

*SeeBeyond ICAN Suite*

# SWIFT OTD Library User's Guide

*Release 5.0.1*



The information contained in this document is subject to change and is updated periodically to reflect changes to the applicable software. Although every effort has been made to ensure the accuracy of this document, SeeBeyond Technology Corporation (SeeBeyond) assumes no responsibility for any errors that may appear herein. The software described in this document is furnished under a License Agreement and may be used or copied only in accordance with the terms of such License Agreement. Printing, copying, or reproducing this document in any fashion is prohibited except in accordance with the License Agreement. The contents of this document are designated as being confidential and proprietary; are considered to be trade secrets of SeeBeyond; and may be used only in accordance with the License Agreement, as protected and enforceable by law. SeeBeyond assumes no responsibility for the use or reliability of its software on platforms that are not supported by SeeBeyond.

SeeBeyond, e\*Gate, e\*Way, and e\*Xchange are the registered trademarks of SeeBeyond Technology Corporation in the United States and/or select foreign countries. The SeeBeyond logo, SeeBeyond Integrated Composite Application Network Suite, eGate, eWay, eInsight, eVision, eXchange, eView, eIndex, eTL, ePortal, eBAM, and e\*Insight are trademarks of SeeBeyond Technology Corporation. The absence of a trademark from this list does not constitute a waiver of SeeBeyond Technology Corporation's intellectual property rights concerning that trademark. This document may contain references to other company, brand, and product names. These company, brand, and product names are used herein for identification purposes only and may be the trademarks of their respective owners.

© 2004 by SeeBeyond Technology Corporation. All Rights Reserved. This work is protected as an unpublished work under the copyright laws.

**This work is confidential and proprietary information of SeeBeyond and must be maintained in strict confidence.**

Version 20040611162417.

# Contents

---

## Chapter 1

<b>Introducing the SWIFT OTD Library</b>	<b>6</b>
Overview	6
2003 Library Features	7
Library Versions and Access	7

---

## Chapter 2

<b>Installing the SWIFT OTD Library</b>	<b>8</b>
Supported Operating Systems	8
System Requirements	9
External System Requirements	9
Installation Procedures	9
After Installation	10
Increasing the Enterprise Designer Heap Size	10

---

## Chapter 3

<b>Using the SWIFT OTD Library</b>	<b>12</b>
SWIFT Message Type OTDs	12
SWIFT Message Structure	12
OTD and Collaboration Locations in Enterprise Designer	13
SWIFT Message Type Reference	14
Category 1 Messages	15
Category 2 Messages	16
Category 3 Messages	17
Category 4 Messages	18
Category 5 Messages	19
Category 6 Messages	22
Category 7 Messages	23
Category 8 Messages	24
Category 9 Messages	25
Category n Messages	26

---

**Chapter 4**

<b>Using Message Validation Features</b>	<b>27</b>
<b>Library Validation Features: Overview</b>	<b>27</b>
Basic Validation Features	27
Validation Components	28
Validation Collaboration Definitions	28
Validation Operation	28
Library Methods	29
<b>Message Format Validation Rules</b>	<b>30</b>
MFVR Validation Rules	31
MFVR Validation Collaboration Definitions	31
MFVR Validation Methods	32
MFVR Errors	32
<b>Market Practice Rules</b>	<b>32</b>
MPR Validation Collaboration Definitions	33
MPR Validation Methods	33
MPR Errors	34
<b>Using the Validation Project Examples</b>	<b>35</b>
Projects and the Enterprise Designer	35
Importing Sample Projects	36
SWIFT Gold Validation Project	37
eInsight Validation Project	38
Using eGate With eInsight	38
Using eInsight With eGate Components	38
SWIFT OTD Library With eInsight	39
Using a Business Process	40
Configuring the Modeling Elements	40
Copying the Output File	41
Unmarshaling and Marshaling the Data	42
Returning the Value	43
Creating the Project's Environment	43
Deploying a Project	43
Basic Steps	44
Alerting and Logging	44
<b>BICDirService Browser Lookup</b>	<b>44</b>
Updating BICDirService	45
Live Updating	45
Source of Information	45
Update Operation	45
BICDirService Wildcard Support	46
BICDirService Method Operation	46
Lookup Method Definitions	47
Validation Method Definitions	47
BICDir Exceptions	47
BICDirService Sample Project	48
Using the BICDirService Lookup Feature	48
<b>Error Message Information</b>	<b>50</b>
Error Messages	50

## Contents

Format of Error Messages	50
Debug Flags	50
Message Examples	51
Parse Debug Level Message Example	51
Debug Level Message Example	53

---

## Chapter 5

<b>Using SWIFT Funds OTD Library</b>	<b>54</b>
SWIFT OTD Library Funds Features: Overview	54
SWIFT Funds Message Templates	56
SWIFT Standards and Message Types	56

---

## Chapter 6

<b>Using OTD Library Java Classes</b>	<b>58</b>
SWIFT OTD Library Classes: Overview	58
Relation to OTD Message Types	58
SWIFT OTD Library Javadoc	58
OTD Library Java Classes	59

<b>Index</b>	<b>60</b>
--------------	-----------

# Introducing the SWIFT OTD Library

This guide explains how to install and operate the SeeBeyond® Integrated Composite Application Network Suite™ (ICAN) SWIFT OTD Library.

This chapter provides a brief overview of operations, components, and general features of the Object Type Definition (OTD) Library.

### Chapter Topics

- [“Overview” on page 6](#)
- [“2003 Library Features” on page 7](#)
- [“Library Versions and Access” on page 7](#)

---

## 1.1 Overview

The Society for Worldwide Interbank Financial Telecommunication (SWIFT) OTD Library contains template OTDs for use with the SeeBeyond ICAN Suite. These OTDs correspond to the SWIFT user-to-user message types employed by its SWIFT network. The library provides an individual OTD for each SWIFT message type, as defined in the SWIFT standards documentation.

Each OTD in the SWIFT OTD Library represents a corresponding SWIFT message type. [Chapter 3](#) shows a complete list of these OTDs. You can use these OTDs to transport SWIFT message data in ICAN.

This user's guide explains how to use these OTDs in eGate, as well as the features available with them.

---

## 1.2 2003 Library Features

The SWIFT OTD Library (2003 version) allows you to use the following features:

- SWIFT Funds OTDs
- Message Format Validation Rules (MFVRs)
- Market Practice Rules (MPRs)
- BICDirService (Blank Identifier Code Directory Service) Lookup

See later chapters in this guide for complete explanations of how to use these features.

---

## 1.3 Library Versions and Access

SWIFT periodically revises their message types, adding to or subtracting from the total set of Message Types, and modifying the definitions of individual message types. New sets are identified with the year they are issued, such as 2001, 2002, or 2003.

SeeBeyond releases a new SWIFT OTD Library corresponding to each revised set of SWIFT message types. The current release includes templates supporting the 2001 through 2003 message type sets. This user's guide explains the 2003 version OTDs.

You must install each year's version via a separate **.sar** file (see [Chapter 2](#)). However, the Funds, Validation, and BICDirService (see [Chapter 4](#)) features can only be used with the 2003 version OTDs.

# Installing the SWIFT OTD Library

This chapter lists supported operating systems, shows requirements, and explains how to install the SWIFT OTD Library.

## Chapter Topics

- [“Supported Operating Systems” on page 8](#)
- [“System Requirements” on page 9](#)
- [“External System Requirements” on page 9](#)
- [“Installation Procedures” on page 9](#)
- [“After Installation” on page 10](#)
- [“Increasing the Enterprise Designer Heap Size” on page 10](#)

*Note:* See the *ICAN Suite Installation Guide* for complete eGate installation instructions.

---

## 2.1 Supported Operating Systems

The SWIFT OTD Library is available for the following operating systems:

- Windows 2000, Windows XP, Windows Server 2003
- HP Tru64 V5.1A
- HP-UX 11.0 and 11i (PA-RISC)
- IBM AIX 5.1L and 5.2
- Red Hat Enterprise Linux AS 2.1 (Intel x86)
- Red Hat Linux 8 (Intel x86)
- Sun Solaris 8 and 9



---

## 2.2 System Requirements

To use the SWIFT OTD Library, you need:

- eGate Logical Host.
- SeeBeyond eWay capable of connecting to an external network, for example, the File eWay.

### Logical Host requirements

The OTD library must be configured and administered using the eGate Enterprise Designer. For complete information on the Enterprise Designer system requirements, see the *ICAN Suite Installation Guide*.

### BICDirService requirements

To use the BICDirService feature of the OTD library (see [“BICDirService Browser Lookup” on page 44](#)), you must also have already installed the SeeBeyond ICAN Suite’s eVision Studio and eInsight Business Process Manager.

---

## 2.3 External System Requirements

If you are using the BICDirService feature, you need the update CD-ROM supplied by SWIFT. The Java constructor for the **BICDir** class loads the required data from the following SWIFT-supplied files:

- **FI.dat**
- **CU.dat**

See [“Updating BICDirService” on page 45](#) for details.

---

## 2.4 Installation Procedures

The Enterprise Manager, a Web-based application, is used to select and upload Object Type Definition (OTD) libraries (**.sar** files for the libraries) from the installation CD-ROM to the Repository.

The SWIFT OTD Library is installed using the Enterprise Manager. You must be installing eGate or have eGate already installed. Follow the instructions in the installation steps in the *ICAN Suite Installation Guide* and include the following steps:

- 1 During the procedures for uploading files to the Repository using the Enterprise Manager, select and upload one or more of the following files:
  - ♦ **SWIFT\_OTD\_Library2003.sar** (to install the 2003 OTD Library)
  - ♦ **SWIFT\_OTD\_Library2002.sar** (to install the 2002 OTD Library)
  - ♦ **SWIFT\_OTD\_Library2001.sar** (to install the 2001 OTD Library)

- 2 To use the sample Projects, you must also upload the following file:
  - ♦ **FileeWay.sar** (to install the File eWay)

**Note:** *The Funds, Validation, and BICDirService features can only be used with the 2003 OTDs (see [Chapter 4](#)).*

- 3 If you are installing eGate, continue as instructed in the *ICAN Suite Installation Guide*.

---

## 2.5 After Installation

Once the OTD library is installed and configured, its files must then be incorporated into a Project before it can perform its intended functions. See the *eGate Integrator User's Guide* for more information on incorporating OTDs into an eGate Project.

---

## 2.6 Increasing the Enterprise Designer Heap Size

Because of the size of the SWIFT OTD Library, the Enterprise Designer **Heap Size** may need to be increased before using eGate with the library. If the heap size is not increased, you may receive an **OutOfMemoryError** message, when you try to activate a SWIFT OTD Project.

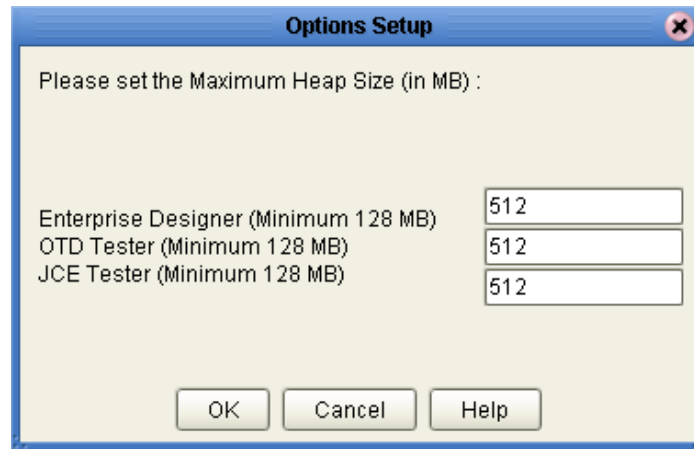
If you receive this message during Project activation, you must increase the heap size before you can activate any SWIFT OTD Projects. This action resets the Enterprise Designer's maximum memory size.

**To increase the heap size from the Enterprise Designer**

- 1 From the Enterprise Designer Menu bar click **Tools** and select **Options**. The **Options Setup** dialog box appears.

- 2 Increase the configured heap size for the Enterprise Designer, OTDTester, and JCE Tester to 512 MB as shown in Figure 1. Click **OK**.

**Figure 1** Options Setup: Heap Size



- 3 Close and restart the Enterprise Designer to allow your changes to take effect.

If an **OutOfMemoryError** message occurs while you are trying to open the Enterprise Designer, the heap size settings may be changed before starting the Enterprise Designer. You can increase the heap size values found in the **heapSize.bat** file.

Go to the following directory and file:

**<eGate Install Directory>/eDesigner/bin/heapSize.bat**

Change the following lines:

- **set eDesigner\_heap\_size=128**
- **set OTDTester\_heap\_size=128**
- **set JCETester\_heap\_size=128**

To read as follows:

- **set eDesigner\_heap\_size=512**
- **set OTDTester\_heap\_size=512**
- **set JCETester\_heap\_size=512**

# Using the SWIFT OTD Library

This chapter explains, lists, and provides a cross-reference for, the SWIFT OTD Library message types.

## Chapter Topics

- [“SWIFT Message Type OTDs” on page 12](#)
- [“SWIFT Message Type Reference” on page 14](#)

---

## 3.1 SWIFT Message Type OTDs

This section provides a general overview of the SWIFT message types and their OTDs.

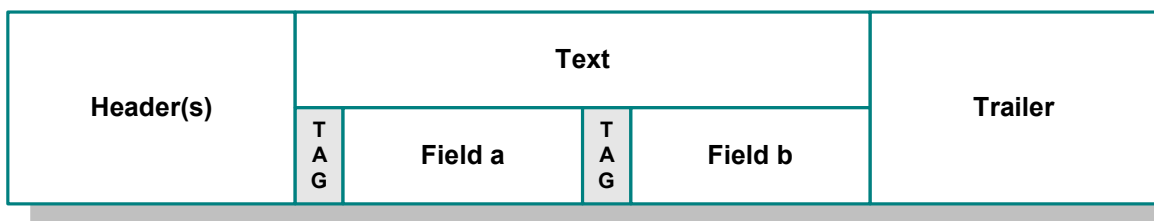
### 3.1.1 SWIFT Message Structure

A message used by the SWIFT network has a maximum of five components, as follows:

- Basic header block
- Application header block
- User header block (optional)
- Text block
- Trailer block

See Figure 2 for a diagram of the SWIFT message structure.

**Figure 2** SWIFT Message Structure



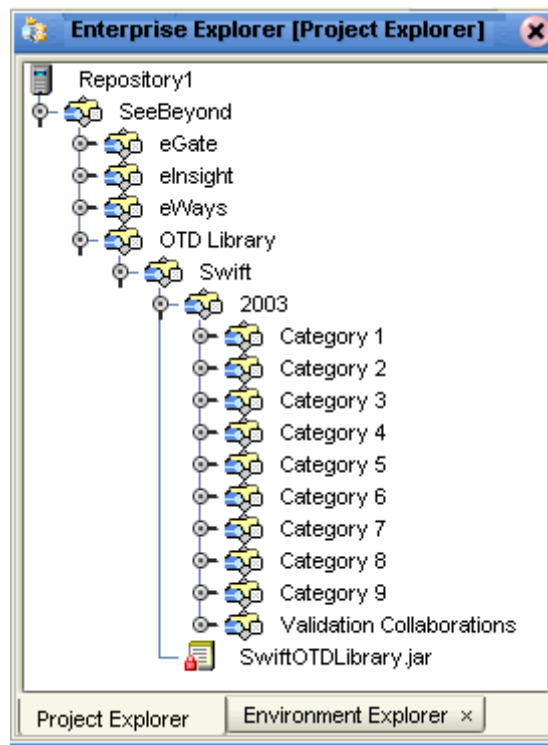
Each field component in the text block is preceded by a field tag. There are no field tags in the header and trailer blocks. The one exception to this format is MT 121, EDIFACT FINPAY, which has a single text field with no field tag identifier.

Information about a field common to all message types in which that field is used is found in the *Standards - General Field Definitions* volume of the *SWIFT User Handbook*. Information about a field specific to its use with a particular message type is found in the field specifications section of the *Standards* volume of the *SWIFT User Handbook* for that message type.

### 3.1.2 OTD and Collaboration Locations in Enterprise Designer

You can find the SWIFT OTDs, including the Fund OTDs, in the eGate Enterprise Designer's Project Explorer, as shown in Figure 3. This figure also shows the location of the Java-based Validation Collaboration Definitions.

Figure 3 SWIFT OTD Locations



The **Validation Collaborations** directory contains the Collaboration Definitions that enable the validation features of the SWIFT OTD Library. See [Chapter 4](#) for details.

The **Category 5** directory contains the SWIFT Funds message template OTDs in the library. See [Chapter 5](#) for details.

The **SwiftOTDLibrary.jar** file allows you to update the BICDirService feature. See [Chapter 4](#) for details.

---

## 3.2 SWIFT Message Type Reference

SWIFT groups message types into the following categories:

### Customer Payments and Cheques

- See [“Category 1 Messages” on page 15.](#)

### Financial Institution Transfers

- See [“Category 2 Messages” on page 16.](#)

### Treasury Markets: Foreign Exchange and Derivatives

- See [“Category 3 Messages” on page 17.](#)

### Collections and Cash Letters

- See [“Category 4 Messages” on page 18.](#)

### Securities Markets

- See [“Category 5 Messages” on page 19.](#)

### Treasury Markets: Precious Metals and Syndications

- See [“Category 6 Messages” on page 22.](#)

### Documentary Credits and Guarantees

- See [“Category 7 Messages” on page 23.](#)

### Travellers Cheques

- See [“Category 8 Messages” on page 24.](#)

### Cash Management and Customer Status

- See [“Category 9 Messages” on page 25.](#)

### Common Group Messages

- See [“Category n Messages” on page 26.](#)

The final category (**Category n**) in the previous list comprises a group of message types common to all the other categories. The OTDs in the SWIFT OTD Library are grouped into the same categories. The remainder of this chapter explains these categories and the message types within each category.

The 2001, 2002, and 2003 versions of the SWIFT OTD Library are available. You must install each version via a separate **.sar** file (see [Chapter 2](#)). However, the Funds, Validation, and BICDirService features can only be used with the 2003 OTDs (see [Chapter 4](#)).

This chapter explains only the 2003 SWIFT message types. For explanations of the 2001 and 2002 versions, see the SWIFT Alliance Web site.

### 3.2.1 Category 1 Messages

Table 1 explains the Category 1 message types, Customer Payments and Cheques, with the type designation MT 1xx.

**Table 1** Customer Payments and Cheques

<b>SWIFT Message Type</b>	<b>Description</b>
MT 101	Request for Transfer
MT 102	Multiple Customer Credit Transfer
MT 102+	Multiple Customer Credit Transfer (STP)
MT 103	Single Customer Credit Transfer
MT 103+	Single Customer Credit Transfer (REMIT, STP)
MT 104	Customer Direct Debit
MT 105	EDIFACT Envelope
MT 106	EDIFACT Envelope
MT 107	General Direct Debit Message
MT 110	Advice of Cheque(s)
MT 111	Request for Stop Payment of a Cheque
MT 112	Status of a Request for Stop Payment of a Cheque
MT 121	Multiple Interbank Funds Transfer (EDIFACT FINPAY)
MT 190	Advice of Charges, Interest and Other Adjustments
MT 191	Request for Payment of Charges, Interest and Other Expenses
MT 192	Request for Cancellation
MT 195	Queries
MT 196	Answers
MT 198	Proprietary Message
MT 199	Free Format Message

### 3.2.2 Category 2 Messages

Table 2 explains the Category 2 message types, Financial Institution Transfers, with the type designation MT 2xx.

**Table 2** Financial Institution Transfers

<b>SWIFT Message Type</b>	<b>Description</b>
MT 200	Financial Institution Transfer for its Own Account
MT 201	Multiple Financial Institution Transfer for its Own Account
MT 202	General Financial Institution Transfer
MT 203	Multiple General Financial Institution Transfer
MT 204	Financial Markets Direct Debit Message
MT 205	Financial Institution Transfer Execution
MT 206	Cheque Truncation Message
MT 207	Request for Financial Institution Transfer
MT 210	Notice to Receive
MT 256	Advice of Non-Payment of Cheques
MT 290	Advice of Charges, Interest and Other Adjustments
MT 291	Request for Payment of Charges, Interest and Other Expenses
MT 292	Request for Cancellation
MT 293	Information Service Message
MT 295	Queries
MT 296	Answers
MT 298	Proprietary Message
MT 299	Free Format Message



### 3.2.3 Category 3 Messages

Table 3 explains the Category 3 message types, Treasury Markets, Foreign Exchange, Money Markets, and Derivatives, with the type designation MT 3xx.

**Table 3** Treasury Markets, Foreign Exchange, Money Markets, and Derivatives

<b>SWIFT Message Type</b>	<b>Description</b>
MT 300	Foreign Exchange Confirmation
MT 303	Forex/Currency Option Allocation Instruction
MT 304	Advice/Instruction of a Third Party Deal
MT 305	Foreign Currency Option Confirmation
MT 306	Foreign Currency Option
MT 307	Advice/Instruction of a Third Party FX Deal
MT 308	Instruction for Gross/Net Settlement of Third Party FX Deals
MT 320	Fixed Loan/Deposit Confirmation
MT 321	Instruction to Settle a Third Party Loan/Deposit
MT 330	Call/Notice Loan/Deposit Confirmation
MT 340	Forward Rate Agreement Confirmation
MT 341	Forward Rate Agreement Settlement Confirmation
MT 350	Advice of Loan/Deposit Interest Payment
MT 360	Single Currency Interest Rate Derivative Confirmation
MT 361	Cross Currency Interest Rate Swap Confirmation
MT 362	Interest Rate Reset/Advice of Payment
MT 364	Single Currency Interest Rate Derivative Termination/Recouping Confirmation
MT 365	Single Currency Interest Rate Swap Termination/Recouping Confirmation
MT 380	Foreign Exchange Order
MT 381	Foreign Exchange Order Confirmation
MT 390	Advice of Charges, Interest and Other Adjustments
MT 391	Request for Payment of Charges, Interest and Other Expenses
MT 392	Request for Cancellation
MT 395	Queries
MT 396	Answers
MT 398	Proprietary Message
MT 399	Free Format Message

### 3.2.4 Category 4 Messages

Table 4 explains the Category 4 message types, Collections and Cash Letters, with the type designation MT 4xx.

**Table 4** Collections and Cash Letters

<b>SWIFT Message Type</b>	<b>Description</b>
MT 400	Collections: Advice of Payment
MT 405	Collections: Clean Collection
MT 410	Collections: Acknowledgment
MT 412	Collections: Advice of Acceptance
MT 416	Collections: Advice of Non-Payment/Non-Acceptance
MT 420	Collections: Tracer
MT 422	Collections: Advice of Fate and Request for Instructions
MT 430	Collections: Amendment of Instructions
MT 450	Cash Letters: Cash Letter Credit Advice
MT 455	Cash Letters: Cash Letter Credit Adjustment Advice
MT 456	Cash Letters: Advice of Dishonor
MT 490	Advice of Charges, Interest and Other Adjustments
MT 491	Request for Payment of Charges, Interest and Other Expenses
MT 492	Request for Cancellation
MT 495	Queries
MT 496	Answers
MT 498	Proprietary Message
MT 499	Free Format Message

### 3.2.5 Category 5 Messages

Table 5 explains the Category 5 message types, Securities Markets, with the type designation MT 5xx.

**Table 5** Securities Markets

<b>SWIFT Message Type</b>	<b>Description</b>
MT 500	Instruction to Register
MT 501	Confirmation of Registration or Modification
MT 502	Order to Buy or Sell
MT 503	Collateral Claim
MT 504	Collateral Proposal
MT 505	Collateral Substitution
MT 506	Collateral and Exposure Statement
MT 507	Collateral Status and Processing Advice
MT 508	Intra-Position Advice
MT 509	Trade Status Message
MT 510	Registration Status and Processing Advice
MT 513	Client Advice of Execution
MT 514	Trade Allocation Instruction
MT 515	Client Confirmation of Purchase or Sale
MT 516	Securities Loan Confirmation
MT 517	Trade Confirmation Affirmation
MT 518	Market-Side Securities Trade Confirmation
MT 519	Modification of Client Details
MT 524	Intra-Position Instruction
MT 526	General Securities Lending/Borrowing Message
MT 527	Triparty Collateral Instruction
MT 528	ETC Client-Side Settlement Instruction
MT 529	ETC Market-Side Settlement Instruction
MT 535	Statement of Holdings
MT 536	Statement of Transactions
MT 537	Statement of Pending Transactions
MT 538	Statement of Intra-Position Advice
MT 540	Receive Free
MT 541	Receive Against Payment Instruction
MT 542	Deliver Free
MT 543	Deliver Against Payment Instruction

**Table 5** Securities Markets (Continued)

<b>SWIFT Message Type</b>	<b>Description</b>
MT 544	Receive Free Confirmation
MT 545	Receive Against Payment Confirmation
MT 546	Deliver Free Confirmation
MT 547	Deliver Against Payment Confirmation
MT 548	Settlement Status and Processing Advice
MT 549	Request for Statement/Status Advice
MT 558	Triparty Collateral Status and Processing Advice
MT 559	Paying Agent's Claim
MT 564	Corporate Action Notification
MT 565	Corporate Action Instruction
MT 566	Corporate Action Confirmation
MT 567	Corporate Action Status and Processing Advice
MT 568	Corporate Action Narrative
MT 569	Triparty Collateral and Exposure Statement
MT 574 (IRSLST)	IRS 1441 NRA Beneficial Owners' List
MT 574 (W8BENO)	IRS 1441 NRA Beneficial Owner Withholding Statement
MT 575	Statement of Combined Activity
MT 576	Statement of Open Orders
MT 577	Statement of Numbers
MT 578	Statement of Allegement
MT 579	Certificate Numbers
MT 581	Collateral Adjustment Message
MT 582	Reimbursement Claim or Advice
MT 584	Statement of ETC Pending Trades
MT 586	Statement of Settlement Allegements
MT 587	Depository Receipt Instruction
MT 588	Depository Receipt Confirmation
MT 589	Depository Receipt Status and Processing Advice
MT 590	Advice of Charges, Interest and Other Adjustments
MT 591	Request for Payment of Charges, Interest and Other Expenses
MT 592	Request for Cancellation
MT 595	Queries
MT 596	Answers

**Table 5** Securities Markets (Continued)

<b>SWIFT Message Type</b>	<b>Description</b>
MT 598	Proprietary Message
MT 599	Free Format Message

### 3.2.6 Category 6 Messages

Table 6 explains the Category 6 message types, Treasury Markets, Precious Metals, with the type designation MT 6xx.

**Table 6** Treasury Markets, Precious Metals

<b>SWIFT Message Type</b>	<b>Description</b>
MT 600	Precious Metal Trade Confirmation
MT 601	Precious Metal Option Confirmation
MT 604	Precious Metal Transfer/Delivery Order
MT 605	Precious Metal Notice to Receive
MT 606	Precious Metal Debit Advice
MT 607	Precious Metal Credit Advice
MT 608	Statement of a Metal Account
MT 609	Statement of Metal Contracts
MT 643	Notice of Drawdown/Renewal
MT 644	Advice of Rate and Amount Fixing
MT 645	Notice of Fee Due
MT 646	Payment of Principal and/or Interest
MT 649	General Syndicated Facility Message
MT 690	Advice of Charges, Interest and Other Adjustments
MT 691	Request for Payment of Charges, Interest and Other Expenses
MT 692	Request for Cancellation
MT 695	Queries
MT 696	Answers
MT 698	Proprietary Message
MT 699	Free Format Message

### 3.2.7 Category 7 Messages

Table 7 explains the Category 7 message types, Treasury Markets, Syndication, with the type designation MT 7xx.

**Table 7** Treasury Markets, Syndication

<b>SWIFT Message Type</b>	<b>Description</b>
MT 700	Issue of a Documentary Credit
MT 701	Issue of a Documentary Credit
MT 705	Pre-Advice of a Documentary Credit
MT 707	Amendment to a Documentary Credit
MT 710	Advice of a Third Bank's Documentary Credit
MT 711	Advice of a Third Bank's Documentary Credit
MT 720	Transfer of a Documentary Credit
MT 721	Transfer of a Documentary Credit
MT 730	Acknowledgment
MT 732	Advice of Discharge
MT 734	Advice of Refusal
MT 740	Authorization to Reimburse
MT 742	Reimbursement Claim
MT 747	Amendment to an Authorization to Reimburse
MT 750	Advice of Discrepancy
MT 752	Authorization to Pay, Accept or Negotiate
MT 754	Advice of Payment/Acceptance/Negotiation
MT 756	Advice of Reimbursement or Payment
MT 760	Guarantee
MT 767	Guarantee Amendment
MT 768	Acknowledgment of a Guarantee Message
MT 769	Advice of Reduction or Release
MT 790	Advice of Charges, Interest and Other Adjustments
MT 791	Request for Payment of Charges, Interest and Other Expenses
MT 792	Request for Cancellation
MT 795	Queries
MT 796	Answers
MT 798	Proprietary Message
MT 799	Free Format Message

### 3.2.8 Category 8 Messages

Table 8 explains the Category 8 message types, Travellers Cheques, with the type designation MT 8xx.

**Table 8** Travellers Cheques

<b>SWIFT Message Type</b>	<b>Description</b>
MT 800	T/C Sales and Settlement Advice [Single]
MT 801	T/C Multiple Sales Advice
MT 802	T/C Settlement Advice
MT 810	T/C Refund Request
MT 812	T/C Refund Authorization
MT 813	T/C Refund Confirmation
MT 820	Request for T/C Stock
MT 821	T/C Inventory Addition
MT 822	Trust Receipt Acknowledgment
MT 823	T/C Inventory Transfer
MT 824	T/C Inventory Destruction/Cancellation Notice
MT 890	Advice of Charges, Interest and Other Adjustments
MT 891	Request for Payment of Charges, Interest and Other Expenses
MT 892	Request for Cancellation
MT 895	Queries
MT 896	Answers
MT 898	Proprietary Message
MT 899	Free Format Message



### 3.2.9 Category 9 Messages

Table 9 explains the Category 9 message types, Cash Management and Customer Status, with the type designation MT 9xx.

**Table 9** Cash Management and Customer Status

<b>SWIFT Message Type</b>	<b>Description</b>
MT 900	Confirmation of Debit
MT 910	Confirmation of Credit
MT 920	Request Message
MT 935	Rate Change Advice
MT 940	Customer Statement Message
MT 941	Balance Report
MT 942	Interim Transaction Report
MT 950	Statement Message
MT 960	Request for Service Initiation Message
MT 961	Initiation Response Message
MT 962	Key Service Message
MT 963	Key Acknowledgement Message
MT 964	Error Message
MT 965	Error in Key Service Message
MT 966	Discontinue Service Message
MT 967	Discontinuation Acknowledgement Message
MT 970	Netting Statement
MT 971	Netting Balance Report
MT 972	Netting Interim Statement
MT 973	Netting Request Message
MT 985	Status Inquiry
MT 986	Status Report
MT 990	Advice of Charges, Interest and Other Adjustments
MT 991	Request for Payment of Charges, Interest and Other Expenses
MT 992	Request for Cancellation
MT 995	Queries
MT 996	Answers
MT 998	Proprietary Message
MT 999	Free Format Message

### 3.2.10 Category n Messages

Table 10 explains the Category n message types, Common Group Messages, common to all message types, with the type designation MT n9x, where n equals 1 through 9.

**Table 10** Common Group Messages

<b>SWIFT Message Type</b>	<b>Description</b>
MT n90	Advice of Charges, Interest and Other Adjustments
MT n91	Request for Payment of Charges, Interest and Other Expenses
MT n92	Request for Cancellation
MT n93	Information Service Message
MT n95	Queries
MT n96	Answers
MT n98	Proprietary Message
MT n99	Free Format Message

# Using Message Validation Features

This chapter explains how to use specialized message validation features and Projects available with the SWIFT OTD Library.

## Chapter Topics

- [“Library Validation Features: Overview” on page 27](#)
- [“Message Format Validation Rules” on page 30](#)
- [“Market Practice Rules” on page 32](#)
- [“BICDirService Browser Lookup” on page 44](#)
- [“Using the Validation Project Examples” on page 35](#)
- [“Error Message Information” on page 50](#)

---

## 4.1 Library Validation Features: Overview

This section generally explains the validation features available with the SWIFT OTD Library and how they operate.

### 4.1.1 Basic Validation Features

The SWIFT OTD Library accomplishes validation operations via Java-based Collaboration Definitions packaged with the library. These Collaboration Definitions have the following validation features provided to enhance their use:

- **Message Format Validation Rules (MFVRs):** Set of functions that accurately test the semantic validity of a given subset of the SWIFT messages.
- **Market Practice Rules (MPRs):** Set of functions that accurately test the semantic and syntactical validity of a particular subset of the SWIFT messages called the 500 series.
- **BICDirService (Bank Identifier Code Directory Service) Lookup:** A set of methods that provide search and validation functionality for SWIFT’s BIC codes and ISO currency and country codes. The information used to look up and validate comes from SWIFT.

These validation features share the following use characteristics:

- Each available method and function is fully incorporated into and used by the appropriate SWIFT message OTD.
- You can modify the validation rules, if desired.
- Validation methods and functions have no dependencies outside SWIFT data files and the individual OTD.

Installing the OTD library allows eGate and any eWay you use with the library to provide full support for these features. The rest of this chapter provides a summary of how these features operate with the SWIFT OTD Library.

## Validation Components

In addition to components described under [“Basic Validation Features” on page 27](#), the SWIFT OTD Library also contains the following basic components:

- **SWIFT OTDs (2001, 2002, and 2003):** OTDs in the SWIFT OTD Library that represent standard SWIFT message types. See [Chapter 3](#) for details. The validation features are only available with the 2003 OTDs.
- **Funds OTDs:** Specialized OTDs that allow you to automate the specialized funds operations. This category contains both FIN- and XML-based OTDs.
- **Validation Collaboration Definitions:** Validation eGate components provided for each SWIFT message type. See the next section for details.
- **Sample Validation Projects:** Projects provided as examples of validation implementation. See [“Using the Validation Project Examples” on page 35](#) for details.

## Validation Collaboration Definitions

A validation Collaboration Definition is provided for each SWIFT message type. These Collaboration Definitions, when combined with eGate Services, become Java-based Collaborations that verify the syntax of the SWIFT messages.

This verification is done by parsing the data into a structure that conforms to the SWIFT standard specifications. The validation functions use these Collaborations to access specific data that is then verified according the algorithms of the MFVR or MPR specifications.

For lists of these Collaboration Definitions, see [“MFVR Validation Collaboration Definitions” on page 31](#) and [“MPR Validation Collaboration Definitions” on page 33](#).

## Validation Operation

You can combine the library’s validation features in any way desired, to meet your specific needs. The SWIFT OTD Library packages a prebuilt implementation that takes SWIFT messages from a file and validates them individually, then writes the results to a specified set of files. One set contains valid messages, and the other contains the invalid ones, along with messages indicating the errors generated.

## Validation Project examples

The validation Collaboration Definitions are part of the OTD Library and packaged with validation Project examples you can import into eGate.

### Basic validation steps

Each validation Collaboration Definition has only the applicable tests for a specific OTD/message type, but they all operate according to the same general format, as follows:

- The Service first tests a message to make sure it is syntactically correct, by parsing it into the OTD.
- If the message fails, the message and its parser error are sent to an error Queue. If the message is valid, all applicable MRVR or MPR functions are applied to the message.
- Any and all errors produced from these tests are accumulated, and the combined errors, as well as the message, are written to an error Queue for later processing. As long as no error is fatal, all applicable tests are applied.
- Again any and all errors produced from these tests are accumulated, and the combined errors and message are written to the error Queue for later processing.
- If no errors are found in a message, it is sent to a Queue for valid messages.

For an explanation of using these Collaboration Definitions and the validation Project examples, see [“Using the Validation Project Examples” on page 35](#).

## 4.1.2 Library Methods

The SWIFT OTD Library provides a set of run-time methods that allow you to manipulate OTD data in a variety of ways. The following methods are the most frequently used with validation operations:

- **set()**: Allows you to set data on a parent node using a byte array or a string as a parameter.
- **value()**: Lets you get the string value of data in a node at any tree level.
- **getLastSuccessInfo()**: Returns a string that represents information about the last node in the tree that was successfully parsed.
- **command()**: Allows you to pass flags as parameters, which set levels that determine the quantity of debug information you receive (see [“Debug Flags” on page 50](#) for details).
- **marshalToString()** and **unmarshalFromString()**: Returns string data from or accepts string data to a desired node.

In addition, the library has methods that allow you to perform basic but necessary operations with the OTDs. See Table 11.

**Table 11** Basic OTD Methods

Method	Description
<b>add()</b>	Adds a repetition to a given child node.
<b>append()</b>	Adds given data at the end of existing data.
<b>copy()</b>	Copies given data at a specified point.
<b>count()</b>	Gives the count of node repetitions.
<b>delete()</b>	Erases data at a specified point.
<b>get()</b>	Retrieves data from a node.
<b>has()</b>	Checks whether a specified child node is present.
<b>insert()</b>	Inserts given data at a specified point.
<b>length()</b>	Returns the length of data contained in an object.
<b>marshal()</b>	Serializes internal data into an output stream.
<b>remove()</b>	Removes a given child node repetition.
<b>reset()</b>	Clears out any data held by an OTD.
<b>size()</b>	Returns the current number of repetitions for the current child node.
<b>unmarshal()</b>	Parses given input data into an internal data tree.

To help in your use of the SWIFT OTD Library and its features, the library includes a **Javadoc**. You can see this document for complete details on all of these methods. See [Chapter 6](#) for more information on this document and how to use it.

---

## 4.2 Message Format Validation Rules

The MFVR support for the SWIFT OTD Library is a set of functions, collectively known as the message format validation rules methods. These functions accurately test the semantic validity of a given subset of the SWIFT messages. Validation is performed according to standards provided in SWIFT’s publication, the *Message Format Validation Rules Guide* (current version).

There is one validation method for each MFVR message type and its corresponding OTD. Each method is called on a particular OTD and is used to validate the data of a given instance of that message type. Because business practices vary greatly between organizations, you can modify any of these functions.

For examples of how the MFVR validation process works, you can import the sample validation Projects. For details, see [“Using the Validation Project Examples” on page 35](#).

## 4.2.1 MFVR Validation Rules

SWIFT's MFVR validation rules are known as semantic verification rules (SVRs) or semantic rules, as opposed to the syntactic rules, which verify the syntax of the fields only. Syntactic verification is built into each OTD.

SWIFT defines a total of 299 SVRs that are validated by the FIN network engine. SWIFT Alliance Access or IBM's Merva products do not implement these rules, mainly because there is no functional model, and the implementation work is mostly manual. Each message type has to be validated against a subset of these rules.

In addition this set of 299 SVRs, SWIFT has defined a new series of rules to help enable straight-through processing (STP) in the securities industry. The OTD methods that validate for MFVR compliance also validate for compliance with STP rules.

## 4.2.2 MFVR Validation Collaboration Definitions

MFVR Java-based Collaboration Definitions are created for the following message types and their corresponding OTDs in the library:

- MT 100: Customer Transfer
- MT 103: Single Customer Credit Transfer
- MT 199: Free Format Message
- MT 202: General Financial Institution Transfer
- MT 300: Foreign Exchange Confirmation
- MT 535: Statement of Holdings
- MT 536: Statement of Transactions
- MT 537: Statement of Pending Transactions
- MT 540: Receive Free
- MT 541: Receive Against Payment Instruction
- MT 543: Deliver Against Payment Instruction
- MT 545: Receive Against Payment Confirmation
- MT 547: Deliver Against Payment Confirmation
- MT 548: Settlement Status and Processing Advice
- MT 900: Confirmation of Debit
- MT 910: Confirmation of Credit
- MT 940: Customer Statement Message
- MT 950: Statement Message

See [Chapter 3](#) for a complete list of the OTDs/SWIFT message types in the library.

### 4.2.3 MFVR Validation Methods

The MFVR methods adhere to SWIFT's current *Message Format Validation Rules Guide*, including those in any updates section in the back of the manual. The methods implement all of the "special functions" as defined in the guide, which are required by the validation rules.

The SVR methods also implement the semantic validation "rules" or functionality used in the validation functions, as defined by the current *Message Format Validation Rules Guide*.

Using this semantic validation, eGate can verify the contents of each message before it is sent into the FIN network, saving time and usage fees.

### 4.2.4 MFVR Errors

These errors result from the application the Semantic Validation Rules. Multiple errors are possible, and they are given in the order in which they occurred and with the sequences, fields, or subfields used to determine them.

For example, an MFRVR failure on a 535 Collaboration OTD appears as follows:

```
MFVR MT535 Error
SVR Rule 103 - Error code: D031001 = Since field :94a:: is present
in Sequence B, then fields :93B::AGGR and :94a::SAFE are not
allowed in any occurrence of Subsequence B1a.mt_535.Mt_535.Data[1].
SubSafekeepingAccount mt_535.Mt_535.Data[1].SubSafekeepingAccount[0].
SubSeqB1[0].SubSeqB1a.Balance

SVR Rule 104 - Error code: D04-
1001 = Since field :93B::AGGR is present in Subsequence B1a, then
:field 94a::SAFE must be present in the same Subsequence B1a.
mt_535.Mt_535.Data[1].SubSafekeepingAccount[0].SubSeqB1[0].
SubSeqB1a.Balance
```

For more information on error messages, see ["Error Message Information" on page 50](#).

---

## 4.3 Market Practice Rules

The MPR support for the SWIFT OTD Library is a set of functions collectively known as the MPR methods. These methods accurately test the semantic and syntactical validity of the SWIFT messages' 500 series (according to SWIFT's current publication of the ISO 15022 *Market Practice Rules – Description and Pseudo Code, v1.1*).

The validations done by SWIFT are not sufficient for certain scenarios. The MPRs represent *best practices* to avoid creating messages that are invalid at the semantic level, as defined by the Securities Market Practice Group. This group is an organization of SWIFT members seeking to create best practices to enable straight-through processing, that is, the automatic handling of requests without human intervention.



An MPR-invalid message is *not* invalid on the network. This type of invalidation only means that the subject message does not adhere to these best practices for specifying information. Also, MPR-invalid messages may not be assimilated into the receiving system because certain information is not supplied and/or may not be in a machine-usable form. Members of the Securities Market Practice Group or anyone following its practices would reject messages that failed to meet these standards.

For an example of how the MPR validation process works, you can import the sample validation Projects. For details, see [“Using the Validation Project Examples” on page 35](#).

### 4.3.1 MPR Validation Collaboration Definitions

MPR Java-based Collaboration Definitions are created for the following message types and their corresponding OTDs in the library:

- MT 535: Statement of Holdings
- MT 536: Statement of Transactions
- MT 537: Statement of Pending Transactions
- MT 540: Receive Free
- MT 541: Receive Against Payment Instruction
- MT 542: Deliver Free
- MT 543: Deliver Against Payment Instruction
- MT 544: Receive Free Confirmation
- MT 545: Receive Against Payment Confirmation
- MT 546: Deliver Free Confirmation
- MT 547: Deliver Against Payment Confirmation
- MT 548: Settlement Status and Processing Advice

See [Chapter 3](#) for a complete list of the SWIFT message types in the OTD library.

### 4.3.2 MPR Validation Methods

The SWIFT OTD Library’s MPR methods and their validations are optional and can be taken out. However, these validations can help you to achieve STP with your trading partners and reduce excess labor and processing time, because you have fewer messages rejected.

The following MPRs are created as methods:

- S101, Specify Trade Date: Abbreviated as **S101-SpecifyTradeDate**.
- S103, Specify the Identification of the Financial Instruments with a valid ISIN and valid country code: Abbreviated as **S103-SpecifyFinancialInstrumentWithISINAndCountryCode**.
- S106, Specify the delivering or receiving agent with a BIC: Abbreviated as **S106-SpecifyDeliveringReceivingAgentWithBIC**.
- S108, Specify the client of the delivering agent (except if place of settlement is US or DK): Abbreviated as **S108-SpecifyClientOfDeliveringAgent**.
- S109, Specify the client of the receiving agent (except if place of settlement is US or DK): Abbreviated as **S109-SpecifyClientOfReceivingAgent**.
- S111, Specify place of settlement with a BIC: Abbreviated as **S111-SpecifyPlaceOfSettlementWithBIC**.
- S114, Do not specify any Financial Instrument Attributes if the ISIN is already specified: Abbreviated as **S114-DoNotSpecifyAnyFinancialInstrumentAttributesWithISIN**.
- S117, Specify related reference: Abbreviated as **S117-SpecifyRelatedReference**.
- S120, Specify correct Safekeeping Account: Abbreviated as **S120-SpecifySafekeepingAccount**.
- S124, Specify DEAG's safekeeping Account at the place of settlement: Abbreviated as **S124-SpecifyDEAGSafekeepingAccountAtSettlementPlace**.
- S128, Specify REAG safekeeping Account at the place of settlement: Abbreviated as **S128-SpecifyREAGSafekeepingAccountSettlementPlace**.
- S132: When the place of settlement is Italy, specify the deal price: Abbreviated as **S132-WhenSettlementInItalySpecifyDealPrice**.
- S134, Specify the safekeeping account of the beneficiary agent with Receiving agent: Abbreviated as **S134-SpecifySafekeepingAccountOfBeneficiaryAgentWithReceivingAgent**.

### 4.3.3 MPR Errors

MPR errors are the result of the application of specific MPRs. As with the MFVRs, these rules are cumulative and are accompanied by the paths to the data used in the rule.

The following examples show possible error messages resulting from the MPR rules for a 541 Collaboration OTD:

```
MPR S101 - Error code: 1002 = The receive against payment message
(MT-541) should specify trade date(field :98A).
    mt_541.Mt_541.Data.TradeDetails
    mt_541.Mt_541.Data.TradeDetails[0].DateTime
```

```
MPR S108 - Error code: 1005 = The receive against payment message
(MT-541) should specify the client of the delivering agent with Option
P, not with Option Q, if the place of settlement is not US or DK.
    mt_541.Mt_541.Data.SettlementDetails[0].SubSeqE1
    mt_541.Mt_541.Data.SettlementDetails[0].SubSeqE1[0].Party
    mt_541.Mt_541.Data.SettlementDetails[1].SubSeqE1
    mt_541.Mt_541.Data.SettlementDetails[1].SubSeqE1[0].Party
    mt_541.Mt_541.Data.SettlementDetails[1].SubSeqE1[0].Party.PartyP
    mt_541.Mt_541.Data.SettlementDetails[2].SubSeqE1
    mt_541.Mt_541.Data.SettlementDetails[2].SubSeqE1[0].Party
    mt_541.Mt_541.Data.SettlementDetails[3].SubSeqE1
    mt_541.Mt_541.Data.SettlementDetails[3].SubSeqE1[0].Party
    mt_541.Mt_541.Data.SettlementDetails[3].SubSeqE1[0].Party.PartyP
    mt_541.Mt_541.Data.SettlementDetails[0].SubSeqE1
    mt_541.Mt_541.Data.SettlementDetails[0].SubSeqE1[0].Party
    mt_541.Mt_541.Data.SettlementDetails[0].SubSeqE1[0].Party.PartyQ
```

For more information on error messages, see [“Error Message Information” on page 50](#).

---

## 4.4 Using the Validation Project Examples

Two sample Projects are packaged with the SWIFT OTD Library, one with Java-based Collaboration Services and the other with an eInsight Business Process Service. This section introduces you to both validation Project examples. The operation of the Business Process example is functionally similar to the SWIFT Gold Validation Services.

These Projects are packaged as samples and provided with the library. They demonstrate both the MFVR and MPR validation operations.

### 4.4.1 Projects and the Enterprise Designer

A Project contains all of the eGate components you designate to perform one or more desired processes in eGate. Each eGate Project is created using the Enterprise Designer’s Project canvas.

The Project canvas contains windows that represent the various stages of your Project. The types of windows in your Project canvas area include:

- **Connectivity Map Canvas:** Contains the eGate business logic components, such as Collaborations, Topics, Queues, and eWays, that you include in the structure of the Project.
- **OTD Editor:** Contains the source files used to create Object Type Definitions (OTDs) to use with a Project.
- **Business Process Canvas:** Allows you to use eInsight's Business Process management features.
- **Collaboration Editor (Java):** Allows you to create and/or modify Business Rules to implement the business logic of a Project's Java-enabled Collaboration Definition.

## 4.4.2 Importing Sample Projects

Before you can view or work with a sample Project, you must first import it into eGate, using the Enterprise Designer.

**Note:** *The sample .zip file you first download may contain more than one Project and/or additional files. If this is the case, you must unzip this file first, find the desired Project file, then import the Project file. For details on how to download this file, see the [SeeBeyond ICAN Suite Installation Guide](#).*

The container file you are looking for is **SWIFT\_OTD\_Library\_Sample.zip**. The Project file names are:

- **SwiftGoldCertificationProject.zip:** Collaboration (Java) Project that enables SWIFT Gold Services.
- **SampleBPELProject.zip:** eInsight Project that enables Business Process Services.
- **SwiftBICDirLookupProject.zip:** Specialized Project that enables the BICDirService feature (see ["BICDirService Browser Lookup" on page 44](#) for details).

You can name the imported Projects as desired.

### To import a sample Project

- 1 Save any changes not saved previously.
- 2 From the Enterprise Designer's **Project Explorer** pane, right-click the desired Repository and select **Import**.
- 3 On the **Import Manager** window, browse to the directory that contains the sample Project .zip file.
- 4 Select the sample Project file and click **Open**.
- 5 Click **Import**. If the import was successful, click **OK** on the **Import Status** dialog box.
- 6 Close the **Import Manager** window.

**Important:** An imported Project does not contain an environment or a deployment profile. After importing a Project, you must use the Enterprise Designer to create these functions for the Project. See [“Creating the Project’s Environment” on page 43](#) and [“Deploying a Project” on page 43](#). For additional information, see the *eGate Integrator User’s Guide* and *SeeBeyond ICAN Suite Deployment Guide*.

You must check out the major eGate components before you can change them. For details, see the *eGate Integrator Tutorial*.

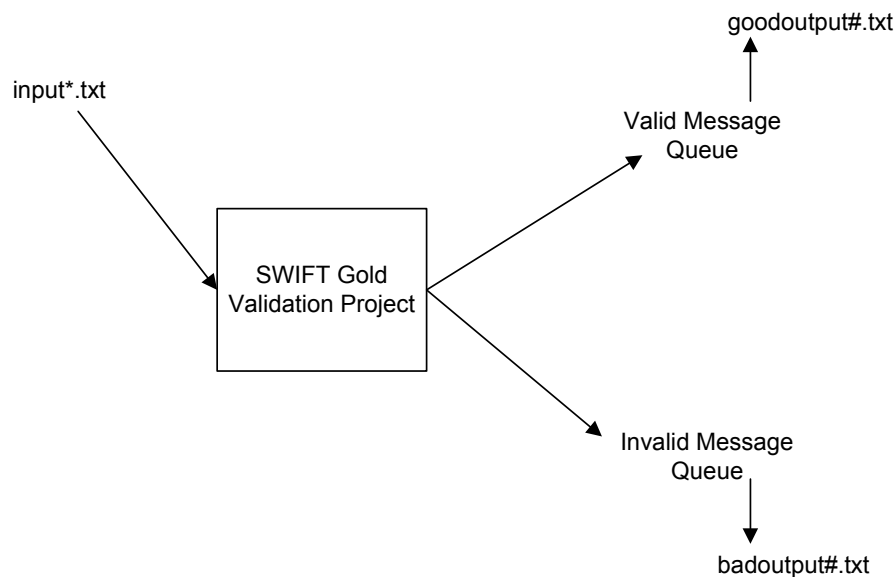
### 4.4.3 SWIFT Gold Validation Project

The SWIFT Gold validation Project demonstrates the validation features of the SWIFT OTD Library. Specifically, this Project employs the Java-based validation Collaborations and their Definitions.

This Project is SWIFT Gold-certified and represents a possible validation process within an institution's network. For example, the Project could be getting SWIFT messages from other systems and/or from Java-based Collaborations within eGate.

The Project uses a common process infrastructure to identify and isolate invalid messages. The process keeps these messages readily available for further use. It also passes valid messages on to their destinations. Figure 4 shows how the Project performs validations.

**Figure 4** SWIFT Gold Validation Project Operation



Inbound messages for validation enter the Project in the format `input*.txt`, with “\*” being the message name. Outbound messages go into two separate locations and have the file names `goodoutput#.txt` and `badoutput#.txt`, with “#” being a number eGate attaches to the file name to distinguish among the different messages.

Error messages are appended to the end of each of the **badoutput#.txt** files. You must name the source and destination file directories in setting properties for the File eWays you use with the Project. See the *File eWay Intelligent Adapter User's Guide* for details.

Also, see “**Validation Operation**” on page 28 for a more detailed explanation of the validation operation.

#### 4.4.4 eInsight Validation Project

Also packaged with the SWIFT OTD Library is a sample Project template using an eInsight Business Process Service instead of Java-based Collaboration Services. Before using this Project, you must first import it into eGate. See “**Importing Sample Projects**” on page 36 for details on how to import a Project.

*Note:* You must have the *eInsight.sar* file installed to use the features available with this Project. See the *SeeBeyond ICAN Suite Installation Guide* for complete installation procedures.

This Project employs simple unmarshal and marshal operations with the library OTDs. You can modify this Project to meet your own needs, as desired. The Project is packaged with the following sample input data file:

**inputBPELmt541.~in**

#### 4.4.5 Using eGate With eInsight

You can set up and deploy an eGate component using eInsight. Once you have associated the desired component (a Service in this Project) with a Business Process, the eInsight engine can automatically invoke that component during run time, using a Business Process Execution Language (BPEL) interface.

### Using eInsight With eGate Components

Examples of eGate components that can interface with eInsight in this way are:

- Java Messaging Service (JMS)
- Object Type Definitions (OTDs)
- An eWay
- eGate Services

Using the eGate Enterprise Designer and its eInsight canvas, you can add a desired operation to a Business Process, then associate that process with an eGate component, for example, a Service. In the Enterprise Designer, associate the Business Process and Service icons using drag-and-drop operations.

See the *eInsight Business Process Manager User's Guide* for details.

## SWIFT OTD Library With eInsight

You can add SWIFT OTD Library objects to an eInsight Business Process during the system design phase. To make this association, select the desired operation, for example **Marshal** or **Unmarshal**, under the OTD in the Enterprise Explorer and drag it onto the eInsight Business Process canvas. In turn, you can activate a Business Process in eGate by dragging it onto a Service or onto the Business Process canvas.

At run time, the eInsight Engine is able to invoke each of the steps in order as set up in the Business Process. Using the engine’s BPEL interface, eInsight in turn invokes the SWIFT OTD Library operations, as well as any eWays in the current Project.

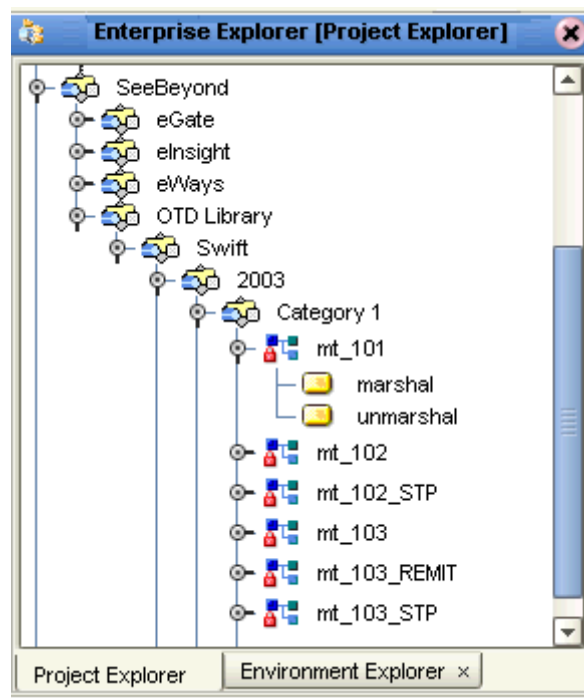
Table 12 shows the eInsight Business Process operations available to the SWIFT OTD Library, as well as a description of these operations.

**Table 12** Available eInsight SWIFT OTD Business Process Operations

eInsight Business Process Operation	Description
Unmarshal	Parses the SWIFT message/OTD for validation.
Marshal	Readies the SWIFT message for writing, along with errors if any are found.

Figure 5 shows the Enterprise Designer’s **Project Explorer** with the SWIFT OTD Library Business Process operations exposed under the OTD icon.

**Figure 5** Project Explorer With Business Process Operations

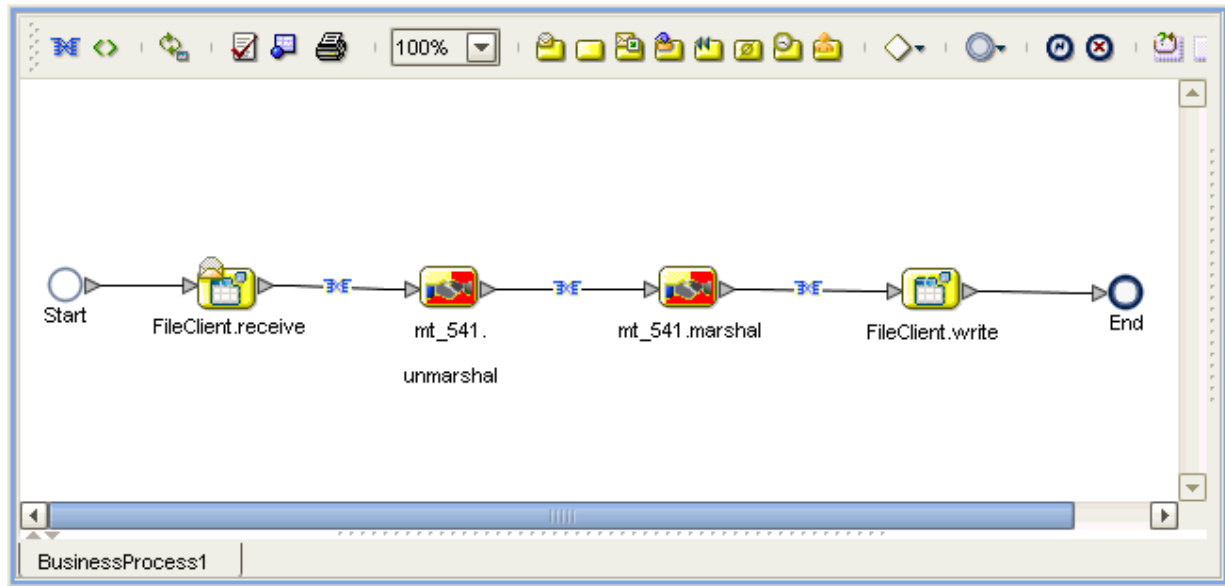


### 4.4.6 Using a Business Process

Once you have designed your Business Process for this sample, you can use the eGate Enterprise Designer to create it. See [“Validation Operation” on page 28](#) for a description of the sample’s Business Process.

Figure 6 shows how the Business Process operations appear on the sample Project’s Business Process canvas.

**Figure 6** Sample Project Business Process



### 4.4.7 Configuring the Modeling Elements

Business Rules are defined and configured between the Business Process Activities located on the modeling canvas. The sample Project contains the Business Rules that govern each of the Activities listed in a Business Process flow.

Each of the small blue icons placed on an arrow represents a Business Rule. The Business Rules found in the sample Project include:

- [“Copying the Output File” on page 41](#)
- [“Unmarshaling and Marshaling the Data” on page 42](#)
- [“Returning the Value” on page 43](#)

Double-clicking on one of these icons opens the Business Rule Designer pane.

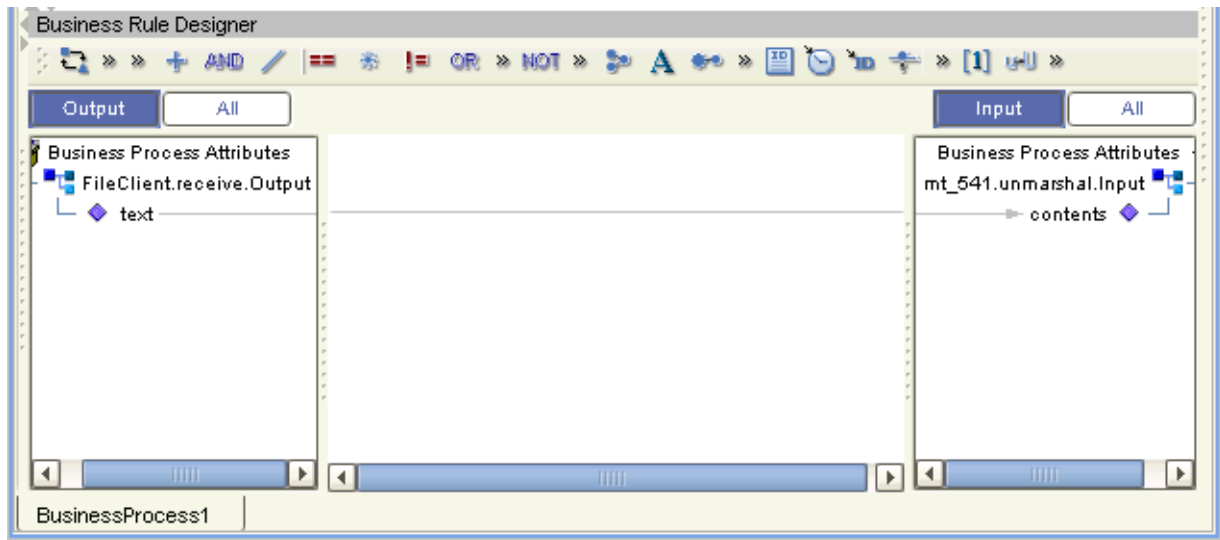
**Note:** A detailed description of the steps required to configure modeling elements is found in the *eInsight Business Process Manager User’s Guide*.



## Copying the Output File

The **FileClient.receive.Output** container copies the output file containing the message to be used. The Business Process copies the message content to an input container, **mt\_541.unmarshal**, to be unmarshaled. See Figure 7.

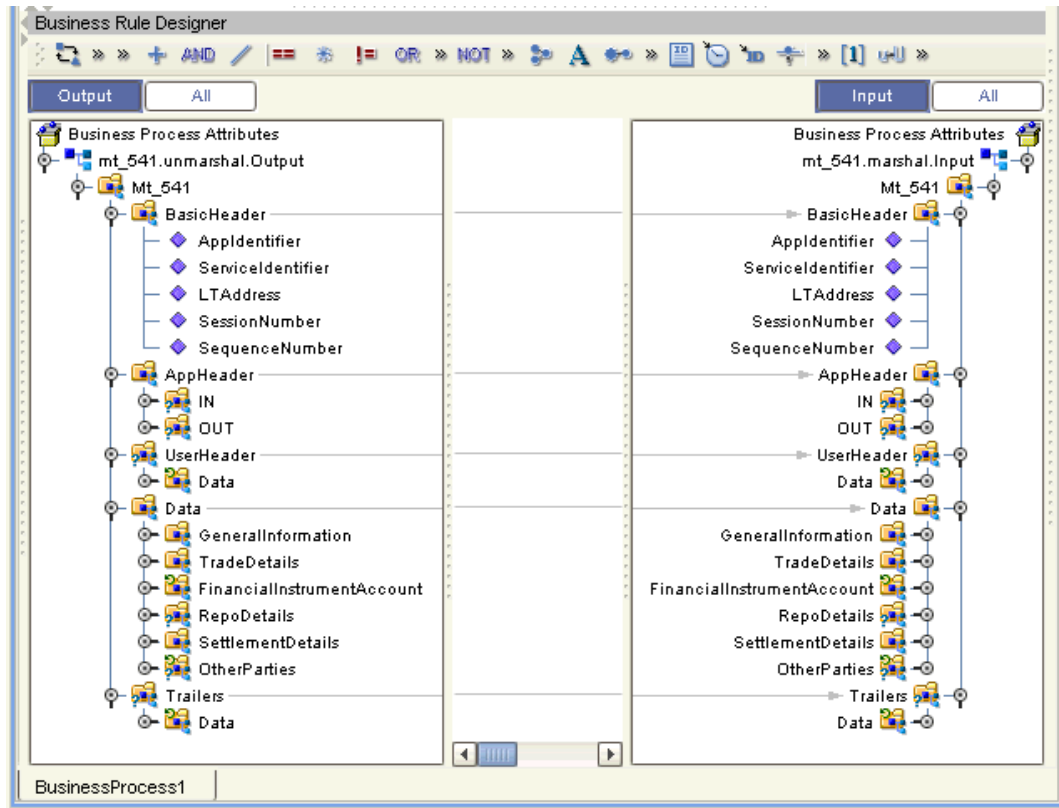
**Figure 7** Copying the Output File



## Unmarshaling and Marshaling the Data

The Business Process unmarshals the data and marshals the data, using the **mt\_541.unmarshal** and **mt\_541.marshal** operations. The Business Process then writes the results to the **FileClient.write.Output** container. See Figure 8.

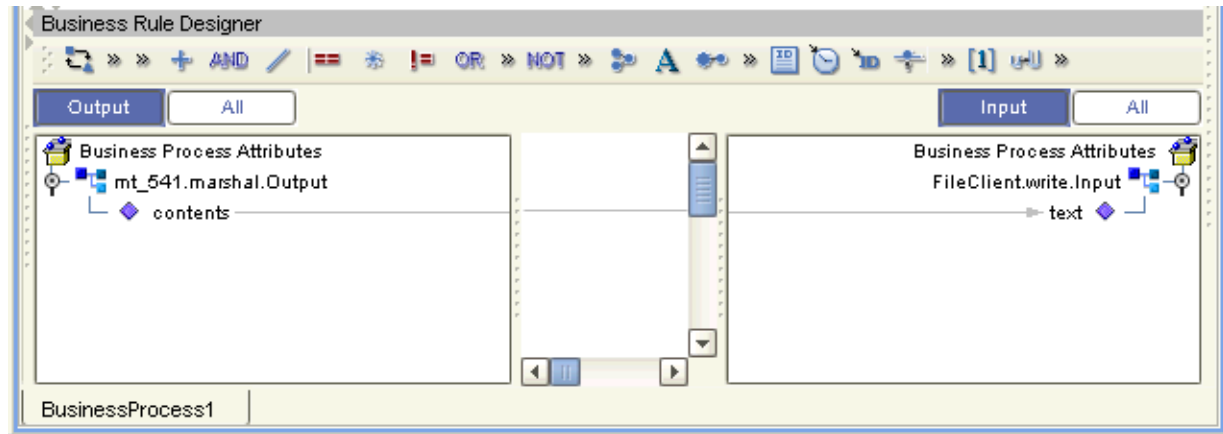
**Figure 8** Unmarshaling and Marshaling the Data



## Returning the Value

The OTD output container writes the resulting value to a text file using the `FileClient.write.Input` container. See Figure 9.

**Figure 9** Returning the Requested Value



### 4.4.8 Creating the Project's Environment

This section provides general procedures for creating an Environment for your Project after you have imported it into eGate. For a complete explanation, see the *eGate Integrator Tutorial*.

To create an Environment

- 1 From the Enterprise Designer, click the **Environment Explorer** tab on the Enterprise Explorer.
- 2 Under the current **Repository** icon in the **Environment Explorer**, create a new environment for your Project and name it as desired.
- 3 In the **Environment Explorer**, right-click the **Environment** icon and select the desired external systems from the pop-up menu. Give them the same names as you did the corresponding external applications on the Connectivity Map.
- 4 Use the same pop-up menu to create a Logical Host for your Project, and name it as desired.
- 5 Click **Save** and return to the **Project Explorer** tab.

### 4.4.9 Deploying a Project

This section provides general procedures for Project deployment after you have created a Project's environment.

## Basic Steps

For a complete explanation of how to deploy and run an eGate Project, see the *eGate Integrator Tutorial*.

### To deploy the Project

- 1 From the **Project Explorer**, select the current Project and right-click, choosing **New > Deployment Profile** from the pop-up menus.
- 2 From the **Create a Deployment Profile** dialog box, enter the name of the current Project and select the Environment you created for this Project.
- 3 Click **OK**.

The Deployment Profile canvas appears as follows:

- ♦ The Project's external applications and Services show up as icons on the left side of the canvas.
  - ♦ The external systems and Logical Host you created under "**Creating the Project's Environment**" on page 43 show up as windows on the right side of the canvas.
- 4 Set up your Deployment Profile by dragging the icons on the left into the corresponding window on the right.
  - 5 Click **Save All** then **Activate**.

When the Project has been activated, a pop-up message appears stating the activation was successful.

For additional information, see the *eGate Integrator User's Guide* and *SeeBeyond ICAN Suite Deployment Guide*.

### To run the Project

For instructions on how to run a Project, see the *eGate Integrator Tutorial*.

## Alerting and Logging

eGate provides an alerting and logging feature. This feature allows monitoring of messages and captures any adverse messages in order of severity based on configured severity level and higher. To enable Logging, see the *eGate Integrator User's Guide*.

---

## 4.5 BICDirService Browser Lookup

The BICDirService browser lookup support for the SWIFT OTD Library is based on a set of methods combined into a service. This service provides Internet browser lookup and validation functionality for SWIFT's BIC codes, as well as ISO currency and country codes.

This feature allows you to either check or look up any of these codes while eGate creates and/or converts SWIFT messages. These check and lookup operations are carried out by the eGate Java-based validation Collaboration Definitions.

With SeeBeyond's ICAN Suite eVision Studio, in conjunction with eInsight, you can also use the BICDirService feature via a browser-based lookup window. You must install **eVision.sar** and **eInsight.sar** with eGate in order to use this feature.

### 4.5.1 Updating BICDirService

The BICDirService feature is a database service. The data files used to populate BICDirService must be updated periodically from SWIFT's source CD-ROM issued once every four months.

#### Live Updating

An update method permits the live updating of the search structure within the class, from new SWIFT data files. Once this method has been invoked, all currently in-process lookups and validations complete, but no new ones are allowed to start until the update is done. During an update, pending lookups and validations block any additional operations until that update is finished.

Since the number of changes from version to version of the SWIFT data files is relatively low, this method considers the modification flag by selectively updating only those records in which the modification flag is either a "A" (addition), "M" (modification), or "D" (deletion).

For more information on BICDirService methods, see "[BICDirService Method Operation](#)" on page 46.

#### Source of Information

The Java constructor for the **BICDir** class loads the required data from the following SWIFT-supplied files:

- **FI.dat**
- **CU.dat**

The constructor takes an argument from the directory that contains these two files. It then opens each file and loads the appropriate fields into a searchable structure. For more details on these files, see the SWIFT document *BIC Directory, CD-ROM Format Technical Specifications, 9757* (current version) for actual file layout and positioning information.

The data used to look up and validate comes from SWIFT's own BIC bank files containing its BIC codes and its currency and country codes. When necessary, SWIFT updates these files with a new version of its lookup tables, to keep them current. You can upload these files to eGate and control when updates to the system occur and access these files via SWIFT's update CD-ROM.

#### Update Operation

The BICDirService feature allows multiple simultaneous objects to access its methods with near-local object response times. Besides having these files loaded at run-time, you can also update the "live" service whenever new update files arrive.

The SWIFT standards are not always sufficiently complete to enable STP. Currently a message can pass network validation but fail at the receiving end because of incompatible definitions or codes, or missing data. The result is expense to manually repair or follow up on these messages and possible retransmission of the message.

The SWIFT OTD Library's BICDirService ensures that valid, up-to-date BIC, country, and currency codes are present in eGate-processed messages. This feature increases the likelihood that a given message can flow "straight through".

You must update the BICDirService information before running components that might use the feature, for example, the validation Collaboration Definitions and the BICDirService lookup Project.

#### To update BIC information

- 1 Go to the **SwiftOTDLibrary.jar** file in the Enterprise Designer's **Project Explorer**. The file is located under **SeeBeyond/OTD Library/Swift**.
- 2 Right-click the **SwiftOTDLibrary.jar** icon (see **Figure 3 on page 13**).
- 3 Select the **Update BIC Files** option from the pop-up menu.
- 4 In the resulting **Open** dialog box, navigate to the location of the **CU.dat** file on the SWIFT update CD-ROM.
- 5 Select the file and upload it.
- 6 Select **Update BIC Files** again.
- 7 Navigate to the location of the **FI.dat** file.
- 8 Select the file and upload it.
- 9 When you are finished updating, you must redeploy and reactivate the BICDirService Lookup Project. See **"Deploying a Project" on page 43** for details.

This procedure updates the BICDirService feature.

### 4.5.2 BICDirService Wildcard Support

You can add an asterisk (\*) at the end of an **Institution** name, for institutions with very long names. For example, if a you enter **BRE BANK S.A.**, no results are returned. However, if you add "\*" after the name, valid results are returned.

You need to enter enough of the name so that you get results for the particular institution you are looking for. For example, if you just enter **Bank\*** you get results for every institution starting with the word **Bank**.

### 4.5.3 BICDirService Method Operation

The BICDirService methods are static methods of a single Java class, the **BICDir** class. There is one method per each required lookup and validation. In certain circumstances, the method returns multiple results as a byte array.

The **BICDir** methods are not dependent on any module other than SWIFT data files, eGate Collaborations, and eVision. Also, the **BICDir** methods are able to run outside of the eGate framework, as well as inside.

## Lookup Method Definitions

The **BICDir** class has the following lookup methods:

### Look up BIC by Institution Name

Takes a string and returns a byte array of BICs (one element is possible). The signature is:

```
BIC[] getBIC(institutionName*);
```

### Look up BIC by Institution Name, City and Country

Takes three strings, an institution name, city, and country, and returns a byte array of BICs (one element is possible). The signature is:

```
BIC[] getBIC(institutionName*, city*, country*);
```

### Look up Institution Name by BIC

Takes a BIC string, either a BIC 8 or BIC11, and returns a byte array of institution names (one element is possible). The signature is:

```
institutionName[] getInstitutionName(BIC);
```

### Look up Currency Code by Country Code

Takes a string, a country code, and returns the currency code. The signature is:

```
currencyCode getCurrencyCode(countryCode);
```

### Look up Country Code by Currency Code

Takes a string, a currency code, and returns the country code. The signature is:

```
countryCode getCountryCode(currencyCode);
```

## Validation Method Definitions

The **BICDir** class has the following validation methods:

### Validate BIC

Takes a string, either a BIC 8 or BIC11, and returns **true** or **false**. The signature is:

```
boolean validateBIC(BIC);
```

### Validate Currency Code

Takes a string, a currency code, and returns **true** or **false**. The signature is:

```
boolean validateCurrencyCode(currencyCode);
```

### Validate Country Code

Takes a string, a country code, and returns **true** or **false**. The signature is:

```
boolean validateCountryCode(countryCode);
```

## BICDir Exceptions

The purpose of the exceptions is to give you some indication of what error has occurred and how to rectify it.

### Error message framework

These error messages are implemented using the **log4j** framework. **STC.OTD.SWIFT.BICDirService** is used as the logging category.

### Error message general form

The message of **BICDir** exception takes the following general form:

**"BICDirService Error ["XX"] –" *error-message***

Where:

- **""**: Marks static text.
- **XX**: Stands for a unique number assigned to each error message.
- ***error-message***: A descriptive narrative derived from the condition that caused the error, and a possible solution to rectify it.

## 4.5.4 BICDirService Sample Project

Before using the BICDirService lookup feature, you *must* import the Project sample template file **SwiftBICDirLookupProject.zip** (see [“Importing Sample Projects” on page 36](#)). This Project enables the BICDirService and must be running before you can use the feature.

## 4.5.5 Using the BICDirService Lookup Feature

This section explains how to use the lookup feature with the ICAN Suite’s eVision (along with eInsight), after you have imported the BICDirService sample Project template.

### To use the BICDirService lookup feature

- 1 Create an Environment for the sample Project (see [“Creating the Project’s Environment” on page 43](#)).
- 2 Create a Deployment for the sample Project (see [“Deploying a Project” on page 43](#)).
- 3 Activate the Project.
- 4 Upon successful activation, you receive the following message:

**eVision application url: http://localhost:18003/swift**

- 5 Copy and paste this message into a text file for later use.
- 6 Download and install the Logical Host, if you have not done so already. For instructions on how to do this operation, see the *SeeBeyond ICAN Suite Installation Guide*.
- 7 Run the Logical Host.

Running a Project’s Logical Host runs the Project. For instructions on how to run a Logical Host, see the *eGate Integrator Tutorial*.

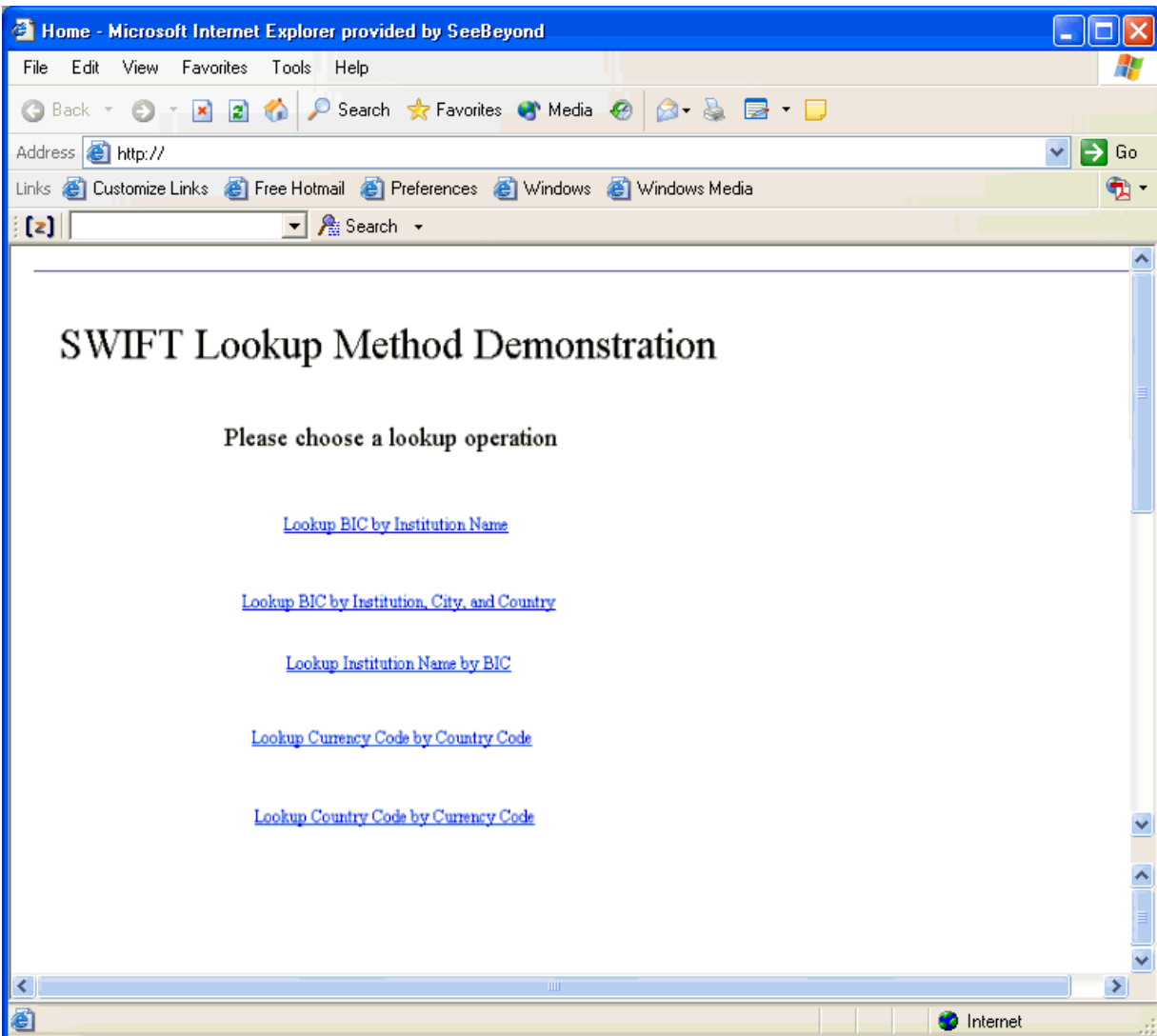


- 8 When the Logical Host screen displays that the Environment is ready, open a browser and enter the URL listed previously, from your text file.
- 9 Press **Enter** from your browser to display the BICDirService window.

Click one of the operations listed on the startup page to begin using that operation.

The BICDirService lookup startup window appears in the browser (see [Figure 10 on page 49](#)). To proceed, you can follow the prompts and instructions shown in your browser. Your browser interface allows you to use the SWIFT lookup methods to access the desired information.

**Figure 10** BICDirService Lookup Startup Window



Once you are finished with an operation, you must click the **Home** link to return to the startup page before doing another operation. You cannot back up.

---

## 4.6 Error Message Information

This section explains the SWIFT OTD Library validation error files and messages.

### 4.6.1 Error Messages

There are separate error messages and reporting mechanisms for each type of validation performed by a Service. The library's error parser provides the following debug levels:

- **Regular Information:** Gives general information, and if an error occurs, the path to the node or piece of data that caused the error.
- **Debug:** Gives all of the node information generated for the parse, that is, each field and subfield.
- **Parser Debug:** Combines the debug level with information about what the parser is matching with and the data being used. In general, you only need to use this level for situations where the error cannot be determined using the other levels because of the quantity of data. However, this level does give the exact location and nature of the failure.

These messages appear at the end of each message with an error, with the error file output.

### Format of Error Messages

Each message has the following general format:

- List of parser errors
- Failed MFVR list
- Failed MPR list
- Actual message

### Debug Flags

You can control the amount of debugging information you receive by using the debug flags as parameters when you call the **command()** method. The the library's **Javadoc** for more information on using this method.

If the flag is passed as a parameter, the flag's function is ON. If the parameter is not present, the flag's function is OFF. The debug flags with explanations are:

- **A or a**: Turns the abbreviation of path names on or off. This flag reduces the path output when you are printing to a Regular Information set.
- **D or d**: Turns Debug (mid-level) debugging on or off. If switched on, this flag generates more debug data than the Regular Information level, but less than the Parser Debug level.
- **I or i**: Turns Regular Information level debugging on or off.
- **L or l**: Turns the saving and display of the last successfully parsed node on or off. When a parse has failed, this flag's information is the last item printed by the current root node.
- **P or p**: Turns the Parser Debug-level information on or off. If switched on, this flag generates the maximum information about what the internal parser is doing.

## 4.6.2 Message Examples

An example of a regular information-level parse error (cannot find a required field) is:

```
at 0: com.stc.swift.runtime.SwiftUnmarshalException: mt_103.Mt_103:
0: Failed to parse required child(Data).
```

An example of a parse error with the debug level enabled (cannot find a required field) is:

```
at 146: null: com.stc.swift.runtime.SwiftUnmarshalException:
mt_543.Mt_543.Data.GeneralInformation.FunctionOfTheMessage: 146:
Failed to parse required child(String2).
```

Given this path to the data, you can determine where in the message the parser failed by looking at:

- The *SWIFT User Handbook*
- Structure of the OTD in the Enterprise Designer's OTD Editor
- **Javadoc** for the OTD

See [“MFVR Errors” on page 32](#) and [“MPR Errors” on page 34](#) for MFVR- and MPR-specific error information. For more detailed error information, see [“Error Message Information” on page 50](#).

## Parse Debug Level Message Example

The following example shows error message output at the parse debug level:

```
[main] PARSE - Swift: matchDelimSkip("{1:}") --> true.
[main] PARSE - Swift: getData("F|A|L") --> "F".
[main] DEBUG - Swift: mt_502.Mt_502.BasicHeader.AppIdentifier: 3:
Mapped data("F").
[main] DEBUG - Swift: mt_502.Mt_502.BasicHeader.AppIdentifier: 3:
Mapped rep[0].
[main] PARSE - Swift: getData(charSet, 2, 2) --> "01".
```

```
[main] DEBUG - Swift: mt_502.Mt_502.BasicHeader.ServiceIdentifier: 4:
The following is the last field successfully parsed the 4th 22a:
[main] PARSE - Swift: matchDelimSkip("22H::") --> true.
[main] PARSE - Swift: getData(charSet, 4, 4) --> "PAYM".
[main] DEBUG - Swift:
mt_502.Mt_502.Data.OrderDetails.Indicator.IndicatorH.String3: 218:
Mapped data("PAYM").
[main] PARSE - Swift: matchDelimSkip("/") --> true.
[main] DEBUG - Swift:
mt_502.Mt_502.Data.OrderDetails.Indicator.IndicatorH.String3: 218:
Mapped rep[0].
[main] PARSE - Swift: getData(charSet, 4, 4) --> "APMT".
[main] DEBUG - Swift:
mt_502.Mt_502.Data.OrderDetails.Indicator.IndicatorH.String5: 224:
Mapped data("APMT").
[main] DEBUG - Swift:
mt_502.Mt_502.Data.OrderDetails.Indicator.IndicatorH.String5: 224:
Mapped rep[0].
[main] PARSE - Swift: matchDelimSkip("
:") --> true.
[main] DEBUG - Swift:
mt_502.Mt_502.Data.OrderDetails.Indicator.IndicatorH: 213: Mapped
rep[0].
```

The message goes on for several more lines, not indicating any error. Then the parser is looking for any more 22a's, F or H, and does not find one. See the following example:

```
[main] DEBUG - Swift: mt_502.Mt_502.Data.OrderDetails.Indicator[3]:
159: Mapped rep[3].
[main] PARSE - Swift: matchDelimSkip("22F::") --> false.
[main] PARSE - Swift:
mt_502.Mt_502.Data.OrderDetails.Indicator.IndicatorF: 231: Failed to
find BeginDelimiter("22F::").
[main] PARSE - Swift: matchDelimSkip("22H::") --> false.
[main] PARSE - Swift:
mt_502.Mt_502.Data.OrderDetails.Indicator.IndicatorH: 231: Failed to
find BeginDelimiter("22H::").
```

The parser then looks for a 98a either option A | B | C as follows:

```
[main] PARSE - Swift: matchDelimSkip("98A::") --> false.
[main] PARSE - Swift:
mt_502.Mt_502.Data.OrderDetails.DateTime[0].DateTimeA: 231: Failed to
find BeginDelimiter("98A::").
[main] PARSE - Swift: matchDelimSkip("98B::") --> false.
[main] PARSE - Swift:
mt_502.Mt_502.Data.OrderDetails.DateTime[0].DateTimeB: 231: Failed to
find BeginDelimiter("98B::").
[main] PARSE - Swift: matchDelimSkip("98C::") --> false.
[main] PARSE - Swift:
mt_502.Mt_502.Data.OrderDetails.DateTime[0].DateTimeC: 231: Failed to
find BeginDelimiter("98C::").
```

The parser finds no repetitions, which does not fit in the required range of 1 to 3 as described in the following example, so at this point, the parser fails, because no expected repetitions were found:

```
[main] PARSE - Swift: mt_502.Mt_502.Data.OrderDetails: 231: Failed to
match minimum repetitions[ 1 < 0 <= 3 ].

[main] PARSE - Swift: mt_502.Mt_502.Data.OrderDetails: 145: Failed to
parse required child(DateTime).
[main] PARSE - Swift: mt_502.Mt_502.Data: 145: Failed to match
minimum repetitions[ 1 < 0 <= 1 ].
[main] PARSE - Swift: mt_502.Mt_502.Data: 73: Failed to parse
required child(OrderDetails).
[main] PARSE - Swift: mt_502.Mt_502: 67: Failed to match minimum
repetitions[ 1 < 0 <= 1 ].
[main] PARSE - Swift: mt_502.Mt_502: 0: Failed to parse required
child(Data).
[main] LAST - Swift: Last match: mt_502.Mt_502.
Exception in thread "main" at 0: null:
com.stc.swift.runtime.SwiftUnmarshalException: mt_502.Mt_502: 0:
Failed to parse required child(Data).
    at
com.stc.swift.runtime.SwiftOtdRep.throwExcept(SwiftOtdRep.java:1977)
    at
com.stc.swift.runtime.SwiftOtdRep.parseChildren(SwiftOtdRep.java:1577
)
    at
com.stc.swift.runtime.SwiftOtdRep.parse(SwiftOtdRep.java:1486)
    at
com.stc.swift.runtime.SwiftOtdRep.unmarshal(SwiftOtdRep.java:1339)
```

## Debug Level Message Example

The following example shows a debug-level run for Collaboration OTD 502, Test 17:

```
{1:F01ABNANL2AAXXX1234123456}{2:I502BSUIFRPPXXXXN}{3:{108:Test 17}}{4
:
:16R:GENL
:20C::SEME//TEST 17
:23G:NEWM
:22F::TRTR//TRAD
:16S:GENL
:16R:ORDRDET
:22H::BUSE//SUBS
:22F::TOOR//MAKT
:22F::TILI//GTHD
:22H::PAYM//APMT
:16R:TRADPRTY
:95P::BUYR//RABONL2A
:16S:TRADPRTY
:16R:TRADPRTY
:95P::INVE//PECSBEBB
:35B:ISIN LU0111807797
:16S:TRADPRTY
:16S:ORDRDET
-}
```

# Using SWIFT Funds OTD Library

This chapter explains how to use specialized funds features available with the SWIFT OTD Library and eGate Integrator.

## Chapter Topics

- [“SWIFT OTD Library Funds Features: Overview” on page 54](#)
- [“SWIFT Funds Message Templates” on page 56](#)
- [“SWIFT Standards and Message Types” on page 56](#)

---

## 5.1 SWIFT OTD Library Funds Features: Overview

The SWIFT OTD Library Object Type Definitions (OTDs) contain specialized OTDs that allow you to automate the following funds operations:

- Orders to buy and sell
- Client confirmations
- Checking order status
- Statement of holdings, for fund balances reconciliation

In the past, many funds industry players have asked SWIFT to help automate these operations by providing standards and connectivity between funds distributors, transfer agents, funds management companies, and other intermediaries like funds processing hubs. To meet these needs, SWIFT has developed standards and message templates based on these standards.

The SWIFT OTD library contains the following types of Funds OTDs:

- FIN based
- Extensible Markup Language (XML) based

Table 13 lists the Funds OTDs and shows which ones are FIN-based and which are XML-based.

**Table 13** FIN- and XML-based Funds OTDs

OTD Name	Base
if5001_FUNDS	XML
if5002_FUNDS	XML
if5003_FUNDS	XML
if5010_FUNDS	XML
if5011_FUNDS	XML
if5012_FUNDS	XML
if5013_FUNDS	XML
if5014_FUNDS	XML
if5015_FUNDS	XML
if5020_FUNDS	XML
if5021_FUNDS	XML
if5022_FUNDS	XML
if5023_FUNDS	XML
if5024_FUNDS	XML
if5025_FUNDS	XML
if5030_FUNDS	XML
if5031_FUNDS	XML
if5032_FUNDS	XML
if5040_FUNDS	XML
if5041_FUNDS	XML
mt_502_FUNDS	FIN
mt_509_FUNDS	FIN
mt_515_FUNDS	FIN
mt_535_FUNDS	FIN
mt_574_IRSLIST	FIN
mt_574_W8BENO	FIN

The rest of this chapter explains how to use the library's Funds OTDs.

---

## 5.2 SWIFT Funds Message Templates

To answer market demand for standards, SWIFT has developed standard order initiation and order confirmation messages for equities. In addition, SWIFT has created templates for these types of messages dedicated to investment funds.

These standard funds template message formats have been reviewed and approved by leading investment funds industry players. These messages are already available on the SWIFT network. They help facilitate the automation of subscriptions, redemptions, switches, and cancellations of orders for open-ended investment funds. These message templates also help to automate basic operations related to such orders, including cash payment flows, as well as the reconciliation of investment fund balances.

These funds message templates have a variety of features that help automate and expedite your order processing, for example:

- Expressing your order in a cash amount or a number of units
- Expressing your order in a currency other than the denomination currency of the funds
- Including any settlement information
- Indicating, where applicable, your dividend standing instruction
- Identifying any distributor linked to the order (up to three levels)
- Specifying the applicable commission (in & or amount, net or gross)
- Identifying the final beneficial owner when required (for example, registration in a final beneficial owner's name)

The SWIFT OTD Library's Funds OTDs let you implement and automate all of these features in eGate.

---

## 5.3 SWIFT Standards and Message Types

The SWIFT OTD Library's Funds OTDs are based on the SWIFT standards and message structure descriptions provided in the current version of their funds templates document. The funds template update frequency is expected to be on a yearly basis, with backward compatibility. The Funds OTDs are updated on the same cycle.



The SWIFT OTD Library includes the following Funds OTDs specialized for the associated SWIFT message types and their funds operations:

- **MT 502 (order to buy and sell):** For funds subscription, redemption, switch, and cancellation
- **MT 515 (client confirmation):** For confirmation of the funds subscription, redemption, switch and cancellation
- **MT 509 (order status):** For status update on orders (for example, rejection or acknowledgement of receipt)
- **MT 535 (statement of holdings):** For funds balances reconciliation

See [Chapter 3](#) for a complete list of the SWIFT message types in the library.

# Using OTD Library Java Classes

This chapter provides an overview of the Java classes/interfaces and methods contained in the SWIFT OTD Library. These methods are used to extend the functionality of the library.

## Chapter Topics

- [“SWIFT OTD Library Classes: Overview” on page 58](#)
- [“OTD Library Java Classes” on page 59](#)

---

## 6.1 SWIFT OTD Library Classes: Overview

The SWIFT OTD Library exposes various Java classes to add extra functionality to the library and its Object Type Definitions (OTDs). Some of these classes contain methods that allow you to set data in the library OTDs, as well as get data from them.

### 6.1.1 Relation to OTD Message Types

The nature of this data transfer depends on the available nodes and features in each of the individual SWIFT OTD message types. For more information on the SWIFT OTD Library’s messages and message types, see [Chapter 3](#).

### 6.1.2 SWIFT OTD Library Javadoc

For a complete list and explanation of the Java classes and methods in the OTDs, refer to the **Javadoc**. You can download the **Javadoc** while you are installing the OTD Library, from the same eGate Enterprise Manager page where you downloaded this user’s guide. For complete instructions, see the *SeeBeyond ICAN Suite Installation Guide*.

After you extract the **SWIFT\_OTD\_Library\_Javadoc.zip** file, double-click on the **JavadocOverview.html** file to begin using the **Javadoc**. This file contains complete instructions on how to use this document, as well a link that takes you to the additional **Javadoc** files.

*Note:* The **Javadoc** for the SWIFT OTD Library is a large set of files and may operate slowly in your browser.

---

## 6.2 OTD Library Java Classes

The **Javadoc** shows a Java class for each OTD in the SWIFT OTD Library. For example, the class **Mt\_101** includes the OTD for the MT 101 SWIFT message type. See [Chapter 3](#) for a complete list of the SWIFT message types/OTDs in the library.

In addition to the classes for OTDs, there are the following Java classes with methods for run-time operation:

- **SwiftMarshalException**
- **SwiftOtdChild**
- **SwiftOtdInputStream**
- **SwiftOtdLocation**
- **SwiftOtdRep**
- **SwiftParseUtils**
- **SwiftUnmarshalException**

# Index

## A

Acknowledgment 23  
 Acknowledgment of  
   Guarantee Message 23  
   Trust Receipt 24  
 Advice of  
   Charges, Interest and Other Adjustments 15, 16,  
     17, 18, 20, 22, 23, 24, 25, 26  
   Cheque 15  
   Discharge 23  
   Discrepancy 23  
   Loan/Deposit Interest Payment 17  
   Non-Payment of Cheques 16  
   Payment/Acceptance/Negotiation 23  
   Rate and Amount Fixing 22  
   Reduction or Release 23  
   Refusal 23  
   Reimbursement Claim 20  
   Reimbursement or Payment 23  
   Third Bank's Documentary Credit 23  
 Advice/Instruction of a  
   Third Party Deal 17  
   Third Party FX Deal 17  
 Allegement Statement 20  
 Amendment to  
   Authorization to Reimburse 23  
   Documentary Credit 23  
 Answers 15, 16, 17, 18, 20, 22, 23, 24, 25, 26  
 Authorization to  
   Pay, Accept or Negotiate 23  
   Reimburse 23

## B

Balance Report 25  
 BICDir exceptions 47  
 BICDir Service requirements 9  
 BICDirService Browser Lookup 44  
 BICDirService Method Operation 46  
 Business Process Canvas 36  
 Business Process Execution Language (BPEL) 38  
 Buy/Sell Order 19

## C

Call/Notice Loan/Deposit Confirmation 17  
 Cancellation  
   Request 15, 17, 18, 20, 22, 23, 24, 25, 26  
 Cash Letters  
   Advice of Dishonor 18  
   Cash Letter Credit Adjustment Advice 18  
   Cash Letter Credit Advice 18  
 Certificate Numbers 20  
 Cheque  
   Advice of 15  
   Advice of Non-Payment 16  
   Stop Payment Request 15  
   Stop Payment Status 15  
   Truncation Message 16  
 Claim  
   Collateral 19  
   Paying Agent's 20  
   Reimbursement 20, 23  
 Client Advice of Execution 19  
 Client Confirmation of Purchase or Sale 19  
 Collaboration Editor (Java) 36  
 Collateral  
   Adjustment Message 20  
   Claim 19  
   Proposal 19  
   Status and Processing Advice 19  
   Substitution 19  
 Collateral and Exposure Statement 19  
 Collections  
   Acknowledgment 18  
   Advice of Acceptance 18  
   Advice of Fate and Request for Instructions 18  
   Advice of Non-Payment/Non-Acceptance 18  
   Advice of Payment 18  
   Amendment of Instructions 18  
   Clean Collection 18  
   Tracer 18  
 Combined Activity Statement 20  
 Confirmation of  
   Call/Notice Loan/Deposit 17  
   Corporate Action 20  
   Credit 25  
   Debit 25  
   Deliver Against Payment 20  
   Deliver Free 20  
   Depository Receipt 20  
   Fixed Loan/Deposit 17  
   Foreign Currency Option 17  
   Foreign Exchange 17  
   Foreign Exchange Order 17  
   Forward Rate Agreement 17  
   Forward Rate Agreement Settlement 17

- Market-Side Securities Trade 19
  - Precious Metal Option 22
  - Precious Metal Trade 22
  - Purchase or Sale 19
  - Receive Against Payment 20
  - Receive Free 20
  - Securities Loan 19
  - Confirmation of Registration or Modification 19
  - Connectivity Map Canvas 36
  - Corporate Action
    - Confirmation 20
    - Instruction 20
    - Narrative 20
    - Notification 20
    - Status and Processing Advice 20
  - Credit
    - Confirmation 25
    - Transfer 15
    - Transfer (REMIT) 15
    - Transfer (STP) 15
  - Cross Currency Interest Rate Swap Confirmation 17
  - Customer
    - Direct Debit 15
    - Statement Message 25
- D**
- Debit Confirmation 25
  - debug example 53
  - Deliver Against Payment
    - Confirmation 20
    - Instruction 19
  - Deliver Free 19
    - Confirmation 20
  - Depository Receipt
    - Confirmation 20
    - Instruction 20
    - Status and Processing Advice 20
  - Discharge, Advice of 23
  - Discontinuation Acknowledgement Message 25
  - Discontinue Service Message 25
  - Discrepancy, Advice of 23
  - Documentary Credit
    - Advice of Third Bank's 23
    - Amendment 23
    - Issue 23
    - Pre-Advice 23
    - Transfer 23
  - Drawdown/Renewal Notice 22
- E**
- EDIFACT
    - Envelope 15
  - FINPAY 15
  - eInsight Engine and components 38
  - eInsight Validation Project, introduction 38
  - eInsight with SWIFT OTD Library
    - overview 39
    - using Business Process canvas 40
  - Error in Key Service Message 25
  - Error Message 25
  - error messages 50
  - errors
    - MFVR 32
    - MPR 34
  - ETC
    - Client-Side Settlement Instruction 19
    - Market-Side Settlement Instruction 19
    - Pending Trades Statement 20
  - external system requirements 9
- F**
- Fee Due Notice 22
  - field tag 12
  - files, SWIFT supplied 45
  - Financial Institution
    - Transfer 16
    - Transfer Execution 16
    - Transfer for its Own Account 16
    - Transfer Request 16
  - Financial Markets Direct Debit Message 16
  - Fixed Loan/Deposit Confirmation 17
  - Foreign Currency
    - Option 17
    - Option Confirmation 17
  - Foreign Exchange
    - Confirmation 17
    - Order 17
    - Order Confirmation 17
  - Forex/Currency Option Allocation Instruction 17
  - format of error message 50
  - Forward Rate Agreement
    - Confirmation 17
    - Settlement Confirmation 17
  - Free Format Message 15, 16, 17, 18, 21, 22, 23, 24, 25, 26
  - fund OTDs 54
- G**
- General
    - Direct Debit Message 15
    - Financial Institution Transfer 16
    - Securities Lending/Borrowing Message 19
    - Syndicated Facility Message 22
  - Guarantee 23

Amendment 23  
 Message Acknowledgment 23

## H

header block 12  
 heap size  
   adjusting heap memory size 10  
 Holdings Statement 19

## I

import sample Project 36  
 Information Service Message 16, 26  
 Initiation Message  
   Request 25  
   Response 25  
 Initiation Response Message 25  
 installation 9  
 Instruction for Gross/Net Settlement of Third Party  
 FX Deals 17  
 Instruction to  
   Register 19  
   Settle a Third Party Loan/Deposit 17  
 Interest Rate Reset/Advice of Payment 17  
 Interim Transaction Report 25  
 Intra-Position  
   Advice 19  
   Advice Statement 19  
   Instruction 19  
 Inventory Addition 24  
 IRS 1441 NRA  
   (IRSLST) 20  
   (W8BENO) 20  
 Issue of a Documentary Credit 23

## J

Java methods and classes  
   overview 58  
 Javadoc 58  
 Javadoc, obtaining 58

## K

Key Acknowledgement Message 25  
 Key Service Message 25  
   Error 25

## L

live updating 45  
 Logical Host requirements 9

lookup method definitions, BIC 47

## M

market practice rules (MPRs), overview 32  
 Market-Side Securities Trade Confirmation 19  
 Message  
   Categories 14  
   Cheque Transaction 16  
   Collateral Adjustment 20  
   Customer Statement 25  
   Financial Markets Direct Debit 16  
   Free Format 15, 16, 17, 18, 21, 22, 23, 24, 25, 26  
   General Direct Debit 15  
   General Syndicated Facility 22  
   Netting Request 25  
   Proprietary 15, 16, 17, 18, 21, 22, 23, 24, 25, 26  
   Statement 25  
   Trade Status 19  
   Types 14  
 message format validation rules (MFVRs), overview  
 30  
 message mypes 6, 7, 13  
 message type 59  
 methods list, MPR 34  
 Modification of Client Details 19  
 Multiple  
   Customer Credit Transfer 15  
   Customer Credit Transfer (STP) 15  
   Financial Institution Transfer for its Own  
   Account 16  
   General Financial Institution Transfer 16  
 Multiple Interbank Funds Transfer 15

## N

Netting  
   Balance Report 25  
   Interim Statement 25  
   Request Message 25  
   Statement 25  
 Notice of  
   Corporate Action 20  
   Drawdown/Renewal 22  
   Fee Due 22  
   T/C Inventory Destruction/Cancellation 24  
 Notice to Receive 16  
 Numbers Statement 20

## O

Object Type Definition (OTD) 6  
 Open Orders Statement 20

## Index

operating systems  
supported 8  
Options Setup  
dialog box 10  
Order to Buy or Sell 19  
OTD and Collaboration Locations in Enterprise Designer 13  
OTD Editor 36  
OutOfMemoryError  
increase heap size 10

## P

parse debug output example 51  
Paying Agent's Claim 20  
Payment Advice 23  
Payment of Principal and/or Interest 22  
Pending Transactions Statement 19  
Pre-Advice of a Documentary Credit 23  
Precious Metal  
Account Statement 22  
Contracts Statement 22  
Credit Advice 22  
Debit Advice 22  
Notice to Receive 22  
Option Confirmation 22  
Trade Confirmation 22  
Transfer/Delivery Order 22  
Principal and/or Interest Payment 22  
Project  
canvas 35  
deploying 43  
External Environment 43  
Project sample  
importing 36  
Proposal  
Collateral 19  
Proprietary Message 15, 16, 17, 18, 21, 22, 23, 24, 25, 26

## Q

Queries 15, 16, 17, 18, 20, 22, 23, 24, 25, 26

## R

Rate Change Advice 25  
Receive Against Payment  
Confirmation 20  
Instruction 19  
Receive Free 19  
Confirmation 20  
Refusal, Advice of 23

Registration Status and Processing Advice 19  
Reimbursement  
Advice 23  
Authorization 23  
Authorization Amendment 23  
Claim 23  
Claim or Advice 20  
Request for  
Cancellation 15, 16, 17, 18, 20, 22, 23, 24, 25, 26  
Financial Institution Transfer 16  
Payment of Charges, Interest and Other Expenses 15, 16, 17, 18, 20, 22, 23, 24, 25, 26  
Statement/Status Advice 20  
Stop Payment of a Cheque 15  
T/C Stock 24  
Transfer 15  
Request for Service Initiation Message 25  
Request Message 25

## S

Securities  
General Lending/Borrowing Message 19  
Loan Confirmation 19  
Market-Side Trade Confirmation 19  
Settlement  
Allegements Statement 20  
ETC Client-Side Instruction 19  
ETC Market-Side Instruction 19  
Status and Processing Advice 20  
Single Currency  
Interest Rate Derivative Confirmation 17  
Interest Rate Derivative Termination/Recouping Confirmation 17  
Interest Rate Swap Termination/Recouping Confirmation 17  
Single Customer Credit Transfer 15  
Single Customer Credit Transfer (REMIT) 15  
Single Customer Credit Transfer (STP) 15  
Statement Message 25  
Statement of  
Allegation 20  
Combined Activity 20  
ETC Pending Trades 20  
Holdings 19  
Intra-Position Advice 19  
Numbers 20  
Open Orders 20  
Pending Transactions 19  
Precious Metal Account 22  
Precious Metal Contracts 22  
Settlement Allegements 20  
Transactions 19  
Triparty Collateral and Exposure 20

## Index

Statement/Status Advice Request 20  
Status  
    Inquiry 25  
    Report 25  
Status of a Request for Stop Payment of a Cheque 15  
supported operating systems 8  
SWIFT  
    Message Categories 14  
    Message Types 14  
SwiftOTDLibrary.jar 46  
system requirements 9

## T

T/C 24  
    Inventory Destruction/Cancellation Notice 24  
    Inventory Transfer 24  
    Multiple Sales Advice 24  
    Refund Authorization 24  
    Refund Confirmation 24  
    Refund Request 24  
    Sales and Settlement Advice 24  
    Settlement Advice 24  
    Stock Request 24  
text block 12  
Trade  
    Allocation Instruction 19  
    Confirmation Affirmation 19  
    Status Message 19  
Trailer block 12  
Transactions Statement 19  
Transfer  
    Customer Credit 15  
    Customer Credit (REMIT) 15  
    Customer Credit (STP) 15  
    Documentary Credit 23  
    Financial Institution 16  
    General Financial Institution 16  
    Multiple General Financial Institution 16  
    Request 15  
    T/C Inventory 24  
Triparty  
    Collateral and Exposure Statement 20  
    Collateral Instruction 19  
    Collateral Status and Processing Advice 20  
Trust Receipt Acknowledgment 24

## U

update method operation 45  
updating BIC information 46  
updating BICDirService 45  
Using BICDir Service lookup feature 48

## V

validation Collaboration Definitions  
    MFVR 31  
    MPR 33  
validation components, OTD 28  
validation features 27  
validation method definitions, BIC 47  
validation methods  
    MFVR 32  
    MPR 33  
validation Projects, overview 35  
validation rules, MFVR 31

## W

wildcards in lookups 46