FIN_INBOUND=FALSE**.

2. Save and close the **enveloping.properties.in** file.
3. Stop Gentran Integration Suite.
4. Run the setupfiles script using one of the following steps:
   - (UNIX or Linux) - From the ***install_dir***/bin directory, run the **setupfiles.sh** command.
   - (Windows) - From the ***install_dir***\bin directory, run the **setupfiles.cmd** command.
5. Start Gentran Integration Suite.

Configuring Outbound Address Verification

To enable or disable *outbound* address verification, use the Outbound SWIFT envelope parameter **Validate Sender and Receiver**. This parameter allows you to enable (by choosing **Yes**) or disable (by choosing **No**) address verification. The default is **No** (sender and receiver verification is disabled). See *Outbound SWIFT envelope* on page 37 for more information.

Maintaining the External Code Lists

You need to populate the four external code lists to use in conjunction with the SWIFT_FINMessageTypes code list (all code lists are automatically installed with Gentran Integration Suite—the SWIFT_FINMessageTypes code list already contains a list of valid FIN message types) to perform SWIFTNet validations.

The five external code lists for which you need to populate SenderID, ReceiverID, and Description are:

- SWIFT_Addresses
- SWIFT_BaseAddresses
- SWIFT_Currencies
- SWIFT_Countries
- SWIFT_ClearingCodes

SWIFT_Addresses Code List

To populate the SWIFT_Addresses code list:

1. From the Trading Partner menu, select **Code Lists**.
2. Next to Alphabetically by List Name, select **S** and click **Go!**

3. Next to SWIFT_Addresses, click **Source Manager**.

   Note: This code list (along with the SWIFT_BaseAddresses code list) is used for sender and receiver verification when address verification is enabled. This code list is also used for verification of those addresses that are contained within the body of a message.

4. In the Code List Source Manager page, next to the appropriate list, click **Edit**.
5. In the Naming page, click **Next**.
6. In the Codes page, click **add** to add a new pair of codes.
7. In the Input Code Data dialog box, complete the following fields, and click **Save**:

Field	Description
Sender Code	Type the 11-character address. Required. **Note:** This code list is used to check the sender and receiver IDs within the message. The code used is the same code list that is used for message validations.
Receiver Code	Type the 11-character address. Required **Note:** This code list is used to check the sender and receiver IDs within the message. For envelope validation to work, the addresses you enter must have **X** in the LT (9th) position. For example, STERCOMMXUSA. The code used is the same code list that is used for message validations.
Description	Type a meaningful description. Required.
Text1	This is the subtype of the address. The value in this field should be the subtype four-character abbreviation that identifies the type of financial institution. For example, CSDS represents Clearing Houses, Central Depositories. Required.
Text2	Description or data relating to the sender and receiver code. Optional.
Text3	Description or data relating to the sender and receiver code. Optional.
Text4	Description or data relating to the sender and receiver code. Optional.

8. Repeat steps 6 - 7 as needed to create additional code pairs. When you are finished, click **Next**.
9. Click **Save** to save the code list.
10. In the Codes page, click **Save**.
11. In the Code List Source Manager page, click **Return**.

SWIFT_BaseAddresses Code List

To populate the SWIFT_BaseAddresses code list:

Note: This code list (along with the SWIFT_Addresses code list) is used for sender and receiver verification when address verification is enabled. You only need to populate the SWIFT_BaseAddresses code list when you have enabled address verification. See *Enabling and Disabling Address Verification* on page 15.

1. From the Trading Partner menu, select **Code Lists**.
2. Next to Alphabetically by List Name, select **S** and click **Go!**

3. Next to SWIFT_BaseAddresses, click **Source Manager**.

   Note: This code list (along with the SWIFT_Addresses code list) is used for sender and receiver verification when address verification is enabled.

4. In the Code List Source Manager page, next to the appropriate list, click **Edit**.
5. In the Naming page, click **Next**.
6. In the Codes page, click **add** to add a new pair of codes.
7. In the Input Code Data dialog box, complete the following fields, and click **Save**:

Field	Description
Sender Code	Type the eight-character address (the BIC minus the branch code). Required. **Note:** This code list is used to check the Sender Branch Code value. For example, if the sender ID is STERCOMMXUSA, the branch code is "USA," and you would type **STERCOMM** for the Sender Code parameter.
Receiver Code	Type the eight-character address (the BIC minus the branch code). Required **Note:** This code list is used to check the Sender Branch Code value. For example, if the sender ID is STERCOMMXUSA, the branch code is "USA," and you would type **STERCOMM** for the Sender Code parameter.
Description	Type a meaningful description. Required.
Text1	Description or data relating to the sender and receiver code. Optional.
Text2	Description or data relating to the sender and receiver code. Optional.
Text3	Description or data relating to the sender and receiver code. Optional.
Text4	Description or data relating to the sender and receiver code. Optional.

8. Repeat steps 6 - 7 as needed to create additional code pairs. When you are finished, click **Next**.
9. Click **Save** to save the code list.
10. In the Codes page, click **Save**.
11. In the Code List Source Manager page, click **Return**.

SWIFT_Currencies Code List

To populate the SWIFT_Currencies code list:

1. From the Trading Partner menu, select **Code Lists**.
2. Next to Alphabetically by List Name, select **S** and click **Go!**
3. Next to SWIFT_Currencies, click **Source Manager**.
4. In the Code List Source Manager page, next to the appropriate list, click **Edit**.
5. In the Naming page, click **Next**.
6. In the Codes page, click **add** to add a new pair of codes.

7. In the Input Code Data dialog box, complete the following fields, and click **Save**:

Field	Description
Sender Code	Contains the currency code (same value as in the Receiver Code parameter). Required.
Receiver Code	Contains the currency code (same value as in the Sender Code parameter). Required.
Description	Meaningful descriptive text. Required.
Text1	The number of fractional decimal digits supported for the identified currency. Required. For example, if Sender Code=USD, Receiver Code=USD, and Text1=2, this indicates that the currency code is USD which supports 2 decimal digits. **Note:** Include the maximum number of digits after the decimal point that the currency supports.
Text2	Description or data relating to the sender and receiver code. Optional.
Text3	Description or data relating to the sender and receiver code. Optional.
Text4	Description or data relating to the sender and receiver code. Optional.

8. Repeat steps 6 - 7 as needed to create additional code pairs. When you are finished, click **Next**.
9. Click **Save** to save the code list.
10. In the Codes page, click **Save**.
11. In the Code List Source Manager page, click **Return**.

SWIFT_Countries Code List

You can populate the SWIFT_Countries code list in two ways—automatically and manually. To automatically populate the SWIFT_Countries code list, you can use the HIPAA codelist conversion map for countries to populate it. However, prior to using the HIPAA codelist conversion map, you need to modify the import file created using the conversion map (the import file is used to populate the Countries codelist). To modify the import file you need to change the codelist name from Countries to **SWIFT_Countries**. If you need to load several countries in the code list, you should use the automatic method. But you only need a few countries in the code list then using manual entry is more efficient.

To populate the SWIFT_Countries code list automatically:

1. Download or save the Countries map from the ***install_dir*/installed_data/b2b/maps** directory to the **tp_import** directory where Gentran Integration Suite is installed.
2. From a command line, go to the **tp_import** directory.
3. Type the following command to start the conversion and import process, where <map name> is the name of the map to use during translation (without the file extension) and <code list path and filename> is the fully qualified name of the code list to translate, including filename extension, if any:
   - If you are using Windows, **hipaaconvert.cmd [-import] <map name> <code list path and filename>**

- If you are using UNIX, **hipaaconvert.sh [-import] <map name> <code list path and filename>**

  Do not specify the file extension for the map name when importing a code list—just indicate the base name of the map.

  The [-import] parameter is optional. You can convert the code list file without importing it. If you do not use the [-import] parameter during conversion, you can import the resulting XML file into Gentran Integration Suite using the import utility.

4. Once the utility completes, a translation report (hipaaconvert.rpt) and an input file (hipaaconvert.xml) are created. If no translation errors are reported, the code list was successfully generated (and imported if you used the [-import] parameter). A code list will not be imported if there are translation errors.
5. Then, to modify the import file you need to change the codelist name from Countries to **SWIFT_Countries** so it can be used for SWIFTNet messages.

To populate the SWIFT_Countries code list manually:

1. From the Trading Partner menu, select **Code Lists**.
2. Next to Alphabetically by List Name, select **S** and click **Go!**
3. Next to SWIFT_Currencies, click **Source Manager**.
4. In the Code List Source Manager page, next to the appropriate list, click **Edit**.
5. In the Naming page, click **Next**.
6. In the Codes page, click **add** to add a new pair of codes.
7. In the Input Code Data dialog box, complete the following fields, and click **Save**:

Field	Description
Sender Code	Contains the country code (same value as in the Receiver Code parameter). Required. For example, **US**.
Receiver Code	Contains the country code (same value as in the Sender Code parameter). Required. For example, **US**.
Description	Meaningful descriptive text. Required.
Text1	Not used.
Text2	Not used.
Text3	Not used.
Text4	Not used.

8. Repeat steps 6 - 7 as needed to create additional code pairs. When you are finished, click **Next**.
9. Click **Save** to save the code list.
10. In the Codes page, click **Save**.
11. In the Code List Source Manager page, click **Return**.

SWIFT_ClearingCodes Code List

You can populate the SWIFT_ClearingCodes code list in two ways—automatically and manually. To automatically populate the SWIFT_Countries code list, you can use the HIPAA codelist conversion map for countries to populate it. However, prior to using the HIPAA codelist conversion map, you need to modify the import file created using the conversion map (the import file is used to populate the Countries codelist). To modify the import file you need to change the codelist name from Countries to **SWIFT_Countries**. If you need to load several countries in the code list, you should use the automatic method. But you only need a few countries in the code list then using manual entry is more efficient.

To populate the SWIFT_Countries code list automatically:

1. Download or save the Countries map from the ***install_dir*/installed_data/b2b/maps** directory to the **tp_import** directory where Gentran Integration Suite is installed.
2. From a command line, go to the **tp_import** directory.
3. Type the following command to start the conversion and import process, where <map name> is the name of the map to use during translation (without the file extension) and <code list path and filename> is the fully qualified name of the code list to translate, including filename extension, if any:
   - If you are using Windows, **hipaaconvert.cmd [-import] <map name> <code list path and filename>**
   - If you are using UNIX, **hipaaconvert.sh [-import] <map name> <code list path and filename>**

   Do not specify the file extension for the map name when importing a code list—just indicate the base name of the map.

   The [-import] parameter is optional. You can convert the code list file without importing it. If you do not use the [-import] parameter during conversion, you can import the resulting XML file into Gentran Integration Suite using the import utility.
4. Once the utility completes, a translation report (hipaaconvert.rpt) and an input file (hipaaconvert.xml) are created. If no translation errors are reported, the code list was successfully generated (and imported if you used the [-import] parameter). A code list will not be imported if there are translation errors.
5. Then, to modify the import file you need to change the codelist name from Countries to **SWIFT_Countries** so it can be used for SWIFTNet messages.

To populate the SWIFT_Countries code list manually:

1. From the Trading Partner menu, select **Code Lists**.
2. Next to Alphabetically by List Name, select **S** and click **Go!**
3. Next to SWIFT_Currencies, click **Source Manager**.
4. In the Code List Source Manager page, next to the appropriate list, click **Edit**.
5. In the Naming page, click **Next**.
6. In the Codes page, click **add** to add a new pair of codes.
7. In the Input Code Data dialog box, complete the following fields, and click **Save**:

Field	Description
Sender Code	Contains the country code (same value as in the Receiver Code parameter). Required. For example, **US**.

Field	Description
Receiver Code	Contains the country code (same value as in the Sender Code parameter). Required. For example, **US**.
Description	Meaningful descriptive text. Required.
Text1	Not used.
Text2	Not used.
Text3	Not used.
Text4	Not used.

8. Repeat steps 6 - 7 as needed to create additional code pairs. When you are finished, click **Next**.
9. Click **Save** to save the code list.
10. In the Codes page, click **Save**.
11. In the Code List Source Manager page, click **Return**.
12. If you want to disable the automatic BIC+ validation, access the **translator_swift_2006.properties** file to the following:

```
fieldTags.BICPlus.102_not_STP=53A,52C57A,57C
fieldTags.BICPlus.102_STP=52A,57A
fieldTags.BICPlus.103_not_STP=52A,52D56A,56C,57A,57C,57D
fieldTags.BICPlus.103_STP=52A,56A,57A
```

Creating Envelopes

Inbound envelopes define expected header and trailer information for inbound messages. This information helps Gentran Integration Suite route and process the messages. Outbound envelopes specify information about messages that enables them to be sent to and received by trading partners, and they gather and provide the appropriate data used to create the header.

To create an envelope:

1. From the Admin Console, select **Trading Partner > Document Envelopes > Envelopes**.
2. Under Create, next to New Envelope, click **Go!**
3. On the Envelope Standards page, select **SWIFT** and click **Next**.
4. Select the level of envelope you want to create, inbound or outbound, and click **Next**.
5. On the Base Envelope page, do you want this envelope to inherit properties from a base envelope (if available)?
   - If Yes, select a base envelope and click **Next**.
   - If No (you want to create a new envelope), select **Not Applicable** and click **Next**.
6. On the Name page, type a unique name for the envelope, and a description or comments, then click **Next**.

7. Complete the properties for the envelope as necessary and click **Next** after each page until you reach the confirm page. Required fields are highlighted in blue. See *Inbound SWIFT envelope* on page 23 or *Outbound SWIFT envelope* on page 37.
8. Click **Finish** to add the envelope.

Using Base Envelopes

A *base envelope* is a regular envelope that you use as a starting point to create a new envelope. The base envelope maintains a link to the envelope that inherited its properties. If you modify the base envelope, all related envelopes (those that inherited the base envelope properties) are also changed.

When you create an envelope using a base envelope, everything in the new envelope is the same as in the base envelope, except the envelope name, description, and parameters such as unique identification numbers. If you plan to create many envelopes using base envelopes, do not use the base envelopes in production. You should also be sure to note the envelopes that are related to the base envelopes.

To use a base envelope:

1. Create the base envelope, using *Creating Envelopes* on page 22.
2. Specify the name and description of a new envelope.
3. Identify the base envelope that the new envelope uses.

Inbound SWIFT envelope

You only need to create an Inbound SWIFT envelope if you are receiving inbound SWIFTNet messages. However, if you are receiving inbound SWIFTNet messages, you need to create a separate Inbound SWIFT envelope for *each* SWIFTNet message type you will be receiving. The following table describes Inbound SWIFT envelope properties:

Note: An (*) asterisk indicates that a wildcard value can be used with that parameter (for mandatory fields, the wildcard value is an (*) asterisk and for optional fields, the wildcard value is leaving the field blank):

Field or Check Box	Description
* Sender ID	Coded identifier of the supplier number or data sender. Valid value is eight standard characters. Required—must be eight characters. **Note:** This parameter enables you to type in a new ID or pick an ID that has already been used. When you start typing an ID, Gentran Integration Suite returns all matching IDs existing in the system and provide a combo-box from which you can select an ID by double-clicking it. There must be Sender ID Codes in the system for autocomplete to find matches and display a selection list.
* Sender Branch Code	Three-character code that further refines the Sender ID. For example, if the SenderID is ROMANSFL, the branch code might be WAS to indicate the Washington branch. This parameter acts as the Sender ID qualifier for envelope matching. Required.

Field or Check Box	Description
* Receiver ID	Coded identifier of the customer number or data source number. Valid value is eight standard characters. Required—must be eight characters. **Note:** This parameter enables you to type in a new ID or pick an ID that has already been used. When you start typing an ID, Gentran Integration Suite returns all matching IDs existing in the system and provide a combo-box from which you can select an ID by double-clicking it. There must be Receiver ID Codes in the system for autocomplete to find matches and display a selection list.
* Receiver Branch Code	Three-character code that further refines the Receiver ID. For example, if the ReceiverID is ROMANSFL, the branch code might be WAS to indicate the Washington branch. This parameter acts as the Receiver ID qualifier for envelope matching. Required.
Message Type	Transaction message type as determined by the information type in the header of the message group, which includes the message, or determined by the information type in the transaction message. Required.
Enforce Message Size Limit	Specifies that Gentran Integration Suite will check the size of the message, and error out if the message exceeds it. If you select this check box, Gentran Integration Suite gives you the option to set the Maximum Message Size (which defaults to 10,000, the standard limit for SWIFT messages). Valid values are Yes (default) and No. Required.
Maximum Message Size	The maximum size of message that is allowed. The default is 10,000. This parameter only accepts numeric input. Only displayed if you selected Yes for the **Enforce Message Size Limit** parameter. Required.
Reconcile Message User Reference (MUR) against a control number	Whether to reconcile the Message User Reference (MUR) against a control number. Valid values are Yes (default) and No. The MUR is a message identifier separate from the one SWIFT assigns. Required.
Use global control number	Whether to use a global control number. Required. Valid values are: ◆ Yes (default) ◆ Yes (and generate name from data) **Note:** If you select this option, you must select at least one parameter for Primary Name Format. ◆ No **Note:** Only displayed if you selected Yes for **Reconcile message User Reference (MUR) against a control number**.
Perform Duplicate Control Number (MUR) Checking	Determine control number/MUR duplications. Required. Valid values: ◆ Yes ◆ No (default) **Note:** Only displayed if you selected Yes for **Reconcile message User Reference (MUR) against a control number**.

Field or Check Box	Description
Maximum age of Control Number History Records in days	Maximum days that Gentran Integration Suite should retain a history of control numbers to use for duplication determinations. Valid value is nine numerics. Optional. **Note:** Only displayed if you selected Yes for **Reconcile message User Reference (MUR) against a control number**.
Assign control number	Select a control number with this envelope. Optional. Displayed only if **Use Global Control Number** is set to Yes
Local control number	Select a local control number to associate with this envelope. Default is 1. Required. Displayed only if **Use Global Control Number** is set to No.
Primary Name Format	Check boxes to instruct what information to include when generating a name for a primary global control number and finding the correct number to assign based on that name. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated global control number (**Use global control number** is set to Yes (and generate name from data)), Gentran Integration Suite tries to generate and match the following control numbers: ◆ First, it tries to generate and match the control number in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the control number SWIFT_100_Inbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing control number that matches the names generated in steps 1-3, a control number with the name assigned in the first step is created.

Field or Check Box	Description
First Backup Name Format	The first backup name format to use when generating a global control number. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated global control number (**Use global control number** is set to Yes (and generate name from data)), Gentran Integration Suite tries to generate and match the following control numbers: ◆ First, it tries to generate and match the control number in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the control number SWIFT_100_Inbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing control number that matches the names generated in steps 1-3, a control number with the name assigned in the first step is created.

Field or Check Box	Description
Second Backup Name Format	The second backup name format to use when generating a global control number. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated global control number (**Use global control number** is set to Yes (and generate name from data)), Gentran Integration Suite tries to generate and match the following control numbers: ◆ First, it tries to generate and match the control number in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the control number SWIFT_100_Inbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing control number that matches the names generated in steps 1-3, a control number with the name assigned in the first step is created.
Compliance check FIN message	Check the message body for compliance. Required. Valid values are Yes (default) and No.
Keep translated document after compliance check	Whether to keep the translated document after the compliance check.Valid values are Yes and No (default). Required. **Note:** If you select Yes, the translated document replaces the primary document.
Map Name Mode	How to determine which map to use to perform a compliance check. Required. Valid values are: ◆ Specify (default) ◆ Generate from data (this is useful if you want to share across envelopes) **Note:** If you select this option, you must select at least one parameter for Primary Name Format. The generated name will end with FIN.
Map Name	Which map to use to perform a compliance check (if **Compliance Check SWIFT messages** set to Yes and **Map Name Mode** set to Yes). The map must already be checked in to Gentran Integration Suite. Optional.

Field or Check Box	Description
Primary Name Format	Check boxes to instruct what information to include when generating and matching a name for the map. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated map name (**Map Name Mode** is set to Generate from data), Gentran Integration Suite tries to generate and match the following maps: ◆ First, it tries to generate and match the map in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the map named SWIFT_100_Inbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing map that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no matching map is found** is set to Yes.

Field or Check Box	Description
First Backup Name Format	The first backup name format to use when generating a map name. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated map name (**Map Name Mode** is set to Generate from data), Gentran Integration Suite tries to generate and match the following maps: ◆ First, it tries to generate and match the map in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the map named SWIFT_100_Inbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing map that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no matching map is found** is set to Yes.

Field or Check Box	Description
Second Backup Name Format	The second backup name format to use when generating a map name. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated map name (**Map Name Mode** is set to Generate from data), Gentran Integration Suite tries to generate and match the following maps: ◆ First, it tries to generate and match the map in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the map named SWIFT_100_Inbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing map that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no matching map is found** is set to Yes.
Generate an error if no matching map is found	Specifies whether to generate an error if the Map Name selected is not found. Valid values are Yes (default) and No. **Note:** Only displayed if **Map Name Mode** is set to Generate from data.
Determine Error Business Process Name By	How to determine the business process name to use if there were errors in the compliance check. Required. Valid values are: ◆ Specifying a Business Process name (default) ◆ Generating a Business Process name from the data (this is useful if you want to share across envelopes) **Note:** If you select this option, you must select at least one parameter for Primary Name Format. The generated name will end with FIN.

Field or Check Box	Description
Primary Name Format	Check boxes to instruct what information to include when generating and matching a name for the error business process name. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated error business process name (**Determine Error Business Process Name By** is set to Generating a Business Process name from the data), Gentran Integration Suite tries to generate and match the following business process: ◆ First, it tries to generate and match the business process in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the business process SWIFT_100_Inbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing business process that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no generated business process name exists in the system** is set to Yes.

Field or Check Box	Description
First Backup Name Format	The first backup name format to use when generating an error business process name. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated error business process name (**Determine Error Business Process Name By** is set to Generating a Business Process name from the data), Gentran Integration Suite tries to generate and match the following business process: ◆ First, it tries to generate and match the business process in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the business process SWIFT_100_Inbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing business process that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no generated business process name exists in the system** is set to Yes.

Field or Check Box	Description
Second Backup Name Format	The second backup name format to use when generating an error business process name. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated error business process name (**Determine Error Business Process Name By** is set to Generating a Business Process name from the data), Gentran Integration Suite tries to generate and match the following business process: ◆ First, it tries to generate and match the business process in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the business process SWIFT_100_Inbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing business process that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no generated business process name exists in the system** is set to Yes.
Generate an error if no generated business process name exists in the system	Specify whether to generate an error if there is no match to the generated business process name in the system. Valid values are Yes (default) and No. **Note:** Only displayed if **Determine the Error Business Process By** is set to Generating the business process name from the data.
Business Process List	Select a previously created business process to associate with this envelope. Optional. Displayed only if **Determine Error Business Process Name By** is set to Specifying the business process.
Determine the Business Process By	How to determine the business process name to use if there were no errors in the compliance check. Required.Valid values are: ◆ Specifying a business process ◆ Generating the business process name from the data **Note:** If you select this option, you must select at least one parameter for Primary Name Format. The generated name will end with FIN.

Field or Check Box	Description
Primary Name Format	Check boxes to instruct what information to include when generating and matching a name for the business process. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated business process name (**Determine the Business Process By** is set to Generating the business process name from the data), Gentran Integration Suite tries to generate and match the following business process: ◆ First, it tries to generate and match the business process in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the business process SWIFT_100_Inbound_FIN. ◆ Second, it tries to generate and match the first Backup Name. ◆ Third, it tries to generate and match the second Backup Name. ◆ Fourth, if there is not an existing business process that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no generated business process name exists in the system** is set to Yes.

Field or Check Box	Description
First Backup Name Format	The first backup name format to use when generating a business process name. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated business process name (**Determine Business Process Name By** is set to Generating a Business Process name from the data), Gentran Integration Suite tries to generate and match the following business process: ◆ First, it tries to generate and match the business process in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the business process SWIFT_100_Inbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing business process that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no generated business process name exists in the system** is set to Yes.

Field or Check Box	Description
Second Backup Name Format	The second backup name format to use when generating a business process name. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated business process name (**Determine Error Business Process Name By** is set to Generating a Business Process name from the data), Gentran Integration Suite tries to generate and match the following business process: ◆ First, it tries to generate and match the business process in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the business process SWIFT_100_Inbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing business process that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no generated business process name exists in the system** is set to Yes.
Generate an error if no generated business process name exists in the system	Specify whether to generate an error if there is no match to the generated business process name in the system. Valid values are Yes (default) and No. **Note:** Only displayed if **Determine the Business Process By** is set to Generating the business process name from the data.
Business Process List	Select a previously created business process to associate with this envelope Optional. Displayed only if **Determine the Business Process Name** is set to Specifying the business process.
Extraction Options	Business process data extraction. Required. Valid values: ◆ Determined by business process (default) ◆ Extract to a file system directory ◆ Extract to a mailbox
Data Extraction Directory	Directory for data extraction. Displayed only if Extraction Options set to **Extract to a file system directory**. Optional

Field or Check Box	Description
Data Extraction Filename	Filename for data extraction. Displayed only if Extraction Options set to **Extract to a file system directory**. Optional.
Data Extraction Mailbox	Mailbox for data extraction. Displayed only if Extraction Options set to **Extract to a mailbox**. Optional.
Data Extraction Mailbox Message Name	Mailbox message name for data extraction. Displayed only if Extraction Options set to **Extract to a mailbox**. Optional.

Outbound SWIFT envelope

You only need to create an Outbound SWIFT envelope if you are sending outbound SWIFTNet messages. However, if you are sending outbound SWIFTNet messages, you need to create a separate Outbound SWIFT envelope for *each* SWIFTNet message type you will be sending. The following table describes Outbound SWIFT envelope properties:

Note: An (*) asterisk indicates that a wildcard value can be used with that parameter:

Field or Check Box	Description
* Sender ID	Coded identifier of the supplier number or data sender. Valid value is eight standard characters. Required. **Note:** This parameter enables you to type in a new ID or pick an ID that has already been used. When you start typing an ID, Gentran Integration Suite returns all matching IDs existing in the system and provide a combo-box from which you can select an ID by double-clicking it. There must be Sender ID Codes in the system for autocomplete to find matches and display a selection list.
Sender Logical Terminal	Specifies the logical terminal for outbound FIN. This is a single alphanumeric character Required. **Note:** This parameter is necessary to send messages through the WebSphere MQ adapter using the FileAct protocol.
* Sender Branch Code	Three-character code that further refines the Sender ID. For example, if the SenderID is ROMANSFL, the branch code might be WAS to indicate the Washington branch. This parameter acts as the Sender ID qualifier for envelope matching. Required.
* Receiver ID	Coded identifier of the customer number or data source number. Valid value is eight standard characters. Required. **Note:** This parameter enables you to type in a new ID or pick an ID that has already been used. When you start typing an ID, Gentran Integration Suite returns all matching IDs existing in the system and provide a combo-box from which you can select an ID by double-clicking it. There must be Receiver ID Codes in the system for autocomplete to find matches and display a selection list.

Field or Check Box	Description
* Receiver Branch Code	Three-character code that further refines the Receiver ID. For example, if the ReceiverID is ROMANSFL, the branch code might be WAS to indicate the Washington branch. This parameter acts as the Receiver ID qualifier for envelope matching. Required.
Validate Sender and Receiver	Specify whether to validate the sender and receiver. Required. Valid values are Yes and No (default). **Note:** This parameter allows you to enable (Yes) or disable (No, which is the default) address verification. Address verification is performed using the SWIFT_Addresses and SWIFT_BaseAddresses code lists. See *Maintaining the External Code Lists* on page 16 for more information on creating these code lists.
Message Type	Transaction message type as determined by the information type in the header of the message group, which includes the message, or determined by the information type in the transaction message. Required.
Accepter Lookup Alias	Identifying string used with the Sender ID and the Receiver ID to look up this envelope with the EDI Encoder service. This alias associates a message with the service it requires. Valid value must be at least one limited standard character. Required. Default is FIN.
Batching Options	When the EDI Encoder is used to prepare multiple messages for enveloping, you can either choose to process each message individually or concatenate them into a single file, with the messages separated by a '$'. Required. Valid values are: ◆ Use FileAct batching ("$" between messages) (default) ◆ Do not batch messages
Use Correlation Overrides	When to use correlation overrides (when a SWIFT Reviewer chooses not to validate a message on resend). Required. Valid values are: ◆ Never (default) ◆ Only when the envelope field has a wildcard value (*) ◆ Always ◆ Use the default specified in enveloping.properties
Enforce Message Size Limit	Specifies that Gentran Integration Suite will check the size of the message, and error out if the message exceeds it. If you select this check box, give you the option to set the Maximum Message Size (which defaults to 10,000, the standard limit for SWIFT messages). Valid values are Yes (default) and No. Required.
Message User Reference (tag 108)	Indicates how to process the message user reference. The message user reference block is an optional section in the SWIFT envelope that the SWIFT network tracks for acknowledgement purposes. Valid values are: ◆ Don't include a message user reference (default) ◆ Use control number ◆ Use value from process data

Field or Check Box	Description
Service Identifier	A two-character numeric field indicating the type of data. The default is 01 for all Gentran Integration Suite and user-to-user messages, 21 for acknowledgements, and 03 for SELECT commands. Required.
Maximum Message Size	The maximum size of message that is allowed. The default is 10,000. This parameter only accepts numeric input. Only displayed if you selected Yes for the **Enforce Message Size Limit** parameter. Required.
Use global control number	Whether to use a global control number. Required. Valid values are: ◆ Yes (default) ◆ Yes (and generate from data) **Note:** If you select this option, you must select at least one parameter for Primary Name Format. ◆ No Displayed only if **Message User Reference** is set to Use Control Number.
Primary Name Format	Check boxes to instruct what information to include when generating a name for a primary global control number and finding the correct number to assign based on that name. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated global control number (**Use global control number** is set to Yes (and generate name from data)), Gentran Integration Suite tries to generate and match the following control numbers: ◆ First, it tries to generate and match the control number in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the control number SWIFT_100_Outbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing control number that matches the names generated in steps 1-3, a control number with the name assigned in the first step is created.

Field or Check Box	Description
First Backup Name Format	The first backup name format to use when generating a global control number. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated global control number (**Use global control number** is set to Yes (and generate name from data)), Gentran Integration Suite tries to generate and match the following control numbers: ◆ First, it tries to generate and match the control number in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the control number SWIFT_100_Outbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing control number that matches the names generated in steps 1-3, a control number with the name assigned in the first step is created.

Field or Check Box	Description
Second Backup Name Format	The second backup name format to use when generating a global control number. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated global control number (**Use global control number** is set to Yes (and generate name from data)), Gentran Integration Suite tries to generate and match the following control numbers: ◆ First, it tries to generate and match the control number in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the control number SWIFT_100_Outbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing control number that matches the names generated in steps 1-3, a control number with the name assigned in the first step is created.
Global Message User Reference (MUR) Control Number	Select the control number with which to start generating control numbers. Optional. Displayed only if **Message User Reference** is set to Use Control Number.
Process Data element from which to set the Message User Reference	Specify the process data element from which the MUR will be set. Displayed only if **Message User Reference** is set to Use value from process data. Required.
Local Message User Reference (MUR)	The message user reference block is an optional section in the SWIFT envelope that the SWIFT network tracks for acknowledgement purposes. If you choose to use a local control number as the user reference, type the starting value of that control number in this parameter.
Expect an acknowledgement for messages sent using this envelope	Whether to expect an acknowledgement for messages that are sent using this envelope. Valid values are Yes and No (default). Required.
Acknowledgement overdue after (hours)	Amount of time, in hours, within which you must receive an acknowledgment. Valid value is four numeric characters. Optional.
Acknowledgement overdue after (minutes)	Amount of time, in minutes, within which you must receive an acknowledgment. Valid value is four numeric characters. Optional.

Field or Check Box	Description
Translate documents prior to enveloping	Whether to translate the documents prior to enveloping them. Valid values are Yes or No (default). Required.
Map Name Mode	How to determine which map to use to translate the message. Required. Valid values are: ◆ Specify (default) ◆ Generate from data (this is useful if you want to share across envelopes) **Note:** If you select this option, you must select at least one parameter for Primary Name Format. This parameter is displayed only if **Translate document prior to enveloping** set to Yes. The generated name will end with FIN.
Map Name	Which map to use to perform a compliance check (if **Translate document prior to enveloping** set to Yes and **Map Name Mode** set to Yes). The map must already be checked in to Gentran Integration Suite. Optional/
Primary Name Format	Check boxes to instruct what information to include when generating and matching a name for the map. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated map name (**Map Name Mode** is set to Generate from data), Gentran Integration Suite tries to generate and match the following maps: ◆ First, it tries to generate and match the map in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the control number SWIFT_100_Outbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing map that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no matching map is found** is set to Yes.

Field or Check Box	Description
First Backup Name Format	The first backup name format to use when generating a map name. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated map name (**Map Name Mode** is set to Generate from data), Gentran Integration Suite tries to generate and match the following maps: ◆ First, it tries to generate and match the map in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the control number SWIFT_100_Outbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing map that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no matching map is found** is set to Yes.

Field or Check Box	Description
Second Backup Name Format	The second backup name format to use when generating a map name. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated map name (**Map Name Mode** is set to Generate from data), Gentran Integration Suite tries to generate and match the following maps: ◆ First, it tries to generate and match the map in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the control number SWIFT_100_Outbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing map that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no matching map is found** is set to Yes.
Generate an error if no matching map is found	Specifies whether to generate an error if the Map Name selected is not found. Valid values are Yes (default) and No. **Note:** Only displayed if **Map Name Mode** is set to Generate from data.

Field or Check Box	Description
On a translation error, determine the Business Process by	How to determine the business process name to use if there were errors in the translation process. Required. Valid values are: ◆ Specifying a Business Process name (default) ◆ Generating a Business Process name from the data (this is useful if you want to share across envelopes) **Note:** If you select this option, you must select at least one parameter for Primary Name Format. The generated name will end with FIN. **Note:** If you select a translation error business process, when a document with compliance errors is encountered an instance of the error business process is generated, using the non-compliant document as the primary document. The processing of that document within the enveloper is halted at that point, and the next document starts processing. If you do not specify a translation error business process, the enveloper continues to process the non-compliant document. In either case, the status report containing the translation errors is created. If HALT_ON_TRANS_ERROR is set to Yes or True in ProcessData, no more documents are processed after a non-compliant document is encountered. If HALT_ON_TRANS_ERROR is set to No or False in ProcessData, the rest of the documents will be processed after a non-compliant document is encountered. If HALT_ON_TRANS_ERROR is not defined, the behavior depends on the enveloping mode; IMMEDIATE mode behaves as if HALT_ON_TRANS_ERROR is set to True, and DEFERRED mode behaves as if HALT_ON_TRANS_ERROR is set to False.

Field or Check Box	Description
Primary Name Format	Check boxes to instruct what information to include when generating and matching a name for the error business process name. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated error business process name (**On a translation error, determine Error Business Process Name By** is set to Generating a Business Process name from the data), Gentran Integration Suite tries to generate and match the following business process: ◆ First, it tries to generate and match the business process in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the business process SWIFT_100_Outbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing business process that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no generated business process name exists in the system** is set to Yes.

Field or Check Box	Description
First Backup Name Format	The first backup name format to use when generating an error business process name. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated error business process name (**On a translation error, determine Error Business Process Name By** is set to Generating a Business Process name from the data), Gentran Integration Suite tries to generate and match the following business process: ◆ First, it tries to generate and match the business process in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the business process SWIFT_100_Outbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing business process that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no generated business process name exists in the system** is set to Yes.

Field or Check Box	Description
Second Backup Name Format	The second backup name format to use when generating an error business process name. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated error business process name (**On a translation error, determine Error Business Process Name By** is set to Generating a Business Process name from the data), Gentran Integration Suite tries to generate and match the following business process: ◆ First, it tries to generate and match the business process in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the business process SWIFT_100_Outbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing business process that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no generated business process name exists in the system** is set to Yes.
Generate an error if no generated business process name exists in the system	Specify whether to generate an error if there is no match to the generated business process name in the system. Valid values are Yes (default) and No. **Note:** Only displayed if **On a translation error, determine the Error Business Process By** is set to Generating the business process name from the data.
Business Process List	Select a previously created business process to associate with this envelope. Displayed only if **On a translation error, determine the Business Process by** is set to Specifying a business process names. Optional.
Message Priority	Specify the priority of the message delivery. Optional. Valid values are: ◆ Normal (default) ◆ Urgent ◆ System

Field or Check Box	Description
Delivery Monitoring	One digit that indicates how monitoring will be performed by the SWIFT network. Optional. Valid values are: ◆ No Delivery Monitoring (default) ◆ 1 (Warning Message) indicates that a warning message will be given if the message is not delivered within a reasonable period ◆ 2 (Delivery Notification) indicates notification when the message is delivered ◆ 3 (Non-Delivery Warning and Delivery Notification) indicates that both the non-delivery warning and delivery notification will be given **Note:** The allowable options are tied to the **Message Priority**: priority **Urgent** requires that the user select 1 or 3, priority **Normal** enables the user to select 2 or No Delivery Monitoring.
Obsolescence Period	Indicates the time after which a Delayed Message trailer will be added to the message by the SWIFT network if it has not yet been delivered. This is also the period after which a non-delivery warning will be generated by the SWIFT network, if the appropriate choice for the **Delivery Monitoring** parameter. Optional. **Note:** This parameter must be three numerics. Each unit represents five minutes (so, for example, 003 equates to 15 minutes). SWIFT requires leading zeros, so the if the number of minutes is less than 3 digits, you must include leading zeroes.
FIN Copy Service Code (tag 103):	A typical configuration requires that the FIN Copy Service Code tag be included in the envelope (usually set to **COP**). Optional. The SWIFT network support the FIN Copy mode, in which a message is sent to an intermediary for approval before it goes to its final destination (or is just copied to the intermediary without requiring approval).
Banking Priority (tag 113):	A four-character optional tag indicating the banking priority. The allowed values are agreed on by you and your trading partner or partners. Optional.
Validation Flag (tag 119):	Specifies the validation flag. Optional. This is an optional part of the header that can contain a code word to indicate that certain types of validations should be performed on the enveloped message. The valid values for this tag vary depending on the message type.
Payment Release Information (tag 115):	Specifies the payment release information. This is an envelope component used in FIN Copy that contains information from the central institution to the receiver of the payment message. The information from this parameter will be placed by the SWIFT network into the MT 097 FIN Copy Message Authorization/Refusal Notification in Y-copy mode.

Field or Check Box	Description
Include Possible Duplicate Emission (PDE) Trailer	Indicates whether to include a trailer specifying that this message may be a duplicate. Required. This is an optional component of the envelope that indicates that this message may already have been sent. For example, if the system crashes during the delivery of a message, another copy of the message could be sent, with this trailer included to indicate that it may be a duplicate. Valid values are Yes and No (default).
Include Training (TNG) Trailer	Specifies whether to include a training (TNG) trailer. Required. This is an optional component of the envelope that indicates the message contained is being sent for system testing purposes. Valid values are Yes and No (default).
Determine the Business Process Name	How to determine the business process name to use if there were no errors in the compliance check. Required. Valid values are: ◆ Specify a Business Process (default) ◆ Generate Business Process Name (from the data) **Note:** If you select this option, you must select at least one parameter for Primary Name Format. The generated name will end with FIN.
Primary Name Format	Check boxes to instruct what information to include when generating and matching a name for the business process. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated business process name (**Determine the Business Process By** is set to Generating the business process name from the data), Gentran Integration Suite tries to generate and match the following business process: ◆ First, it tries to generate and match the business process in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the business process SWIFT_100_Outbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing business process that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no generated business process name exists in the system** is set to Yes.

Field or Check Box	Description
First Backup Name Format	The first backup name format to use when generating a business process name. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated business process name (**Determine Business Process Name By** is set to Generating a Business Process name from the data), Gentran Integration Suite tries to generate and match the following business process: ◆ First, it tries to generate and match the business process in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the business process SWIFT_100_Outbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing business process that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no generated business process name exists in the system** is set to Yes.

Field or Check Box	Description
Second Backup Name Format	The second backup name format to use when generating a business process name. The system allows for two alternatives if the Primary Name Format is not found—checking for the First Backup Name Format and then, if that is not found, checking for the Second Backup Name Format. Optional. Select from: ◆ Sender ID ◆ Sender Branch Code ◆ Receiver ID ◆ Receiver Branch Code ◆ Message Type **Note:** If you are using a generated business process name (**Determine Error Business Process Name By** is set to Generating a Business Process name from the data), Gentran Integration Suite tries to generate and match the following business process: ◆ First, it tries to generate and match the business process in the primary name format (replacing the values that are selected for the corresponding value in the message). For example, if only message type is selected, and the message type is 100, Gentran Integration Suite checks for the business process SWIFT_100_Outbound_FIN. ◆ Second, it tries to generate and match the First Backup Name. ◆ Third, it tries to generate and match the Second Backup Name. ◆ Fourth, if there is not an existing business process that matches the names generated in steps 1-3, an error is generated. **Note:** Only occurs if **Generate an error if no generated business process name exists in the system** is set to Yes.
Generate an error if no generated business process name exists in the system	Specify whether to generate an error if there is no match to the generated business process name in the system. Required. Valid values are Yes (default) and No. **Note:** Only displayed if **Determine the Business Process By** is set to Generate Business Process Name.
Business Process List	Select a previously created business process to associate with this envelope. Optional. Displayed only if **Determine the Business Process Name** is set to Specify Business Process.
Extraction Options	Business process data extraction. Required. Valid values: ◆ Determined by business process (default) ◆ Extract to a file system directory ◆ Extract to a mailbox
Data Extraction Directory	Directory for data extraction. Displayed only if **Extraction Options** set to Extract to a file system directory. Optional.

Field or Check Box	Description
Data Extraction Filename	Filename for data extraction. Displayed only if **Extraction Options** set to Extract to a file system directory. Optional.
Data Extraction Mailbox	Mailbox for data extraction. Displayed only if **Extraction Options** set to Extract to a mailbox. Optional.
Data Extraction Mailbox Message Name	Mailbox message name for data extraction. Displayed only if **Extraction Options** set to Extract to a mailbox. Optional.

Configuring the EDI Encoder Service for SWIFTNet Outbound Messages

The EDI Encoder service determines which envelope will be used on the document. If translations are specified in an envelope, the service determines which map to use. For SWIFTNet outbound messages you need to configure five parameters to allow the outbound message to be sent correctly. See *EDI Encoder Service* for more information.

To configure the EDI Encoder service, you must specify settings for the following fields in the GPM that match the values you configured in the outbound envelope:

- AccepterLookupAlias
- ReceiverID
- SenderID
- ReceiverIDQual
- SenderIDQual

You need to set the sender and receiver variables based on the desired source and destination for the outbound message. These variables also match the corresponding values in the envelope, except that the envelope allows wildcards (which will match documents encoded with any value for that variable), whereas the value used in the EDI encoder must always be the full eight-character identifier and three-character branch code of the desired source and destination.

Or, if you are editing the EDI Encoder BPML, you need to include these parameters and values, where for xxx you substitute the values set up in the outbound envelope:

```
<operation name="EDI Encode">
  <participant name="EDIEncoder"/>
  <output message="EDIEnc_In">
    <assign to="AccepterLookupAlias">xxx</assign>
    <assign to="ReceiverID">xxx</assign>
    <assign to="SenderID">xxx</assign>
    <assign to="ReceiverIDQual">xxx</assign>
    <assign to="SenderIDQual">xxx</assign>
    <assign to="." from="*"/>
  </output>
  <input message="EDIEnc_Out">
    <assign to="." from="*"/>
  </input>
</operation>
```

SWIFT Business Processes

Overview

To help you accomplish your business goals, Gentran Integration Suite provides two predefined enveloping business processes which are used by Gentran Integration Suite to implement SWIFT processing: SWIFTEnvelope and SWIFTDeenvelope. These predefined business processes are initiated by other processes/services during SWIFT processing and do not require you to modify them.

Gentran Integration Suite also provides predefined business processes which are used by Gentran Integration Suite to implement SWIFT transport, including the SWIFTNetClient business process (using for InterAct transport), and the SWIFTNet ClientFA (used for FileAct transport) business process. These predefined business processes are initiated by other processes/services during SWIFT processing but do require you to modify them.

The following table lists business goals for some of the predefined SWIFT business processes:

Business Process	Business Goals
SWIFTDeenvelope	Extracts SWIFT message types from a message and translates and processes them, according to the content of the envelopes.
SWIFTEnvelope	Applies a SWIFT envelope to one or more SWIFT message types and then uses the envelope data to translate and process them.
SWIFTNetClient	Used for InterAct processing. Contains the necessary parameters so the SWIFTNet Client service can prepare the request and send it (outbound) to the Gentran Integration Suite SWIFTNet MEFG Server. The client application on the SWIFTNet MEFG Server processes this request, performs the necessary communication exchange with the SWIFTNet SAG/SNL instance, and sends the request to the SWIFTNet Network.
SWIFTNetClientFA	Used for FileAct processing. This is a bootstrap business process used by the inbound SWIFTNet request coming in through the SWIFTNet MEFG Server for FileAct. It is a system business process used by the SWIFTNet Server adapter.
handleSWIFTNetServerRequest	Used for InterAct processing. Enables Gentran Integration Suite to receive SWIFTNet messages. This is the bootstrap business process used by the inbound SWIFTNet request through the Gentran Integration Suite SWIFTNet MEFG Server. It is a system business process used by the SWIFTNet Server Adapter, which pre-processes the incoming request, search the SWIFTNet Routing Rule table, and route the request payload to the business process for processing.

Business Process	Business Goals
handleSWIFTNetServerSnfRequest	Used for InterAct processing. Enables Gentran Integration Suite to receive SWIFTNet store-and-forward messages. This is a bootstrap business process used by the inbound SWIFTNet request that includes a store-and-forward option. For the store-and-forward option, an incoming request is not processed immediately, but instead is stored in the responder mailbox in Gentran Integration Suite to be responded to later. Gentran Integration Suite then sends an acknowledgment that the request has been successfully stored to the requestor through the Gentran Integration Suite SWIFTNet MEFG Server.
handleSWIFTNetServerFARequest	Used for FileAct processing. Enables Gentran Integration Suite to receive SWIFTNet messages. This is the bootstrap business process used by the inbound SWIFTNet request through the Gentran Integration Suite SWIFTNet MEFG Server. It is a system business process used by the SWIFTNet Server Adapter, which pre-processes the incoming request, search the SWIFTNet Routing Rule table, and route the request payload to the business process for processing.
handleSWIFTNetServerFASnF Request	Used for FileAct processing. Enables Gentran Integration Suite to receive SWIFTNet store-and-forward messages. This is a bootstrap business process used by the inbound SWIFTNet request that includes a store-and-forward option. For the store-and-forward option, an incoming request is not processed immediately, but instead is stored in the responder mailbox in Gentran Integration Suite to be responded to later. Gentran Integration Suite then sends an acknowledgment that the request has been successfully stored to the requestor through the Gentran Integration Suite SWIFTNet MEFG Server.
handleSWIFTNetServerFAEvent	The handleSWIFTNetServerFAEvent business process is used by the Gentran Integration Suite SWIFTNet MEFG Server. It is a system business process called by the SWIFTNet Server adapter that preprocesses the incoming FileAct Event with a COMPLETED status, searches the SWIFTNet routing rule table and bootstraps the business process for processing

SWIFTDeenvelope Business Process

The SWIFTDeenvelope business process is used to deenvelope SWIFT data. A typical scenario is one in which SWIFT data must be received from a trading partner. The data must be deenveloped to extract identifying batch and interchange data, and the SWIFTDeenvelope business process helps to provide the deenveloping services.

The SWIFTDeenvelope business process is initiated as part of the following inbound process flow:

1. You create a business process that calls the EDI Deenveloping service.
2. The EDI Deenveloping service parses the whole document and extracts messages from it so it can be further processed by Gentran Integration Suite.
3. Passes the SWIFT messages to the SWIFTDeenvelope business process. The Generic Deenvelope service runs as a subprocess.
4. The SWIFTDeenvelope business process looks up the envelope, based on the data found in the header.
5. The envelope specifies what to do with the deenveloped messages.

6. Starts the Invoke Business Process service or a subprocess service to initiate the appropriate business process to handle each deenveloped message.

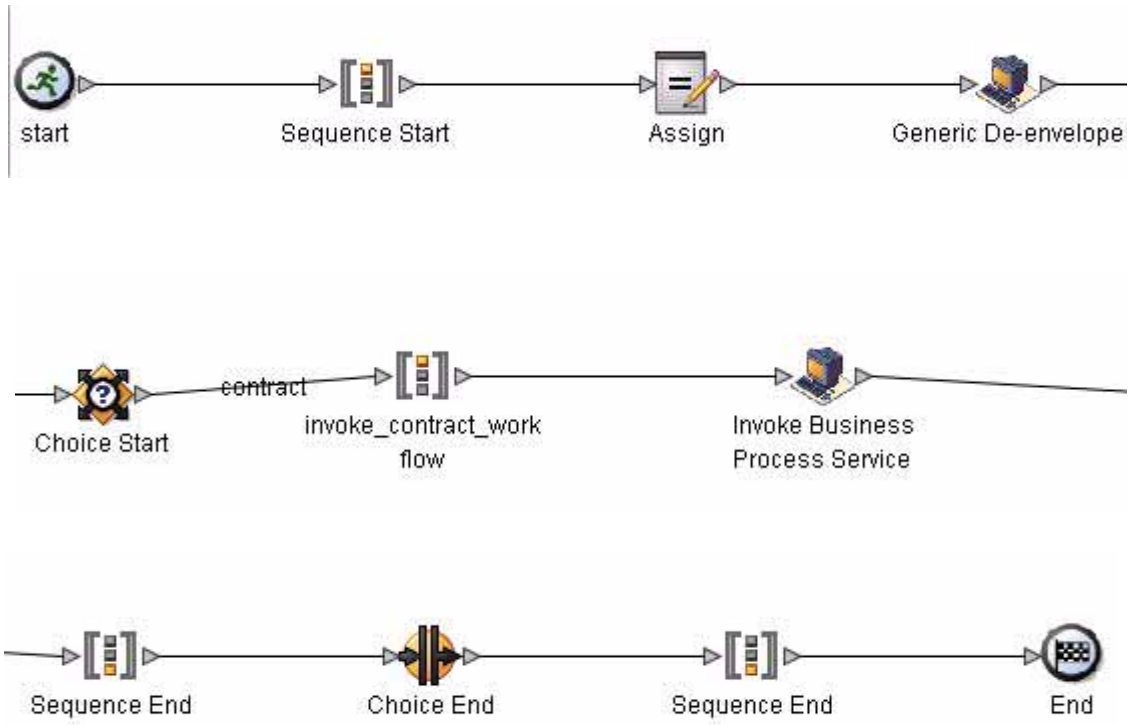
If exceptions occur when running an EDI Deenveloping business process, Gentran Integration Suite generates an EDI Compliance Report.

This table lists the configuration parameters for the SWIFTDeenvelope business process:

Parameter	Default	Description
Document Tracking	False	When document tracking is enabled for a business process, tracking information is carried with the message throughout the process, and the tracking information is persisted about the message regardless of the persistence level you configured globally for Gentran Integration Suite.
Set onfault processing	False	Onfault processing allows the process to immediately execute the on-fault activity specified in the process, even if the process has not yet reached that step in the process. For example, if a process fails at step 3, but the on-fault activity is specified in step 7, if onfault processing is enabled, the process proceeds to the step 7 on-fault rather than halting at step 3.
Queue	4	Gentran Integration Suite enables you to set performance optimizations by queue, defining queue levels to allocate resources. This number indicates the previously allocated queue level that you want for this business process model for processing.
Persistence Level	System Default	The level of data to retain for generating a status report that describes each step that the business process completes. System default indicates that, for the data, configuration is already defined in Gentran Integration Suite to retain data.
Recovery Level	Manual	The level of recovery for this business process if the business process should halt during execution. Manual requires you to resume or restart the business process manually.
Document Storage Type	System Default	The level of document storage for messages that process when the business process runs. System Default specifies to store messages in the file system or database, according to how you configured archiving and purging in Gentran Integration Suite.
Life Span	Life Span Days — 2 Life Span Hours — 0 Life Span Type — System Level Removal Method — Archive	The length of time, in days and hours, to retain the data in Gentran Integration Suite, along with the life span type and removal method.

Parameter	Default	Description
Complete by Deadline	None Available **Note:** To set a deadline you must change it in the business process.	Complete by – The deadline time, in hours and minutes, by which the business process must complete process once it starts. ◆ First Notification: Hours and Minutes – Whether to receive notification before a business process deadline. ◆ Second Notification: Hours and Minutes – Whether to receive another notification before a business process deadline.
Event Reporting Level	Full	The level of event reporting that is retrieved for this business process when it runs. Full specifies to generate events for the business process, including the business process start and end time, start and end times for all services or services running as a result of this business processes, and any resulting errors and exceptions.

The following figure shows the business process model in the GPM, which makes up the SWIFT Deenvelope business process:



The following BPML code makes up the SWIFTDeenvelope business process:

```
<process name="SWIFTDeenvelope">
  <rule name="contract">
```

```
    <condition>CONTRACT_FOUND = &amp;quot;YES&amp;quot;</condition>
  </rule>

  <sequence>
    <assign to="RunInValidationMode">FALSE</assign>
    <operation>
      <participant name="DeenvelopeGeneric"/>
      <output message="Xout">
        <assign to="map_name">SWIFT_FIN_INBOUND</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <choice>
      <select>
        <case ref="contract" activity="invoke_contract_workflow"/>
      </select>

      <sequence name="invoke_contract_workflow">
        <operation>
          <participant name="InvokeBusinessProcessService"/>
          <output message="Xout">
            <assign to="INVOKE_MODE">ASYNC</assign>
            <assign to="." from="*"></assign>
          </output>
          <input message="Xin">
            <assign to="." from="*"></assign>
          </input>
        </operation>

      </sequence>
    </choice>
  </sequence>
</process>
```

Before Using the SWIFTDeenvelope Business Process

Before you use the SWIFTDeenvelope business process, you must complete the following task:

1. Create a SWIFT inbound envelope for each SWIFT message type that you are receiving. See *Inbound SWIFT envelope* on page 23.

SWIFTEnvelope Business Process

The SWIFTEnvelope business process is initiated when it is called by another business process. The SWIFTEnvelope business process envelopes the SWIFT messages contained in the business process context with outbound SWIFT envelopes that you have preconfigured. You must have created one outbound SWIFT envelope for each SWIFT message type that you are sending.

A typical scenario is one in which SWIFT data must be sent to a trading partner. To prepare for this, the data must be enveloped to provide identifying batch and interchange data. The SWIFTEnvelope business process helps to provide these enveloping services.

The SWIFTEnvelope business process is initiated as part of the following outbound process flow:

1. You create a business process that calls the EDI Encoder service or Document Extraction service.
2. The EDI Encoder service or Document Extraction service looks up the envelope to apply and prepares the document to be enveloped.
3. Either the business process calls the EDI Enveloping service or the Document Extraction service is configured to perform enveloping.
4. The EDI Enveloping service or Document Extraction service starts the SWIFTEnvelope business process (which runs the Generic Envelope service as a subprocess to extract the name of the business process).
5. The SWIFTEnvelope business process searches the envelope definition to retrieve information to envelope each message.

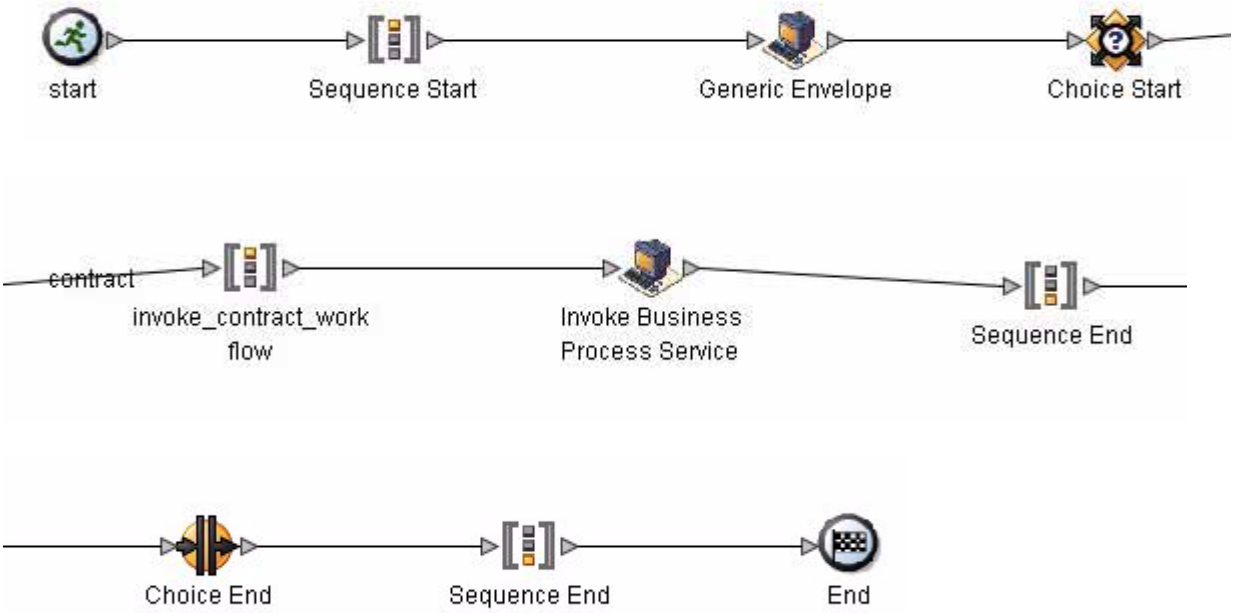
Note: The Sender ID, Receiver ID, and Lookup Alias in your Outbound Envelope definition must match the parameters that you define for this outbound business process.

This table lists the configuration parameters for the SWIFTEnvelope business process:

Parameter	Default	Description
Document Tracking	False	When document tracking is enabled for a business process, tracking information is carried with the message throughout the process, and the tracking information is persisted about the message regardless of the persistence level you configured globally for Gentran Integration Suite.
Set onfault processing	False	Onfault processing allows the process to immediately execute the on-fault activity specified in the process, even if the process has not yet reached that step in the process. For example, if a process fails at step 3, but the on-fault activity is specified in step 7, if onfault processing is enabled, the process proceeds to the step 7 on-fault rather than halting at step 3.
Queue	4	Gentran Integration Suite enables you to set performance optimizations by queue, defining queue levels to allocate resources. This number indicates that previously allocated queue level that you want for this business process model for processing.
Persistence Level	System Default	The level of data to retain for generating a status report that describes each step that the business process completes. System default indicates that, for the data, configuration is already defined in Gentran Integration Suite to retain data.
Recovery Level	Manual	The level of recovery for this business process if the business process should halt during execution. Manual requires you to resume or restart the business process manually.

Parameter	Default	Description
Document Storage Type	System Default	The level of document storage for messages that process when the business process runs. System Default specifies to store messages in the file system or database, according to how you configured archiving and purging in Gentran Integration Suite.
Life Span	Life Span Days — 2 Life Span Hours — 0 Life Span Type — System Level Removal Method — Archive	The length of time, in days and hours, to retain the data in Gentran Integration Suite, along with the life span type and removal method.
Complete by Deadline	None Available **Note:** To set a deadline you must change it in the business process.	Complete by – The deadline time, in hours and minutes, by which the business process must complete process once it starts. ◆ First Notification: Hours and Minutes – Whether to receive notification before a business process deadline. ◆ Second Notification: Hours and Minutes – Whether to receive another notification before a business process deadline.
Event Reporting Level	Full	The level of event reporting that is retrieved for this business process when it runs. Full specifies to generate events for the business process, including the business process start and end time, start and end times for all services or services running as a result of this business processes, and any resulting errors and exceptions.

The following figure shows the business process model in the GPM, which makes up the SWIFT Envelope business process:



The following BPML code makes up the SWIFT Envelope business process:

```
<process name="SWIFTEnvelope">

  <rule name="contract">
    <condition>CONTRACT_FOUND = &quot;YES&quot;</condition>
  </rule>

  <sequence>
    <operation>
      <participant name="EnvelopeGeneric" />
      <output message="Xout" >
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin" >
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <choice>
      <select>
        <case ref="contract" activity="invoke_contract_workflow" />
      </select>
      <sequence name="invoke_contract_workflow">
        <operation>
          <participant name="InvokeBusinessProcessService" />
          <output message="Xout">
```

```
            <assign to="INVOKE_MODE">ASYNC</assign>
            <assign to="." from="*"></assign>
          </output>
          <input message="Xin" >
            <assign to="." from="*"></assign>
          </input>
        </operation>
      </sequence>
    </choice>

  </sequence>
</process>
```

Before Using the SWIFTEnvelope Business Process

Before you use the SWIFTEnvelope business process, you must complete the following task:

1. Create a SWIFT outbound envelope for each SWIFT message type that you are sending. See *Outbound SWIFT envelope* on page 37.

SWIFTNetClient Business Process

The SWIFTNetClient business process contains the necessary parameters so the SWIFTNet Client service can prepare the request and send it to the Gentran Integration Suite SWIFTNet MEFG Server. The client application on the SWIFTNet MEFG Server processes this request, performs the necessary communication exchange with the SWIFTNet SAG/SNL instance, and sends the request to the SWIFTNet Network.

Note: This business process is used for InterAct processing only. It takes the default parameters configured in the SWIFTNet Client service. If a parameter defined in the SWIFTNet Client Service is specified in this business process, the value in the business process overrides the parameter in the SWIFTNet Client service.

The SWIFTNetClient business process is initiated as part of the following outbound process flow:

1. The SWIFTNetClient business process invokes the SWIFTNet Client service and passes it all the necessary parameters to send a request.
2. The client application on the SWIFTNet MEFG Server processes the request.
3. The SWIFTNet Client service invokes the SOAP Outbound service.
4. The SOAP Outbound service:
   - Checks for the presence of SOAPFault in process data, and if it exists, uses the SOAP fault data to generate a SOAP fault.
   - If there is no fault and Gentran Integration Suite is not in intermediate mode, it adds SOAP enveloping, including header blocks, to the primary document.
   - Generates the HTTP response code header.
5. Then the SOAP Outbound service invokes the HTTP Client Begin Session service.
6. The HTTP Client Begin Session service starts an HTTP session and invokes the HTTP Client Post service.

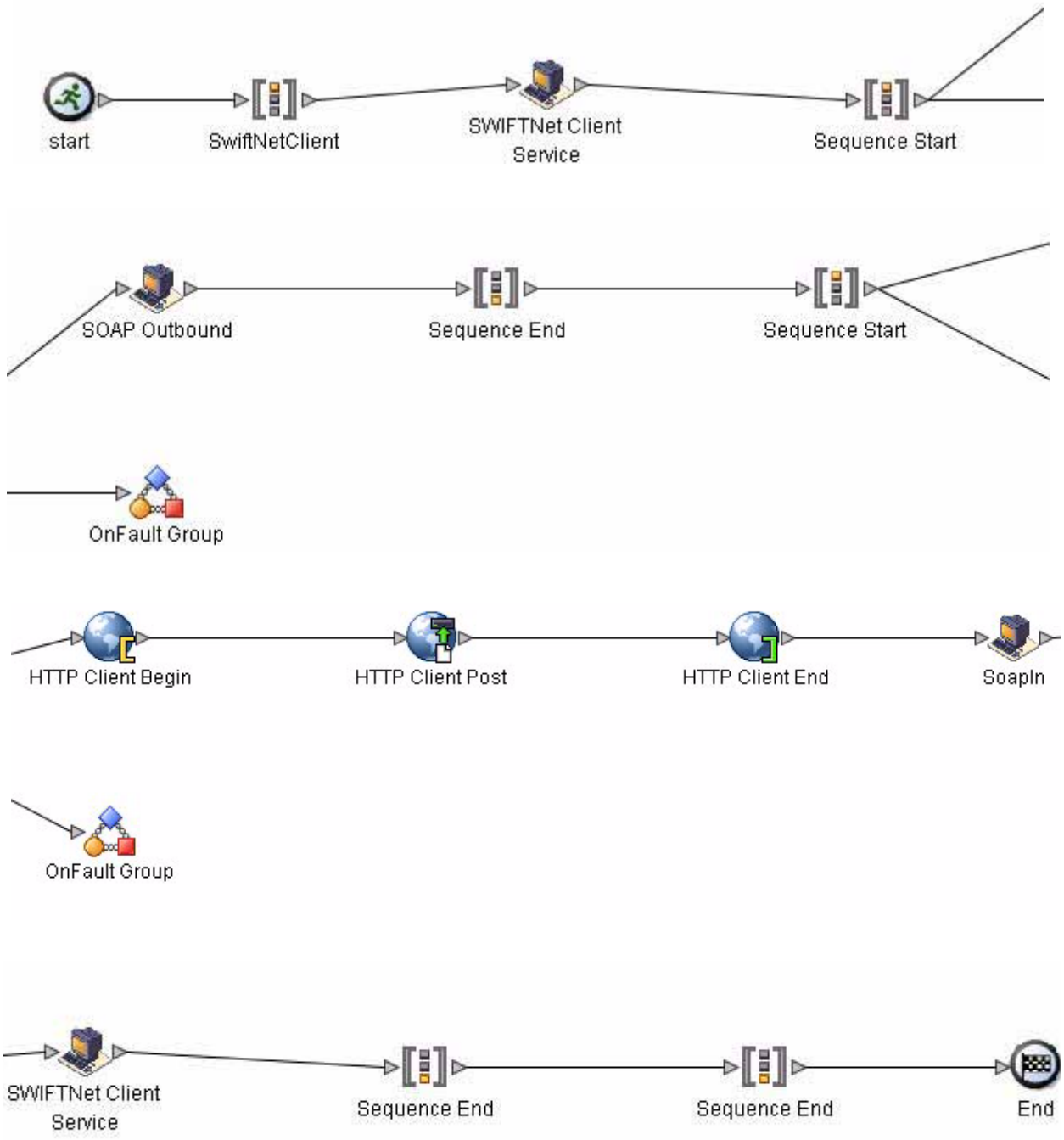
7. The HTTP Client Post service sends HTTP POST requests to the trading partner's HTTP server through the perimeter server.
8. The HTTP Client Post service invokes the HTTP Client End Session service.
9. The HTTP Client End Session service ends the HTTP session with the external trading partner HTTP server and invokes the SOAP Inbound service.
10. The SOAP Inbound service uses the URI to which the document was posted to set the business process that should be invoked to handle the request. The service also adds any SOAP headers targeted at it to process data, removes the SOAP enveloping from the request, and makes the SOAP payload the primary document.
11. The SOAP Inbound service reinvokes the SWIFTNet Client service to pass the request to the SWIFTNet network.

This table lists the configuration parameters for the SWIFTNetClient business process:

Parameter	Default	Description
Document Tracking	False	When document tracking is enabled for a business process, tracking information is carried with the message throughout the process, and the tracking information is persisted about the message regardless of the persistence level you configured globally for Gentran Integration Suite.
Set onfault processing	False	Onfault processing allows the process to immediately execute the on-fault activity specified in the process, even if the process has not yet reached that step in the process. For example, if a process fails at step 3, but the on-fault activity is specified in step 7, if onfault processing is enabled, the process proceeds to the step 7 on-fault rather than halting at step 3.
Start mode	async	Asynchronous initiation is selected by default. Starting business processes asynchronously is recommended. Asynchronous mode is standard Gentran Integration Suite processing, wherein the business process is placed in queue and processed.
Transaction	False	This option instructs Gentran Integration Suite to treat the entire process as a single transaction so that either all of the steps complete, or, in the event of an error, none of them do. When an error occurs, no data is committed; data returns to its pre-process state. By default, this transaction mode is not enabled.
Queue	4	Gentran Integration Suite enables you to set performance optimizations by queue, defining queue levels to allocate resources. This number indicates the previously allocated queue level that you want for this business process model for processing.

Parameter	Default	Description
Persistence Level	System Default	The level of data to retain for generating a status report that describes each step that the business process completes. System default indicates that, for the data, configuration is already defined in Gentran Integration Suite to retain data.
Recovery Level	Manual	The level of recovery for this business process if the business process should halt during execution. Manual requires you to resume or restart the business process manually.
Document Storage Type	System Default	The level of document storage for messages that process when the business process runs. System Default specifies to store messages in the file system or database, according to how you configured archiving and purging in Gentran Integration Suite.
Life Span	Life Span Days — 2 Life Span Hours — 0 Life Span Type — System Level Removal Method — Archive	The length of time, in days and hours, to retain the data in Gentran Integration Suite, along with the life span type and removal method.
Complete by Deadline	None Available **Note:** To set a deadline you must change it in the business process.	Complete by – The deadline time, in hours and minutes, by which the business process must complete process once it starts. ◆ First Notification: Hours and Minutes – Whether to receive notification before a business process deadline. ◆ Second Notification: Hours and Minutes – Whether to receive another notification before a business process deadline.
Event Reporting Level	Full	The level of event reporting that is retrieved for this business process when it runs. Full specifies to generate events for the business process, including the business process start and end time, start and end times for all services or services running as a result of this business processes, and any resulting errors and exceptions.

The following figure shows the business process model in the GPM, which makes up the SWIFTNetClient business process:



The following BPML code makes up the SWIFTNetClient business process:

Note: The **bold** lines indicate information that you need to modify to match your installation.

```
<process name="SWIFTNetClient">
   <sequence name="SWIFTNetClientService">
      <operation name="set user token">
         <participant name="SetUserToken"/>
         <output message="SetUserTokenMessage">
            <assign to="USER_TOKEN">admin</assign>
            <assign to="." from="*"/>
         </output>
         <input message="inmsg">
            <assign to="." from="*"/>
         </input>
      </operation>
      <!-- build SWIFTNET request -->
      <operation>
         <participant name="SWIFTNetClientService"/>
         <output message="handleClientRequest">
            <assign to="." from="*"/>
         </output>
         <input message="testing">
            <assign to="." from="*"/>
         </input>
      </operation>
      <operation>
         <participant name="SOAPOutbound"/>
         <output message="output">
            <assign to="." from="*"/>
            <assign to="SOAP_MODE">send</assign>
            <assign
to="SOAPEnvNSURI">http://schemas.xmlsoap.org/soap/envelope/</assign>
            <assign
to="SOAPRequestURL">http://00.000.00.000:00000/soap/SWIFTNetClientRequest</assign>
         </output>
         <input message="input">
            <assign to="." from="*"/>
         </input>
      </operation>
      <!-- Sequence to send HTTP request -->
      <sequence>
         <operation name="HTTP Client Begin">
            <participant name="HTTPClientBeginSession"/>
            <output message="HttpClientBeginServiceInputMessage">
               <assign to="." from="PrimaryDocument"/>
               <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
               <assign to="RemoteHost">00.000.00.000</assign>
               <assign to="RemotePort">00000</assign>
            </output>
            <input message="inmsg">
               <assign to="." from="*"/>
            </input>
         </operation>
         <operation name="HTTP Client Post">
            <participant name="HTTPClientPost"/>
            <output message="HttpClientPostServiceInputMessage">
               <assign to="." from="PrimaryDocument"/>
               <assign to="SessionToken" from="SessionToken/text()"/>
```

```
                <assign to="RawResponse">true</assign>
                <assign to="URI">/soap/SWIFTNetClientRequest</assign>
            </output>
            <input message="inmsg">
                <assign to="." from="*"/>
            </input>
        </operation>
        <operation name="HTTP Client End">
            <participant name="HTTPClientEndSession"/>
            <output message="HttpClientEndServiceInputMessage">
                <assign to="SessionToken" from="SessionToken/text()"/>
            </output>
            <input message="inmsg">
                <assign to="." from="*"/>
            </input>
        </operation>
        <onFault>
            <sequence>
                <operation name="HTTP Client End">
                    <participant name="HTTPClientEndSession"/>
                    <output message="HttpClientEndServiceInputMessage">
                        <assign to="SessionToken" from="SessionToken/text()"/>
                    </output>
                    <input message="inmsg">
                        <assign to="." from="*"/>
                    </input>
                </operation>
            </sequence>
        </onFault>
    </sequence>
    <sequence>
        <operation name="SoapIn">
            <participant name="SOAPInbound"/>
            <output message="output">
                <assign to="." from="*"/>
                <assign to="bootstrap">false</assign>
                <assign to="SOAP_INTERMEDIATE_NODE">false</assign>
            </output>
            <input message="input">
                <assign to="." from="*"/>
            </input>
        </operation>
        <onFault>
            <sequence>
                <assign to="SOAPfaultcode">Server</assign>
                <assign to="SOAPfaultstring">There was an error processing the SOAP
request.</assign>
                <assign to="SOAPdetail">An error occurred while processing the SOAP
request.</assign>
                <assign
to="SOAPRequestURL">http://00.000.00.000:00000/soap/SWIFTNetClientRequest</assign>
            </sequence>
        </onFault>
    </sequence>
    <sequence>
        <operation>
```

```
            <participant name="SWIFTNetClientService"/>
            <output message="handleClientResponse">
               <assign to="." from="*"/>
            </output>
            <input message="testing">
               <assign to="." from="*"/>
            </input>
         </operation>
      </sequence>
   </sequence>
</process>
```

Before Using the SWIFTNetClient Business Process

Before you use the SWIFTNetClient business process, you must complete the following tasks:

1. Configure the SWIFTNet Client service (or create a new instance of it) to reflect your installation. See *SWIFTNet Client Service*.
2. Enable Document Tracking in the BPML Editor.
3. Configure the SWIFTNetClient business process to change the IP address and port to that of the remote Gentran Integration Suite SWIFTNet MEFG Server in the following locations (**bold** in business process example above):
   - **SOAPRequestURL** of participant SOAPOutbound
   - **SOAPRequestURL** of onFault
   - **RemoteHost and RemotePort** of participant HTTPClientBeginSession

   Then you can execute the SWIFTNetClient business process as part of your SWIFTNet processing.

SWIFTNetClientFA Business Process

This is a bootstrap business process used by the inbound SWIFTNet request coming in through the SWIFTNet MEFG Server for FileAct. It is a system business process used by the SWIFTNet Server adapter.

Note: This business process is used with FileAct processing only. It takes the default parameters configured in the SWIFTNet Client service. If a parameter defined in the SWIFTNet Client Service is specified in this business process, the value in the business process overrides the parameter in the SWIFTNet Client service.

The SWIFTNetClientFA business process is initiated as part of the following outbound process flow:

1. The SWIFTNetClientFA business process invokes the SWIFTNet Client service and passes it all the necessary parameters to send a request.
2. The client application on the SWIFTNet MEFG Server processes the request.
3. The SWIFTNet Client service invokes the SOAP Outbound service.
4. The SOAP Outbound service:
   - Checks for the presence of SOAPFault in process data, and if it exists, uses the SOAP fault data to generate a SOAP fault.

- If there is no fault and Gentran Integration Suite is not in intermediate mode, it adds SOAP enveloping, including header blocks, to the primary document.
- Generates the HTTP response code header.

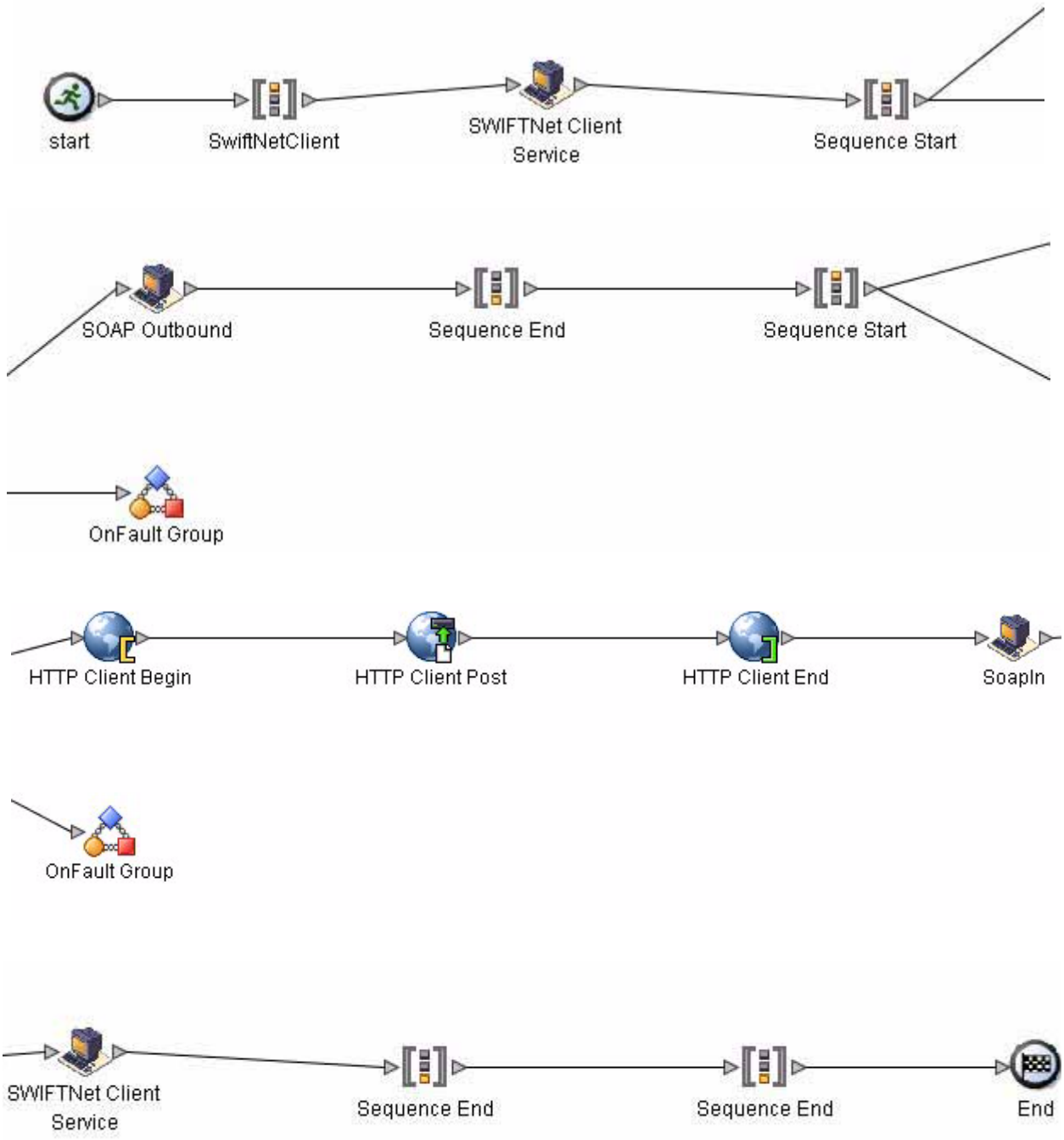
5. Then the SOAP Outbound service invokes the HTTP Client Begin Session service.
6. The HTTP Client Begin Session service starts an HTTP session and invokes the HTTP Client Post service.
7. The HTTP Client Post service sends HTTP POST requests to the trading partner's HTTP server through the perimeter server.
8. The HTTP Client Post service invokes the HTTP Client End Session service.
9. The HTTP Client End Session service ends the HTTP session with the external trading partner HTTP server and invokes the SOAP Inbound service.
10. The SOAP Inbound service uses the URI to which the document was posted to set the business process that should be invoked to handle the request. The service also adds any SOAP headers targeted at it to process data, removes the SOAP enveloping from the request, and makes the SOAP payload the primary document.
11. The SOAP Inbound service reinvokes the SWIFTNet Client service to pass the request to the SWIFTNet network.

This table lists the configuration parameters for the SWIFTNetClientFA business process:

Parameter	Default	Description
Document Tracking	False	When document tracking is enabled for a business process, tracking information is carried with the message throughout the process, and the tracking information is persisted about the message regardless of the persistence level you configured globally for Gentran Integration Suite.
Set onfault processing	False	Onfault processing allows the process to immediately execute the on-fault activity specified in the process, even if the process has not yet reached that step in the process. For example, if a process fails at step 3, but the on-fault activity is specified in step 7, if onfault processing is enabled, the process proceeds to the step 7 on-fault rather than halting at step 3.
Start mode	async	Asynchronous initiation is selected by default. Starting business processes asynchronously is recommended. Asynchronous mode is standard Gentran Integration Suite processing, wherein the business process is placed in queue and processed.
Transaction	False	This option instructs Gentran Integration Suite to treat the entire process as a single transaction so that either all of the steps complete, or, in the event of an error, none of them do. When an error occurs, no data is committed; data returns to its pre-process state. By default, this transaction mode is not enabled.

Parameter	Default	Description
Queue	4	Gentran Integration Suite enables you to set performance optimizations by queue, defining queue levels to allocate resources. This number indicates the previously allocated queue level that you want for this business process model for processing.
Persistence Level	System Default	The level of data to retain for generating a status report that describes each step that the business process completes. System default indicates that, for the data, configuration is already defined in Gentran Integration Suite to retain data.
Recovery Level	Manual	The level of recovery for this business process if the business process should halt during execution. Manual requires you to resume or restart the business process manually.
Document Storage Type	System Default	The level of document storage for messages that process when the business process runs. System Default specifies to store messages in the file system or database, according to how you configured archiving and purging in Gentran Integration Suite.
Life Span	Life Span Days — 2 Life Span Hours — 0 Life Span Type — System Level Removal Method — Archive	The length of time, in days and hours, to retain the data in Gentran Integration Suite, along with the life span type and removal method.
Complete by Deadline	None Available **Note:** To set a deadline you must change it in the business process.	Complete by – The deadline time, in hours and minutes, by which the business process must complete process once it starts. ◆ First Notification: Hours and Minutes – Whether to receive notification before a business process deadline. ◆ Second Notification: Hours and Minutes – Whether to receive another notification before a business process deadline.
Event Reporting Level	Full	The level of event reporting that is retrieved for this business process when it runs. Full specifies to generate events for the business process, including the business process start and end time, start and end times for all services or services running as a result of this business processes, and any resulting errors and exceptions.

The following figure shows the business process model in the GPM, which makes up the SWIFTNetClientFA business process:



The following BPML code makes up the SWIFTNetClientFA business process:

Note: The **bold** lines indicate information that you need to modify to match your installation.

```
<process name="SWIFTNetClientFA">
   <sequence name="SWIFTNetClientService">
      <operation name="set user token">
         <participant name="SetUserToken"/>
         <output message="SetUserTokenMessage">
            <assign to="USER_TOKEN">admin</assign>
            <assign to="." from="*"/>
         </output>
         <input message="inmsg">
            <assign to="." from="*"/>
         </input>
      </operation>
      <!-- build SWIFTNET request -->
      <operation>
         <participant name="SWIFTNetClientService"/>
         <output message="handleClientRequest">
            <assign to="." from="*"/>
            <assign to="physicalFilename" from="''"/>
            <assign to="logicalFilename" from="''"/>
            <assign to="transferInfo" from="''"/>
            <assign to="transferDesc" from="''"/>
            <assign to="fileInfo" from="''"/>
            <assign to="fileDesc" from="''"/>
         </output>
         <input message="testing">
            <assign to="." from="*"/>
         </input>
      </operation>
      <operation>
         <participant name="SOAPOutbound"/>
         <output message="output">
            <assign to="." from="*"/>
            <assign to="SOAP_MODE">send</assign>
            <assign
to="SOAPEnvNSURI">http://schemas.xmlsoap.org/soap/envelope/</assign>
            <assign
to="SOAPRequestURL">http://00.000.00.000:00000/soap/SWIFTNetClientRequest</assign>
         </output>
         <input message="input">
            <assign to="." from="*"/>
         </input>
      </operation>
      <!-- Sequence to send HTTP request -->
      <sequence>
         <operation name="HTTP Client Begin">
            <participant name="HTTPClientBeginSession"/>
            <output message="HttpClientBeginServiceInputMessage">
               <assign to="." from="PrimaryDocument"/>
               <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
               <assign to="RemoteHost">00.000.00.000</assign>
               <assign to="RemotePort">00000</assign>
            </output>
            <input message="inmsg">
               <assign to="." from="*"/>
            </input>
```

```
            </operation>
            <operation name="HTTP Client Post">
               <participant name="HTTPClientPost"/>
               <output message="HttpClientPostServiceInputMessage">
                  <assign to="." from="PrimaryDocument"/>
                  <assign to="SessionToken" from="SessionToken/text()"/>
                  <assign to="RawResponse">true</assign>
                  <assign to="URI">/soap/SWIFTNetClientRequest</assign>
               </output>
               <input message="inmsg">
                  <assign to="." from="*"/>
               </input>
            </operation>
            <operation name="HTTP Client End">
               <participant name="HTTPClientEndSession"/>
               <output message="HttpClientEndServiceInputMessage">
                  <assign to="SessionToken" from="SessionToken/text()"/>
               </output>
               <input message="inmsg">
                  <assign to="." from="*"/>
               </input>
            </operation>
            <onFault>
               <sequence>
                  <operation name="HTTP Client End">
                     <participant name="HTTPClientEndSession"/>
                     <output message="HttpClientEndServiceInputMessage">
                        <assign to="SessionToken" from="SessionToken/text()"/>
                     </output>
                     <input message="inmsg">
                        <assign to="." from="*"/>
                     </input>
                  </operation>
               </sequence>
            </onFault>
         </sequence>
         <sequence>
            <operation name="SoapIn">
               <participant name="SOAPInbound"/>
               <output message="output">
                  <assign to="." from="*"/>
                  <assign to="bootstrap">false</assign>
                  <assign to="SOAP_INTERMEDIATE_NODE">false</assign>
               </output>
               <input message="input">
                  <assign to="." from="*"/>
               </input>
            </operation>
            <onFault>
               <sequence>
                  <assign to="SOAPfaultcode">Server</assign>
                  <assign to="SOAPfaultstring">There was an error processing the SOAP
request.</assign>
                  <assign to="SOAPdetail">An error occurred while processing the SOAP
request.</assign>
```

```
                <assign
to="SOAPRequestURL">http://00.000.00.000:00000/soap/SWIFTNetClientRequest</assign>
            </sequence>
         </onFault>
      </sequence>
      <sequence>
         <operation>
            <participant name="SWIFTNetClientService"/>
            <output message="handleClientResponse">
               <assign to="." from="*"/>
            </output>
            <input message="testing">
               <assign to="." from="*"/>
            </input>
         </operation>
      </sequence>
   </sequence>
</process>
```

Before Using the SWIFTNetClientFA Business Process

Before you use the SWIFTNetClientFA business process, you must complete the following tasks:

1. Configure the SWIFTNet Client service (or create a new instance of it) to reflect your installation. See *SWIFTNet Client Service*.
2. Enable Document Tracking in the BPML Editor.
3. Configure the SWIFTNetClientFA business process to change the IP address and port to that of the remote Gentran Integration Suite SWIFTNet MEFG Server in the following locations (**bold** in business process example above):
   - **SOAPRequestURL** of participant SOAPOutbound
   - **SOAPRequestURL** of onFault
   - **RemoteHost and RemotePort** of participant HTTPClientBeginSession

   Then you can execute the SWIFTNetClientFA business process as part of your SWIFTNet processing.

handleSWIFTNetServerRequest Business Process

The handleSWIFTNetServerRequest business process enables Gentran Integration Suite to receive SWIFTNet messages. This is the bootstrap business process used by the inbound SWIFTNet request through the SWIFTNet MEFG Server. It is a system business process used by the SWIFTNet Server adapter, which pre-processes the incoming request, search the SWIFTNet Routing Rule table, and route the request payload to the business process for processing.

Note: This business process is used with InterAct processing only.

The handleSWIFTNetServerRequest business process is initiated as part of the following inbound process flow:

1. The SWIFTNet Server adapter invokes the handleSWIFTNetServerRequest business process and passes it all the necessary parameters to send a request.

2. The handleSWIFTNetServerRequest business process invokes the SOAP Inbound service.
3. The SOAP Inbound service:
   - Pre-processes the incoming request.
   - Searches the SWIFTNet Routing Rule table.
   - Routes the request payload to the business process for processing.
4. Then the SOAP Outbound service invokes the HTTP Response service.
5. The HTTP Response service sends a response to the request.

This table lists the configuration parameters for the handleSWIFTNetServerRequest business process:

Parameter	Default	Description
Document Tracking	False	When document tracking is enabled for a business process, tracking information is carried with the message throughout the process, and the tracking information is persisted about the message regardless of the persistence level you configured globally for Gentran Integration Suite.
Set onfault processing	False	Onfault processing allows the process to immediately execute the on-fault activity specified in the process, even if the process has not yet reached that step in the process. For example, if a process fails at step 3, but the on-fault activity is specified in step 7, if onfault processing is enabled, the process proceeds to the step 7 on-fault rather than halting at step 3.
Start mode	async	Asynchronous initiation is selected by default. Starting business processes asynchronously is recommended. Asynchronous mode is standard Gentran Integration Suite processing, wherein the business process is placed in queue and processed.
Transaction	False	This option instructs Gentran Integration Suite to treat the entire process as a single transaction so that either all of the steps complete, or, in the event of an error, none of them do. When an error occurs, no data is committed; data returns to its pre-process state. By default, this transaction mode is not enabled.
Queue	4	Gentran Integration Suite enables you to set performance optimizations by queue, defining queue levels to allocate resources. This number indicates the previously allocated queue level that you want for this business process model for processing.
Persistence Level	System Default	The level of data to retain for generating a status report that describes each step that the business process completes. System default indicates that, for the data, configuration is already defined in Gentran Integration Suite to retain data.
Recovery Level	Manual	The level of recovery for this business process if the business process should halt during execution. Manual requires you to resume or restart the business process manually.
Document Storage Type	System Default	The level of document storage for messages that process when the business process runs. System Default specifies to store messages in the file system or database, according to how you configured archiving and purging in Gentran Integration Suite.

Parameter	Default	Description
Life Span	Life Span Days — 2 Life Span Hours — 0 Life Span Type — System Level Removal Method — Archive	The length of time, in days and hours, to retain the data in Gentran Integration Suite, along with the life span type and removal method.
Complete by Deadline	None Available **Note:** To set a deadline you must change it in the business process.	Complete by – The deadline time, in hours and minutes, by which the business process must complete process once it starts. ◆ First Notification: Hours and Minutes – Whether to receive notification before a business process deadline. ◆ Second Notification: Hours and Minutes – Whether to receive another notification before a business process deadline.
Event Reporting Level	Full	The level of event reporting that is retrieved for this business process when it runs. Full specifies to generate events for the business process, including the business process start and end time, start and end times for all services or services running as a result of this business processes, and any resulting errors and exceptions.

The following BPML code makes up the handleSWIFTNetServerRequest business process:

```
<process name="handleSWIFTNetServerRequest">
   <sequence>
      <operation name="set user token">
         <participant name="SetUserToken"/>
         <output message="SetUserTokenMessage">
            <assign to="USER_TOKEN">admin</assign>
            <assign to="." from="*"/>
         </output>
         <input message="inmsg">
            <assign to="." from="*"/>
         </input>
      </operation>
      <operation name="SoapIn">
         <participant name="SOAPInbound"/>
         <output message="output">
            <assign to="." from="*"/>
            <assign to="bootstrap">false</assign>
            <assign to="SOAP_INTERMEDIATE_NODE">false</assign>
         </output>
         <input message="input">
            <assign to="." from="*"/>
         </input>
      </operation>
      <operation>
         <participant name="SWIFTNetServerAdapter"/>
         <output message="handleServerRequest">
```

```
            <assign to="." from="*"/>
         </output>
         <input message="testing">
            <assign to="." from="*"/>
         </input>
      </operation>
      <!-- internal processing by invoking a subprocess -->
      <!-- business-specific processing that will return a response for InterAct -->
      <operation>
         <participant name="InvokeSubProcessService"/>
         <output message="Xout">
            <assign to="INVOKE_MODE">SYNC</assign>
            <assign to="." from="*"/>
         </output>
         <input message="Xin">
            <assign to="." from="*"/>
         </input>
      </operation>
      <!-- this is to construct the server response message back to GIS Server
application -->
      <operation>
         <participant name="SWIFTNetServerAdapter"/>
         <output message="handleServerResponse">
            <assign to="." from="*"/>
            <assign to="interface" from="SwiftServerRequest/interface/text()"/>
            <assign to="messageID" from="SwiftServerRequest/messageID/text()"/>
            <assign to="Status">Accepted</assign>
            <assign to="deliveryNotification"
from="SwiftServerRequest/deliveryNotification/text()"/>
            <assign to="SnF" from="SwiftServerRequest/SnF/text()"/>
         </output>
         <input message="testing">
            <assign to="." from="*"/>
         </input>
      </operation>
      <operation name="SoapOut">
         <participant name="SOAPOutbound"/>
         <output message="output">
            <assign to="." from="*"/>
            <assign to="SOAP_MODE">respond</assign>
         </output>
         <input message="input">
            <assign to="." from="*"/>
         </input>
      </operation>
      <assign to="doc-has-headers">true</assign>
      <operation name="HttpResponse">
         <participant name="HttpRespond"/>
         <output message="Xout">
            <assign to="." from="*"/>
         </output>
         <input message="Xin">
            <assign to="." from="*"/>
         </input>
      </operation>
      <onFault>
```

```
          <!-- On Fault, we will clear PrimDoc, construct Rejected response and
soap-envelope it -->
          <sequence>
             <operation name="ReleasePrimDoc">
                <participant name="ReleaseService"/>
                <output message="outmsg">
                   <assign to="TARGET">/ProcessData/PrimaryDocument</assign>
                   <assign to="." from="*"/>
                </output>
                <input message="inmsg"/>
             </operation>
             <operation>
                <participant name="SWIFTNetServerAdapter"/>
                <output message="handleServerResponse">
                   <assign to="." from="*"/>
                   <assign to="interface"
from="SwiftServerRequest/interface/text()"/>
                   <assign to="messageID"
from="SwiftServerRequest/messageID/text()"/>
                   <assign to="Status">Rejected</assign>
                   <assign to="Description">Unable to get the Server
Response</assign>
                   <assign to="Info">Failure in getting the Server Response</assign>
                   <assign to="deliveryNotification"
from="SwiftServerRequest/deliveryNotification/text()"/>
                   <assign to="SnF" from="SwiftServerRequest/SnF/text()"/>
                </output>
                <input message="testing">
                   <assign to="." from="*"/>
                </input>
             </operation>
             <operation name="SoapOut">
                <participant name="SOAPOutbound"/>
                <output message="output">
                   <assign to="." from="*"/>
                   <assign to="SOAP_MODE">respond</assign>
                </output>
                <input message="input">
                   <assign to="." from="*"/>
                </input>
             </operation>
             <assign to="doc-has-headers">true</assign>
             <operation name="HttpResponse">
                <participant name="HttpRespond"/>
                <output message="Xout">
                   <assign to="." from="*"/>
                </output>
                <input message="Xin">
                   <assign to="." from="*"/>
                </input>
             </operation>
          </sequence>
       </onFault>
    </sequence>
</process>
```

Before Using the handleSWIFTNetServerRequest Business Process

Before you use the handleSWIFTNetServerRequest business process, you must complete the following tasks:

1. Configure the SWIFTNet Server adapter to reflect your installation. See *SWIFTNet Server Adapter.*
2. Enable Document Tracking in the BPML Editor for the handleSWIFTNetServerRequest business process.

handleSWIFTNetServerSnFRequest Business Process

The handleSWIFTNetServerSnFRequest business process enables Gentran Integration Suite to receive SWIFTNet store-and-forward messages. This is a bootstrap business process used by the inbound SWIFTNet request that includes a store-and-forward option. For the store-and-forward option, an incoming request is not processed immediately, but instead is stored in the responder mailbox in Gentran Integration Suite to be responded to later. Gentran Integration Suite then sends an acknowledgment that the request has been successfully stored to the requestor through the Gentran Integration Suite SWIFTNet MEFG Server.

Note: This business process is used with InterAct processing only.

The handleSWIFTNetServerSnFRequest business process is initiated as part of the following inbound process flow:

1. The SWIFTNet Server adapter invokes the handleSWIFTNetServerSnFRequest business process and passes it all the necessary parameters to send a request.
2. The handleSWIFTNetServerSnFRequest business process invokes the SOAP Inbound service.
3. The SOAP Inbound service:
   - Pre-processes the incoming request.
   - Searches the SWIFTNet Routing Rule table.
4. The Mailbox Add service Routes the request payload to the responder mailbox to be responded to later.
5. Then the SOAP Outbound service invokes the HTTP Response service.
6. The HTTP Response service sends an acknowledgement to the requestor that the request has been successfully stored.

This table lists the configuration parameters for the handleSWIFTNetServerSnFRequest business process:

Parameter	Default	Description
Document Tracking	False	When document tracking is enabled for a business process, tracking information is carried with the message throughout the process, and the tracking information is persisted about the message regardless of the persistence level you configured globally for Gentran Integration Suite.
Set onfault processing	False	Onfault processing allows the process to immediately execute the on-fault activity specified in the process, even if the process has not yet reached that step in the process. For example, if a process fails at step 3, but the on-fault activity is specified in step 7, if onfault processing is enabled, the process proceeds to the step 7 on-fault rather than halting at step 3.
Start mode	async	Asynchronous initiation is selected by default. Starting business processes asynchronously is recommended. Asynchronous mode is standard Gentran Integration Suite processing, wherein the business process is placed in queue and processed.
Transaction	False	This option instructs Gentran Integration Suite to treat the entire process as a single transaction so that either all of the steps complete, or, in the event of an error, none of them do. When an error occurs, no data is committed; data returns to its pre-process state. By default, this transaction mode is not enabled.
Queue	4	Gentran Integration Suite enables you to set performance optimizations by queue, defining queue levels to allocate resources. This number indicates the previously allocated queue level that you want for this business process model for processing.
Persistence Level	System Default	The level of data to retain for generating a status report that describes each step that the business process completes. System default indicates that, for the data, configuration is already defined in Gentran Integration Suite to retain data.
Recovery Level	Manual	The level of recovery for this business process if the business process should halt during execution. Manual requires you to resume or restart the business process manually.
Document Storage Type	System Default	The level of document storage for messages that process when the business process runs. System Default specifies to store messages in the file system or database, according to how you configured archiving and purging in Gentran Integration Suite.

Parameter	Default	Description
Life Span	Life Span Days — 2 Life Span Hours — 0 Life Span Type — System Level Removal Method — Archive	The length of time, in days and hours, to retain the data in Gentran Integration Suite, along with the life span type and removal method.
Complete by Deadline	None Available **Note:** To set a deadline you must change it in the business process.	Complete by – The deadline time, in hours and minutes, by which the business process must complete process once it starts. ◆ First Notification: Hours and Minutes – Whether to receive notification before a business process deadline. ◆ Second Notification: Hours and Minutes – Whether to receive another notification before a business process deadline.
Event Reporting Level	Full	The level of event reporting that is retrieved for this business process when it runs. Full specifies to generate events for the business process, including the business process start and end time, start and end times for all services or services running as a result of this business processes, and any resulting errors and exceptions.

The following BPML code makes up the handleSWIFTNetServerSnFRequest business process:

```
<process name="handleSWIFTNetServerSnFRequest">
   <rule name="not_DeliveryNotificationRequest">
      <condition>SwiftServerRequest/deliveryNotification = 'FALSE'</condition>
   </rule>
   <sequence>
      <operation name="set user token">
         <participant name="SetUserToken"/>
         <output message="SetUserTokenMessage">
            <assign to="USER_TOKEN">admin</assign>
            <assign to="." from="*"/>
         </output>
         <input message="inmsg">
            <assign to="." from="*"/>
         </input>
      </operation>
      <operation name="SoapIn">
         <participant name="SOAPInbound"/>
         <output message="output">
            <assign to="." from="*"/>
            <assign to="bootstrap">false</assign>
            <assign to="SOAP_INTERMEDIATE_NODE">false</assign>
         </output>
         <input message="input">
            <assign to="." from="*"/>
         </input>
      </operation>
```

```
      <operation>
         <participant name="SWIFTNetServerAdapter"/>
         <output message="handleServerRequest">
            <assign to="." from="*"/>
         </output>
         <input message="testing">
            <assign to="." from="*"/>
         </input>
      </operation>
      <!-- internal processing for SnF is to put into a Mailbox so that it can
bootstrap internal business process later-->
      <!-- Mailbox path is based on SwiftServerRequest/responderDN/requestorDN/for
InterAct -->
      <choice>
         <select>
            <case ref="not_DeliveryNotificationRequest"
activity="not_DeliveryNotificationRequest"/>
         </select>
         <sequence name="not_DeliveryNotificationRequest">
            <operation name="Mailbox Add Service">
               <participant name="MailboxAdd"/>
               <output message="AddRequest">
                  <assign to="." from="*"/>
                  <assign to="MailboxPath" from="concat('/',
SwiftServerRequest/responderDN/text(),'/',SwiftServerRequest/requestorDN/text())"/>
                  <assign to="ContentType">ascii</assign>
               </output>
               <input message="inmsg">
                  <assign to="AddResults" from="*"/>
               </input>
            </operation>
         </sequence>
      </choice>
      <operation>
         <participant name="SWIFTNetServerAdapter"/>
         <output message="handleServerResponse">
            <assign to="." from="*"/>
            <assign to="interface" from="SwiftServerRequest/interface/text()"/>
            <assign to="messageID" from="SwiftServerRequest/messageID/text()"/>
            <assign to="Status">Accepted</assign>
            <assign to="deliveryNotification"
from="SwiftServerRequest/deliveryNotification/text()"/>
            <assign to="SnF" from="SwiftServerRequest/SnF/text()"/>
         </output>
         <input message="testing">
            <assign to="." from="*"/>
         </input>
      </operation>
      <operation name="SoapOut">
         <participant name="SOAPOutbound"/>
         <output message="output">
            <assign to="." from="*"/>
            <assign to="SOAP_MODE">respond</assign>
         </output>
         <input message="input">
            <assign to="." from="*"/>
```

```
            </input>
        </operation>
        <assign to="doc-has-headers">true</assign>
        <operation name="HttpResponse">
            <participant name="HttpRespond"/>
            <output message="Xout">
                <assign to="." from="*"/>
            </output>
            <input message="Xin">
                <assign to="." from="*"/>
            </input>
        </operation>
        <onFault>
            <sequence>
                <operation name="ReleasePrimDoc">
                    <participant name="ReleaseService"/>
                    <output message="outmsg">
                        <assign to="TARGET">/ProcessData/PrimaryDocument</assign>
                        <assign to="." from="*"/>
                    </output>
                    <input message="inmsg"/>
                </operation>
                <operation>
                    <participant name="SWIFTNetServerAdapter"/>
                    <output message="handleServerResponse">
                        <assign to="." from="*"/>
                        <assign to="interface"
from="SwiftServerRequest/interface/text()"/>
                        <assign to="messageID"
from="SwiftServerRequest/messageID/text()"/>
                        <assign to="Status">Rejected</assign>
                        <assign to="Description">Unable to get the Server
Response</assign>
                        <assign to="Info">Failure in getting the Server Response</assign>
                        <assign to="deliveryNotification"
from="SwiftServerRequest/deliveryNotification/text()"/>
                        <assign to="SnF" from="SwiftServerRequest/SnF/text()"/>
                    </output>
                    <input message="testing">
                        <assign to="." from="*"/>
                    </input>
                </operation>
                <operation name="SoapOut">
                    <participant name="SOAPOutbound"/>
                    <output message="output">
                        <assign to="." from="*"/>
                        <assign to="SOAP_MODE">respond</assign>
                    </output>
                    <input message="input">
                        <assign to="." from="*"/>
                    </input>
                </operation>
                <assign to="doc-has-headers">true</assign>
                <operation name="HttpResponse">
                    <participant name="HttpRespond"/>
                    <output message="Xout">
```

```
                    <assign to="." from="*"/>
                </output>
                <input message="Xin">
                    <assign to="." from="*"/>
                </input>
            </operation>
        </sequence>
    </onFault>
</sequence>
</process>
```

Before Using the handleSWIFTNetServerSnFRequest Business Process

Before you use the handleSWIFTNetServerSnFRequest business process, you must complete the following tasks:

1. Configure the SWIFTNet Server adapter to reflect your installation. See *SWIFTNet Server Adapter.*
2. Enable Document Tracking in the BPML Editor for the handleSWIFTNetServerSnFRequest business process.

handleSWIFTNetServerFARequest Business Process

The handleSWIFTNetServerFARequest business process enables Gentran Integration Suite to receive SWIFTNet messages. This is the bootstrap business process used by the inbound SWIFTNet request through the SWIFTNet MEFG Server. It is a system business process used by the SWIFTNet Server adapter, which pre-processes the incoming request, search the SWIFTNet Routing Rule table, and route the request payload to the business process for processing.

Note: This business process is used with InterAct processing only.

The handleSWIFTNetServerFARequest business process is initiated as part of the following inbound process flow:

1. The SWIFTNet Server adapter invokes the handleSWIFTNetServerFARequest business process and passes it all the necessary parameters to send a request.
2. The handleSWIFTNetServerRequest business process invokes the SOAP Inbound service.
3. The SOAP Inbound service:
   - Pre-processes the incoming request.
   - Searches the SWIFTNet Routing Rule table.
   - Routes the request payload to the business process for processing.
4. Then the SOAP Outbound service invokes the HTTP Response service.
5. The HTTP Response service sends a response to the request.

This table lists the configuration parameters for the handleSWIFTNetServerFARequest business process:

Parameter	Default	Description
Document Tracking	False	When document tracking is enabled for a business process, tracking information is carried with the message throughout the process, and the tracking information is persisted about the message regardless of the persistence level you configured globally for Gentran Integration Suite.
Set onfault processing	False	Onfault processing allows the process to immediately execute the on-fault activity specified in the process, even if the process has not yet reached that step in the process. For example, if a process fails at step 3, but the on-fault activity is specified in step 7, if onfault processing is enabled, the process proceeds to the step 7 on-fault rather than halting at step 3.
Start mode	async	Asynchronous initiation is selected by default. Starting business processes asynchronously is recommended. Asynchronous mode is standard Gentran Integration Suite processing, wherein the business process is placed in queue and processed.
Transaction	False	This option instructs Gentran Integration Suite to treat the entire process as a single transaction so that either all of the steps complete, or, in the event of an error, none of them do. When an error occurs, no data is committed; data returns to its pre-process state. By default, this transaction mode is not enabled.
Queue	4	Gentran Integration Suite enables you to set performance optimizations by queue, defining queue levels to allocate resources. This number indicates the previously allocated queue level that you want for this business process model for processing.
Persistence Level	System Default	The level of data to retain for generating a status report that describes each step that the business process completes. System default indicates that, for the data, configuration is already defined in Gentran Integration Suite to retain data.
Recovery Level	Manual	The level of recovery for this business process if the business process should halt during execution. Manual requires you to resume or restart the business process manually.
Document Storage Type	System Default	The level of document storage for messages that process when the business process runs. System Default specifies to store messages in the file system or database, according to how you configured archiving and purging in Gentran Integration Suite.

Parameter	Default	Description
Life Span	Life Span Days — 2 Life Span Hours — 0 Life Span Type — System Level Removal Method — Archive	The length of time, in days and hours, to retain the data in Gentran Integration Suite, along with the life span type and removal method.
Complete by Deadline	None Available **Note:** To set a deadline you must change it in the business process.	Complete by – The deadline time, in hours and minutes, by which the business process must complete process once it starts. ◆ First Notification: Hours and Minutes – Whether to receive notification before a business process deadline. ◆ Second Notification: Hours and Minutes – Whether to receive another notification before a business process deadline.
Event Reporting Level	Full	The level of event reporting that is retrieved for this business process when it runs. Full specifies to generate events for the business process, including the business process start and end time, start and end times for all services or services running as a result of this business processes, and any resulting errors and exceptions.

The following BPML code makes up the handleSWIFTNetServerFARequest business process:

```
<process name="handleSWIFTNetServerFARequest">
   <sequence>
      <operation name="set user token">
         <participant name="SetUserToken"/>
         <output message="SetUserTokenMessage">
            <assign to="USER_TOKEN">admin</assign>
            <assign to="." from="*"/>
         </output>
         <input message="inmsg">
            <assign to="." from="*"/>
         </input>
      </operation>
      <operation name="SoapIn">
         <participant name="SOAPInbound"/>
         <output message="output">
            <assign to="." from="*"/>
            <assign to="bootstrap">false</assign>
            <assign to="SOAP_INTERMEDIATE_NODE">false</assign>
         </output>
         <input message="input">
            <assign to="." from="*"/>
         </input>
      </operation>
      <operation>
         <participant name="SWIFTNetServerAdapter"/>
         <output message="handleServerRequest">
```

```
            <assign to="." from="*"/>
         </output>
         <input message="testing">
            <assign to="." from="*"/>
         </input>
      </operation>
      <!-- this is to construct the server response message back to GIS Server
application -->
      <operation>
         <participant name="SWIFTNetServerAdapter"/>
         <output message="handleServerResponse">
            <assign to="." from="*"/>
            <assign to="interfaceMode"
from="SwiftServerRequest/interfaceMode/text()"/>
            <assign to="messageID" from="SwiftServerRequest/messageID/text()"/>
            <assign to="Status">Accepted</assign>
            <assign to="deliveryNotification"
from="SwiftServerRequest/deliveryNotification/text()"/>
            <assign to="SnF" from="SwiftServerRequest/SnF/text()"/>
         </output>
         <input message="testing">
            <assign to="." from="*"/>
         </input>
      </operation>
      <operation name="SoapOut">
         <participant name="SOAPOutbound"/>
         <output message="output">
            <assign to="." from="*"/>
            <assign to="SOAP_MODE">respond</assign>
         </output>
         <input message="input">
            <assign to="." from="*"/>
         </input>
      </operation>
      <assign to="doc-has-headers">true</assign>
      <operation name="HttpResponse">
         <participant name="HttpRespond"/>
         <output message="Xout">
            <assign to="." from="*"/>
         </output>
         <input message="Xin">
            <assign to="." from="*"/>
         </input>
      </operation>
      <onFault>
         <!-- On Fault, we will clear PrimDoc, contruct Rejected response and
soap-envelope it -->
         <sequence>
            <operation name="ReleasePrimDoc">
               <participant name="ReleaseService"/>
               <output message="outmsg">
                  <assign to="TARGET">/ProcessData/PrimaryDocument</assign>
                  <assign to="." from="*"/>
               </output>
               <input message="inmsg"/>
            </operation>
```

```
            <operation>
               <participant name="SWIFTNetServerAdapter"/>
               <output message="handleServerResponse">
                  <assign to="." from="*"/>
                  <assign to="interfaceMode"
from="SwiftServerRequest/interfaceMode/text()"/>
                  <assign to="messageID"
from="SwiftServerRequest/messageID/text()"/>
                  <assign to="Status">Rejected</assign>
                  <assign to="Description">Unable to get the Server
Response</assign>
                  <assign to="Info">Failure in getting the Server Response</assign>
                  <assign to="deliveryNotification"
from="SwiftServerRequest/deliveryNotification/text()"/>
                  <assign to="SnF" from="SwiftServerRequest/SnF/text()"/>
               </output>
               <input message="testing">
                  <assign to="." from="*"/>
               </input>
            </operation>
            <operation name="SoapOut">
               <participant name="SOAPOutbound"/>
               <output message="output">
                  <assign to="." from="*"/>
                  <assign to="SOAP_MODE">respond</assign>
               </output>
               <input message="input">
                  <assign to="." from="*"/>
               </input>
            </operation>
            <assign to="doc-has-headers">true</assign>
            <operation name="HttpResponse">
               <participant name="HttpRespond"/>
               <output message="Xout">
                  <assign to="." from="*"/>
               </output>
               <input message="Xin">
                  <assign to="." from="*"/>
               </input>
            </operation>
         </sequence>
      </onFault>
   </sequence>
</process>
```

Before Using the handleSWIFTNetServerFARequest Business Process

Before you use the handleSWIFTNetServerFARequest business process, you must complete the following tasks:

1. Configure the SWIFTNet Server adapter to reflect your installation. See *SWIFTNet Server Adapter.*
2. Enable Document Tracking in the BPML Editor for the handleSWIFTNetServerFARequest business process.

handleSWIFTNetServerFASnFRequest Business Process

The handleSWIFTNetServerFASnFRequest business process enables Gentran Integration Suite to receive SWIFTNet store-and-forward messages. This is a bootstrap business process used by the inbound SWIFTNet request that includes a store-and-forward option. For the store-and-forward option, an incoming request is not processed immediately, but instead is stored in the responder mailbox in Gentran Integration Suite to be responded to later. Gentran Integration Suite then sends an acknowledgment that the request has been successfully stored to the requestor through the Gentran Integration Suite SWIFTNet MEFG Server.

Note: This business process is used for FileAct processing only.

The handleSWIFTNetServerFASnFRequest business process is initiated as part of the following inbound process flow:

1. The SWIFTNet Server adapter invokes the handleSWIFTNetServerFASnFRequest business process and passes it all the necessary parameters to send a request.
2. The handleSWIFTNetServerFASnFRequest business process invokes the SOAP Inbound service.
3. The SOAP Inbound service:
   - Pre-processes the incoming request.
   - Searches the SWIFTNet Routing Rule table.
4. The Mailbox Add service Routes the request payload to the responder mailbox to be responded to later.
5. Then the SOAP Outbound service invokes the HTTP Response service.
6. The HTTP Response service sends an acknowledgement to the requestor that the request has been successfully stored.

This table lists the configuration parameters for the handleSWIFTNetServerFASnFRequest business process:

Parameter	Default	Description
Document Tracking	False	When document tracking is enabled for a business process, tracking information is carried with the message throughout the process, and the tracking information is persisted about the message regardless of the persistence level you configured globally for Gentran Integration Suite.
Set onfault processing	False	Onfault processing allows the process to immediately execute the on-fault activity specified in the process, even if the process has not yet reached that step in the process. For example, if a process fails at step 3, but the on-fault activity is specified in step 7, if onfault processing is enabled, the process proceeds to the step 7 on-fault rather than halting at step 3.
Start mode	async	Asynchronous initiation is selected by default. Starting business processes asynchronously is recommended. Asynchronous mode is standard Gentran Integration Suite processing, wherein the business process is placed in queue and processed.
Transaction	False	This option instructs Gentran Integration Suite to treat the entire process as a single transaction so that either all of the steps complete, or, in the event of an error, none of them do. When an error occurs, no data is committed; data returns to its pre-process state. By default, this transaction mode is not enabled.
Queue	4	Gentran Integration Suite enables you to set performance optimizations by queue, defining queue levels to allocate resources. This number indicates the previously allocated queue level that you want for this business process model for processing.
Persistence Level	System Default	The level of data to retain for generating a status report that describes each step that the business process completes. System default indicates that, for the data, configuration is already defined in Gentran Integration Suite to retain data.
Recovery Level	Manual	The level of recovery for this business process if the business process should halt during execution. Manual requires you to resume or restart the business process manually.
Document Storage Type	System Default	The level of document storage for messages that process when the business process runs. System Default specifies to store messages in the file system or database, according to how you configured archiving and purging in Gentran Integration Suite.

Parameter	Default	Description
Life Span	Life Span Days — 2 Life Span Hours — 0 Life Span Type — System Level Removal Method — Archive	The length of time, in days and hours, to retain the data in Gentran Integration Suite, along with the life span type and removal method.
Complete by Deadline	None Available **Note:** To set a deadline you must change it in the business process.	Complete by – The deadline time, in hours and minutes, by which the business process must complete process once it starts. ◆ First Notification: Hours and Minutes – Whether to receive notification before a business process deadline. ◆ Second Notification: Hours and Minutes – Whether to receive another notification before a business process deadline.
Event Reporting Level	Full	The level of event reporting that is retrieved for this business process when it runs. Full specifies to generate events for the business process, including the business process start and end time, start and end times for all services or services running as a result of this business processes, and any resulting errors and exceptions.

The following BPML code makes up the handleSWIFTNetServerFASnFRequest business process:

```
<process name="handleSWIFTNetServerFASnFRequest">
   <rule name="not_DeliveryNotificationRequest">
      <condition>SwiftServerRequest/deliveryNotification = 'FALSE'</condition>
   </rule>
   <sequence>
      <operation name="set user token">
         <participant name="SetUserToken"/>
         <output message="SetUserTokenMessage">
            <assign to="USER_TOKEN">admin</assign>
            <assign to="." from="*"/>
         </output>
         <input message="inmsg">
            <assign to="." from="*"/>
         </input>
      </operation>
      <operation name="SoapIn">
         <participant name="SOAPInbound"/>
         <output message="output">
            <assign to="." from="*"/>
            <assign to="bootstrap">false</assign>
            <assign to="SOAP_INTERMEDIATE_NODE">false</assign>
         </output>
         <input message="input">
            <assign to="." from="*"/>
         </input>
      </operation>
```

```
       <operation>
           <participant name="SWIFTNetServerAdapter"/>
           <output message="handleServerRequest">
              <assign to="." from="*"/>
           </output>
           <input message="testing">
              <assign to="." from="*"/>
           </input>
       </operation>
       <operation>
           <participant name="SWIFTNetServerAdapter"/>
           <output message="handleServerResponse">
              <assign to="." from="*"/>
              <assign to="interfaceMode"
from="SwiftServerRequest/interfaceMode/text()"/>
              <assign to="messageID" from="SwiftServerRequest/messageID/text()"/>
              <assign to="Status">Accepted</assign>
              <assign to="deliveryNotification"
from="SwiftServerRequest/deliveryNotification/text()"/>
              <assign to="SnF" from="SwiftServerRequest/SnF/text()"/>
           </output>
           <input message="testing">
              <assign to="." from="*"/>
           </input>
       </operation>
       <operation name="SoapOut">
           <participant name="SOAPOutbound"/>
           <output message="output">
              <assign to="." from="*"/>
              <assign to="SOAP_MODE">respond</assign>
           </output>
           <input message="input">
              <assign to="." from="*"/>
           </input>
       </operation>
       <assign to="doc-has-headers">true</assign>
       <operation name="HttpResponse">
           <participant name="HttpRespond"/>
           <output message="Xout">
              <assign to="." from="*"/>
           </output>
           <input message="Xin">
              <assign to="." from="*"/>
           </input>
       </operation>
       <onFault>
           <sequence>
              <operation name="ReleasePrimDoc">
                  <participant name="ReleaseService"/>
                  <output message="outmsg">
                     <assign to="TARGET">/ProcessData/PrimaryDocument</assign>
                     <assign to="." from="*"/>
                  </output>
                  <input message="inmsg"/>
              </operation>
              <operation>
```

```
                <participant name="SWIFTNetServerAdapter"/>
                <output message="handleServerResponse">
                   <assign to="." from="*"/>
                   <assign to="interfaceMode"
from="SwiftServerRequest/interfaceMode/text()"/>
                   <assign to="messageID"
from="SwiftServerRequest/messageID/text()"/>
                   <assign to="Status">Rejected</assign>
                   <assign to="Description">Unable to get the Server
Response</assign>
                   <assign to="Info">Failure in getting the Server Response</assign>
                   <assign to="deliveryNotification"
from="SwiftServerRequest/deliveryNotification/text()"/>
                   <assign to="SnF" from="SwiftServerRequest/SnF/text()"/>
                </output>
                <input message="testing">
                   <assign to="." from="*"/>
                </input>
             </operation>
             <operation name="SoapOut">
                <participant name="SOAPOutbound"/>
                <output message="output">
                   <assign to="." from="*"/>
                   <assign to="SOAP_MODE">respond</assign>
                </output>
                <input message="input">
                   <assign to="." from="*"/>
                </input>
             </operation>
             <assign to="doc-has-headers">true</assign>
             <operation name="HttpResponse">
                <participant name="HttpRespond"/>
                <output message="Xout">
                   <assign to="." from="*"/>
                </output>
                <input message="Xin">
                   <assign to="." from="*"/>
                </input>
             </operation>
          </sequence>
       </onFault>
   </sequence>
</process>
```

Before Using the handleSWIFTNetServerFASnFRequest Business Process

Before you use the handleSWIFTNetServerFASnFRequest business process, you must complete the following tasks:

1. Configure the SWIFTNet Server adapter to reflect your installation. See *SWIFTNet Server Adapter.*
2. Enable Document Tracking in the BPML Editor for the handleSWIFTNetServerFASnFRequest business process.

handleSWIFTNetServerFAEvent Business Process

The handleSWIFTNetServerFAEvent business process is used by the Gentran Integration Suite SWIFTNet MEFG Server. It is a system business process called by the SWIFTNet Server adapter that preprocesses the incoming FileAct Event with a COMPLETED status, searches the SWIFTNet routing rule table and bootstraps the business process for processing.

Note: This business process is used for FileAct processing only.

The handleSWIFTNetServerFAEvent business process is initiated as part of the following inbound process flow:

1. The SWIFTNet Server adapter invokes the handleSWIFTNetServerFAEvent business process and passes it all the necessary parameters to send a request.
2. The handleSWIFTNetServerFAEvent business process preprocesses the incoming FileAct event.
3. The handleSWIFTNetServerFAEvent business process searches the SWIFTNet Routing Rule table.
4. Then the business process bootstraps the appropriate business process for processing.

This table lists the configuration parameters for the handleSWIFTNetServerFAEvent business process:

Parameter	Default	Description
Document Tracking	False	When document tracking is enabled for a business process, tracking information is carried with the message throughout the process, and the tracking information is persisted about the message regardless of the persistence level you configured globally for Gentran Integration Suite.
Set onfault processing	False	Onfault processing allows the process to immediately execute the on-fault activity specified in the process, even if the process has not yet reached that step in the process. For example, if a process fails at step 3, but the on-fault activity is specified in step 7, if onfault processing is enabled, the process proceeds to the step 7 on-fault rather than halting at step 3.
Start mode	async	Asynchronous initiation is selected by default. Starting business processes asynchronously is recommended. Asynchronous mode is standard Gentran Integration Suite processing, wherein the business process is placed in queue and processed.
Transaction	False	This option instructs Gentran Integration Suite to treat the entire process as a single transaction so that either all of the steps complete, or, in the event of an error, none of them do. When an error occurs, no data is committed; data returns to its pre-process state. By default, this transaction mode is not enabled.
Queue	4	Gentran Integration Suite enables you to set performance optimizations by queue, defining queue levels to allocate resources. This number indicates the previously allocated queue level that you want for this business process model for processing.
Persistence Level	System Default	The level of data to retain for generating a status report that describes each step that the business process completes. System default indicates that, for the data, configuration is already defined in Gentran Integration Suite to retain data.
Recovery Level	Manual	The level of recovery for this business process if the business process should halt during execution. Manual requires you to resume or restart the business process manually.
Document Storage Type	System Default	The level of document storage for messages that process when the business process runs. System Default specifies to store messages in the file system or database, according to how you configured archiving and purging in Gentran Integration Suite.

Parameter	Default	Description
Life Span	Life Span Days — 2 Life Span Hours — 0 Life Span Type — System Level Removal Method — Archive	The length of time, in days and hours, to retain the data in Gentran Integration Suite, along with the life span type and removal method.
Complete by Deadline	None Available **Note:** To set a deadline you must change it in the business process.	Complete by – The deadline time, in hours and minutes, by which the business process must complete process once it starts. ◆ First Notification: Hours and Minutes – Whether to receive notification before a business process deadline. ◆ Second Notification: Hours and Minutes – Whether to receive another notification before a business process deadline.
Event Reporting Level	Full	The level of event reporting that is retrieved for this business process when it runs. Full specifies to generate events for the business process, including the business process start and end time, start and end times for all services or services running as a result of this business processes, and any resulting errors and exceptions.

The following BPML code makes up the handleSWIFTNetServerFAEvent business process:

```
<process name="handleSWIFTNetServerFAEvent">
   <rule name="Backend_Workflow">
      <condition>DoFABackendProcess= 'workflow'</condition>
   </rule>
   <rule name="Backend_MailBox">
      <condition>DoFABackendProcess= 'mailbox'</condition>
   </rule>
   <sequence>
      <operation name="set user token">
         <participant name="SetUserToken"/>
         <output message="SetUserTokenMessage">
            <assign to="USER_TOKEN">admin</assign>
            <assign to="." from="*"/>
         </output>
         <input message="inmsg">
            <assign to="." from="*"/>
         </input>
      </operation>
      <operation name="SoapIn">
         <participant name="SOAPInbound"/>
         <output message="output">
            <assign to="." from="*"/>
            <assign to="bootstrap">false</assign>
            <assign to="SOAP_INTERMEDIATE_NODE">false</assign>
         </output>
         <input message="input">
```

```
            <assign to="." from="*"/>
         </input>
      </operation>
      <operation>
         <participant name="SWIFTNetServerAdapter"/>
         <output message="handleServerRequest">
            <assign to="." from="*"/>
         </output>
         <input message="testing">
            <assign to="." from="*"/>
         </input>
      </operation>
      <!-- Process Completed FA Event with incoming, put or get request-->
      <!-- non SnF - bootstrap Workflow -->
      <!-- SnF - put in MailBox -->
      <choice>
         <select>
            <case ref="Backend_Workflow" activity="processBackend_Workflow"/>
            <case ref="Backend_MailBox" activity="processBackend_MailBox"/>
         </select>
         <sequence name="processBackend_Workflow">
            <operation>
               <participant name="InvokeSubProcessService"/>
               <output message="Xout">
                  <assign to="INVOKE_MODE">SYNC</assign>
                  <assign to="." from="*"/>
               </output>
               <input message="Xin">
                  <assign to="." from="*"/>
               </input>
            </operation>
         </sequence>
         <sequence name="processBackend_MailBox">
            <operation name="Mailbox Add Service">
               <participant name="MailboxAdd"/>
               <output message="AddRequest">
                  <assign to="." from="*"/>
                  <assign to="MailboxPath" from="concat('/',
SwiftServerRequest/responderDN/text(),'/',SwiftServerRequest/requestorDN/text())"/>
                  <assign to="ContentType">ascii</assign>
               </output>
               <input message="inmsg">
                  <assign to="AddResults" from="*"/>
               </input>
            </operation>
         </sequence>
      </choice>
      <!-- this is to construct the server response message back to GIS Server
application -->
      <operation>
         <participant name="SWIFTNetServerAdapter"/>
         <output message="handleServerResponse">
            <assign to="." from="*"/>
            <assign to="interfaceMode"
from="SwiftServerRequest/interfaceMode/text()"/>
            <assign to="messageID" from="SwiftServerRequest/messageID/text()"/>
```

```
            <assign to="Status">Accepted</assign>
            <assign to="deliveryNotification"
from="SwiftServerRequest/deliveryNotification/text()"/>
            <assign to="SnF" from="SwiftServerRequest/SnF/text()"/>
         </output>
         <input message="testing">
            <assign to="." from="*"/>
         </input>
      </operation>
      <operation name="SoapOut">
         <participant name="SOAPOutbound"/>
         <output message="output">
            <assign to="." from="*"/>
            <assign to="SOAP_MODE">respond</assign>
         </output>
         <input message="input">
            <assign to="." from="*"/>
         </input>
      </operation>
      <assign to="doc-has-headers">true</assign>
      <operation name="HttpResponse">
         <participant name="HttpRespond"/>
         <output message="Xout">
            <assign to="." from="*"/>
         </output>
         <input message="Xin">
            <assign to="." from="*"/>
         </input>
      </operation>
      <onFault>
         <!-- On Fault, we will clear PrimDoc, contruct Rejected response and
soap-envelope it -->
         <sequence>
            <operation name="ReleasePrimDoc">
               <participant name="ReleaseService"/>
               <output message="outmsg">
                  <assign to="TARGET">/ProcessData/PrimaryDocument</assign>
                  <assign to="." from="*"/>
               </output>
               <input message="inmsg"/>
            </operation>
            <operation>
               <participant name="SWIFTNetServerAdapter"/>
               <output message="handleServerResponse">
                  <assign to="." from="*"/>
                  <assign to="interfaceMode"
from="SwiftServerRequest/interfaceMode/text()"/>
                  <assign to="messageID"
from="SwiftServerRequest/messageID/text()"/>
                  <assign to="Status">Rejected</assign>
                  <assign to="Description">Fail in Backend Process and unable to
get the Server Response</assign>
                  <assign to="Info">Failure in Backend Process and getting the
Server Response</assign>
                  <assign to="deliveryNotification"
from="SwiftServerRequest/deliveryNotification/text()"/>
```

```
                  <assign to="SnF" from="SwiftServerRequest/SnF/text()"/>
               </output>
               <input message="testing">
                  <assign to="." from="*"/>
               </input>
            </operation>
            <operation name="SoapOut">
               <participant name="SOAPOutbound"/>
               <output message="output">
                  <assign to="." from="*"/>
                  <assign to="SOAP_MODE">respond</assign>
               </output>
               <input message="input">
                  <assign to="." from="*"/>
               </input>
            </operation>
            <assign to="doc-has-headers">true</assign>
            <operation name="HttpResponse">
               <participant name="HttpRespond"/>
               <output message="Xout">
                  <assign to="." from="*"/>
               </output>
               <input message="Xin">
                  <assign to="." from="*"/>
               </input>
            </operation>
         </sequence>
      </onFault>
   </sequence>
</process>
```

Before Using the handleSWIFTNetServerFAEvent Business Process

Before you use the handleSWIFTNetServerFAEvent business process, you must complete the following tasks:

1. Configure the SWIFTNet Server adapter to reflect your installation. See *SWIFTNet Server Adapter.*
2. Enable Document Tracking in the BPML Editor for the handleSWIFTNetServerFAEvent business process.

Creating SWIFTNet Maps

- *Overview* on page 102
- *Creating a File Layout from an MT or Market Practice* on page 106
- *Creating a Custom Market Practice or Fund Template* on page 107
- *Creating a SWIFTSolutions Map using SWIFTNet FUNDS, SWIFTNet Trade Services, SWIFTNet Exceptions and Investigations, and SWIFTNet Cash Reporting* on page 109
- *SWIFTNet Syntax Validation* on page 111
- *Creating Extended Rules for SWIFTNet Maps* on page 111
- *Extended Rules Used with SWIFTNet Maps* on page 112
- *Using Autolink* on page 112
- *Using Extended Rule Libraries with SWIFTNet* on page 112

Overview

The Map Editor enables you to map SWIFTNet documents.The Map Editor generates a file layout for you using the components and message types that you select. The Map Editor-generated SWIFT map consists of groups, records, composites, and fields that are comparable to parameters that are defined by SWIFT. See *How SWIFTNet Terminology Correlates with the Map Editor* on page 103.

Note: You need to download the Map Editor component from Gentran Integration Suite to use it.

Map Editor allows you to modify the map components by using the Deactivate, Promote, Split, Copy, Cut, and Paste functions.

You can create a map for all SWIFTNet Standards Release messages loaded into the standards database through the Map Editor. All messages are validated by Gentran Integration Suite for syntax (for example, field types, field lengths, and so forth). See *Overview* on page 6 for information on which messages are validated by Gentran Integration Suite for syntax *and* semantics. If you create a map for a message that is not currently validated by Gentran Integration Suite, when the Map Editor is capable of updating your existing map with message validation rules in a future release, any extended rules you have created will exist outside the validation block (and thus will always run regardless of whether validation is enabled).

When you create a map, for both the input and output sides, you choose whether you want to use a preloaded standard, an existing file format, or wants to create a new file format for that side of the map (including selecting from standards that you have previously downloaded to the standards database. The preloaded standards are automatically loaded when you download the Map Editor to your machine and include SWIFTNet FUNDS, SWIFTNet Trade Services, SWIFTNet Exceptions and Investigations, and SWIFTNet Cash Reporting.

However, if you want to use a specialized version of a message that is not available in the Gentran Integration Suite SWIFTNet data dictionary, it may be appropriate for you to define the SWIFTNet map yourself. Alternatively, you could use an MT that is similar to what is required and customize it yourself.

Gentran Integration Suite supplies you with the following extended rules to be used with SWIFTNet maps and any other data format:

- cerror
- occurrencetotal
- resetoccurrencetotal
- sum
- sumtotal

See *Alphabetic Language Reference* for all the available extended rules.

For this release, to ensure that extended rules for SWIFTNet maps are only run when validation is enabled, place your custom extended rule code in a validation block. See *Creating Extended Rules for SWIFTNet Maps* on page 111.

How SWIFTNet Terminology Correlates with the Map Editor

The terminology used by SWIFTNet differs from that used in the Map Editor. This table lists the SWIFTNet terms and how they correspond to map components in the Map Editor:

SWIFTNet Terminology	Corresponding Map Editor Terminology
MT (File)	SWIFT file format (the top map component on the input and output sides of a map)
Sequence or an implicit group of repeating field tags	Group
Field tag	SWIFT Record
Subfield that is an “OR” option or a group of related subfields that occur in a sequence, and they are also groups of related subfields that occur in a sequence (for example, each SWIFTBIC address is defined as a “composite” consisting of a number of subfields such as branch code, location, and so forth)	Composite
Subfield	Field
Component or a group of SWIFT components that define a SWIFT subfield	Field

SWIFTNet Components in the Map Editor

The following table lists the components that make up the SWIFTNet layout in the Map Editor, the icons that represent the components, and descriptions of the components. For information about adding a map component to a layout, see *Map Editor Basics*.

Component	Icon	Description
SWIFT root element		The *SWIFT root element* represents the MT that Gentran Integration Suite is mapping. At the SWIFT file root element, you define the message type and encoding. It is a group and can contain groups and SWIFT records.
Group		A *group* is a looping structure that contains a sequence or an implicit group of repeating field tags (in Map Editor a group is related records and groups that repeat in sequence until either the group data ends, or the maximum number of times that the loop is permitted to repeat is exhausted). A group that is subordinate to another group is a subgroup (and corresponds to a nested looping structure, a loop within a loop). When a group contains an extended rule or a standard rule, an asterisk appears to the right of the group icon.
SWIFT Record		A *SWIFT record* contains a field tag (in Map Editor a SWIFT record is a group of related fields or composite data elements that combine to communicate useful data). A SWIFT record can occur once or can repeat multiple times. **Note:** If a SWIFT record occurs more than once in a map, it is identified by its name <ID>. The second and subsequent occurrences are identified by <ID>:*n*, where *n* is the number of the occurrence in the map.
Composite		A *composite* is a subfield that is an “OR” option or a group of related subfields that occur in a sequence (in Map Editor a composite is a data element that contains two or more component data elements or subelements). They are also groups of related subfields that occur in a sequence. For example, each SWIFTBIC address is defined as a “composite” consisting of a number of subfields (e.g. branch code, location). A composite can occur once or repeat multiple times. For example, each SWIFTBIC address is defined in the Map Editor as a composite that consists of a number of fields (SWIFT subfields) such as branch code, location, and so forth. **Note:** If a composite occurs more than once in a map, it is identified by its name <ID>. The second and subsequent occurrences are identified by <ID>:*n*, where *n* is the number of the occurrence in the map. A *repeating composite* is a related group of fields that have the ability to loop as a whole (occur more than once) within a particular SWIFT record. To enable a composite to repeat multiple times within a SWIFT record, each occurrence of the composite must be separated by a start and end delimiter. **Note:** SWIFTNet does not use the repeating option, although it is available to you.

Component	Icon	Description
Field		A *field* is a subfield or a group of SWIFT components (the smallest piece of information defined by the SWIFTNet standard) that define a SWIFT subfield. A field can have different meanings depending on the context. In other data formats in the Map Editor, a field is not considered to have useful meaning except in the larger context of the record that contains it. However, fields used in the Map Editor to represent SWIFTNet subfields and components contain useful and discrete information. **Note:** If a field occurs more than once in a map it is identified by its name <ID>. The second and subsequent occurrences are identified by <ID>:*n*, where *n* is the number of the occurrence in the map. A *repeating field* is a field with the ability to loop (occur more than once) within a particular SWIFT record. To enable a single field to repeat multiple times within a SWIFT record, each occurrence of the field must be separated by a start and end delimiter. The use of start and end delimiters help the translator determine where subfields and components are defined within a field tag. When a field has a link performed against it, a red check mark appears over the field icon. When a field contains an extended rule or a standard rule, an asterisk appears to the right of the field icon.

Downloading and Installing the SWIFT Standards Database

Before you install the SWIFT standards database on your desktop, consider these guidelines:

- ✦ Download the Map Editor.
- ✦ For the most current version of the SWIFT standard, contact Sterling Commerce Customer Support.
- ✦ Be sure your desktop meets the Windows Client requirements listed in the Gentran Integration Suite *System Requirements*.

To download and install the standards database:

1. From the Gentran Integration Suite **Deployment** menu, select **Standards**.
2. In the Download and Install section next to Download SWIFT Standards, click **Go!**
3. In the **File Download** dialog box, select a download option, then click **OK**.
   - ◆ If you choose to run the file click **Run** and the operating system downloads the files immediately.
   - ◆ If you choose to save the file, the operating system prompts you to save the file. Browse to the location where you want to download the file and click **OK**. If you want to continue installing, run the file you just saved from the location you specified.
4. In the Security Warning page, select **Always trust content from Sterling Commerce (Mid America), Inc.** if you do not want to see similar security messages in the future when you download software from Sterling Commerce. Click **Yes**.
5. In the Welcome window, click **Next**.
6. In the Choose Destination Location window, select where you want to install the standards database:
   - ◆ If you accept the default location, click **Next**.

- If you want to specify a different location, click **Browse**, specify the path to the folder, click **OK**, and click **Next**.

If you specify a folder name that does not exist, Gentran Integration Suite displays a message asking if you want to create that folder.

7. In the Select Components window, verify that **SWIFTStandardDatabase** is selected and click **Next**.

   The download wizard installs the standards database (SWIFT_2006.mdb).

8. In the Setup Complete window, click **Finish**.

Creating a File Layout from an MT or Market Practice

When you create a new map, you typically use the map wizard that creates a layout for you based on an MT or Market Practice from the standards database. The wizard saves you the time and effort to create the SWIFT side of the map yourself, and minimizes the risk of having an invalid standard format for a message.

When you create a map, for both the input and output sides, you choose whether you want to use a preloaded standard, an existing file format, or wants to create a new file format for that side of the map (including selecting from standards that you have previously downloaded to the standards database.

Note: You can create a map for all SWIFTNet messages through the Map Editor, and all messages are validated by Gentran Integration Suite for syntax (that is, field types, field lengths, and so forth).

To create a file layout from an MT or Market Practice:

1. From the Map Editor **File** menu, select **New**.
2. In the **New Map Wizard**, complete the questions in the first window and click **Next**.

Note: Be sure that **Sterling Integrator** is selected in the **What type of map are you creating** list.

3. If you are translating from SWIFTNet, in the Input Format window, Create a new data format using this standards, select **SWIFT (Society for Worldwide Interbank Financial Telecommunications)** and click **Messages**.

Note: The **Create a new data format using this** selection allows you to use a preloaded standard (that is, you do not have to download the standard to the Gentran Integration Suite database) or use the standards that were downloaded through the standards database.

4. Click **Next**.
5. Select the type of message to auto-generate:
   - Standard SWIFT Message Type
   - Market Practice or Fund Template
   - Custom Market Practice or Fund Template

Note: If you select Custom Market Practice or Fund Template, you must have already created a custom Market Practice or Fund Template to have it listed as an available option.

6. If you are translating an MT, select the **GIS SWIFT Standard** ODBC data source (which contains the SWIFTNet standards database) and click **Next**.

7. If you are translating a Market Practice, select the **GIS SWIFTMP Customer** ODBC data source (which contains the SWIFTNet preloaded market practices) and click **Next**.
8. If you are translating from an MT, select the MT that you want to use and select the version, and click **Next**.
9. If you are translating from a preloaded Market Practice or fund template, select the MT you want to modify and click **Next**.
10. If you are translating from SWIFTNet, click **Finish** to load the MT you selected.
11. Click **Next**.
12. If you are translating to SWIFTNet, in the Output Format window, select **SWIFT (Society for Worldwide Interbank Financial Telecommunications)** and click **Customize**. If you are translating from another format, select that format and continue to the next screen.
13. Click **Next**.
14. If you are translating to SWIFTNet, select the **GIS SWIFT Standard** ODBC data source (which contains the SWIFTNet standards database) and click **Next**.
15. If you are translating to SWIFTNet, select the MT that you want to use and select the version, and click **Next**.
16. If you are translating to SWIFTNet, click **Finish** to load the MT you selected.
17. Click **Next**.
18. If you are translating to SWIFTNet, click **Finish**. The Map Editor displays the new map in the Map Editor window.
19. Use Autolink to link your fields prior to using the Link function or creating standard or extended rules.
20. Use the Rule Library Manager to add a reusable extended rule library that can then be added to any map so you do not have to redefine extended rules for each map you create.

Note: The SWIFT extended rule libraries that are installed with Gentran Integration Suite, contains all the extended rules necessary to carry out the business logic for SWIFT messages.

Creating a Custom Market Practice or Fund Template

You have the capability to create a custom market practice or fund template by using the Map Editor to create a message template of the same message type you want, modifying it accordingly, and using the SWIFT Rule Importer to import the map into Gentran Integration Suite.

To create a custom Market Practice or Fund Template:

1. From the Map Editor **File** menu, select **New**.
2. In the **New Map Wizard**, complete the questions in the first window and click **Next**.

Note: Be sure that **Sterling Integrator** is selected in the **What type of map are you creating** list.

3. In the Input Format window, Create a new data format using this standards, select **SWIFT (Society for Worldwide Interbank Financial Telecommunications)** and click **Messages**.
4. Click **Next**.
5. Select **Standard SWIFT Message Type** as the type of message to auto-generate.

6. Select the **GIS SWIFT Standard** ODBC data source (which contains the SWIFTNet standards database) and click **Next**.
7. Select the MT that you want to use and select the version, and click **Next**.
8. If you are translating from SWIFTNet and have selected a common group MT, select the MT for copy fields and click **Next**.
9. If you are translating from SWIFTNet, click **Finish** to load the MT you selected.
10. Click **Next**.
11. In the Output Format window, select **SWIFT (Society for Worldwide Interbank Financial Telecommunications)** and click **Customize**. If you are translating from another format, select that format and continue to the next screen.
12. Click **Next**.
13. Select the **GIS SWIFT Standard** ODBC data source (which contains the SWIFTNet standards database) and click **Next**.
14. Select the *same MT that you selected for the Input Format* and select the version, and click **Next**.

Note: By selecting the same MT on both the input and output sides of the map, you are creating a "passthrough" map.

15. If you are translating to SWIFTNet and have selected a common group MT, select the MT for copy fields and click **Next**.
16. Click **Finish** to load the MT you selected.
17. Click **Next**.
18. Click **Finish**. The Map Editor displays the new map in the Map Editor window.
19. On the Output side of the map, right-click the SWIFT Properties icon and select **Properties**.
20. Select the **Message Type** tab.
21. In **Market Practice ID**, type the unique identifier for the market practice. This identifier distinguishes the map from the standard SWIFT message type.

Note: This identifier must be 6 upper-case alphanumerics with no spaces. You create the unique Market Practice ID.

22. Use the **Save As** function to save the modified map under a new name.
23. Open a command window and change to the directory where you installed the SWIFT Standards Database.
24. Then, from the Map Editor **File** menu, select **New** so you can use the customer Market Practice or Fund Template to create a new map.
25. In the **New Map Wizard**, complete the questions in the first window and click **Next**.

Note: Be sure that **Sterling Integrator** is selected in the **What type of map are you creating** list.

26. In the Input Format window, Create a new data format using this standards, select **SWIFT (Society for Worldwide Interbank Financial Telecommunications)** and click **Messages**.
27. Click **Next**.
28. Select **Custom Market Practice or Fund Template** as the type of message to auto-generate.

29. Select the **GIS SWIFT Standard** ODBC data source (which contains the SWIFTNet standards database) and click **Next**.
30. Select the custom Market Practice or Fund Template and click **Next**.
31. Follow the prompts to complete the map selection.
32. Click **Finish**. The Map Editor displays the new map in the Map Editor window.
33. Use Autolink to link your fields prior to using the Link function or creating standard or extended rules.
34. Modify the map to implement the additional restrictions related to the desired Market Practice or Fund Template.
35. Use the Rule Library Manager to add a reusable extended rule library that can then be added to any map so you do not have to redefine extended rules for each map you create.

Note: The SWIFT extended rule libraries that are installed with Gentran Integration Suite contain all the extended rules necessary to carry out the business logic for SWIFT messages.

36. For the SWIFT Rule Importer command script, set the Java_home variable.
37. Run the SWIFT Rule Importer from the command prompt using the following syntax:

```
SWIFTRuleImporter MapName.mxl
```

38. Save your map using the following naming convention:

```
SWIFT_<version>_<messageType>[_<messageExtension>]_<marketPracticeID>.mxl
```

Note: In this syntax, version is synonymous with year. For Market Practices, you do not need a messageExtension since all market practice maps are in the 5xx series. The map must be saved in MXL format.

39. Compile the map.
40. If necessary, update your property files with code word and qualifier validations that are specific to the Market Practice.

Note: The properties file to update is **translator_swift_mp_customer_year.properties**, located in the **<install_dir>/properties** directory.

Creating a SWIFTSolutions Map using SWIFTNet FUNDS, SWIFTNet Trade Services, SWIFTNet Exceptions and Investigations, and SWIFTNet Cash Reporting

When you create a SWIFTSolutions map, for both the input and output sides, you choose whether you want to use a preloaded standard, an existing file format, or wants to create a new file format for that side of the map (including selecting from standards that you have previously downloaded to the standards database. The preloaded standards are downloaded when you download the Map Editor to your machine and include SWIFTNet FUNDS, SWIFTnet Trade Services, SWIFTNet Exceptions and Investigations, and SWIFTNet Cash Reporting.

For a complete list of the preloaded standards consult the Map Editor New Map Wizard.

Note: If the map you are creating contains greater than 20,000 objects, you will receive a message noting that this map contains a very large number of objects. For best performance, it is recommended that

you consider whether any unnecessary objects in the map can be removed, do not expand the entire object tree—expand only the section of the tree you are currently mapping, consider using the “Show links to or from the currently selected element” option instead of the “Show links to or from all visible elements” option, and save the map using the.MAP file format (using the Save As function).

To create a map using a preloaded SWIFTNet Fund template:

1. From the Map Editor **File** menu, select **New**.
2. In the **New Map Wizard**, complete the questions on the first screen.

Note: To use XML schemas, the map type must be Sterling Integrator.

3. If you are translating from a SWIFTNet Fund template, from the **Create a new format using this standard** list, select **SWIFTNet Funds** (or SWIFTnet Trade Services, SWIFTNet Exceptions and Investigations, or SWIFTNet Cash Reporting) and click **Messages**. If you are translating from another format, select that format and continue to the next screen.
4. Complete the Map Wizard by selecting the version of the standard, the desired message, the maximum length of the data elements, and whether to build codelists for enumerated attributes, and click **Next**.

Note: The default for the **Build code lists for enumerations** check box is cleared because using code lists may be redundant and thus your map is smaller and more efficient without them.

5. Click **Next**.
6. If you chose to import an XML schema, do not modify any of the subelements selected in the display. *This list is included for your information only.* Click **Next**.

   If you are using an XML schema and the XML parser detects any errors, the messages are displayed in an error window.

7. Click **Finish**.
8. If you are translating to a SWIFTNet Fund template, on the Output screen (from the **Create a new format using this standard** list), select select **SWIFTNet Funds** (or SWIFTnet Trade Services, SWIFTNet Exceptions and Investigations, or SWIFTNet Cash Reporting) and click **Messages**. If you are translating from another format, select that format and continue to the next screen.
9. Complete the Map Wizard by selecting the version of the standard, the desired message, the maximum length of the data elements, and whether to build codelists for enumerated attributes, and click **Next**.

Note: The default for the **Build code lists for enumerations** check box is cleared because using code lists may be redundant and thus your map is smaller and more efficient without them.

10. Click **Next**.
11. If you chose to import an XML schema, do not modify any of the subelements selected in the display. *This list is included for your information only.* Click **Next**.

    If you are using an XML schema and the XML parser detects any errors, the messages are displayed in an error window.

12. Click **Finish**.

    The **XML Load Warnings** dialog box opens if there are any errors. If the Map Editor made changes to the DTD to make it compliant with Gentran Integration Suite, it indicates the changes. Click **OK**.

13. Continue with the **New Map Wizard** as directed. When you click **Finish**, the Map Editor displays the new map in the Map Editor window.

SWIFTNet Syntax Validation

SWIFT field tag syntax validation is defined in the Map Editor and stored stored in the translator_swift_2007.properties file. This properties file contains the validation syntax for the field tags, as defined by the SWIFT standards (for example, field 73 is 35x[‘CRLF’35x]0-5), and may not be edited. The validation syntax for any given record is displayed in the Field Syntax parameter on the SWIFT Record Properties dialog box (SWIFT Validation tab). This parameter is automatically populated when you create a map using the SWIFTNet standards data dictionary. The convention of the syntax conforms exactly to the SWIFT documentation.

SWIFT also uses special validation functions such as <CUR>, <SWIFTBIC>, and <NON-SWIFTBIC>, which are components of a field tag. Each of these special functions has an expanded syntax that is validated by the Gentran Integration Suite translator. When these special validation functions are used, the special function syntax is displayed in the **Field Syntax** text box on the **SWIFT Record Properties** dialog box. The currency (<CUR>) and sender/receiver addresses (<SWIFTBIC>, <NON-SWIFTBIC>) must also be validated against legal code word lists. These code lists are defined in the SWIFTNet standards data dictionary and used by Gentran Integration Suite.

Note: You will need to maintain these code lists in Gentran Integration Suite (not in the Map Editor). See *Creating and Using Code Lists* for more information.

Additionally, the Map Editor and the translator allows complex syntax validation, which can be used for any data format. The Map Editor provides the Free Format option on the Field Properties Validation tab to enable you to specify that any characters are acceptable in the field, and the translator does not check the characters for compliance. To define a complex syntax, you just need to use the Free Format option for a String data type and define the syntax.

To define a complex syntax when you create a SWIFT map without using the SWIFTNet standards data dictionary or want to edit the syntax for a record:

1. In the Map Editor, right-click the record for which you want to define or edit the syntax.
2. In the **SWIFT Record Properties** dialog box, click the **SWIFT Validation** tab.
3. In **Field Syntax**, type the necessary complex syntax such as **3a[2!n]**, which indicates that the String should consist of zero to three uppercase letters followed by an optional two digits.
4. Click **OK** to save the information.

Creating Extended Rules for SWIFTNet Maps

In the next release of SWIFTNet there will be support to update existing maps to include or add any message rule validation changes. This future update feature will impact you only if you want custom code in the extended rules (in existing maps) included in the validation block that only runs when validation is enabled. The update feature places custom extended rule code outside of the validation block. Thus, if you want to create custom rules used only during validation, then you need to follow the example below when you write extended rules for SWIFTNet maps to ensure they are only run when validation is enabled:

```
if validation = 1 then
```

```
begin
   {type your custom validation extended rule code}
end
```

Extended Rules Used with SWIFTNet Maps

The following statements are used with SWIFTNet maps, though they can also be used with any other data format:

- cerror
- occurrencetotal
- resetoccurrencetotal
- sum
- sumtotal

See *Alphabetic Language Reference* for information on all the available extended rules.

Using Autolink

The Autolink function automatically creates links between input and output fields that have the same name or which contain logically equivalent business data. This functionality can be used regardless of which format you have selected for the input and output sides of your map. See *Using Autolink with the Map Editor* for more information on using this function. You can choose to link by either Field Name or Business Name.

Just like with the Link function, the link between two map components is represented visually with a connecting line. See *Creating Simple Links* for more information on the Link function.

Note: To increase the likelihood that the links in your maps are valid, in the Preferences dialog box Confirmations tab, select the link objects at different levels and link objects with different maximum usages confirmations. See *Using Autolink with the Map Editor* for more information on using this function.

Using Extended Rule Libraries with SWIFTNet

This section describes how to use the extended rule library and the properties of the dialog boxes that comprise its functionality. A rules library (used with SWIFTNet and any other data format) contains a list of rules in a separate file outside of the Map Editor source. Map Editor stores the name of the library in its source file, so when you open a map the library is also loaded. When you compile a map, the library rules that are referenced in the map are also compiled. This enables you to create a library of extended rules and then add it to any other map, so you do not have to recreate those extended rules after the first time. You can use this functionality with any data format.

Note: The SWIFT extended rules libraries are automatically installed with Gentran Integration Suite (and checked in), contains all the extended rules necessary to carry out the business logic for SWIFT messages.

Semantic validation rules for Market Practices are stored in an extended rule library separate from the library that implements the SWIFT standard semantic validation rule

Only the extended rule libraries referenced by a map are compiled into the TXO translation object.

This functionality minimizes the impact to users when, for example, SWIFT updates their messages—without the rule library you would need to update the extended rules for each updated map (correlating to the updated messages), but using the extended rule library you just update the library and then use the library with all the applicable maps.

When you view the checked in libraries through the Extended Rule Library check in interface, you are also able to obtain a list of all the maps that use each library.

The extended rules library can contain many rules. An extended rule consists of a declarations section followed by a statements section. The *declarations* section is required only if you use additional variables. The declarations section is where you declare the names and types of any variables you use either in the extended rule. The *statements* section is where you define the actions that you want the extended rule to run.

When calling a rule library function, you can pass parameters.

You must declare any variables that are not already defined as part of the input or output specification of the map before you use those variables in an extended rule. For the extended rule libraries, you typically use global variables that are passed as parameters.

Rule libraries are versioned resources. When you create a new rule library you need to check it in to Gentran Integration Suite just like you need to check in maps. This also enables you to check out, version, and delete extended rule libraries. Furthermore, when you view the checked in libraries through the Extended Rule Library check in interface, you can also see all the maps that use each library. This is very important because it enables you to easily view a list of the maps that will need to be recompiled if you change an extended rule in a library (you would recompile all the maps that use that particular library).

Additionally, you can import and export extended rule libraries into Gentran Integration Suite using the Resource Manager.

You can call an extended rule from a library in any extended rule in a map.

See *Extended Rule Libraries* for more information on this functionality.

The extended rule libraries that are preloaded for use with SWIFTNet are as follows:

Library	ERL File Name	Description
SWIFT_2006	SWIFT_2006.erl	Contains rules to validate SWIFT 2006 MT messages.
SWIFT_2007	SWIFT_2007.erl	Contains rules to validate SWIFT 2007 MT messages.

Calling a Rule from an Extended Rule Library in a Map

You can call a rule from any extended rule library (that is currently checked in to Gentran Integration Suite) in any extended rule in your map. The syntax you use to call a rule from a library is:

```
call library_name.rule_name(parameter1, parameter2, parameter3)
```

In this syntax, **library_name** is the name of the extended rule library. For example, if the library is SWIFT_2006.erl, the library_name is **SWIFT**. The **rule_name** is the name of the rule that you defined in the Library Rule dialog box. There is no limit on the number of parameters you can use.

Note: You can have multiple rule libraries with the same name and different version numbers, but you can only use one rule library of the same name in a map (the last version of that rule library that was checked in to the system).

The syntax you use to call a rule with a return value set is:

```
integer i;
i = call library_name.rule_name parameter1
```

In this syntax, **i** is the return value set.

Searching for SWIFTNet Correlations

Correlation searches rely on name-value pairs that define the specific data items you are trying to locate. These defined name-value pairs are tracking points for business processes and documents. The SWIFTNet Correlation search feature offers the following additional benefits:

- ✦ You can receive the results of search queries more quickly.
- ✦ You can further refine correlation searches by specifying a start and end date/time range.
- ✦ You can further refine SWIFT correlation searches by specifying SWIFT-specific criteria.

Searching for SWIFT Messages Using Correlations

To perform an advanced search for documents:

1. From the **Administration** menu, select **Business Process** > **Monitor** > **Advanced Search** > **SWIFTNet Correlation**.
2. In the Document Advanced Search page, specify any combination of the following search criteria, as appropriate:

Field	Description	Action
Requestor DN	Distinguished name of the requestor.	Type the name of the requestor. **Note:** This DN must be registered with the SAG instance using SWIFTNet Alliance Webstation.
Responder DN	Distinguished name of the responder.	**Note:** This DN must be registered with the SAG instance using SWIFTNet Alliance Webstation.
Service Name	Name of the service to which both SWIFT correspondents have subscribed.	Type the service name. **Note:** This must be a SWIFTNet service to which you are subscribed.
Request Type	Request type supported by the message exchange.	Type the request type.
Interface	SWIFTNet message type.	Select either InterAct or FileAct.
Operation	The SWIFTNet operation to send an InterAct or FileAct message.	Select from the following: ◆ get (FileAct) ◆ put (FileAct) ◆ sync (synchronous — Interact) ◆ async (asynchronous — Interact)
SnF	Indicates if the file transfer is done using the store-and-forward method.	Select either True (use Store-and-Forward) or False (do not use Store-and-Forward).
Direction	Indicates the direction of the messages.	Select either Inbound or Outbound.

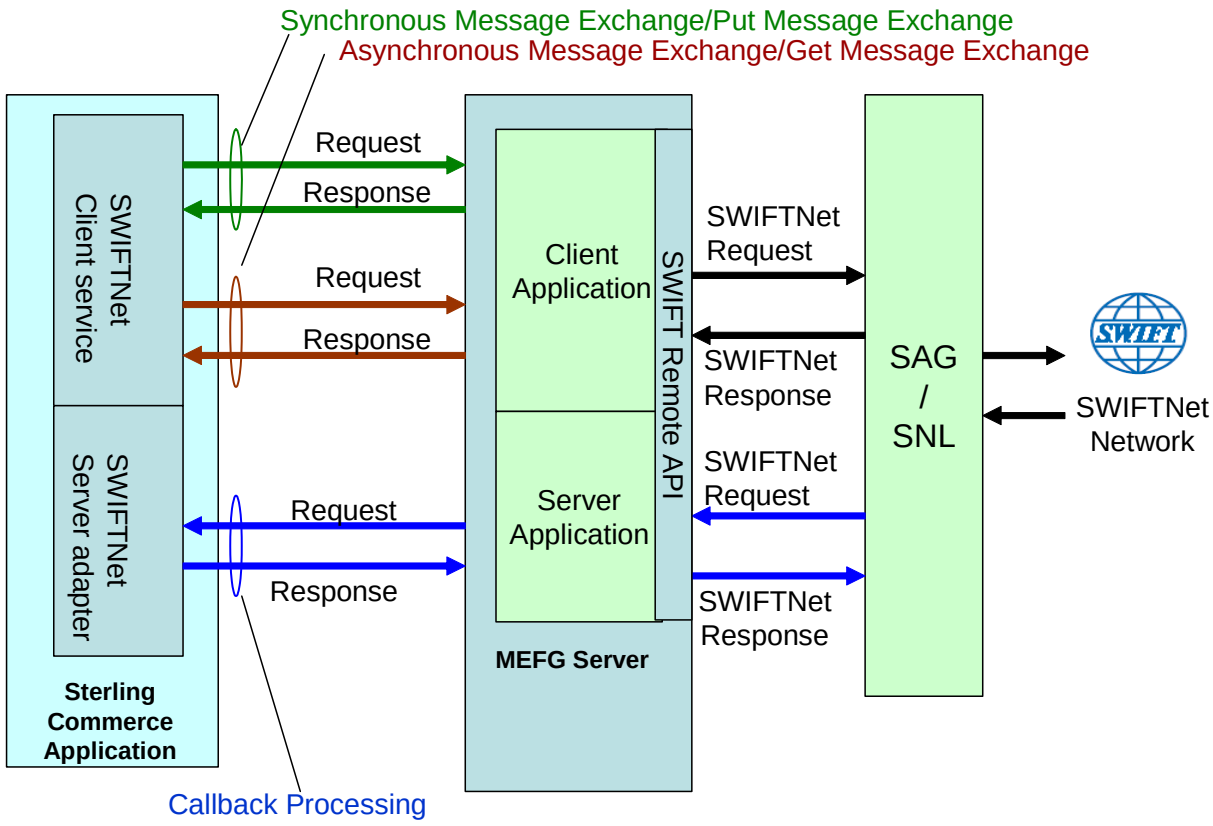
Field	Description	Action
Status	The status of the messages.	Select one of the following formats: ◆ Success ◆ Failed ◆ In Process
Activity From	Documents in progress or completed after the specified start date and time.	Using the following formats, type a starting date and time range and select A.M. or P.M.: ◆ Date – MM/DD/YYYY ◆ Time – HR:MN:SC
Activity To	Documents in progress or completed before the specified end date and time.	Using the following formats, type an end date and time range and select A.M. or P.M.: ◆ Date – MM/DD/YYYY ◆ Time – HR:MN:SC
File Name (FileAct Only)	Search for SWIFTNet records which are associated with a specific file.	Type the name of the file for which you want to search the associated records. **Note:** This search is valid for FileAct messages only.
File Size From	Search for SWIFTNet records which are associated to files with a specified size range.	Type the beginning size range of the files.
File Size To	Search for SWIFTNet records which are associated to files with a specified size range.	Type the endpoint of the range for the file size.
File Size	Measurement of file size	Select from the following: ◆ bytes ◆ KB ◆ MB
Results per page	Number of documents you want to display on the results page.	Select the value to indicate the number of documents to display. Valid values are: ◆ 5 ◆ 15 ◆ 30 ◆ 50

3. Click **Go!** The SWIFTNet Correlation page opens, listing the documents that match your search criteria.

Overview of SWIFTNet Transport

The Gentran Integration Suite enables you to send messages to SWIFTNet using either the InterAct or FileAct protocol. Gentran Integration Suite enables you to connect to SWIFTNet through the InterAct protocol for real-time messaging, store-and-forward messaging and real-time query and response. It supports delivery notification, non-repudiation, and message priority.

This diagram illustrates the process flow between Gentran Integration Suite and the SWIFTNet network through the FileAct and InterAct protocols:



InterAct and FileAct Protocol

When you use the InterAct or FileAct protocol to transport messages to and from SWIFTNet, the SWIFTNet MEFG Server serves requests and receives messages to and from SWIFTNet, through a client application and a server application that communicate with the SWIFTNet network through the InterAct protocol. The SWIFTNet MEFG Server operates independently from Gentran Integration Suite and includes all the APIs necessary to communicate with the SWIFTNet network.

The SWIFTNet Server adapter is responsible for receiving request messages from SWIFTNet through the Gentran Integration Suite SWIFTNet MEFG Server and sending responses back. The SWIFTNet Server adapter is comprised of two parts: the service part and the adapter part. The service part is used in a BPML

that does not require configuration except for enabling it for document tracking. The adapter part is configured through the Admin Console or the GPM, and this adapter is responsible for starting and stopping the SWIFTNet MEFG Server from Gentran Integration Suite using the Command Line 2 adapter (CLA2), which is build into the SWIFTNet Server adapter. A Command Line 2 client operates in remote installations (with the SWIFTNet MEFG Server) to enable Gentran Integration Suite to run a program from a command line in a business process.

The SWIFTNet Client service is responsible for sending SWIFTNet InterAct messages (both requests and responses) to SWIFTNet which are initiated by Gentran Integration Suite. The SWIFTNet Client service enables you to use InterAct messaging with a store-and-forward option. The benefits of using store-and-forward include:

- The sender and receiver do not need to be online at the same time, as is necessary for real-time messaging.
- The sender is notified if a message cannot be delivered (and can optionally be notified when messages are delivered).

The SWIFTNet Client service enables you to use either synchronous or asynchronous messaging.

The SWIFTNet Routing Rule is used by the SWIFTNet Client service to route incoming request to a business process for processing. It uses four parameters:

- RequesterDN
- ResponderDN
- Service name
- Request type

These parameters are used to map an incoming request to a business process. The SWIFTNet Routing Rule page enables you to assign any business process to a set of routing parameters.You need to create a SWIFTNet routing rule and associate it with an appropriate business process to process incoming SWIFTNet requests.

Additional FileAct Protocol Options

To send and receive SWIFTNet messages through the FileAct protocol, you will either use the SWIFTNet MEFG Server or use one of the following two methods to connect Gentran Integration Suite to SWIFT:

- **WebSphere MQ Adapter or WebSphere MQ Suite** (see *Configuring the WebSphere MQ Adapter/Suite to Communicate with SWIFT* on page 142)—enables you to configure Gentran Integration Suite to send and receive InterAct and FileAct files to/from SWIFTNet through the WebSphere MQ Interface for SWIFTAlliance Access (MQSA). Also enables you to send and receive FIN messages to/from SWIFTNet though SWIFTNet Alliance Access (SAA) and the SWIFTNet Remote API Host Adapter (RAHA).
- **Connect:Direct for SWIFTNet**—enables you to send and receive FileAct files to/from SWIFTNet through the SWIFTNet Remote API (SWIFTNet RA) and the SWIFTNet Remote API Host Adapter (RAHA).

SWIFTNet MEFG Server

- *Overview* on page 119
- *Client Application* on page 120
- *Server Application* on page 121
- *SWIFTNet MEFG Server Installation* on page 121
- *Monitoring the Status of the SWIFTNet MEFG Server* on page 127
- *Starting and Stopping the SWIFTNet MEFG Server* on page 127

Overview

The Gentran Integration Suite SWIFTNet MEFG Server serves requests and receives messages to and from SWIFTNet, through a client application and a server application that communicate with the SWIFTNet network through the InterAct or FileAct protocol. The SWIFTNet MEFG Server operates independently from Gentran Integration Suite and includes all the APIs necessary to communicate with the SWIFTNet network.

Gentran Integration Suite enables you to use either InterAct or FileAct messaging with a store-and-forward option. The benefits of using store-and-forward include:

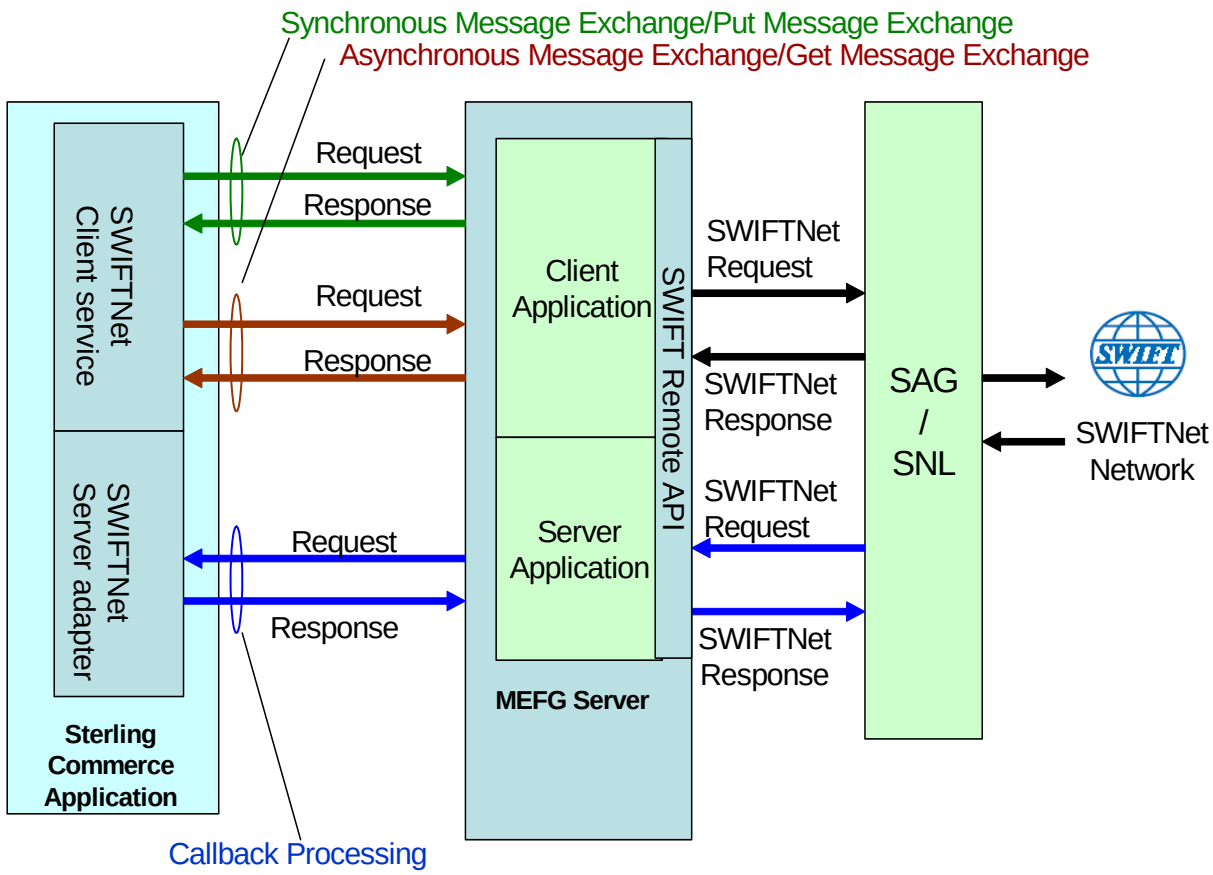
- The sender and receiver do not need to be online at the same time, as is required for real-time messaging.
- The sender is notified in the event delivery fails (and can optionally be notified upon delivery of the message).

The SWIFTNet client application serves requests to the SWIFTNet network through the SWIFTNet Alliance Gateway/SWIFTNet Net Link (SAG/SNL) instance. The client application listens for request from the Gentran Integration Suite SWIFTNet Client service, and interacts with SWIFTNet to obtain responses.

The SWIFTNet MEFG Server application receives requests from SWIFTNet. The server application listens for requests from SWIFTNet and interacts with Gentran Integration Suite to obtain responses. A request from the server application to Gentran Integration Suite calls the SWIFTNet Server adapter to process the request.

The SWIFTNet MEFG Server server application is started by enabling (and stopped by disabling) the Gentran Integration Suite SWIFTNet Server adapter. The starting and stopping of the server application is handled through the Command Line 2 adapter, which is built into the SWIFTNet Server adapter.

This diagram illustrates the process flow between Gentran Integration Suite and the SWIFTNet network through the SWIFTNet MEFG Server:



The administration of the SWIFTNet MEFG Server is through SWIFTNet Server adapter, including enabling and disabling the SWIFTNet MEFG Server.

Client Application

The client application can exchange messages in synchronous or asynchronous mode for InterAct processing or can exchange messages in Put or Get mode for FileAct processing.

Synchronous Message Exchange

When the client application is communicating in synchronous mode messaging, the SWIFTNet Client service prepares the request and sends it to the SWIFTNet MEFG Server. Then, the client application on the SWIFTNet MEFG Server processes the request, performs the necessary communication exchange with the SWIFTNet SAG/SNL instance and sends the request to the SWIFTNet network.

In synchronous mode, the client application is blocked until a response is received from the responder through the SAG/SNL instance. Once a response is received, it is sent to Gentran Integration Suite by the client application, and the response is placed in the primary document.

Asynchronous Message Exchange

In asynchronous mode, the SWIFTNet Client service prepares the request and sends it to the SWIFTNet MEFG Server. The client application on the SWIFTNet MEFG Server processes the request, performs the necessary communication exchange with the SWIFTNet SAG/SNL instance, and sends the request to the SWIFTNet network.

In asynchronous mode, the client application receives a response handle from the SAG/SNL instance. Using this response handle, the client application periodically checks with the SWIFTNet network to determine if a response is available . Once a response is received by Gentran Integration Suite, it is placed in the primary document.

Configuring the Client Application

There is no configuration necessary for the client application (either in synchronous or asynchronous mode or in Put or Get mode), but you must appropriately configure the SWIFTNet Client service to use the client application. See *SWIFTNet Client Service*.

Server Application

When a request from the SWIFTNet network arrives, the SWIFTNet SAG/SNL sends it to the SWIFTNet MEFG Server server application. The server application processes the request and forwards the request to Gentran Integration Suite. When Gentran Integration Suiter receives the request, it invokes the SWIFTNet Server adapter to process the request. Based on the SWIFTNet routing rule that you configure (using requestorDN, responderDN, service name and request type to map to a business process), a business process is invoked to produce the response. When a response is received from Gentran Integration Suite, the server application will formulate the SWIFTNet response and send it back to SWIFTNet via the SWIFTNet SAG/SNL instance.

Configuring the Server Application

There is no configuration necessary for the server application, but you must appropriately configure the SWIFTNet Server adapter to use the server application. See *SWIFTNet Server Adapter*.

SWIFTNet MEFG Server Installation

The SWIFTNet MEFG Server installation consists of a sequence of related tasks. This table outlines the process flow you must follow to install the SWIFTNet MEFG Server:

Task Number	Description	For more information
1	Configure SAG/SNL.	*Configuring SAG/SNL* on page 122
2	Install and configure the SWIFTNet remote API.	*Installing the SWIFTNet Remote API (RA)* on page 124
3	Install the SWIFTNet MEFG Server.	*Installing the SWIFTNet MEFG Server* on page 125

Task Number	Description	For more information
4	Install the Command Line 2 Adapter Client,	*Starting the Command Line 2 Client* on page 127
5	Configure and enable the SWIFTNet Server Adapter and start the SWIFTNet MEFG Server. **Note:** To monitor the status of the SWIFTNet MEFG Server, you need to select **Show Advanced Status** when you configure the SWIFTNet Server adapter.	*SWIFTNet Server Adapter*.
6	Configure and enable the SWIFTNet Client Service.	*SWIFTNet Client Service*.
7	Configure the SWIFTNet routing rule.	*SWIFTNet Routing Rule.*
8	Configure Mailboxes (only if you are executing store-and-forward).	*Using Mailboxes.*
9	Configure the SWIFTNetClient business process.	*SWIFTNetClient Business Process.*

Prerequisites

The following prerequisites must be met for the SWIFTNet MEFG Server to operate:

- The Command Line 2 adapter client (CLA2Client) must be running to receive commands from Gentran Integration Suite to start and stop the SWIFTNet MEFG Server.
- The Command Line 2 adapter client (CLA2Client) must be deployed on the same machine as the SWIFTNet MEFG Server.
- SWIFTNet Remote API (RA) must be installed on the same machine as the SWIFTNet MEFG Server.
- The SAG/SNL installed and configured with appropriate message partners and endpoints. See *Configuring SAG/SNL* on page 122.
- Create a configuration file to configure the SWIFTNet MEFG Server to work Gentran Integration Suite and the SAG/SNL instance. See *SWIFTNet Server Adapter*.
- You must have a SWIFTNet Subscription for the InterAct and/or FileAct protocols.
- You must install the SWIFTNet MEFG Server on either the Sun_Solaris operating system version 9.0, Windows 9 operating system, or AIX operating system.

Configuring SAG/SNL

Complete the following steps to configure SAG/SNL for use with the SWIFTNet MEFG Server:

1. Create a configuration file to configure the SWIFTNet MEFG Server to work Gentran Integration Suite and the SAG/SNL instance. See *SWIFTNet Server Adapter*.
2. Log in as an administrator to the SWIFTAlliance Workstation.
3. Go to **Gateway Admin - Application Interface** and create the client and server message partners.

   Note: The client (type = Client) and server (type = Server) message partner names must match the names in the SWIFTNet MEFG Server configuration (<**SagMessagePartnerClientName**> and <**SagMessagePartnerServerName**>).

4. In the Application Interface module, for the server message partner, configure the parameters as follows:

Parameter	Configuration
Name	Name from the SWIFTNet MEFG Server configuration (<SagMessagePartnerServerName>).
Type	Server
Status	Enabled
Unit	None
Host Adapter	Remote API Host Adapter
Supported Message Formats	Select **Strict SNL Format**.
Additional Processing	Select **Remote API Host Adapter**.

5. In the Application Interface module, for the client message partner, configure the parameters as follows:

   Note: The Application Interface must be started.

Parameter	Configuration
Name	Name from the SWIFTNet MEFG Server configuration (<SagMessagePartnerClientName>).
Type	Client
Status	Enabled
Unit	None
Default Message Format for Emission (from Message Partner)	Strict SNL Format **Note:** Strict SNL Format is required by the API.
Supported Message Formats	Select **Strict SNL Format**. **Note:** Strict SNL Format is required by the API.
Additional Processing	**Note:** Do not select any additional processing options.

6. In the Endpoints module, for the server message partner, configure the endpoint parameters as follows to define where to route the messages:

Parameter	Configuration
Name	Name from the SWIFTNet MEFG Server configuration.
Destination	Application Interface:<Name from the SWIFTNet MEFG Server configuration>
Status	Enabled

7. In the Endpoints module, for the server message partner, configure the routing detail parameters as follows:

Parameter	Configuration
From	SWIFTNet Interface
Sequence	**Note:** This is the sequence number.
Name	Name from the SWIFTNet MEFG Server configuration (<SNLEndPoint>).
Status	Enabled
SNL Endpoint	None
Service Name	None
Request Type	None
Requestor DN	Relation: **Equals (=**) Parameter: **o=administrator,o=swift**
Responder DN	Relation: **Equals (=**) Parameter: **o=administrator,o=swift**
Traffic Type	None
Delivery Mode	None
Priority	None

8. In the Endpoints module, for the server message partner, configure the destination detail parameters as follows:

Parameter	Configuration
Interface	Application Interface
Application	Name of the SWIFTNet MEFG Server server application (<SagMessagePartnerServerName>).
Mode	Strict

Installing the SWIFTNet Remote API (RA)

You need to install the SWIFTNet Remote API on the machine on which the SWIFTNet MEFG Server will be installed. This is the software distributed by SWIFT, the API that the SWIFTNet MEFG Server uses to connect to the SWIFTNet SAG/SNL instance to link into a SAG.

Complete the following steps to install the RA:

1. Install the remote API on the machine where you will install SWIFTNet MEFG Server.
2. Configure the RA to point to the SAG instance you will be accessing. See the documentation for the SWIFTNet Remote API for more information.
3. If you are installing on the Windows operating system, add **SYSTEM** to the Security settings, allow SYSTEM **Full Control** and select A**llow inheritable permissions from parent to propagate to this object**. See the documentation for the SWIFTNet Remote API for more information.

Note: To confirm that the change was made, right-click **<install_dir>SWIFTAlliance\RA\li**b directory and select **Properties**. Select the **Security** tab and verify that SYSTEM is listed as having Full Control.

Installing the SWIFTNet MEFG Server

Complete the following steps to install the SWIFTNet MEFG Server:

Note: The installation script and the binary install are located on the Gentran Integration Suite installation CD.

1. Log on to your UNIX system using the same account as the one you used to install and configure SWIFTNet RA.
2. Type the following command to change to the directory where the install binaries are located:

   ```
   cd <Gentran Integration Suite installation directory>/packages
   ```

3. Type the following commands to invoke the installation script:

   ```
   Is swiftnet*.jar [the siwftnet[version].jar is displayed]
   java -jar swiftnet[version].jar
   ```

4. Press **Enter**.
5. Type in destination directory where you want to install the SWIFTNet MEFG Server and press **Enter**.
6. When you are prompted for confirmation, type **y** and press **Enter**.

   Note: To change the destination directory, type **n** and press **Enter**, and repeat step 5. If the destination directory does not have enough free disk space, the script suggests you delete enough files to provide the necessary disk space and then exits.

   The installation script copies files from the CD to the destination directory and verifies that the correct number of files and blocks are copied. .

   You are notified that the installation is complete with the following message: Installation of Gentran Integration Suite SWIFTNet MEFG Server component is finished. You will need to configure this application in your Gentran Integration Suite user interface.
7. If you are installing on the Windows operating system, you are required to enter the account user name in the format **DomainName\Username**. If it is a local user, type **.\Administrator**.
8. Type **Yes** to confirm and press **Enter**.
9. Type the correct account password, confirm the password, and press **Enter**.

   This installs the following service instances (you can verify this by checking **Control Panel** > **Administrative tools** > **Services**):

   - MEFGCommServer Service
   - MEFGSwiftnetServer Service Instance 1
   - MEFGSwiftnetServer Service Instance 2

Note: If you have already installed any of these services, you will be notified through an error message containing Error Code 1073 that the service or services are already installed. If this occurs, you can unregister the services as follows:

a. Go to <MEFG Server directory>/bin.

b. Type the following and press **Enter** after each line:

To uninstall the MEFG Comm Server Service, type `MEFGCommServer.exe -u`

To uninstall the MEFG SWIFTNet Server Service Instance 1, type `MEFGSwiftnetServer.exe -u s1`

To uninstall the MEFG SWIFTNet Server Service Instance 2, type `MEFGSwiftnetServer.exe -u s2`

10. If you are installing on a Windows operating system, you must set up the following environment variables under the System variables:

- PATH : append <MEFG installdir>/bin;<Swift RA API installdir>/bin;<Swift RA API installdir>/lib
- SWNET_HOME : <Swift RA API installdir>
- SWNET_CFG_PATH : <Swift RA API installdir>/Ra1/cfg;
- Allow the defined user to start MEFGSwiftnetServer and MEFGCommServer through the following steps:

a. Select **Control Panel > Administrative Tools > Local Security Settings**.

b. Select **Local Policies > User Rights Assignment**.

c. Double-click **Log on as a service** and assign the account user (that you entered during the installation process) to this setting.

11. If you are installing on a Windows AIX operating system, you must modify the startup script **MEFGCommServer.sh** in the **<Swiftnet Server install>/bin** directory by passing option **-o 3**.

In the **MEFGCommServer.sh** startup script, locate this line:

```
${SWV2DIR}/bin/${DAEM_NAME} ${PORT} -s 0 -a ${SWV2DIR} 1>/dev/null
2>/dev/null &
```

And change it by adding the following (in boldface type):

```
${SWV2DIR}/bin/${DAEM_NAME} ${PORT} -o 3 -s 0 -a ${SWV2DIR} 1>/dev/null
2>/dev/null &
```

12. Configure the SWIFTNet RA

Configuring Fail-over Processing Using the SWIFTNet MEFG Server

To set up the SWIFTNet MEFG Server in a dual-active SAG configuration for fail-over processing, specify the following application interface definitions for the SWIFTNet Server adapter:

- ✦ active-active Configuration
- ✦ RA1 definitions for primary SAG (s1)
- ✦ RA2 definitions for alternate SAG (s2)

Note: Certificates and profiles must be available on the SAG where they are used. For fail-over processing, Puts and Gets try to connect to the first SAG specified for s1 and s2. If the connection fails, the Put and Get try to connect to the next SAG. If this connection also fails, the cycle is repeated if retry has been enabled.

Starting the Command Line 2 Client

The Command Line 2 adapter client (CLA2Client) must be installed and run on a remote server. Complete the following steps to start the remote adapter implementation version of the Command Line 2 adapter:

1. Locate the client jar (CLA2Client.jar) in your Gentran Integration Suite installation that contains the necessary classes.
2. Move the client jar to the machine that will be running the remote Command Line 2 adapter client.

   Note: This is the machine on which the SWIFTNet MEFG Server is installed.
3. Start the remote adapter implementation using the following command:

   ```
   [path to java bin]/java -jar [path to CLA2 Client jar file]/CLA2Client.jar
   <port> [debug]
   ```

   Note: The port (above) will be used when you configure the SWIFTNet Server adapter.

   This is an example of the command to start the remote adapter implementation:

   ```
   GISServer/java -jar CLA2Client.jar 15699 debug
   ```

   Note: The [debug] option is not required, but you may find it helpful. When you upgrade Gentran Integration Suite, you will also need to obtain the corresponding new CLA2Client.jar file to avoid receiving a ClassConflict error.

Monitoring the Status of the SWIFTNet MEFG Server

To monitor the status of the SWIFTNet MEFG Server, you need to select **Show Advanced Status** when you configure the SWIFTNet Server adapter:

1. Select **Deployment > Services > Configuration**.
2. Search for SWIFTNet Server adapter or select it from the list and click **Go!**.

   Note: When you select the SWIFTNet Server adapter, make sure you also select the **Show Advanced Status** check box prior to clicking **Go!**. This enables you to view the Advanced Status column on the Services Configuration page to see whether the SWIFTNet MEFG Server is stopped or started.
3. Click **Edit**.
4. Specify field settings in the Admin Console. See *SWIFTNet Server Adapter.*
5. On the Confirm page, verify that the **Enable Service for Business Processes** check box is selected. This enables the adapter instance.

Starting and Stopping the SWIFTNet MEFG Server

To start and stop the SWIFTNet MEFG Server:

1. Select **Deployment > Services > Configuration**.

2. Search for SWIFTNet Server adapter or select it from the list and click **Go!**.

   Note: When you select the SWIFTNet Server adapter, make sure you also select the **Show Advanced Status** check box prior to clicking **Go!**. This enables you to view the Advanced Status column on the Services Configuration page to see whether the SWIFTNet MEFG Server is stopped or started.

3. Once the SWIFTNet Client adapter is configured and saved, click the **Enabled** check box on the Services Configuration page. This starts the SWIFTNet MEFG Server.

   Note: To stop the SWIFTNet MEFG Server, clear the **Enabled** check box on the Services Configuration page

Document Tracking for SWIFTNet Transport

- *Overview* on page 129
- *Monitoring the SWIFTNet Data Flow* on page 129
- *Monitoring the SWIFTNet Communication Session Records* on page 131

Overview

The document tracking support within the SWIFTNet Client service and SWIFTNet Server adapter provides you with a document-centric view of the whole process of SWIFTNet messaging. This gives you the ability to monitor the workflow not only from business process point of view, but also from the actual document point of view. In a single view, you can see how the document is transformed/translated from one form to another within Gentran Integration Suite, and how the request and response document correlate with each other.

To enable this feature, all the business processes that are that are related to the SWIFTNet workflow must have the Document Tracking option enabled when you check in or edit the business processes. To do so and enable the framework to track, select the **Document Tracking** check box on the Process Levels page. Set the following options as needed and leave the rest of the business process parameters as the defaults:

- On the **Deadline Settings** page, set the deadline and notification options, if necessary.
- On the **Life Span** page, set the life span, if necessary.

Monitoring the SWIFTNet Data Flow

The Data Flow Monitoring feature now enables you to view inbound and outbound SWIFTNet data flows. For both inbound and outbound, you can correlate and view the transformation of the selected document, and you can also see the SWIFTNet Message ID that is related to the document.

Note: Message ID is a unique identifier in SWIFTNet that is required to enable the user to discern duplicate documents—you can tell if a document is a duplicate of another document if the Message IDs of both documents are the same. The Message ID is extremely useful in enabling you to reconcile documents.

To perform an advanced search for SWIFTNet business process data flows:

1. From the **Business Process** menu, select **Advanced Search** > **Data Flows**.

2. In the Business Process Monitor Data Flows page, specify any combination of the following search criteria, as appropriate:

Field	Description
Search	
Endpoint	The remote endpoint of the data flows to search for. Host name or IP address. Optional.
Direction	Direction of the data flows to search for. Optional. Valid values are: ◆ Inbound ◆ Outbound
Protocol	Protocol for the data flows to search for. Optional. Valid values are: ◆ AS2 ◆ HTTP ◆ FTP ◆ SFTP ◆ MBI ◆ Connect:Direct ◆ WebDAV ◆ SWIFTNet
Status	Current or final status of a data flow. Optional. Select one of the following options: ◆ Normal ◆ Error
Document Name	For data flows associated with a specific document, enter the document name. Optional.
Data Size	Range of size of the data transferred to search for. From/To in bytes, KB, MB, or GB. Optional.
DateRange	From – The beginning date and time for data flows to search for To – The end date and time for data flows to search for **Note:** Select the calendar icon to the right of the date to access calendar information. Optional.
Save search results values by using tag	Enter a string for use in repeating the search in another session. Required.

Field	Description
Results per page	Select how many results to display per page. Required. Valid values are: ◆ 10 ◆ 25 ◆ 50 ◆ 100 ◆ 200 ◆ 250 ◆ 400 ◆ 500 Default is 10.
List Directly	
By Data Flow ID	Type the data flow ID for which you want to search.

3. Click **Go!** The Monitor page opens, listing the business process data flows that match your search criteria.
4. Click the Root Document Name corresponding to the data flow you want to view. There are two different types of data flows:
   - If the Root Document Name is **SWIFTNetRequest**, it corresponds to an outbound data flow in which Gentran Integration Suite is acting as the client making a SWIFTNet request to the SWIFTNet Alliance Gateway (SAG).
   - If the Root Document Name is **PsHttpDocument_node_***, it corresponds to an inbound data flow in which Gentran Integration Suite is acting as the server processing the request from the client.

Monitoring the SWIFTNet Communication Session Records

Gentran Integration Suite creates communication session records for any associated authentication, authorization, file transfer, or non-file transfer records, even if a document is not transferred and no data flow record is created. For example, session data can include a user connecting to a mailbox using FTP, receiving messages, and then quitting the FTP session.

To view SWIFTNet communications sessions records:

1. From the **Administration** menu, select **Business Processes > Advanced Search > Communication Sessions**.

   Complete the fields using the following descriptions:

Field	Description
Endpoint	The remote endpoint of the communication sessions to search for. Host name or IP address. Optional.

Field	Description
Protocol	Protocol for the communication sessions to search for. Optional. Valid values are: ◆ AS2 ◆ HTTP ◆ FTP ◆ SFTP ◆ MBI ◆ Connect:Direct ◆ WebDAV ◆ SWIFTNet
Date Range	From - The beginning date and time to search for communication sessions To - The end date and time to search for communication sessions **Note:** Select the calendar icon to the right of the date to access calendar information. Optional.
Principal	Search for communication sessions associated with a Principal participant. Optional.
Secure Mode	Search for communication sessions in a secure mode. Optional. Valid values are: ◆ SSL ◆ CCC
Save search results values by using tag	Enter a string for use in repeating the search in another session. Required.
Results per page	Select how many results to display per page. Required. Valid values are: ◆ 10 ◆ 25 ◆ 50 ◆ 100 ◆ 200 ◆ 250 ◆ 400 ◆ 500 Default is 10.
List Directly	By Communication Session ID

SWIFT Editor

Gentran Integration Suite provides you with an online editing interface, the SWIFT Editor, which enables you to correct a SWIFTNet message that was returned due to an error either in translation (a validation check failed) or transmission (including a process failure such as a negative acknowledgement or NAK). Gentran Integration Suite maintains a link between every SWIFT message that is returned for reprocessing and its historical predecessor, so there is a record of every rejected message all the way back to its initial submission, including references to each person that modified the message and the return code (error status and reason).

The SWIFT Editor also enables you to search through the following code lists quickly and easily:

- SWIFT Addresses List
- SWIFT Base Addresses List
- SWIFT Clearing Code Code List
- SWIFT Currency Code List
- SWIFT Country Code List

The editing process requires two different roles (performed by two separate people), per the SWIFT guidelines:

- Editor—This is the person actually editing a message.
- Reviewer—This is the person who can submit modified messages to be resent.

The combination of both roles provides the “four eyes” validation required by SWIFT.

The editing process is as follows:

1. An E-mail alert is sent to the configured address when an outbound translation has errors during FIN enveloping or when a NAK is received.
2. The Editor searches for the returned message through the EDI Correlations search (looking for messages with Ready For Edit status), and accesses the message in the SWIFT Editor.
3. After the Editor repairs the message (or verifies that it does not need repair) and saves it, a separate E-mail alert is sent to the address configured for the Reviewer (the person responsible for auditing the repair and resending the message).
4. The Reviewer receives the E-mail alert and audits the message (looking for messages with Ready For Resend status). The Reviewer accesses the SWIFT Editor in read-only mode. The error report specifies the original errors and the modified and/or added fields as links to their position in the document.
5. After reviewing a repair message, the Reviewer determines how to handle it. The Review can:
   - Reject changes and mark the message for further edit.
   - Abort the repair process entirely.
   - Resend the message using the specified business process.

Note: If the message is resent, the default business process to be used is SWIFTEnvelope (recommended). The message can be resent with validation turned off.

Editor Tasks

The Editor repairs and saves SWIFT messages that are returned or rejected because of translation or transmission errors.

The following caveats apply:

- The error report displayed in the SWIFT Editor when a Reviewer accesses a repaired document shows all errors and links them to their position in the SWIFT Editor.
- All groups and records can be collapsed and expanded as needed to focus viewing.
- Placing the mouse pointer over an error icon in the SWIFT Editor pops up the error message for reference.
- The SWIFT Editor allows users to modify field values and add or delete instances of repeating groups and records.

To search for messages in Ready to Edit status (Editor role task):

1. From the **Administration** menu, select **Business Process** > **Monitor** > **Advanced Search** > **EDI Correlation**.
2. In the Search Option area, specify the following search criteria, as appropriate:

Field	Description	Action
All Level Options		
Location	EDI correlations maintained in a specific location.	Select **Live Tables** – Display live (active) EDI correlations.
Search Level Type	EDI processing level.	Select **Transaction** – For the search query, display results from the transaction level.
Test Mode	Mode of the Gentran Integration Suite system where documents that contain the EDI correlations were created.	If desired, select one of the following options: ◆ Any (default) ◆ Test ◆ Production ◆ Information ◆ Interchange is a test ◆ Syntax only test ◆ Echo request ◆ Echo response Optional.
Direction	Flow of the documents that contain the EDI correlations.	If desired, select one of the following options: ◆ Any (default) ◆ Inbound ◆ Outbound Optional.

Field	Description	Action
Sender ID	ID for the organization that is sending documents.	Type the identifier of the sender. Optional.
Receiver ID	ID for the receiving organization.	Type the identifier of the receiver. Optional.
Sender ID Qualifier	Qualifier used with the Sender ID to define the organization that is sending documents.	Type the qualifier of the sender. Optional.
Receiver ID Qualifier	Qualifier used with the Receiver ID for the receiving organization.	Type the qualifier of the receiver. Optional.
Start Date	Documents in progress or completed after the specified start date and time.	Using the following formats, type a starting date and time range and select A.M. or P.M.: ◆ Date – MM/DD/YYYY ◆ Time – HR:MN:SC **Note:** Defaults to a range of the last 24 hours. Optional.
End Date	Documents in progress or completed before the specified end date and time.	Using the following formats, type an end date and time range and select A.M. or P.M.: ◆ Date – MM/DD/YYYY ◆ Time – HR:MN:SC **Note:** Defaults to a range of the last 24 hours. Optional.
Transaction Level Options		
Transaction Set ID	ID of the transaction set indicated in the document.	If desired, type the ID of the transaction set. Optional.
Compliance Status	Status of compliance checking at the transaction set level.	If desired, select one of the following options: ◆ Any (default) ◆ OK ◆ NOT OK Optional.
Message Repair Status	Status of message repair (for SWIFT documents only).	Select **Ready for Edit** status.

3. Click **Go!** to display the EDI correlation records that match your search criteria.
4. For SWIFT documents, on the EDI Correlation Transaction Results page, click **info** in the Detail column for the document you want to edit.
5. Next to Document Repair Status, select the **ReadyForEdit** link to access the SWIFT Editor.

Note: If the Document Repair Status is ReadyForEdit and is a link, you have the necessary permissions set to access and edit the document in the SWIFT Message Editor. If you have corrected a failed

document and saved it, the status is changed to ReadyforResend and an E-mail is sent to the address specified in Enveloping properties.

6. Review the errors in the Error Report (left side of window), select each error link, and repair the error as necessary. This may include changing the contents of the field, or adding or deleting fields.

Note: To add an occurrence of a repeating field or group, select **Add** at the appropriate juncture of the message structure. To delete an occurrence of a field or group, select **Delete** where appropriate.

7. When you are finished editing the message, click **Save**.
8. Review the changes on the Confirm page, and click **Finish**. This sends an E-mail to the Review, notifying them of that the message has been repaired.
9. If desired, in the EDI Correlation Transaction Detail Results page, click **info** to the right of Document Correlations.
10. In the Document Correlation Details page, view details about the message you selected, and to see the correlation between the message and corresponding EDI document or data. The details available include:
    - time stamp
    - scope
    - process ID
    - document name
    - data value

Note: When you access the returned SWIFT message through the Document Correlation Details page, you can view a tree view on the left that allows you to link directly to the previous version(s) of the message. The right pane of the tree view displays the correlation details of the SWIFT message.

11. Select the document link at the top right to view the SWIFT message as text.

Reviewer Tasks

The Reviewer repairs and saves SWIFT messages that are returned or rejected because of translation or transmission errors.

The following caveat applies:

- ✦ The error report displayed in the SWIFT Editor when a Reviewer accesses a repaired document shows all errors and links them to their position in the SWIFT Editor.
- ✦ All groups and records can be collapsed and expanded as needed to focus viewing.
- ✦ Placing the mouse pointer over an error icon in the SWIFT Editor Change Report (displayed on the left side of the window) pops up the error message for reference.

To search for messages in Ready to Resend status (Reviewer role task):

1. From the **Administration** menu, select **Business Process** > **Monitor** > **Advanced Search** > **EDI Correlation**.

2. In the Search Option area, specify the following search criteria, as appropriate:

Field	Description	Action
All Level Options		
Location	EDI correlations maintained in a specific location.	Select **Live Tables** – Display live (active) EDI correlations.
Search Level Type	EDI processing level.	Select **Transaction** – For the search query, display results from the transaction level.
Test Mode	Mode of the Gentran Integration Suite system where documents that contain the EDI correlations were created.	If desired, select one of the following options: ◆ Any (default) ◆ Test ◆ Production ◆ Information ◆ Interchange is a test ◆ Syntax only test ◆ Echo request ◆ Echo response Optional.
Direction	Flow of the documents that contain the EDI correlations.	If desired, select one of the following options: ◆ Any (default) ◆ Inbound ◆ Outbound Optional.
Sender ID	ID for the organization that is sending documents.	Type the identifier of the sender. Optional.
Receiver ID	ID for the receiving organization.	Type the identifier of the receiver. Optional.
Sender ID Qualifier	Qualifier used with the Sender ID to define the organization that is sending documents.	Type the qualifier of the sender. Optional.
Receiver ID Qualifier	Qualifier used with the Receiver ID for the receiving organization.	Type the qualifier of the receiver. Optional.
Start Date	Documents in progress or completed after the specified start date and time.	Using the following formats, type a starting date and time range and select A.M. or P.M.: ◆ Date – MM/DD/YYYY ◆ Time – HR:MN:SC **Note:** Defaults to a range of the last 24 hours. Optional.

Field	Description	Action
End Date	Documents in progress or completed before the specified end date and time.	Using the following formats, type an end date and time range and select A.M. or P.M.: ◆ Date – MM/DD/YYYY ◆ Time – HR:MN:SC **Note:** Defaults to a range of the last 24 hours. Optional.
Transaction Level Options		
Transaction Set ID	ID of the transaction set indicated in the document.	If desired, type the ID of the transaction set. Optional.
Compliance Status	Status of compliance checking at the transaction set level.	If desired, select one of the following options: ◆ Any (default) ◆ OK ◆ NOT OK Optional.
Message Repair Status	Status of message repair (for SWIFT documents only).	Select **Ready for Resend** status.

3. Click **Go!** to display the EDI correlation records that match your search criteria.
4. For SWIFT documents, on the EDI Correlation Transaction Results page, click **info** in the Detail column for the document you want to audit.
5. Next to Document Repair Status, select the **ReadyForResend** link to access the SWIFT Editor.

Note: If the Document Repair Status is ReadyForResend and is a link, you have the necessary permissions set to audit the document in the SWIFT Message Editor.

6. Review the changes made in the Change Report (left side of window), select each link, and review the change as necessary.
7. When you are finished auditing the entire message, click **Next**.
8. Specify how you want to handle this message:
   - Mark this document for further edit (sends the message back to the Editor for further editing)
   - Resend this document (sends the document using the business process you select)
   - Abort (terminates the audit process)
9. If you are resending the message, select the business process from the **Execute Business Process** list.
10. If you want to disable validation for the resending of this message, select **Turn Off Validation**.

Note: You will select this option if the you want to send a message that does not exactly conform to SWIFT validation rules.

11. Review the Confirm page, and click **Finish**.

SWIFTNet Routing Rule

The SWIFTNet Routing Rule is created by you as the Responder, because you are expecting a server request message from SWIFTNet. This rule is used by the SWIFTNet Server adapter to manage interactive messages from the Gentran Integration Suite SWIFTNet MEFG Server. The rule routes an incoming request message to a user-defined business process based on the following parameters:

- requestorDN
- responderDN
- requestType
- serviceName

You configure these four parameters and create the SWIFTNet routing rule through the SWIFTNet Routing Rule interface. The SWIFTNet Routing Rule page enables you to assign any business process to a set of Requestor, Responder, Service, and Request Type. The rules are applied to the routing of SWIFTNet messages (Funds, MX, or generic XML format), and the server response messages are constructed and sent back to SWIFTNet through the Gentran Integration Suite SWIFTNet MEFG Server.

You can also export and import SWIFTNet routing rules.

Creating a SWIFTNet Routing Rule and Associating it with a Business Process

To create a SWIFTNet routing rule and associate it with an appropriate business process to process incoming SWIFTNet requests:

1. If you have not done so already, modify the SWIFTNetClient business process or create a new instance of it. See *SWIFTNetClient Business Process* on page 63.
2. Create a business process to which the inbound messages will be routed by the SWIFTNet Server adapter.
3. From the Gentran Integration Suite **Deployment** menu, select **SWIFTNet Routing Rule**.
4. To the right of **Create new SWIFTNet routing rule,** click **Go!**.
5. Complete the following parameters and click **Next**:

Parameter	Description
SWIFTNet Routing Rule Name	Type the name of the SWIFTNet routing rule. Required.
Requestor DN	Type the distinguished name of the requestor. Required. **Note:** This value should be taken from the Requestor's SWIFTNet Client service configuration (or from the BPML parameters passed to the service), in accordance with the arrangement made between requestor and responder.

Parameter	Description
Responder DN	Type the distinguished name of the responder. Required. **Note:** This value should be taken from the Requestor's SWIFTNet Client service configuration (or from the BPML parameters passed to the service), in accordance with the arrangement made between requestor and responder.
Service Name	Type the name of the SWIFTNet Client service instance configuration that you created in step 1. Required. **Note:** This value should be taken from the Requestor's SWIFTNet Client service configuration (or from the BPML parameters passed to the service), in accordance with the arrangement made between requestor and responder.
Request Type	Type the request type (message type and name) supported by the file transfer responder. Optional. **Note:** This value should be taken from the Requestor's SWIFTNet Client service configuration (or from the BPML parameters passed to the service), in accordance with the arrangement made between requestor and responder.
Business Process	Select the business process that will be invoked to manage the request and handle the response to a SWIFTNet message (this is the business process that you created in step 2 that will be called when the requestor DN, responder DN, service name, and requestor type match). Required.

6. Click **Finish** to save the routing rule. The rule is now in effect for all incoming SWIFTNet messages (Funds, MX, or generic XML format).

Searching for a SWIFTNet Routing Rule

To edit or delete a SWIFTNet routing rule you must first specify the appropriate rule. You can locate a specific routing rule in two ways:

- Search for the routing rule by name.
- Select the routing rule from an alphabetical list.

Searching for the routing rule by name is more precise and provides fewer results. Searching from an alphabetical list will result in a list of all routing rules or all rules beginning with a specified letter or digit.

Once you search for the routing rule, you can easily edit or delete it from the SWIFTNet Routing Rule interface.

Searching for a Routing Rule by Name

To search for a routing rule by name:

1. From the Gentran Integration Suite **Deployment** menu, select **SWIFTNet Routing Rule**.

2. In the Search section, type the name of the routing rule. Case does not matter and you can type part of a name.

   Gentran Integration Suite returns a list of matches unless no routing rules meet the criteria you specified.

Searching for a Routing Rule from a List

To select a routing rule from a list:

1. From the Gentran Integration Suite **Deployment** menu, select **SWIFTNet Routing Rule**.
2. In the List section, select one of the following:
   - Alphabetically – Select **All** and click **Go!**
   - Alphabetically – Select a specific letter or digit (0 - 9) and click **Go!**

   Gentran Integration Suite returns a list of matches unless no routing rules meet your criteria.

Exporting and Importing a SWIFTNet Routing Rule

The Gentran Integration Suite Import/Export feature enables you to save time and increase the accuracy of duplicating supported resources on different Gentran Integration Suite environments that are set up for unique purposes. To import and export resources from one Gentran Integration Suite environment to another Gentran Integration Suite environment, both environments must be the same version.

Configuring the WebSphere MQ Adapter/Suite to Communicate with SWIFT

The WebSphere MQ adapter and the WebSphere MQ Suite enable you to configure Gentran Integration Suite to send and receive FIN messages to/from SWIFTNet through the WebSphere MQ Interface for SWIFTAlliance Access (MQSA). Also enables you to send and receive FIN messages to/from SWIFTNet though SWIFTNet Alliance Access (SAA) and the SWIFTNet Remote API Host Adapter (RAHA).

This table describes the tasks necessary to configure Gentran Integration Suite to communicate with SWIFTNet through either the WebSphere MQ adapter or the WebSphere MQ Suite:

Number	Task	For More Information
1	Configure Gentran Integration Suite to retrieve messages through SAA and MQSA.	*Configuring Gentran Integration Suite to Retrieve Messages* on page 142
2	Configure Gentran Integration Suite to send messages through SAA and MQSA.	*Configuring Gentran Integration Suite to Send Messages* on page 143
3	Disable the UMID and Block S options for all queues within the MQ interface.	*Configuring the UMID and Block S Options* on page 144

Configuring Gentran Integration Suite to Retrieve Messages

Complete these steps to configure Gentran Integration Suite to retrieve messages from SWIFTNet through SAA and MQSA:

Note: See *WebSphere MQ Adapter* and *WebSphere MQ Suite* documentation for more details on the general MQ options.

1. Configure an MQ Adapter service instance:
2. Go to **Deployment** > **Services** > **Configuration**.
3. Next to New Service, click **Go!**.
4. Select **WebSphere MQ Adapter** and click **Next**.
5. Type a name for the service and a description, and click **Next**.
6. On the “WebSphere MQ Parameters page, type the following information and click **Next**:
   - Set **Host Name** to the name or IP address of the machine hosting the WebSphere MQ that receives messages from SWIFT.
   - Set **Listening Port** to the port number for the MQ installation (if it is something other than the default).
   - Set **Queue Manager** to the name of the queue manager that contains the SWIFTNet message queues.
   - Set **Queue Name** to the name of the queue set up to receive messages from SWIFT.

- Set **Server Connection Channel** to the name of the connection channel associated with the queue manager.
- Set **User ID** and **Password** to the login information for the queue that is configured to receive messages, if required.
- Select the **Receiving messages from WebSphere MQ (Sync)** or **Receiving messages from WebSphere MQ (Async)** option, depending which mode you wish to use.
- Set the other parameters based on the specific MQ configuration desired.

7. Click **Finish** to save the configuration.
8. Create a business process to retrieve the messages. The following example business process retrieves a message from MQ in synchronous mode, assuming the service configuration **FromSAA** was created in step 1. Optional parameters have been set to make the business process wait for up to 10 seconds for a message to appear before timing out:

```
<process name="FromSAABP">
   <operation name="WebSphere MQ Adapter">
      <participant name="FromSAA"/>
      <output message="WebsphereMQInputMessage">
         <assign to="." from="*"></assign>
         <assign to="rcv_MQGMO_wait">Yes</assign>
         <assign to="rcv_MQGMO_waitInterval">10000</assign>
      </output>
      <input message="inmsg">
         <assign to="." from="*"></assign>
      </input>
   </operation>
</process>
```

Configuring Gentran Integration Suite to Send Messages

Complete the following steps to configure Gentran Integration Suite to send messages to SWIFTNet through SAA and MQSA:

1. Configure an MQ Adapter service instance:
2. Go to **Deployment > Services > Configuration**.
3. Next to New Service, click **Go!**.
4. Select **WebSphere MQ Adapter** and click **Next**.
5. Type a name for the service and a description, and click **Next**.
6. On the “WebSphere MQ Parameters page, type the following information and click **Next**:
   - Set **Host Name** to the name or IP address of the machine hosting the WebSphere MQ that receives messages from SWIFT.
   - Set **Listening Port** to the port number for the MQ installation (if it is something other than the default).
   - Set **Queue Manager** to the name of the queue manager that contains the SWIFTNet message queues.

- Set **Queue Name** to the name of the queue set up to receive messages from SWIFT.
- Set **Server Connection Channel** to the name of the connection channel associated with the queue manager.
- Set **User ID** and **Password** to the login information for the queue that is configured to receive messages, if required.
- Select the **Sending messages to WebSphere MQ** option.
- Set the other parameters based on the specific MQ configuration desired.

7. Click **Finish** to save the configuration.
8. Create a business process to send the message. The following example business process sends a message from MQ in synchronous mode, assuming the service configuration **ToSAA** was created in step 1. .

```
<process name="ToSAABP">
    <operation name="WebSphere MQ Adapter">
      <participant name="ToSAA"/>
      <output message="WebsphereMQInputMessage">
        <assign to="." from="*"></assign>
        <assign to="snd_MQMD_msgType">DATAGRAM</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
</process>
```

Configuring the UMID and Block S Options

Gentran Integration Suite does not support the UMID and S block options, so those must be disabled for all Queues within the MQSeries Interface.

Complete these steps to disable the **UMID** and **Block S** options:

1. Open the SWIFTAlliance Workstation application.
2. Double-click the **MQSeries Interface**.
3. Open each queue and verify the following options:
   - For both From MQ and To MQ queues, **Include UMID** must be set to No.
   - For the To MQ queues, **Include Block S** must be set to Without Block S.

SWIFTNet Error Codes

- *Overview* on page 145
- *Text Validation Error Codes* on page 145
- *Specific Error Codes for MUG-textval Rules* on page 158
- *Special Error Codes for Value-Added Service Messages* on page 159
- *Message Syntax and Semantic Rule Codes* on page 160
- *Knn: Code Word Validation in Generic Fields* on page 170
- *Header Validation Error Codes* on page 173

Overview

The Translation service produces a Translation Status report. The report contains information about the translation of the document and any compliance errors. Errors from the input side of the map are listed under the Input tab. Errors from the output side of the map are listed under the Output tab.

Compliance errors can also be specified with the extended rule *cerror*.

This section describes all the translation status errors that may be generated when using SWIFTNet.

Text Validation Error Codes

The following table contains the error codes for SWIFTNet text validation:

SWIFTNet Error Code	Translator Report Error Number	Error Description
M50	901	Message length exceeded
M60	902	Non-SWIFTNet character encountered (a character not included in the <X>, <Y>, <Z> character sets. Also see error code T32.)
T01	904	Code word error. This check applies to: ◆ Field 39B in MT700,705,707,710,720,740,747. ◆ Field :22H:COAL subfield 3 in MT503,504.
T02	905	Unable to determine the reason for NAK. Please contact your CSC immediately for advice.
T03	906	Code word error. This check applies to: ◆ Field 26C, subfield 3, in MT600,601,604,605,606,607,608,609. ◆ Field 38B, subfield 1, in MT405. ◆ Field :22H:COLA subfield 3 in MT503,504,505,506,507.

SWIFTNet Error Code	Translator Report Error Number	Error Description
T04	907	Code word error. This check applies to: ◆ Field 26C, subfield 4, in MT600,601,604,605,606,607,608,609. ◆ Field 38B, subfield 2, in MT405. ◆ Field :22H:COLA subfield 3 in MT503,504,505,506,507. ◆ Field :22H:REDE subfield 3 in MT528,529,536,537,548,575,578,584,586.
T05	908	Code word error. This check applies to: ◆ Field 68B, subfield 4, in MT609. ◆ Field 38B, subfield 3, in MT405. ◆ Field :22H:DEPO subfield 3 in MT503,504,505,506.
T06	909	Code word error. This check applies to: ◆ Subfield 1, field 32F ◆ Subfield 5, field 68B,68C ◆ Subfield 3, field 60F,60M,62F,62M,64,65 in MT608 or in any message appended in common groups n92,n95,n96.
T07	910	Code word error. This check applies to: Subfield 3, field 33G. Field :22H:INOU subfield 3 in MT503,505,527,558.
T08	911	Code word error. This check applies to: ◆ Field 23, in MT102_STP,609. ◆ Field 26G, in MT550, or in any message appended in common groups n92,n95,n96. ◆ Field 71A in MT100,101,102,103_not_STP,103_STP,104,107,405,740, or in any message appended in common groups n92,n95,n96. ◆ Field :22H:REPR subfield 3 in MT527,558,569.
T09	912	Code word error. This check applies to: ◆ Subfield 3, field 23 in MT305,601. ◆ Subfield 3, field 68C in MT609, or in any message appended in common groups n92,n95,n96. ◆ Field :25D:COLL subfield 3 in MT507 when Data Source Scheme (DSS) is not present.
T10	913	The repetitive sequence occurred more than the maximum number of times permitted.
T11	914	The repetitive sequence occurred less than the minimum number of times required.

SWIFTNet Error Code	Translator Report Error Number	Error Description
T12	915	Field, line, subfield, or component content error. Or, the format Reject/Return is not allowed for field 72 in this MT. Or, when "ISIN" is used at the beginning of line one in field 35B it must never be composed of lower-case letters, nor a mix of upper and lower-case letters. Or, in the ISITC MT521 or MT523 in field 35B the second line is mandatory and must not begin with any of the code words defined for the third and subsequent lines.
T13	916	The field tag is not expected at this location in this MT. Either a mandatory field is missing, the sequence of fields is not correct, the specified field is not allowed at this point in the MT, or the specified field is not a defined SWIFTNet field (for example, the field tag is invalid), an end-of-text sequence (CRLF-) was encountered when it was not expected, or more than one end-of-text sequence occurs in this message. Or in a common group message (i.e. n92,n95,n96) within the list of "Copy of any field(s) of the original message", there are generic fields and non-generic fields that are not allowed to be mixed in the same message. Or in one of the following ISO15022 messages: MT502, 503, 504, 505, 506, 507, 508, 509, 513, 514, 515, 518, 524, 527, 528, 529, 530, 535, 536, 537, 538, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 558, 564, 565, 566, 567, 568, 569, 575, 576, 578, 584, 586, 587, 588, 589, an optional sequence of fields was used, however a field or field qualifier which is required within that sequence is missing, or field 16R is present but its related code word is inappropriate. Or in a qualifier table, for a particular generic field, in a "repeatable" order, there is a list of "OR" qualifiers but more than one qualifier has been used in the repetitions of the generic field in this sequence.

SWIFTNet Error Code	Translator Report Error Number	Error Description
T14	917	Subfield[N] (negative indicator) must not be used when the amount, number, or number count component is equal to zero. This check applies to: ◆ Field 19A in MT502,513-515,518,527-529,535-537,540-548,558,564,566-567, 569,575,578,584,586-588. ◆ Field 32H in MT320,330,362,571. ◆ Field 32N in MT646. ◆ Field 33N in MT646, ◆ Field 34E in MT320,330,341,571. ◆ Field 34N in MT646 ◆ Field 35H in MT571,581. ◆ Field 37G in MT320,330,362,644. ◆ Field 37M in MT340,341,362,644. ◆ Field 37R in MT341,360,361,362,644. ◆ Fields 60A,62A,62B in MT572. ◆ Field 60B in MT571,572. ◆ Field 92A in MT502,506,513-515,518,527-529,540-547,558,564-566,568-569, 576,578,584,586-588. ◆ Field 92E in MT564,566. ◆ Field 93B in MT535,536,564-566,568. ◆ Field 93C in MT535,564-566,568. ◆ Field 93D in MT575. ◆ Field 98D in MTs 564, 566. ◆ Field 98E in MTs 513-515, 518, 528, 529, 540-548 ◆ 578, 586.Field 99A in MT506,513-515,518,528-529,535-536,540-547,569,575,578,586. ◆ Any of the above fields in common groups n92,n95,n96.
T15	918	Sign is not valid. <SIGN> must be either '+' or '-'.
T16	919	Time offset is not valid. <OFFSET> has the same format as time <HHMM>.
T17	920	Field, line, subfield, or component consists of blanks, <CRLF>, or it is missing a mandatory line, subfield, or component.

SWIFTNet Error Code	Translator Report Error Number	Error Description
T18	921	Component is not in the format 3!n and/or it is not within the range 100-999. This check applies to: ◆ Field 11R,11S: the first component must have the format 3!n and must be within the range 100-999. ◆ Field 12, and MT105,106: this component must have the format 3!n and must be within the range 100-999. ◆ Field 12, and MT NOT=(M105,106): this component must have the format 3!n. ◆ Refer to T88 for additional special exception checking. ◆ Field 61: if the first character of subfield 6 is 'S' then the next three characters must have the format 3!n and must be within the range 100-999.
T19	922	Code word error. This check applies to: ◆ Subfield 1 of field 87E,87F. ◆ Field :25D:4!c//<Status> subfield 3 (Status) in MT507 when Data Source ◆ Scheme (DSS) is not present. ◆ Field :22H:4!c//<Indicator> subfield 3 (Indicator) in MT 307,321.
T20	923	Code word error in subfield 1, component 3, of field 32K or 33K.
T21	924	(Available).
T22	925	A common reference mismatch exists between field 22, subfield 2, component 2 and one of the following: ◆ Field 36 in MT305. ◆ Field 33G subfield 2 in MT600. ◆ Field 32B subfield 2 in MT601. Or, a common reference mismatch exists between field 22C, component 2, and one of the following: ◆ Field 30P, YYMM of YYYYMMDD in MT360-362,364-365. ◆ Field 36 sequence B in MT300. ◆ Field 36 sequence B in MT303. ◆ Field 36A sequence C in MT303. ◆ Field 36 sequence D or field 37U sequence G in MT306. ◆ Field 37G sequence B in MT320,330. ◆ Field 37M sequence B in MT340,341. ◆ Field 37J sequence B in MT350.

SWIFTNet Error Code	Translator Report Error Number	Error Description
T23	926	Subfield 8 in field 61, subfield 5 in field 66A, or subfield 6 in field 26C is too long or contains only '//', or subfield 2 in fields 26A or 26B is too long or contains only '/'.
T24	927	Subfield 7 in field 61, subfield 4 in field 66A, subfield 5 in field 26C, subfield 1 in fields 26A or 26B is missing or is too long.
T25	928	Subfields 7 or 8 in field 61, subfield 4 or 5 in field 66A, subfield 5 or 6 in field 26C, subfield 1 or 2 in fields 26A or 26B has improper content.
T26	929	Subfield 7 or 8 in field 61, subfield 4 or 5 in field 66A, subfield 5 or 6 in field 26C, subfield 1 or 2 in fields 26A or 26B has improper content. This check applies to: ◆ fields 20,20C,21,21A,21F,21G,21P,21R.
T27	930	BIC incorrectly formatted or invalid.
T28	931	SWIFTNet BIC is not a valid destination.
T29	932	SWIFTNet BIC contains an invalid branch code.
T30	933	Excessive lines, subfields, or components were found in this field.
T31	934	The line, subfield, or component separator or delimiter is missing or incorrect.
T32	935	An expected subfield, component, or component separator was not found.
T33	936	The length of the field, line, subfield, or component contents is too long, or, the component consists of one or more hidden characters, or, the component consists of one or more imbedded characters which are inconsistent with the defined field format, or the characters do not belong to the correct character set.
T34	937	The length of the field, line, subfield, or component contents is too short.
T35	938	Code word error in subfield 1, field 26C.

SWIFTNet Error Code	Translator Report Error Number	Error Description
T36	939	Code word error. This check applies to: ◆ Field 14A in MT360,361. ◆ Field 14D in MT320,330,340,350,360,361. ◆ Field 14J in MT360,361. ◆ Field 17A in MT206,300,303,306,360,361,405. ◆ Field 17B in MT500-505,513-515,518-519,527-529,535-538,540,547,558,564,569,574W8BENO,574IRSLST,575-578,584,586-588. ◆ Field 17F in MT304,306,340,360,361,405. ◆ Field 17G,17N,17O in MT304. ◆ Field 17T,17U in MT300. ◆ Field 22A in MT293,300,303,304,306,320,330,340,341,350,360,361,362,364,365. ◆ Field 23B in MT 103_not_STP,103_STP,303. ◆ Field 94A in MT 300, 303, 304, 305, 306, 320, 330, 340, 341, 350, 360, 361, 362, 364, 365, 600, 601.
T37	940	Code word error. This check applies to: ◆ Subfield 2, field 35H when used with MT581 and with any message appended in common groups n92,n95,n96. ◆ Subfield 2, field 35T when used with MT552 and with any message appended in common groups n92,n95,n96.
T38	941	Illogical time specified.
T39	942	Code word error in subfield 2, field 66A.
T40	943	Missing amount/number or incorrect amount/number first character.
T41	944	Code word error in subfield 3, field 66A.
T42	945	Code word error in subfield 3, field 35U.
T43	946	The decimal separator in the amount/number subfield or component is missing, is not a valid character, or more than one separator is present.
T44	947	The SWIFTNet BIC exists but it is not enabled for FIN, or it is not a cutover.
T45	948	Invalid non-SWIFTNet BIC.
T46	949	A Test-and-Training destination must not be used in a LIVE message.

SWIFTNet Error Code	Translator Report Error Number	Error Description
T47	950	In an ERI field the data part of a code word /OCMT/, or /CHGS/ was being validated but the ending separator (such as, the third '/') is missing. This error is also a code word error that applies to: ◆ Field 14G, subfield 1, in MT360-361. ◆ Field 22, subfield 1, in MT305. ◆ Field 23A, subfield 1, in MT360-362,364-365. ◆ Field 23E. subfield 1, in MT101,103_not_STP,104,107,206,207,256,405,416. ◆ Field 35B, in ISITC MT521,523: invalid or duplicated code word in line 3 or following. ◆ Field 39P, subfield 1, in MT303. ◆ Field 57D, line 1, in ISITC MT521,523. ◆ Field 61, subfield 9, duplicated code words: /OCMT/ or /CHGS/. ◆ Field 72 (narrative), duplicated code words: /OCMT/ or /CHGS/. ◆ Field 72 (structured), duplicated code words: /OCMT/, /CHGS/, or /INS/. ◆ Field 77A, duplicated code words: /OCMT/ or /CHGS/. ◆ Field 72, in ISITC MT521,523: invalid or duplicated code word, or in ISITC MT523 mandatory code word is missing. ◆ Field 77D, line 1, sequence C in MT303. ◆ Field 77D, lines 1-6, in ISITC MT521,523: invalid or duplicated code word. ◆ Field 77H, subfield 1, in MT306,340,360,361. ◆ Field 79, duplicated code words: /OCMT/ or /CHGS/. ◆ Field 85D, line 1, in ISITC MT521. ◆ Field 86, duplicated code words: /OCMT/ or /CHGS/. ◆ Field 87D, line 1, in ISITC MT521,523. ◆ Field 88D, line 1, in ISITC MT521,523.

SWIFTNet Error Code	Translator Report Error Number	Error Description
T48	951	Code word error. This check applies to: ◆ Field 14G, subfield 2, in MT360-361. ◆ Field 22K, subfield 1, in MT306. ◆ Field 23A, subfield 2, in MT360-362,364-365. ◆ Field 23D, in MT340-341. ◆ Field 23E, subfield 1, in MT103_STP. ◆ Field 31P, subfield 2, in ISITC MT521,523. ◆ Field 35B, in ISITC MT521,523: invalid code word in line 1. ◆ Field 38E, subfield 2, MT360-361. ◆ Field 38G, subfield 2 or 4, MT340,360-361. ◆ Field 38H, subfield 2 or 4, MT360-361.
T49	952	(Available).
T50	953	Date error, or the value of "year" (YY) in a Value Date component <DATE2> is invalid.
T51	954	Code word 'C','D','RC','RD','EC','ED' error.
T52	955	Invalid currency code or price code 'PCT','REN', or 'YLD'.
T53	956	Code word error in subfield 6, component 1 of field 61.
T54	957	The format of the first line of Field 50F (Party Identifier) is invalid. This check applies to: ◆ Field 50F in MTs 101, 102, 102_STP, 103, 103_STP, 210, 910. ◆ This check applies to all MTs containing field 50F, when appended to Common Group MTs n92, n95, n96.
T55	958	Code word error. This check applies to Subfield 1, component 1: ◆ Field 50F in MTs 101, 102, 102_STP, 103, 103_STP, 210, 910. ◆ This check applies to all MTs containing field 50F, when appended to Common Group MTs n92, n95, n96.
T56	959	Code word error. This check applies to Subfield 2, component 1: ◆ Field 50F in MTs 101, 102, 102_STP, 103, 103_STP, 210, 910. ◆ This check applies to all MTs containing field 50F, when appended to Common Group MTs n92, n95, n96.
T57	960	Code word error in subfield 2 of fields 31H, 31J, or 31X.
T58	961	Code word error. This check applies to: ◆ Subfield 1 of field 35A,35N,35P,35S. ◆ Subfield 2 of field 35H,35T. ◆ Subfield 1 in the 2nd occurrence of field 35A in MT550.

SWIFTNet Error Code	Translator Report Error Number	Error Description
T59	962	(Available).
T60	963	Code word error. This check applies to: ◆ Field 26F in MT306. ◆ Field 40A in MT700,705.
T61	964	Code word 'D' or 'M' error. This check applies to: ◆ Field 32K, subfield 1, in MT400,405,410,416,420,422,430. ◆ Field 33k, subfield 1, in MT430. ◆ Field 37(A-F), subfield 2, in MT516,644,645,646. ◆ Field 38J, subfield 1, in MT320,330.
T62	965	Either the first subfield <DATE2>[<HHMM>] or the second subfield 7!a but not both must be present. If optional subfield 1 is used, component 1 <DATE2> of this subfield must be present. This check applies to: Fields 31H,31J,31X.
T63	966	Error in component 2 of field 22C or subfield 2, component 2, of field 22. When the last character of this component is zero '0' and the preceding character is not one '1' then the entire component must consist of zeros.
T64	967	Code word error. This check applies to: ◆ Field 12F in MT306. ◆ Field 40B, line 1, in MT710,720.
T65	968	(Available).
T66	969	Code word error. This check applies to: ◆ Field 12E in MT306. ◆ Field 40B, line 2, in MT710,720.
T67	970	Code word error. This check applies to: ◆ Field 17R in MT320,330. ◆ Field 17V in MT306. ◆ Field 49 in MT700,710,720.
T68	971	Code word error. This check applies to: ◆ Field 24D, subfield 1, in MT300,306,320,330,240,360,361. ◆ Field 41A, subfield 2, in MT700,705,710,720,740. ◆ Field 41D, subfield 2, in MT700,705,710,720,740.
T69	972	(Available).
T70	973	Either the Account number, the Place, or both must be present.

SWIFTNet Error Code	Translator Report Error Number	Error Description
T71	974	Code word error. This check applies to: ◆ Field 22D in MT360,361,364,365,405. ◆ Field 22E in MT405. ◆ Field 22G in MT306.
T72	975	Code word error. This check applies to: Field 22J in MT306. Field 23C, subfield 1, in MT405. Field 23F, subfield 1, in MT405.
T73	976	Invalid country code. Please refer to the BIC Directory General Information -Country Codes-.
T74	977	The currency code must be the same for each indicated subfield in the field.
T75	978	In MT405,n92,n95,n96: Field 38B. When subfields 1 and 2 contain "MONT/OTHR" then subfield 3 is mandatory, otherwise subfield 3 is not allowed.
T76	979	The first character in the first line of this field must be a '/', and there must be at least another line, but not more than 5 lines. This check applies to field 50H.
T77	980	If the first character of the first line of this component or sub-component is a '/', then there must be at least another line, but not more than 5 lines. Otherwise, no more than 4 lines are allowed. This check applies to: Fields 42D,50K,(50-58)D,59,(82-88)D, and subfield 2 of field 87F.

SWIFTNet Error Code	Translator Report Error Number	Error Description
T78	981	Invalid or duplicated code word, or a mandatory code word is missing. This check applies to: ◆ Field 53J, subfield 1 of each line, in MTs 300, 303 and 304, 306, 320, 330, 340, 341, 350. ◆ Field 56J, subfield 1 of each line, in MTs 300, 303 and 304, 306, 320, 330, 340, 341, 350. ◆ Field 57J, subfield 1 of each line, in MTs 300, 303 and 304, 306, 320, 330, 340, 341, 350. ◆ Field 58J, subfield 1 of each line, in MTs 300, 303 and 304, 306, 320, 330, 340, 341, 350. ◆ Field 82J, subfield 1 of each line, in MTs 300, 303 and 304, 305, 306, 320, 330, 350, 600, 601. ◆ Field 83J, subfield 1 of each line, in MTs 300, 303 and 304, 305, 306, 320, 330, 350, 600, 601. ◆ Field 84J, subfield 1 of each line, in MTs 300, 306, 320, 330. ◆ Field 85J, subfield 1 of each line, in MTs 300, 320, 330. ◆ Field 86J, subfield 1 of each line, in MTs 306, 320, 330, 340, 341, 350. ◆ Field 87J, subfield 1 of each line, in MTs 300, 303 and 304, 305, 306, 320, 330, 350, 600, 601. ◆ Field 88J, subfield 1 of each line, in MTs 300, 320.
T79	982	(Available).
T80	983	<Field 72 Reject/Return> or <Field 79 Reject Return> code word error, or mandatory code word missing, or code word not in proper sequence.
T81	984	Format of <Field 72 Reject/Return> is not allowed in this message. This check applies to: Field 72, MT102_STP,103_STP.
T82	985	ERI format is not allowed in this message. This check applies to: Field 72, MT102_STP,103_STP.
T83	986	(Available).
T84	987	(Available).
T85	988	Code word error. This check applies to Field 23G, subfield 2, in MTs 307, 308, 321, 380, 381, 500-510, 513-515, 517-519, 524, 527-529, 530, 535-538, 540-549, 558, 564-569, 574IRSLST, 574W8BENO, 575, 576, 578, 584, 586-589, or in any message appended in common groups n92, n95 or n96.

SWIFTNet Error Code	Translator Report Error Number	Error Description
T86	989	Code word error. This check applies to: Field 23G, subfield 1, in MTs 307, 308, 321, 380, 381, 500-510, 513-515, 517-519, 524, 527-529, 530, 535-538, 540-549, 558, 564-569, 574IRSLST, 574W8BENO, 575, 576, 578, 584, 586-589, or in any message appended in common groups n92, n95 or n96.
T87	990	In a generic field the colon ':' delimiter is not present at the expected position.
T88	991	Field 12 in MT570 may only consist of 571,572,573,577. Field 12 in MT920 may only consist of 940,941,942,950. Field 12 in MT973 may only consist of 971,972,998. Field 13A in MT507 may only consist of 503,504,505. Field 13A in MT549 may only consist of 509,535-538,548,567,575-577,584, 586,589.
T89	992	In a generic field either the qualifier is invalid, the qualifier is duplicated, a mandatory qualifier is missing, or the qualifier format is not valid.
T90	993	In a generic field either the issuer code format is invalid, the mandatory issuer code is missing, or the generic field format is invalid.
T91	994	In a generic field the slash '/' delimiter is not present at the expected position.
T92	995	Code word error. This check applies to: ◆ Field 16R in ISO15022. ◆ Field 16S in ISO15022.
T93	996	Code word error. This check applies to: Field 22B in MT320,330,364,365.
T94	997	In field 22, subfield 2, or in field 22C, the components 1 and 3, the values '0' and '1' are not permitted in <LC1> and <LC2>. However, the value '0' is allowed in the second (rightmost) position if the sender or receiver is a test and training user.
T95	998	In field 22, subfield 2, or in field 22C, the components 1 and 3 do not contain the bank code and location code of the message sender, and/or the bank code and location code of the message receiver.
T96	999	In field 22, subfield 2, or in field 22C, the components 1 and 3 are not in alphabetical sequence.
T97	1000	Code word error. This check applies to: Field 28E, subfield 2, in MT206,506,535-538,569,574WBENO,574IRSLST,575, 576,584,586.
T98	1001	(Available).

SWIFTNet Error Code	Translator Report Error Number	Error Description
T99	1002	A special function has been declared in the validation syntax that is not recognized. **Note:** If you receive this error, it indicates that a special function was declared in a syntax that was not recognized by the translator.

Specific Error Codes for MUG-textval Rules

The following table contains the specific error codes for SWIFTNet MUG-textval rules:

SWIFTNet Error Code	Translator Report Error Number	Error Description
G01	1003	AU/PDS: in MT100,103,202, the letter option of the "selected field" is not A or D.
G02	1004	AU/PDS: in MT100,103,202, the format of the "selected field" option A is not valid.
G03	1005	AU/PDS: in MT100,103,202, the format of the "selected field" option D is not valid.
G04	1006	AU/PDS: the "selected field" is missing. At least one of the following fields must be present: MT100,103: fields 56a, 57a.
G05	1007	LVTS: if 2 LVTS members and the first 6 characters of their destination ID are different, exchange a FIN message type 100,103,205, and the currency code used in tag 32A is "CAD" then the tag 103 must be present in the User Header and it must contain the code "CAD".
G06	1008	REMIT: in a FIN message MT103, the field 77T and the tag 119 with the code word "REMIT" (in the User Header) must either both be present or absent.
G07	1009	CLS: in an MT300 eligible for the FIN-Copy service CLS or CLT, any field 53 present in sequence B must be used with the letter option 'A'.
G08	1010	CLS: in an MT300 eligible for the FIN-Copy service CLS or CLT, both fields 57 in sequences B1 and B2 (index 20,24) must be used with the letter option 'A', field 57a: of subsequence B1 must contain the "CLSB" BIC bank code.
G09	1011	CLS: in an MT300 eligible for the FIN-Copy service CLS or CLT, if the tag 17U is used it must contain the value "N".
G10	1012	CLS: in an MT300 eligible for the FIN-Copy service CLS or CLT, any field 56 present in sequence B must be used with the letter option 'A'.
G11	1013	CLS: in an MT300 eligible for the FIN-Copy service CLS or CLT, if field 82 is present in sequence A it must be used with letter option 'A'.

SWIFTNet Error Code	Translator Report Error Number	Error Description
G12	1014	CLS: in an MT300 eligible for the FIN-Copy service CLS or CLT, if field 87 is present in sequence A it must be used with letter option 'A'.
G13	1015	CLS: in an MT304 sent to the CLSB server, field 94A must contain the code "ASET".
G14	1016	CLS: in an MT304 sent to the CLSB server, field 82 in sequence A must be used with letter option 'A'.
G15	1017	CLS: in an MT304 sent to the CLSB server, field 87 in sequence A must be used with letter option 'A'.
G16	1018	CLS: in an MT304 sent to the CLSB server, any field 53 present in sequence B must be used with the letter option 'A'.
G17	1019	CLS: in an MT304 sent to the CLSB server, both fields 57 in sequence B must be used with letter option 'A' and must contain the ""CLSB"" BIC bank code. **Note:** Field 57A, index 19, is mandatory.

Special Error Codes for Value-Added Service Messages

The following table contains the special error codes for SWIFTNet value-added service messages:

SWIFTNet Error Code	Translator Report Error Number	Error Description
B01	1020	PAC Trailer used for non-Premium service message. Message has PAC trailer but sender or receiver, or both, are not members of Premium service.
B02	1021	(Available)
B03	1022	103:LCH present in the message, but sender, receiver, or both are not members of LCH, or the message type is not allowed for LCH, or, '103:TPS' present in the message, but sender, receiver, or both are not members of TPS, or the message type is not allowed for TPS.
B04	1023	(Available)
B05	1024	A system error has occurred. The user should contact their local Customer Service Center for further information.

Message Syntax and Semantic Rule Codes

The following section contains the error codes for SWIFTNet message syntax and semantic rules.

The C, D, and E error code ranges refer to rule numbers (these represent the message validation rules), are documented as follows:

- Rules 000-099 as C00-C99
- Rules 100-199 as D00-D99
- Rules 200-299 as E00-E99

Rules 000-099

SWIFTNet Error Code	Translator Report Error Number
C00	1025
C01	1026
C02	1027
C03	1028
C04	1029
C05	1030
C06	1031
C07	1032
C08	1033
C09	1034
C10	1035
C11	1036
C12	1037
C13	1038
C14	1039
C15	1040
C16	1041
C17	1042
C18	1043
C19	1044
C20	1045
C21	1046
C22	1047

SWIFTNet Error Code	Translator Report Error Number
C23	1048
C24	1049
C25	1050
C26	1051
C27	1052
C28	1053
C29	1054
C30	1055
C31	1056
C32	1057
C33	1058
C34	1059
C35	1060
C36	1061
C37	1062
C38	1063
C39	1064
C40	1065
C41	1066
C42	1067
C43	1068
C44	1069
C45	1070
C46	1071
C47	1072
C48	1073
C49	1074
C50	1075
C51	1076
C52	1077
C53	1078

SWIFTNet Error Code	Translator Report Error Number
C54	1079
C55	1080
C56	1081
C57	1082
C58	1083
C59	1084
C60	1085
C61	1086
C62	1087
C63	1088
C64	1089
C65	1090
C66	1091
C67	1092
C68	1093
C69	1094
C70	1095
C71	1096
C72	1097
C73	1098
C74	1099
C75	1100
C76	1101
C77	1102
C78	1103
C79	1104
C80	1105
C81	1106
C82	1107
C83	1108
C84	1109

SWIFTNet Error Code	Translator Report Error Number
C85	1110
C86	1111
C87	1112
C88	1113
C89	1114
C90	1115
C91	1116
C92	1117
C93	1118
C94	1119
C95	1120
C96	1121
C97	1122
C98	1123
C99	1124

Rules 100-199

SWIFTNet Error Code	Translator Report Error Number
D00	1125
D01	1126
D02	1127
D03	1128
D04	1129
D05	1130
D06	1131
D07	1132
D08	1133
D09	1134
D10	1135
D11	1136
D12	1137

SWIFTNet Error Code	Translator Report Error Number
D13	1138
D14	1139
D15	1140
D16	1141
D17	1142
D18	1143
D19	1144
D20	1145
D21	1146
D22	1147
D23	1148
D24	1149
D25	1150
D26	1151
D27	1152
D28	1153
D29	1154
D30	1155
D31	1156
D32	1157
D33	1158
D34	1159
D35	1160
D36	1161
D37	1162
D38	1163
D39	1164
D40	1165
D41	1166
D42	1167
D43	1168

SWIFTNet Error Code	Translator Report Error Number
D44	1169
D45	1170
D46	1171
D47	1172
D48	1173
D49	1174
D50	1175
D51	1176
D52	1177
D53	1178
D54	1179
D55	1180
D56	1181
D57	1182
D58	1183
D59	1184
D60	1185
D61	1186
D62	1187
D63	1188
D64	1189
D65	1190
D66	1191
D67	1192
D68	1193
D69	1194
D70	1195
D71	1196
D72	1197
D73	1198
D74	1199

SWIFTNet Error Code	Translator Report Error Number
D75	1200
D76	1201
D77	1202
D78	1203
D79	1204
D80	1205
D81	1206
D82	1207
D83	1208
D84	1209
D85	1210
D86	1211
D87	1212
D88	1213
D89	1214
D90	1215
D91	1216
D92	1217
D93	1218
D94	1219
D95	1220
D96	1221
D97	1222
D98	1223
D99	1224

Rules 200-299

SWIFTNet Error Code	Translator Report Error Number
E00	1225
E01	1226
E02	1227

SWIFTNet Error Code	Translator Report Error Number
E03	1228
E04	1229
E05	1230
E06	1231
E07	1232
E08	1233
E09	1234
E10	1235
E11	1236
E12	1237
E13	1238
E14	1239
E15	1240
E16	1241
E17	1242
E18	1243
E19	1244
E20	1245
E21	1246
E22	1247
E23	1248
E24	1249
E25	1250
E26	1251
E27	1252
E28	1253
E29	1254
E30	1255
E31	1256
E32	1257
E33	1258

SWIFTNet Error Code	Translator Report Error Number
E34	1259
E35	1260
E36	1261
E37	1262
E38	1263
E39	1264
E40	1265
E41	1266
E42	1267
E43	1268
E44	1269
E45	1270
E46	1271
E47	1272
E48	1273
E49	1274
E50	1275
E51	1276
E52	1277
E53	1278
E54	1279
E55	1280
E56	1281
E57	1282
E58	1283
E59	1284
E60	1285
E61	1286
E62	1287
E63	1288
E64	1289

SWIFTNet Error Code	Translator Report Error Number
E65	1290
E66	1291
E67	1292
E68	1293
E69	1294
E70	1295
E71	1296
E72	1297
E73	1298
E74	1299
E75	1300
E76	1301
E77	1302
E78	1303
E79	1304
E80	1305
E81	1306
E82	1307
E83	1308
E84	1309
E85	1310
E86	1311
E87	1312
E88	1313
E89	1314
E90	1315
E91	1316
E92	1317
E93	1318
E94	1319
E95	1320

SWIFTNet Error Code	Translator Report Error Number
E96	1321
E97	1322
E98	1323
E99	1324

Knn: Code Word Validation in Generic Fields

The two digits **nn** indicate the field ID.

SWIFTNet Error Code	Translator Report Error Number
K00	1325
K01	1326
K02	1327
K03	1328
K04	1329
K05	1330
K06	1331
K07	1332
K08	1333
K09	1334
K10	1335
K11	1336
K12	1337
K13	1338
K14	1339
K15	1340
K16	1341
K17	1342
K18	1343
K19	1344
K20	1345
K21	1346

SWIFTNet Error Code	Translator Report Error Number
K22	1347
K23	1348
K24	1349
K25	1350
K26	1351
K27	1352
K28	1353
K29	1354
K30	1355
K31	1356
K32	1357
K33	1358
K34	1359
K35	1360
K36	1361
K37	1362
K38	1363
K39	1364
K40	1365
K41	1366
K42	1367
K43	1368
K44	1369
K45	1370
K46	1371
K47	1372
K48	1373
K49	1374
K50	1375
K51	1376
K52	1377

SWIFTNet Error Code	Translator Report Error Number
K53	1378
K54	1379
K55	1380
K56	1381
K57	1382
K58	1383
K59	1384
K60	1385
K61	1386
K62	1387
K63	1388
K64	1389
K65	1390
K66	1391
K67	1392
K68	1393
K69	1394
K70	1395
K71	1396
K72	1397
K73	1398
K74	1399
K75	1400
K76	1401
K77	1402
K78	1403
K79	1404
K80	1405
K81	1406
K82	1407
K83	1408

SWIFTNet Error Code	Translator Report Error Number
K84	1409
K85	1410
K86	1411
K87	1412
K88	1413
K89	1414
K90	1415
K91	1416
K92	1417
K93	1418
K94	1419
K95	1420
K96	1421
K97	1422
K98	1423
K99	1424

Header Validation Error Codes

The following table contains the specific H and U error codes for SWIFTNet header validations that Gentran Integration Suite supports:

SWIFTNet Error Code	Translator Error Code or Description of How This Validation is Handled and Enforced	Error Description
H01	Translator Error Code 1425—enforced by the envelope map	Basic Header no present or format error block 1
H02	Translator Error Code 1426 (inbound)—Gentran Integration Suite generates this error	Application Identifier not ‘A’ (GPA) or ‘F’ (FIN)
H03	Translator Error Code 1427(inbound and outbound)—Gentran Integration Suite generates this error	Invalid Service Message identifier (must be 01 or 21)
H4-H9	Not used	Available
H10	Translator Error Code 1434—not generated or enforced by Gentran Integration Suite	Bad LT address or application not enabled for the LT

SWIFTNet Error Code	Translator Error Code or Description of How This Validation is Handled and Enforced	Error Description
H15	Translator Error Code 1439—not generated or enforced by Gentran Integration Suite	Bad session number
H20	Translator Error Code 1444—not generated or enforced by Gentran Integration Suite	Error in the ISN
H21	Translator Error Code 1445 (outbound)—Gentran Integration Suite generates this error	Error in the message sender's branch code
H25	Translator Error Code 1449—enforced by the envelope map	Application header format error or not present when mandatory
H26	Translator Error Code 1450—enforced by the envelope map	Input/output identifier not "I" (on input from LT)
H30	Translator Error Code 1454 (inbound and outbound)—Gentran Integration Suite generates this error	Message type for a FIN message not found in code list.
H40	Translator Error Code 1464 (inbound and outbound)—Gentran Integration Suite generates this error	Message priority other than S (System) for a message type < 100, or message priority of S for message type < 100
H50	Translator Error Code 1474 (inbound and outbound)—Gentran Integration Suite generates this error	Receiver ID (ID + 'X' + branch code) not found in code list.
H51	Translator Error Code 1475 (outbound)—Gentran Integration Suite generates this error	Message Type < 100 must have a receiver ID of SWFTXXX and a branch code of XXX.
H52	Translator Error Code 1476—not generated or enforced by Gentran Integration Suite	MT 072, selection of Test and Training mode/version, MT 077 Additional Selection Criteria for FIN are not allowed while a FIN session is open
H55	Translator Error Code 1479—not generated or enforced by Gentran Integration Suite	Message type not allowed for fallback session for MT 030
H80	Translator Error Code 1504—enforced by the envelope map	Delivery option error
H81	Translator Error Code 1505—enforced by the envelope map	Obsolescence period error
H98	Translator Error Code 1522 (inbound and outbound)—Gentran Integration Suite generates this error	Sender ID (ID + 'X' + branch code) not found in code list.

SWIFTNet Error Code	Translator Error Code or Description of How This Validation is Handled and Enforced	Error Description
H99	Translator Error Code 1523—enforced by the envelope map	Error can be one of the following: ◆ Invalid receiver destination (invalid character or LT identification is not “X”) ◆ Invalid date or time (not numeric or not within range)
U00	Translator Error Code 1524—enforced by the envelope map	Bad block 3 format
U01	Translator Error Code 1525—enforced by the envelope map	Bad bank priority
U02	Translator Error Code 1526—enforced by the envelope map	Bad MUR
U03	Translator Error Code 1527—not generated or enforced by Gentran Integration Suite	Neither bank priority nor MUR present
U07	Translator Error Code 1531 (outbound)—Gentran Integration Suite generates this error	User Header not permitted for user-to-system messages (that is, message type < 100)

SWIFTNet Error Code	Translator Error Code or Description of How This Validation is Handled and Enforced	Error Description
U08	Translation Error Code 1532 (inbound and outbound)—Gentran Integration Suite generates this error	Tag 119 is not one of the following: ◆ REMIT (any message) ◆ RFDD (any message) ◆ STP (102 or 103) ◆ COMM (503, 504, 505, 506, or 507) ◆ CRPR (503, 504, 505, 506, or 507) ◆ CRSP (503, 504, 505, 506, or 507) ◆ CRTL (503, 504, 505, 506, or 507) ◆ EXTD (503, 504, 505, 506, or 507) ◆ FIXI (503, 504, 505, 506, or 507) ◆ FORX (503, 504, 505, 506, or 507) ◆ LIQU (503, 504, 505, 506, or 507) ◆ OTCD (503, 504, 505, 506, or 507) ◆ PAYM (503, 504, 505, 506, or 507) ◆ REPO (503, 504, 505, 506, or 507) ◆ SBSB (503, 504, 505, 506, or 507) ◆ SCRP (503, 504, 505, 506, or 507) ◆ SECL (503, 504, 505, 506, or 507) ◆ SLEB (503, 504, 505, 506, or 507) ◆ TCRP (503, 504, 505, 506, or 507) ◆ W8BENO (574) ◆ IRSLST (574)
U09	Translation Error Code 1533 (inbound and outbound)—Gentran Integration Suite generates this error	Tag 119 present for a message type other than 102, 103, 104, 503, 504, 505, 506, 507, 521, 523, and 574.