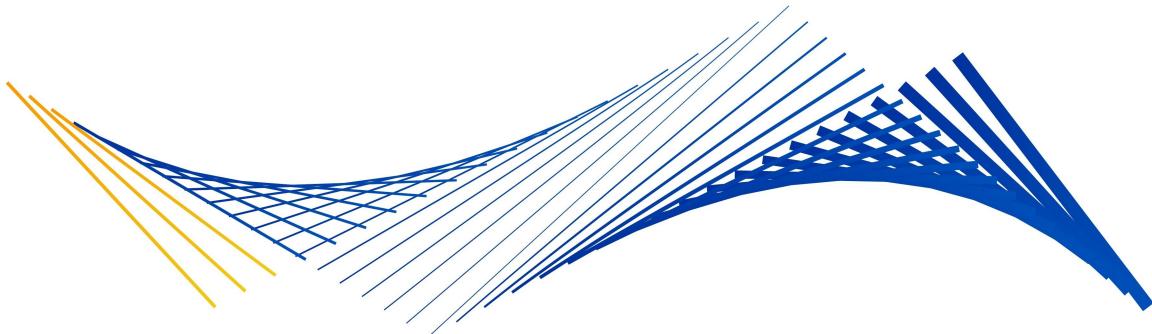




# Real Time System Interface for VROL

Release 19.2

*Effective: 18 October 2019*



### **Important Note on Confidentiality and Copyright**

**Notice:** The Visa Confidential label signifies that the information in this document is proprietary and CONFIDENTIAL to Visa. It is distributed to Visa participants for use exclusively in managing their Visa programs. It must not be duplicated, published, distributed or disclosed, in whole or in part, to merchants, cardholders or any other person without prior written permission from Visa.

The trademarks, logos, trade names and service marks, whether registered or unregistered (collectively the "Trademarks") are Trademarks owned by Visa. All other trademarks not attributed to Visa are the property of their respective owners.

**Note:** *This document is not part of the Visa Rules. In the event of any conflict between any content in this document, any document referenced herein, any exhibit to this document, or any communications concerning this document, and any content in the Visa Rules, the Visa Rules shall govern and control.*

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN: THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. VISA MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

If you have technical questions or questions regarding a Visa service or questions about this document, please contact your Visa representative.

# Contents

## About This Guide

Audience . . . . .	1
Assumptions . . . . .	1
Summary of Changes . . . . .	1
Changes . . . . .	2
Document Organization . . . . .	2
Related Publications. . . . .	4
Document Conventions . . . . .	4
Website Links . . . . .	5

## Chapter 1 • Service Overview

Systems Interfaces . . . . .	1-1
VROL Organization Types. . . . .	1-1
VROL Organization Setup. . . . .	1-2
Supported Transaction Types . . . . .	1-2
Dispute Resolution Process . . . . .	1-4
RTSI Flow—Overview . . . . .	1-5
Dispute Process Flow . . . . .	1-6
Allocation Dispute Process Workflow . . . . .	1-7

---

Collaboration Dispute Process Workflow . . . . .	1-7
--	-----

## **Chapter 2 • RTSI Overview**

Types of Web Services Supported . . . . .	2-1
High-Level Characteristics . . . . .	2-1
Supported Standards . . . . .	2-2
XML Data Validation . . . . .	2-3
JSON Data validation . . . . .	2-3
Image Requirements . . . . .	2-4
Attaching Images to Questionnaires or Existing Cases . . . . .	2-5
Hybrid Image Uploads . . . . .	2-6
Service Description Files . . . . .	2-6
SOAP-based Services . . . . .	2-6
Application Services . . . . .	2-6
REST-based Services . . . . .	2-7
Application Services . . . . .	2-7
Security Services . . . . .	2-7
RTSI Clients . . . . .	2-7
Client Requirements . . . . .	2-7
SOAP-based Services . . . . .	2-7
REST-based Services . . . . .	2-7
Generating Proxy Classes . . . . .	2-8
Authentication and Authorization . . . . .	2-8
Internet Access . . . . .	2-8
RTSI Requests . . . . .	2-9
SOAP-based Services . . . . .	2-9
Transaction Inquiry Web Service . . . . .	2-9
Imaging Web Service . . . . .	2-10

---

REST-based Services . . . . .	2-10
Transaction Inquiry Web Service . . . . .	2-11
Imaging Web Service . . . . .	2-12
HTTP Header . . . . .	2-12
SOAP-based Services . . . . .	2-12
SOAP Envelope . . . . .	2-12
SOAP Body . . . . .	2-12
RTSI SOAP Operation (Method) . . . . .	2-12
SOAP Action (<soapaction>) . . . . .	2-13
REST-based Services . . . . .	2-13
RTSI Responses . . . . .	2-13
SOAP-based Services . . . . .	2-13
Transaction Inquiry Web Service . . . . .	2-14
Imaging Web Service . . . . .	2-15
REST-based Services . . . . .	2-15
Transaction Inquiry Web Service . . . . .	2-19
Imaging Web Service . . . . .	2-21

## **Chapter 3 • RTSI Request Operations**

RTSI Calls . . . . .	3-1
SOAP-based Services . . . . .	3-1
REST-based Services . . . . .	3-2
Transaction Operations and Actions . . . . .	3-2
Important XML/JSON Fields in Transactions . . . . .	3-4
Image Handling . . . . .	3-4
SOAP Message Package . . . . .	3-4
Content-Type Information . . . . .	3-5
SOAP Part . . . . .	3-5

---

Attachment Part . . . . .	3-5
Multipart Message. . . . .	3-6
Error Handling. . . . .	3-7
Reject Messages . . . . .	3-9
General Workflow Considerations. . . . .	3-10
Case Matching . . . . .	3-10
Stage-State check . . . . .	3-10
Locked Fields. . . . .	3-11
Selected TI Check . . . . .	3-11
Locked Fields Check. . . . .	3-12
Other Transaction Specific Checks . . . . .	3-12
Other Miscellaneous Checks . . . . .	3-12

## **Chapter 4 • Transaction Inquiries**

Synchronous vs. Asynchronous Processing . . . . .	4-1
TI Process . . . . .	4-2
Transaction Inquiry Flows . . . . .	4-2
Request TI. . . . .	4-6
Request Original Transaction Information . . . . .	4-8
Request Associated Transactions. . . . .	4-8
Update Associated Transactions Selection Data . . . . .	4-10
Request More Merchant Information (MMI) . . . . .	4-11
TI Operations . . . . .	4-11
SISubmitTranInquiryRequest . . . . .	4-11
Request Message—SOAP-based Services . . . . .	4-14
Response Message—Synchronous Processing—SOAP-based Services. . . . .	4-15
Response Message—Asynchronous Processing—SOAP-based Services . . . . .	4-16
Request Message—REST-based Services . . . . .	4-16

---

Response Message—Synchronous Processing—REST-based Services . . . . .	4-17
Response Message—Asynchronous Processing—REST-based Services . . . . .	4-18
SIGetTransInquiryResultsRequest . . . . .	4-18
Request Message—SOAP-based Services . . . . .	4-19
Response Message—SOAP-based Services . . . . .	4-19
Request Message—REST-based Services . . . . .	4-20
Response Message—REST-based Services. . . . .	4-20
SIGetTransDetailsRequest . . . . .	4-20
Request Message—SOAP-based Services . . . . .	4-21
Response Message—SOAP-based Services . . . . .	4-22
Request Message—REST-based Services . . . . .	4-22
Response Message—REST-based Services. . . . .	4-23
SIGetTransactionsFromCaseRequest . . . . .	4-23
Request Message—SOAP-based Services . . . . .	4-24
Response Message—SOAP-based Services . . . . .	4-24
Request Message—REST-based Services . . . . .	4-24
Response Message—REST-based Services. . . . .	4-25
SIGetAssociatedTransactionListRequest . . . . .	4-25
Request Message—SOAP-based Services . . . . .	4-26
Response Message—SOAP-based Services . . . . .	4-27
Request Message—REST-based Services . . . . .	4-28
Response Message—REST-based Services. . . . .	4-28
SIAssociatedTranSelectionRequest . . . . .	4-29
Request Message—SOAP-based Services . . . . .	4-30
Response Message—SOAP-based Services . . . . .	4-30
Request Message—REST-based Services . . . . .	4-31
Response Message—REST-based Services. . . . .	4-31
SIGetAssociatedTranSelectionRequest . . . . .	4-31

---

Request Message—SOAP-based Services . . . . .	4-32
Response Message—SOAP-based Services . . . . .	4-32
Request Message—REST-based Services . . . . .	4-32
Response Message—REST-based Services. . . . .	4-33
SIGetMMIRequest . . . . .	4-33
Request Message—SOAP-based Services . . . . .	4-33
Response Message—SOAP-based Services . . . . .	4-33
Request Message—REST-based Services . . . . .	4-34
Response Message—REST-based Services. . . . .	4-34
Inquiry Data Lifetime . . . . .	4-34
Request/Response Tracking . . . . .	4-34

## **Chapter 5 • Case Creation**

Case Creation Process . . . . .	5-1
Create, Update Case . . . . .	5-1
Create Case Operations . . . . .	5-2
SICreateCaseFromTransactionRequest . . . . .	5-2
Request Message—SOAP-based Services . . . . .	5-3
Response Message—SOAP-based Services . . . . .	5-3
Request Message—REST-based Services . . . . .	5-3
Response Message—REST-based Services. . . . .	5-3
SISubmitManualCaseCreationRequest . . . . .	5-3
Request Message—SOAP-based Services . . . . .	5-4
Response Message—SOAP-based Services . . . . .	5-4
Request Message—REST-based Services . . . . .	5-5
Response Message—REST-based Services. . . . .	5-5
Update Existing Case . . . . .	5-5
Get Case Details . . . . .	5-7

Add Image to Case . . . . .	5-8
SIUploadImageRequest . . . . .	5-9
Request Message—SOAP-based Services . . . . .	5-9
Response Message—SOAP-based Services . . . . .	5-9
Request Message—REST-based Services . . . . .	5-10
Response Message—REST-based Services. . . . .	5-10
SIGetImageRequest . . . . .	5-10
Request Message—SOAP-based Services . . . . .	5-11
Response Message—SOAP-based Services . . . . .	5-11
Request Message—REST-based Services . . . . .	5-11
Response Message—REST-based Services. . . . .	5-12
Update Case Resolution Status. . . . .	5-12
SIUpdateCaseResolutionStatusRequest . . . . .	5-12
Request Message—SOAP-based Services . . . . .	5-12
Response Message—SOAP-based Services . . . . .	5-13
Request Message—REST-based Services . . . . .	5-13
Response Message—REST-based Services. . . . .	5-13

## Chapter 6 • Request for Copies

RFC Process. . . . .	6-2
Issuer Steps . . . . .	6-2
Create RFC . . . . .	6-2
Retrieve RFC Network Approval Status . . . . .	6-3
Check for RFC Rejects . . . . .	6-3
Retrieve RFC Responses . . . . .	6-3
Acquirer Steps . . . . .	6-4
Retrieve RFC Advices and RFCs, Create RFC Responses . . . . .	6-4
RFC Operations . . . . .	6-5

---

SISubmitRFCRequest . . . . .	6-6
Request Message—SOAP-based Services . . . . .	6-6
Response Message—SOAP-based Services . . . . .	6-6
Request Message—REST-based Services . . . . .	6-6
Response Message—REST-based Services. . . . .	6-7
SIGetRFCDetailsRequest . . . . .	6-7
Request Message—SOAP-based Services . . . . .	6-7
Response Message—SOAP-based Services . . . . .	6-8
Request Message—REST-based Services . . . . .	6-8
Response Message—REST-based Services. . . . .	6-9
Network Approval Status Messages . . . . .	6-9
SIGetRFCAdviceRequest . . . . .	6-14
Request Message—SOAP-based Services . . . . .	6-14
Response Message—SOAP-based Services . . . . .	6-14
Request Message—REST-based Services . . . . .	6-15
Response Message—REST-based Services. . . . .	6-15
SISubmitRFCFulfillmentRequest . . . . .	6-15
Request Message—SOAP-based Services . . . . .	6-16
Response Message—SOAP-based Services . . . . .	6-16
Request Message—REST-based Services . . . . .	6-17
Response Message—REST-based Services. . . . .	6-17
SISubmitRFCNonFulfillmentRequest . . . . .	6-17
Request Message—SOAP-based Services . . . . .	6-18
Response Message—SOAP-based Services . . . . .	6-18
Request Message—REST-based Services . . . . .	6-19
Response Message—REST-based Services. . . . .	6-19
SIGetRFCFulfillmentRequest . . . . .	6-19
Request Message—SOAP-based Services . . . . .	6-19

---

Response Message—SOAP-based Services . . . . .	6-20
Request Message—REST-based Services . . . . .	6-20
Response Message—REST-based Services. . . . .	6-20
SIGetRFCNonFulfillmentRequest . . . . .	6-21
Request Message—SOAP-based Services . . . . .	6-21
Response Message—SOAP-based Services . . . . .	6-21
Request Message—REST-based Services . . . . .	6-21
Response Message—REST-based Services. . . . .	6-22

## **Chapter 7 • Dispute Questionnaires**

Dispute Workflows . . . . .	7-1
Dispute Process . . . . .	7-2
Dispute Process Flows . . . . .	7-3
Initiate and Submit Dispute . . . . .	7-11
Initiate Dispute . . . . .	7-12
Submit Dispute . . . . .	7-14
Submit and Retrieve Fraud Bundle Disputes . . . . .	7-16
Submit Fraud Bundle . . . . .	7-17
Retrieve Fraud Bundle Dispute Details . . . . .	7-19
Retrieve Dispute from Queue . . . . .	7-19
Get Dispute Details . . . . .	7-19
Respond to a Received Dispute . . . . .	7-20
Get Dispute Response Details. . . . .	7-21
Retrieve Acceptance Notification Details. . . . .	7-22
Recall Dispute . . . . .	7-22
Delete Dispute . . . . .	7-23
Check for Dispute Rejects . . . . .	7-23
Check for Disputes Eligible for Rapid Dispute Resolution. . . . .	7-23

Rapid Dispute Resolution Process. . . . .	7-24
Submit Disputes Using Fraud on Authorization Transactions . . . . .	7-25
Dispute Operations . . . . .	7-25
SInitiateDisputeFromTransactionOrCaseRequest . . . . .	7-26
Request Message—SOAP-based Services . . . . .	7-28
Response Message—SOAP-based Services . . . . .	7-29
Request Message—REST-based Services . . . . .	7-29
Response Message—REST-based Services. . . . .	7-29
SISubmitDisputeQuestionnaireRequest . . . . .	7-30
Request Message—Fraud—SOAP-based Services . . . . .	7-30
Response Message—Fraud—SOAP-based Services . . . . .	7-31
Request Message—Authorization—SOAP-based Services . . . . .	7-31
Response Message—Authorization—SOAP-based Services . . . . .	7-32
Request Message—Processing Error—SOAP-based Services . . . . .	7-33
Response Message—Processing Error—SOAP-based Services. . . . .	7-34
Request Message—Consumer—SOAP-based Services . . . . .	7-35
Response Message—Consumer—SOAP-based Services . . . . .	7-36
Request Message—Fraud—REST-based Services . . . . .	7-36
Response Message—Fraud—REST-based Services. . . . .	7-37
Request Message—Authorization—REST-based Services . . . . .	7-38
Response Message—Authorization—REST-based Services . . . . .	7-39
Request Message—Processing Error—REST-based Services . . . . .	7-40
Response Message—Processing Error—REST-based Services . . . . .	7-41
Request Message—Consumer—REST-based Services. . . . .	7-41
Response Message—Consumer—REST-based Services . . . . .	7-42
SIGetDisputeDetailsRequest . . . . .	7-42
Request Message—Fraud—SOAP-based Services . . . . .	7-43
Response Message—Fraud—SOAP-based Services . . . . .	7-43

---

Request Message—Authorization-SOAP-based Services . . . . .	7-45
Response Message—Authorization—SOAP-based Services . . . . .	7-45
Request Message—Processing Error-SOAP-based Services. . . . .	7-46
Response Message—Processing Error—SOAP-based Services. . . . .	7-46
Request Message—Consumer-SOAP-based Services . . . . .	7-48
Response Message—Consumer—SOAP-based Services . . . . .	7-48
Request Message—REST-based Services . . . . .	7-50
Response Message—Fraud—REST-based Services. . . . .	7-50
Response Message—Authorization—REST-based Services . . . . .	7-52
Response Message—Processing Error—REST-based Services . . . . .	7-53
Response Message—Consumer—REST-based Services . . . . .	7-54
SISubmitFraudBundleDisputeQuestionnaireRequest . . . . .	7-56
Request Message—SOAP-based Services . . . . .	7-57
Response Message—SOAP-based Services . . . . .	7-57
Request Message—REST-based Services . . . . .	7-58
Response Message—REST-based Services. . . . .	7-58
SIGetFraudBundleDisputeQuestionnaireStatusRequest . . . . .	7-58
Request Message—SOAP-based Services . . . . .	7-59
Response Message—SOAP-based Services . . . . .	7-59
Request Message—REST-based Services . . . . .	7-60
Response Message—REST-based Services. . . . .	7-60
SICreateDisputeResponseRequest . . . . .	7-60
Request Message—SOAP-based Services . . . . .	7-61
Response Message—SOAP-based Services . . . . .	7-62
Request Message—REST-based Services . . . . .	7-63
Response Message—REST-based Services. . . . .	7-64
SIGetDisputeResponseDetailsRequest . . . . .	7-64
Request Message—SOAP-based Services . . . . .	7-65

Response Message—SOAP-based Services . . . . .	7-65
Request Message—REST-based Services . . . . .	7-67
Response Message—REST-based Services . . . . .	7-67
SIGetAcceptanceNotificationDetailsRequest. . . . .	7-69
Request Message—SOAP-based Services . . . . .	7-69
Response Message—SOAP-based Services . . . . .	7-70
Request Message—REST-based Services . . . . .	7-71
Response Message—REST-based Services. . . . .	7-71
SIChangeDisputeStatusRequest . . . . .	7-72
Request Message—Recall action—SOAP-based Services . . . . .	7-73
Response Message—Recall action—SOAP-based Services . . . . .	7-73
Request Message—Recall action—REST-based Services. . . . .	7-74
Response Message—Recall action—REST-based Services . . . . .	7-74
Request Message—Delete action—SOAP-based Services . . . . .	7-75
Response Message—Delete action—SOAP-based Services. . . . .	7-75
Request Message—Delete action—REST-based Services . . . . .	7-75
Response Message—Delete action—REST-based Services . . . . .	7-75
SIGetRdrDetailsRequest . . . . .	7-76
Request Message—SOAP-based Services . . . . .	7-76
Response Message—SOAP-based Services . . . . .	7-76
Request Message—REST-based Services . . . . .	7-77
Response Message—REST-based Services. . . . .	7-77

## Chapter 8 • Dispute Financials

Dispute Financial Process. . . . .	8-2
Dispute Financial Flows . . . . .	8-2
Retrieve Dispute Financial from Queue . . . . .	8-4
Get Dispute Financial Details . . . . .	8-5

---

Get Dispute Financial Advice Details . . . . .	8-5
Dispute Financial Operations . . . . .	8-5
SIGetDisputeFinancialDetailsRequest . . . . .	8-5
Request Message—SOAP-based Services . . . . .	8-6
Response Message—SOAP-based Services . . . . .	8-6
Request Message—REST-based Services . . . . .	8-11
Response Message—REST-based Services. . . . .	8-11
SIGetDisputeFinancialAdviceDetailsRequest. . . . .	8-17
Request Message—SOAP-based Services . . . . .	8-17
Response Message—SOAP-based Services . . . . .	8-17
Request Message—REST-based Services . . . . .	8-19
Response Message—REST-based Services. . . . .	8-19
Financial Fields Returned in Immediate Response . . . . .	8-21

## **Chapter 9 • Adjustments**

Adjustment Process . . . . .	9-1
Send Adjustment . . . . .	9-1
Retrieve Adjustment Network Approval Status. . . . .	9-2
Check for Adjustment Rejects. . . . .	9-2
Repair Adjustment Rejects . . . . .	9-2
Retrieve Adjustments . . . . .	9-2
Adjustment Operations . . . . .	9-3
SISubmitAdjustmentRequest . . . . .	9-3
Request Message—SOAP-based Services . . . . .	9-3
Response Message—SOAP-based Services . . . . .	9-4
Request Message—REST-based Services . . . . .	9-4
Response Message—REST-based Services. . . . .	9-5
SIGetAdjustmentDetailsRequest . . . . .	9-5

---

Request Message—SOAP-based Services . . . . .	9-5
Response Message—SOAP-based Services . . . . .	9-6
Request Message—REST-based Services . . . . .	9-6
Response Message—REST-based Services. . . . .	9-7
Network Approval Status Messages . . . . .	9-7

## **Chapter 10 • Dispute Pre-Filings**

Pre-Arbitration Process. . . . .	10-1
Associated Transaction Research (ATR) during Pre-Arbitration . . . . .	10-2
Pre-Arbitration Flows. . . . .	10-2
Create Pre-Arbitration . . . . .	10-10
Retrieve Pre-Arbitration from Queue . . . . .	10-12
Get Pre-Arbitration Details . . . . .	10-12
Respond to a Received Pre-Arbitration . . . . .	10-12
Get Pre-Arbitration Response Details . . . . .	10-13
Retrieve Acceptance Notification Details . . . . .	10-14
Delete Pre-Arbitration . . . . .	10-14
Recall Pre-Arbitration. . . . .	10-15
Check for Pre-Arbitration Rejects . . . . .	10-15
Pre-Compliance Process . . . . .	10-16
Pre-Compliance Flows . . . . .	10-16
Create Pre-Compliance . . . . .	10-22
Retrieve Pre-Compliance from Queue . . . . .	10-23
Get Pre-Compliance Details . . . . .	10-24
Respond to a Received Pre-Compliance . . . . .	10-24
Get Pre-Compliance Response Details. . . . .	10-25
Retrieve Acceptance Notification Details. . . . .	10-26
Delete Pre-Compliance . . . . .	10-26

---

Recall Pre-Compliance . . . . .	10-26
Check for Pre-Compliance Rejects . . . . .	10-27
Pre-Filing Operations . . . . .	10-27
SICreateDisputePreArbRequest . . . . .	10-28
Request Message—SOAP-based Services . . . . .	10-29
Response Message—SOAP-based Services . . . . .	10-29
Request Message—REST-based Services . . . . .	10-30
Response Message—REST-based Services . . . . .	10-30
SIGetDisputePreArbDetailsRequest . . . . .	10-30
Request Message—SOAP-based Services . . . . .	10-31
Response Message—SOAP-based Services . . . . .	10-31
Request Message—REST-based Services . . . . .	10-33
Response Message—REST-based Services. . . . .	10-33
SICreateDisputePreArbResponseRequest . . . . .	10-36
Request Message—SOAP-based Services . . . . .	10-36
Response Message—SOAP-based Services . . . . .	10-37
Request Message—REST-based Services . . . . .	10-37
Response Message—REST-based Services. . . . .	10-37
SIGetDisputePreArbResponseDetailsRequest . . . . .	10-37
Request Message—SOAP-based Services . . . . .	10-38
Response Message—SOAP-based Services . . . . .	10-38
Request Message—REST-based Services . . . . .	10-40
Response Message—REST-based Services. . . . .	10-40
SICreateDisputePreCompRequest . . . . .	10-42
Request Message—SOAP-based Services . . . . .	10-43
Response Message—SOAP-based Services . . . . .	10-43
Request Message—REST-based Services . . . . .	10-44
Response Message—REST-based Services. . . . .	10-44

SIGetDisputePreCompDetailsRequest . . . . .	10-44
Request Message—SOAP-based Services . . . . .	10-45
Response Message—SOAP-based Services . . . . .	10-45
Request Message—REST-based Services . . . . .	10-46
Response Message—REST-based Services. . . . .	10-46
SICreateDisputePreCompResponseRequest . . . . .	10-48
Request Message—SOAP-based Services . . . . .	10-48
Response Message—SOAP-based Services . . . . .	10-49
Request Message—REST-based Services . . . . .	10-49
Response Message—REST-based Services. . . . .	10-50
SIGetDisputePreCompResponseDetailsRequest . . . . .	10-50
Request Message—SOAP-based Services . . . . .	10-51
Response Message—SOAP-based Services . . . . .	10-51
Request Message—REST-based Services . . . . .	10-52
Response Message—REST-based Services. . . . .	10-52
SIGetAcceptanceNotificationDetailsRequest. . . . .	10-54
Request Message—SOAP-based Services . . . . .	10-55
Response Message—SOAP-based Services . . . . .	10-56
Request Message—REST-based Services . . . . .	10-57
Response Message—REST-based Services. . . . .	10-57
SIChangeDisputeStatusRequest . . . . .	10-58
Request Message—Recall action—SOAP-based Services . . . . .	10-59
Response Message—Recall action—SOAP-based Services . . . . .	10-60
Request Message—Recall action—REST-based Services. . . . .	10-60
Response Message—Recall action—REST-based Services . . . . .	10-60
Request Message—Delete action—SOAP-based Services . . . . .	10-61
Response Message—Delete action—SOAP-based Services. . . . .	10-61
Request Message—Delete action—REST-based Services . . . . .	10-61

---

Response Message—Delete action—REST-based Services . . . . .	10-62
--	-------

## Chapter 11 • Dispute Case Filings

Dispute Case Filing Process—Summary. . . . .	11-2
Dispute Case Filing Currencies . . . . .	11-3
Dispute Case Filing Process—Details. . . . .	11-3
Dispute Case Filing Flows . . . . .	11-4
Create Dispute Case Filing . . . . .	11-10
Retrieve Dispute Case Filing . . . . .	11-11
Submit Dispute Case Filing Withdrawal Request . . . . .	11-12
Retrieve Dispute Case Filing Withdrawal Request. . . . .	11-12
Retrieve Contact Message, Create Contact Message Response. . . . .	11-13
Create Dispute Case Filing Appeal, Retrieve Dispute Case Filing Appeal Request. . . . .	11-14
Retrieve Acknowledgment/Rejection/Final Decision Letters . . . . .	11-15
Delete Case Filing . . . . .	11-15
Check for Case Filing Rejects . . . . .	11-15
Dispute Case Filing Operations. . . . .	11-16
SISubmitDisputeFilingRequest. . . . .	11-16
Request Message—Arbitration—SOAP-based Services . . . . .	11-17
Request Message—Compliance—SOAP-based Services . . . . .	11-17
Response Message—SOAP-based Services . . . . .	11-19
Request Message—Arbitration—REST-based Services . . . . .	11-19
Request Message—Compliance—REST-based Services . . . . .	11-20
Response Message—REST-based Services. . . . .	11-20
SIGetDisputeFilingDetailsRequest . . . . .	11-20
Request Message—SOAP-based Services . . . . .	11-21
Response Message—Arbitration—SOAP-based Services. . . . .	11-21

Response Message—Compliance—SOAP-based Services . . . . .	11-24
Request Message—REST-based Services . . . . .	11-26
Response Message—Arbitration—REST-based Services . . . . .	11-26
Response Message—Compliance—REST-based Services . . . . .	11-28
SIGetContactMessageDetailsRequest . . . . .	11-30
Request Message—SOAP-based Services . . . . .	11-31
Response Message—SOAP-based Services . . . . .	11-31
Request Message—REST-based Services . . . . .	11-33
Response Message—REST-based Services. . . . .	11-33
SISubmitContactMessageResponseRequest . . . . .	11-36
Request Message—SOAP-based Services . . . . .	11-36
Response Message—SOAP-based Services . . . . .	11-36
Request Message—REST-based Services . . . . .	11-37
Response Message—REST-based Services. . . . .	11-37
SIGetContactMessageResponseDetailsRequest. . . . .	11-37
Request Message—SOAP-based Services . . . . .	11-38
Response Message—SOAP-based Services . . . . .	11-38
Request Message—REST-based Services . . . . .	11-39
Response Message—REST-based Services. . . . .	11-39
SIChangeDisputeStatusRequest . . . . .	11-41
Request Message—Delete action—SOAP-based Services . . . . .	11-42
Response Message—Delete action—SOAP-based Services. . . . .	11-42
Request Message—Delete action—REST-based Services . . . . .	11-43
Response Message—Delete action—REST-based Services . . . . .	11-43

## Chapter 12 • Miscellaneous Fees

Miscellaneous Fees Process . . . . .	12-2
Send Misc Fees . . . . .	12-2

---

Retrieve Misc Fees Network Approval Status . . . . .	12-2
Check for Misc Fee Rejects. . . . .	12-2
Retrieve Misc Fees. . . . .	12-3
Miscellaneous Fee Operations . . . . .	12-3
SISubmitMiscFeeRequest . . . . .	12-3
Request Message—SOAP-based Services . . . . .	12-4
Response Message—SOAP-based Services . . . . .	12-4
Request Message—REST-based Services . . . . .	12-5
Response Message—REST-based Services. . . . .	12-5
SIGetMiscFeeDetailsRequest . . . . .	12-5
Request Message—SOAP-based Services . . . . .	12-6
Response Message—SOAP-based Services . . . . .	12-6
Request Message—REST-based Services . . . . .	12-7
Response Message—REST-based Services. . . . .	12-7
Network Approval Status Messages . . . . .	12-7

## **Chapter 13 • Fraud Reporting**

Fraud Disputes . . . . .	13-1
Fraud Report Process . . . . .	13-2
Send Fraud Reports . . . . .	13-2
Retrieve Delivered Fraud Reports from Queue. . . . .	13-2
Retrieve Fraud Report Information . . . . .	13-2
Check for Fraud Report Rejects . . . . .	13-3
Repair a Rejected Fraud Advice . . . . .	13-3
Fraud Report Operations . . . . .	13-3
SISubmitFraudReportRequest . . . . .	13-4
Non-NRI FRS Fraud With Notification Code=1 and ASN=4001 . . . . .	13-5

Non-NRI FRS Fraud With Notification Code=1 and Auto-Assign ASN=true . . . . .	13-6
NRI FRS Fraud . . . . .	13-7
SIGetFraudReportDetailsRequest. . . . .	13-9
Non-NRI FRS Fraud . . . . .	13-9
NRI Fraud . . . . .	13-10
SIGetQueueRequest . . . . .	13-11

## Chapter 14 • Exception File Listings

Exception File Listing Process . . . . .	14-2
Exception File Listing Operations . . . . .	14-2
SISubmitExceptionRequest . . . . .	14-2
ListNegativeExceptionFile . . . . .	14-3
ListVIPExceptionFile (non-Cardholder Transaction Amount Limit transaction) . . . . .	14-5
ListVIPExceptionFile (Cardholder Transaction Amount Limit transaction) . . . . .	14-6
DeleteExceptionFile . . . . .	14-7
SIGetExceptionDetailsRequest. . . . .	14-8
Request Message—SOAP-based Services . . . . .	14-8
Response Message—SOAP-based Services . . . . .	14-8
Request Message—REST-based Services . . . . .	14-9
Response Message—REST-based Services. . . . .	14-9
SIIInquiryExceptionRequest . . . . .	14-9
Request Message—SOAP-based Services . . . . .	14-10
Response Message—SOAP-based Services . . . . .	14-10
Request Message—REST-based Services . . . . .	14-10
Response Message—REST-based Services. . . . .	14-11

---

## **Chapter 15 • Stop Payments**

Stop Payment Process . . . . .	15-1
Stop Payment Operations. . . . .	15-2
SISubmitStopPaymentOrderRequest . . . . .	15-2
Request Message—SOAP-based Services . . . . .	15-2
Response Message—SOAP-based Services . . . . .	15-2
Request Message—REST-based Services . . . . .	15-3
Response Message—REST-based Services. . . . .	15-3
SIGetStopPaymentDetailsRequest . . . . .	15-3
Request Message—SOAP-based Services . . . . .	15-3
Response Message—SOAP-based Services . . . . .	15-4
Request Message—REST-based Services . . . . .	15-4
Response Message—REST-based Services. . . . .	15-4
SISubmitStopPaymentInquiryRequest . . . . .	15-4
Request Message—SOAP-based Services . . . . .	15-5
Response Message—SOAP-based Services . . . . .	15-5
Request Message—REST-based Services . . . . .	15-6
Response Message—REST-based Services. . . . .	15-7

## **Chapter 16 • Accounting Entry Memos**

AEM Process . . . . .	16-1
AEM Operations . . . . .	16-1
SISubmitAccountingEntryRequest . . . . .	16-2
Request Message—SOAP-based Services . . . . .	16-2
Response Message—SOAP-based Services . . . . .	16-2
Request Message—REST-based Services . . . . .	16-3
Response Message—REST-based Services. . . . .	16-3
SIGetAccountingEntryDetailsRequest . . . . .	16-4

---

Request Message—SOAP-based Services . . . . .	16-4
Response Message—SOAP-based Services . . . . .	16-4
Request Message—REST-based Services . . . . .	16-4
Response Message—REST-based Services. . . . .	16-5

## **Chapter 17 • Queues**

Queue Process. . . . .	17-6
Queue Actions. . . . .	17-6
Queue Operations . . . . .	17-8
SIGetQueueRequest . . . . .	17-8
Request Message—SOAP-based Services . . . . .	17-8
Response Message—P Format—SOAP-based Services . . . . .	17-9
Response Message—R Format—SOAP-based Services . . . . .	17-10
Response Message—S Format—SOAP-based Services . . . . .	17-11
Response Message—T Format—SOAP-based Services . . . . .	17-12
Response Message—K Format—SOAP-based Services . . . . .	17-13
Response Message—D Format—SOAP-based Services . . . . .	17-14
Response Message—J Format—SOAP-based Services . . . . .	17-15
Response Message—V Format—SOAP-based Services . . . . .	17-16
Response Message—Z Format—SOAP-based Services . . . . .	17-17
Response Message—H Format—SOAP-based Services . . . . .	17-18
Request Message—REST-based Services . . . . .	17-19
Response Message—P Format—REST-based Services . . . . .	17-19
Response Message—R Format—REST-based Services . . . . .	17-20
Response Message—S Format—REST-based Services . . . . .	17-21
Response Message—T Format—REST-based Services . . . . .	17-22
Response Message—D Format—REST-based Services . . . . .	17-23
Response Message—J Format—REST-based Services. . . . .	17-24

---

Response Message—V Format—REST-based Services . . . . .	17-25
Response Message—Z Format—REST-based Services . . . . .	17-26
Response Message—K Format—REST-based Services . . . . .	17-27
Response Message—H Format—REST-based Services . . . . .	17-28
Remove Transactions From Queue . . . . .	17-28
SICloseRFCRequest . . . . .	17-28
Request Message—SOAP-based Services . . . . .	17-29
Response Message—SOAP-based Services . . . . .	17-29
Request Message—REST-based Services . . . . .	17-29
Response Message—REST-based Services. . . . .	17-29
SICloseMiscFeeRequest . . . . .	17-29
Request Message—SOAP-based Services . . . . .	17-30
Response Message—SOAP-based Services . . . . .	17-30
Request Message—REST-based Services . . . . .	17-30
Response Message—REST-based Services. . . . .	17-30
SICloseMessageRequest. . . . .	17-30
Request Message—RPP—SOAP-based Services. . . . .	17-31
Response Message—RPP—SOAP-based Services . . . . .	17-31
Request Message—RPP—REST-based Services . . . . .	17-31
Response Message—RPP—REST-based Services . . . . .	17-31
SICloseTransactionRequest . . . . .	17-31
Request Message—SOAP-based Services . . . . .	17-32
Response Message—SOAP-based Services . . . . .	17-32
Request Message—REST-based Services . . . . .	17-32
Response Message—REST-based Services. . . . .	17-33
SIAcceptDisputeRequest. . . . .	17-33
Request Message—SOAP-based Services . . . . .	17-34
Response Message—SOAP-based Services . . . . .	17-35

Request Message—REST-based Services . . . . .	17-35
Response Message—REST-based Services . . . . .	17-35
Ignore Rejected Transactions . . . . .	17-35
Ignore Reject Operation . . . . .	17-36
SIGIgnoreRejectRequest . . . . .	17-36
Request Message—SOAP-based Services . . . . .	17-36
Response Message—SOAP-based Services . . . . .	17-36
Request Message—REST-based Services . . . . .	17-37
Response Message—REST-based Services . . . . .	17-37
SISIGIgnoreGMFPRequest . . . . .	17-37
Change Queue Status . . . . .	17-38
Change Queue Status Operation . . . . .	17-38
SIGChangeQueueStatusRequest . . . . .	17-39
Request Message—SOAP-based Services . . . . .	17-39
Response Message—SOAP-based Services . . . . .	17-39
Request Message—REST-based Services . . . . .	17-39
Response Message—REST-based Services . . . . .	17-39

## Chapter 18 • RTSI Batch Queues

Batch Queue Process . . . . .	18-3
Retrieve Transactions From a Batch Queue . . . . .	18-3
Mark Transactions as <i>Read</i> . . . . .	18-3
Batch Queue Operations . . . . .	18-4
SIGGetBatchQueueRequest (Retrieving Transaction From a Queue) . . . . .	18-4
Request Message—SOAP-based Services . . . . .	18-5
Response Message—SOAP-based Services . . . . .	18-5
Response Message—Action—SOAP-based Services . . . . .	18-7
Request Message—REST-based Services . . . . .	18-9

---

Response Message—REST-based Services. . . . .	18-9
Response Message—Action—REST-based Services . . . . .	18-11
SIMarkBatchQueueItemAsReadRequest (Marking a Transaction as <i>Read</i> ) . . . . .	18-13
Request Message—SOAP-based Services . . . . .	18-14
Response Message—SOAP-based Services . . . . .	18-14
Request Message—REST-based Services . . . . .	18-15
Response Message—REST-based Services. . . . .	18-15

## Chapter 19 • Hypersearch Requests

Hypersearch Process . . . . .	19-1
Hypersearch Operation . . . . .	19-2
Request Message—SOAP-based Services . . . . .	19-3
Response Message—SOAP-based Services . . . . .	19-3
Request Message—REST-based Services. . . . .	19-5
Response Message—REST-based Services . . . . .	19-5
Hypersearch Response Elements . . . . .	19-6

## Chapter 20 • Messages

Message Process . . . . .	20-1
Send Message Initiate . . . . .	20-2
Retrieve Message and Message Responses. . . . .	20-3
Send Message Respond. . . . .	20-3
Message Operations . . . . .	20-4
SICreateMessageRequest . . . . .	20-4
Request Message—Initiate RPP Message—SOAP-based Services . . . . .	20-4
Response Message—Initiate RPP Message—SOAP-based Services . . . . .	20-4
Request Message—Initiate RPP Message—REST-based Services. . . . .	20-5
Response Message—Initiate RPP Message—REST-based Services . . . . .	20-5

---

Request Message—Respond to RPP Message—SOAP-based Services . . . . .	20-5
Response Message—Respond to RPP Message—SOAP-based Services . . . . .	20-6
Request Message—Respond to RPP Message—REST-based Services . . . . .	20-7
Response Message—Respond to RPP Message—REST-based Services . . . . .	20-7
<b>SIGetMessageRequest . . . . .</b>	<b>20-8</b>
Request Message—SOAP-based Services . . . . .	20-8
Response Message—Initiate RPP Message—SOAP-based Services . . . . .	20-8
Response Message—Respond to RPP Message, including the history of the Message thread—SOAP-based Services . . . . .	20-9
Request Message—REST-based Services . . . . .	20-9
Response Message—Initiate RPP Message—REST-based Services . . . . .	20-10
Response Message—Respond to RPP Message, including the history of the Message thread—REST-based Services . . . . .	20-11
Add Image to Message Transaction . . . . .	20-11
<b>SIUploadMessageImageRequest . . . . .</b>	<b>20-12</b>
Request Message—SOAP-based Services . . . . .	20-12
Response Message—SOAP-based Services . . . . .	20-12
Request Message—REST-based Services . . . . .	20-13
Response Message—REST-based Services. . . . .	20-13
<b>SIGetMessageImageRequest . . . . .</b>	<b>20-13</b>
Request Message—SOAP-based Services . . . . .	20-14
Response Message—SOAP-based Services . . . . .	20-14
Request Message—REST-based Services . . . . .	20-14
Response Message—REST-based Services. . . . .	20-15

## **Chapter 21 • Purchase Inquiries**

Purchase Inquiry Process . . . . .	21-2
Request Latest PI Data from Existing Case . . . . .	21-2
Request Latest PI Data from TI Result if No Case Exists . . . . .	21-2

---

Retrieve Historical PI Data in Existing Case . . . . .	21-3
Retrieve Merchant Notification Details . . . . .	21-4
Purchase Inquiry Operations. . . . .	21-4
SISubmitPurchaseInquiryRequest . . . . .	21-4
Request Message—SOAP-based Services . . . . .	21-5
Response Message—SOAP-based Services . . . . .	21-6
Request Message—REST-based Services . . . . .	21-6
Response Message—REST-based Services. . . . .	21-7
SIGetPurchaseInquiryRequest . . . . .	21-7
Request Message—SOAP-based Services . . . . .	21-8
Response Message—SOAP-based Services . . . . .	21-8
Request Message—REST-based Services . . . . .	21-8
Response Message—REST-based Services. . . . .	21-9
SIGetMerchantNotificationDetailsRequest . . . . .	21-9
Request Message—SOAP-based Services . . . . .	21-9
Response Message—SOAP-based Services . . . . .	21-10
Request Message—REST-based Services . . . . .	21-10
Response Message—REST-based Services. . . . .	21-11

## **Chapter 22 • Collaboration**

Collaboration Types. . . . .	22-1
Collaboration Background . . . . .	22-2
Collaboration Process . . . . .	22-2
Send Collaboration Initiate. . . . .	22-3
Retrieve Collaboration or Collaboration Response . . . . .	22-3
Send Collaboration Response (Recipient) or Continue Collaboration (Initiator) . .	22-4
Send Collaboration Recall (Initiator only). . . . .	22-4
Collaboration Operations . . . . .	22-4

SISubmitCollaborationRequest . . . . .	22-4
Request Message—Initiate or Continue Collaboration (Initiator)—SOAP-based Services . . . . .	22-5
Request Message—Respond to Collaboration (Recipient)—SOAP-based Services . . . . .	22-7
Request Message—Initiate or Continue Collaboration (Initiator)—REST-based Services . . . . .	22-12
Request Message—Respond to Collaboration (Recipient)—REST-based Services . . . . .	22-13
Request Message—Recall Collaboration (Initiator only)—SOAP-based Services . . . . .	22-16
Response Message—SOAP-based Services . . . . .	22-16
Request Message—Recall Collaboration (Initiator only)—REST-based Services . . . . .	22-17
Response Message—REST-based Services . . . . .	22-17
SIGetCollaborationDetailsRequest . . . . .	22-17
Request Message—SOAP-based Services . . . . .	22-18
Response Message—Initiate or Continue Collaboration (Initiator)—SOAP-based Services . . . . .	22-18
Response Message—Respond to Collaboration (Recipient)—SOAP-based Services . . . . .	22-20
Response Message—Recall Collaboration—SOAP-based Services . . . . .	22-21
Request Message—REST-based Services . . . . .	22-21
Response Message—Initiate or Continue Collaboration (Initiator)—REST-based Services . . . . .	22-22
Response Message—Respond to Collaboration (Recipient)—REST-based Services . . . . .	22-23
Response Message—Recall Collaboration—REST-based Services . . . . .	22-24

## Chapter 23 • Dispute Quick Forms

Dispute Quick Form Process . . . . .	23-1
Initiate Dispute Quick Form . . . . .	23-1

---

Repair Dispute Quick Form. . . . .	23-2
Dispute Quick Form Operation. . . . .	23-3
SISubmitQFDisputeQuestionnaireRequest . . . . .	23-3
Request Message—Fraud—SOAP-based Services . . . . .	23-4
Request Message—Authorization—SOAP-based Services . . . . .	23-6
Request Message—Processing Error—SOAP-based Services . . . . .	23-6
Request Message—Consumer—SOAP-based Services . . . . .	23-8
Response Message—SOAP-based Services . . . . .	23-9
Request Message—Fraud—REST-based Services . . . . .	23-10
Request Message—Authorization—REST-based Services . . . . .	23-11
Request Message—Processing Error—REST-based Services . . . . .	23-12
Request Message—Consumer—REST-based Services. . . . .	23-13
Response Message—REST-based Services. . . . .	23-14

## **Chapter 24 • RTSI Report Distribution**

RTSI Report Distribution Process . . . . .	24-1
Report Distribution Operations. . . . .	24-1
SIGetActiveScheduleRequest . . . . .	24-2
Request Message—SOAP-based Services . . . . .	24-2
Response Message—SOAP-based Services . . . . .	24-2
Request Message—REST-based Services . . . . .	24-3
Response Message—REST-based Services. . . . .	24-3
SIRetrieveReportOutputRequest . . . . .	24-3
Request Message—SOAP-based Services . . . . .	24-4
Response Message—SOAP-based Services . . . . .	24-4
Request Message—REST-based Services . . . . .	24-4
Response Message—REST-based Services. . . . .	24-5

**Chapter 25 • Visa Fraud Monitoring Program**

Auto Disputes on VFMP-Reported Transactions . . . . .	25-1
VFMP Process . . . . .	25-2
Retrieve VFMP Transactions . . . . .	25-2
Retrieve VFMP Transaction Details . . . . .	25-2
VFMP Operations. . . . .	25-3
SIGetGMFPDetailsRequest . . . . .	25-3

**Chapter 26 • Security**

Password Policy . . . . .	26-1
Encryption. . . . .	26-1
Authentication . . . . .	26-1
Password Types . . . . .	26-2
Password Requirements . . . . .	26-2
Visa Security Policies . . . . .	26-3
RTSI Security . . . . .	26-3
Client Requirements . . . . .	26-4
System ID . . . . .	26-4
TrustStore and KeyStore . . . . .	26-4

**Appendix A • Systems Interface URLs**

Systems Interface Request URL. . . . .	A-1
--	-----

**Appendix B • Miscellaneous Useful Information**

Interface Element Spreadsheet (IES) . . . . .	B-1
Existing Members . . . . .	B-2
New Members . . . . .	B-2
Member Testing . . . . .	B-3

---

Common Issues With SI Fields . . . . .	B-3
Locked Fields . . . . .	B-4
Japanese Character Handling . . . . .	B-5
Comparison of <TransactionInformation> vs. TI Details XML Fields . . . . .	B-6

## **Appendix C • XML Processing**

XML Validation Options . . . . .	C-1
XML Usage . . . . .	C-2
Data Element Sequence . . . . .	C-2
Dates and Timestamps . . . . .	C-3
Dates . . . . .	C-3
Timestamps . . . . .	C-3
More XML Resources . . . . .	C-3

## **Appendix D • Sample Code**

Performing Basic Authentication and Assigning Client Certificate . . . . .	D-1
Submitting Images . . . . .	D-1
SOAP Binding . . . . .	D-1
Remote Service Stub . . . . .	D-2
Service . . . . .	D-5
Operations . . . . .	D-5
Client . . . . .	D-8
Receiving Images . . . . .	D-8
SOAP Binding . . . . .	D-8
Remote Service Stub . . . . .	D-10
Service . . . . .	D-11
Operations . . . . .	D-11
Client . . . . .	D-13

HTTP Headers . . . . .	D-13
------------------------	------

## Appendix E • Web Service Clients

Tools . . . . .	E-1
Architecture . . . . .	E-1
Building the Client . . . . .	E-1
Convert WSDL Files . . . . .	E-1
Implement SOAP Handlers . . . . .	E-2
Enable MTOM/XOP for Image Operations . . . . .	E-2
HTTP Headers . . . . .	E-3
Client With XOP, MTOM Binding . . . . .	E-3
Client With download XOP and MTOM Element . . . . .	E-4
Submitting RFC Fulfillment Images . . . . .	E-4
Image Handling Code . . . . .	E-4

## Appendix F • Sample Project Plan

## Appendix G • System Diagnostic Report

## Appendix H • Alternative Web Services

Alternative Transaction Inquiry Web Services. . . . .	H-1
Transaction Inquiry Using One or More VisaNet Transaction IDs . . . . .	H-1
Overview. . . . .	H-1
Procedure . . . . .	H-1
Key Elements and Sample Code for SISubmitMultiTranInquiryRequest . . . . .	H-2
Get Transaction Details Using One or More VisaNet Transaction IDs. . . . .	H-5
Overview. . . . .	H-5
Procedure . . . . .	H-6

---

Key Elements and Sample Code for SIGetAllTransDetailsRequest . . . . .	H-6
Visa Cardholder Purchase Inquiry . . . . .	H-15
Introduction . . . . .	H-15
<b>Visa Cardholder Purchase Inquiry (VCPI) Request and Response</b> . . . . .	H-15
<b>Visa Cardholder Purchase Inquiry (VCPI) Unsuccessful Response</b> . . . . .	H-16
VCPI Operations . . . . .	H-16
SICardholderPurchaseInquiryRequest . . . . .	H-17
SIGetCardholderPurchaseInquiryRequest . . . . .	H-22

## Appendix I • RTSI Implementation and Testing Checklist

Overview . . . . .	I-1
Checklist . . . . .	I-1
Step 1: Assess if a System Interface Solution Is Appropriate. . . . .	I-1
Step 2: Prepare and Identify Internal Project Teams and Resources . . . . .	I-2
Step 3: Development of Your Host System . . . . .	I-2
Step 4: Prepare For Visa Testing and Validation by Reviewing All Testing Training and Documentation . . . . .	I-3
Step 5: Submit the System Interface Application Form. . . . .	I-3
Step 6: Coordinate Visa Configuration and Testing . . . . .	I-4
Step 7: Implement the New Connection . . . . .	I-5

## Appendix J • RTSI Optimization Points

Additional Important Points to Consider in RTSI . . . . .	J-1
SOAP and REST Web Services . . . . .	J-1
Error, Warning, and Information Messages . . . . .	J-1
Image Upload Methods . . . . .	J-2
Ways to Upload Images . . . . .	J-2
Dispute Submission Service . . . . .	J-2
When Member Needs to Pull Associated Transactions . . . . .	J-2

---

When Member Needs to Submit Supporting Functions . . . . .	J-3
Supporting Services vs Dispute Flow Services . . . . .	J-3
Hypersearch . . . . .	J-3
Queues vs Download. . . . .	J-3
Batch to RTSI Synchronous Processing Migration. . . . .	J-3
Best Practices . . . . .	J-4
Dispute Submission . . . . .	J-4
Queues . . . . .	J-4
Batch Queues . . . . .	J-5

## Glossary

## Figures

1-1:	RTSI Flow—Overview . . . . .	1-5
1-2:	Dispute Process Flow . . . . .	1-6
4-1:	Transaction Inquiry—Flow 1 . . . . .	4-3
4-2:	Transaction Inquiry—Flow 1a . . . . .	4-4
4-3:	Transaction Inquiry—Flow 1b . . . . .	4-5
4-4:	Transaction Inquiry—Flow 1c . . . . .	4-6
7-1:	Initiate Dispute Flow. . . . .	7-4
7-2:	Submit Dispute—Flow 1 . . . . .	7-5
7-3:	Submit Dispute—Flow 1a . . . . .	7-6
7-4:	Submit Dispute—Flow 1b . . . . .	7-7
7-5:	Submit Dispute—Flow 1c . . . . .	7-8
7-6:	Submit Dispute—Flow 1d. . . . .	7-9
7-7:	Submit Dispute—Flow 1e . . . . .	7-10
7-8:	Submit Dispute—Flow 1f . . . . .	7-11
8-1:	Dispute Financials—Flow 1a . . . . .	8-3
8-2:	Dispute Financials—Flow 1b . . . . .	8-4
10-1:	Respond to Incoming Questionnaires—Flow 1a . . . . .	10-3
10-2:	Respond to Incoming Questionnaires—Flow 1b. . . . .	10-4
10-3:	Respond to Incoming Questionnaires—Flow 1c. . . . .	10-5

10-4:	Respond to Incoming Questionnaires—Flow 1d.	10-6
10-5:	Respond to Incoming Questionnaires—Flow 1e.	10-7
10-6:	Respond to Incoming Questionnaires—Flow 1f.	10-8
10-7:	Respond to Incoming Questionnaires—Flow 1g.	10-9
10-8:	Respond to Incoming Questionnaires—Flow 1h.	10-10
10-9:	Dispute Pre-Compliance—Flow 1a.	10-17
10-10:	Dispute Pre-Compliance—Flow 1b	10-18
10-11:	Dispute Pre-Compliance—Flow 1c.	10-19
10-12:	Dispute Pre-Compliance—Flow 1d	10-20
10-13:	Dispute Pre-Compliance—Flow 1e.	10-21
10-14:	Dispute Pre-Compliance—Flow 1f.	10-22
11-1:	Dispute Case Filing Process Flow	11-2
11-2:	Dispute Case Filing—Flow 1a	11-5
11-3:	Dispute Case Filing—Flow 1b/1c	11-6
11-4:	Dispute Case Filing—Flow 1d	11-7
11-5:	Dispute Case Filing—Flow 1e	11-8
11-6:	Dispute Case Filing—Flow 1f.	11-9
11-7:	Dispute Case Filing—Flow 1g	11-10
B-1:	MTE Home Page	B-2
B-2:	Locked Fields (1 of 2)	B-4
B-3:	Locked Fields (2 of 2)	B-5
F-1:	Project Approach	F-1
G-1:	System Diagnostic Report	G-2
G-2:	System Diagnostic Report — Results.	G-3

## Tables

A:	Document Conventions.	4
1-1:	Transaction Types Supported by VROL	1-2
2-1:	TIFF Field Specifications	2-4
2-2:	VROL Image Support	2-5
3-1:	Important Attributes.	3-2
3-2:	Transaction Operations.	3-3
3-3:	Transaction Actions	3-3
3-4:	Case matching fields when <VisaCaseNumber> is not used	3-10
7-1:	Available Requests and Responses	7-2
8-1:	Financial Fields.	8-21
17-1:	Queues	17-2
18-1:	Types of Batch Queues.	18-1
19-1:	Hypersearch Response Elements	19-6
A-1:	URL Formats	A-2
B-1:	<TransactionInformation> vs. TI Details Fields—Manual Transaction Data Entry Form	B-6
B-2:	<TransactionInformation> vs. TI Details—Case Details	B-9
F-1:	Sample Project Plan	F-2
H-1:	VCPI Search Parameters	H-16

THIS PAGE INTENTIONALLY LEFT BLANK.

# About This Guide

The *Real Time System Interface for VROL* describes the Real-Time Systems Interface (RTSI) and provides specifications for integrating this interface alternative into a member host system (MHS), dispute tracking, or back-office system.

Code samples are provided for both requests and responses.

## Audience

This guide is intended for technical designers, system developers, database engineers, and project managers who are developing dispute tracking systems that are to be integrated with the Visa Resolve Online Real-Time Systems Interface.

## Assumptions

This document assumes that the reader has a strong understanding of the dispute process, as well as, an understanding of how disputes are tracked and processed by the member's particular system or systems. It is also assumed that the client has a developer-level understanding of XML and XML schema.

It is assumed that the member has obtained a Visa Online (VOL) System ID that has been entitled to access Visa Resolve Online (VROL). A valid VOL user ID and password allows the user to access VROL through any of the three VROL access mechanisms: User Interface (UI), RTSI, and Bulk Systems Interface (SI). For Bulk SI and RTSI, an additional setup is required in the user's org.

## Summary of Changes

Where appropriate, this version of the *Real Time System Interface for VROL* contains information to support the mandatory and optional changes detailed in the *October 2019 Global Technical Letter and Implementation Guide*.

**Note:** Refer to the IES (Interface Element Specification) and Visa Resolve Online Technical Suite Package for additional changes and further details.

## Changes

### Service/Function/Description

Added new sections for Visa Risk Manager (VRM) and ATR enhancements.

Updated existing dispute functionalities for enhancements.

Corrected or updated information for existing functionalities.

Updated existing samples for corrections and enhancements.

Replaced "Global Arbitration and Compliance (GAC)" with "Dispute Resolution Management (DRM)".

Added Format P queue.

Added information about the new Rapid Dispute Resolution (RDR) process.

Added new URLs in [Appendix A, Systems Interface URLs](#).

Added information about attaching images to questionnaires and existing cases.

Changed "Visa Europe" to "Europe".

Updated [Step 7: Implement the New Connection](#) for contact information.

## Document Organization

The *Real Time System Interface for VROL* is organized as follows:

- [\*\*Chapter 1, Service Overview\*\*](#)—Provides an introduction to VROL and the transaction types it supports. Describes the VROL organization types, setup required, and dispute resolution process.
- [\*\*Chapter 2, RTSI Overview\*\*](#)—Provides a summary of the VROL functions supported by RTSI and describes the requirements and changes that members may need to make to their MHS to access and use VROL.
- [\*\*Chapter 3, RTSI Request Operations\*\*](#)—Describes the operations and actions available with RTSI request messages.
- [\*\*Chapter 4, Transaction Inquiries\*\*](#)—Describes how to request and receive information about customer transactions.
- [\*\*Chapter 5, Case Creation\*\*](#)—Describes the process and provides XML samples for creating a case from a transaction inquiry or manually creating a new (empty) case, updating an existing case, or adding an image to a case.
- [\*\*Chapter 6, Request for Copies\*\*](#)—Describes how to perform copy request and fulfillment functions.
- [\*\*Chapter 7, Dispute Questionnaires\*\*](#)—Describes the dispute functions available in RTSI to initiate and respond to disputes for dispute allocation and collaboration workflows.
- [\*\*Chapter 8, Dispute Financials\*\*](#)—Describes how VROL handles assignment of dispute financials and retrieval of dispute financial messages in dispute allocation and collaboration workflows.
- [\*\*Chapter 9, Adjustments\*\*](#)—Describes how to create and process adjustments.
- [\*\*Chapter 10, Dispute Pre-Filings\*\*](#)—Describes the functions available to initiate pre-arbitration and pre-compliance.
- [\*\*Chapter 11, Dispute Case Filings\*\*](#)—Describes how to submit dispute case filing requests for both arbitration and compliance.

- [\*\*Chapter 12, Miscellaneous Fees\*\*](#)—Describes the functions RTSI supports to manage miscellaneous fees.
- [\*\*Chapter 13, Fraud Reporting\*\*](#)—Describes how to submit and receive fraud report information.
- [\*\*Chapter 14, Exception File Listings\*\*](#)—Describes how to submit and receive exception file listing information.
- [\*\*Chapter 15, Stop Payments\*\*](#)—Describes how to add, update, delete, and view a stop payment request.
- [\*\*Chapter 16, Accounting Entry Memos\*\*](#)—Describes how to create and update AEMs, which record how issuers and acquirers track their internal general ledger entries for a case.
- [\*\*Chapter 17, Queues\*\*](#)—Describes how to use the SIGetQueue operation to manage information in VROL.
- [\*\*Chapter 18, RTSI Batch Queues\*\*](#)—Describes how to use the SIGetBatchQueueRequest operation to manage transactions in the RTSI batch queue.
- [\*\*Chapter 19, Hypersearch Requests\*\*](#)—Describes how to perform searches for single or multiple cases in VROL.
- [\*\*Chapter 20, Messages\*\*](#)—Describes how issuers and acquirers internal to VROL can exchange messages.
- [\*\*Chapter 21, Purchase Inquiries\*\*](#)—Describes how to use the Purchase Inquiry (PI) transaction to provide relevant purchase information from participating merchants.
- [\*\*Chapter 22, Collaboration\*\*](#)—Describes how issuers and acquirers internal to VROL can exchange collaboration transactions.
- [\*\*Chapter 23, Dispute Quick Forms\*\*](#)—Describes how issuers can quickly create and submit dispute questionnaires.
- [\*\*Chapter 24, RTSI Report Distribution\*\*](#)—Describes how users can access the details of their scheduled reports and download the reports in CSV format.
- [\*\*Chapter 25, Visa Fraud Monitoring Program\*\*](#)—Describes how issuers can use the Visa Fraud Monitoring Program (VFMP) functionality for disputes.
- [\*\*Chapter 26, Security\*\*](#)—Describes how security is implemented in the Real-Time Systems Interface.
- [\*\*Appendix A, Systems Interface URLs\*\*](#)—Lists URLs to use for production and Member Test Environment (MTE).
- [\*\*Appendix B, Miscellaneous Useful Information\*\*](#)—Provides information about the Interface Element Spreadsheet, Member Testing, Locked Fields, common issues with SI fields, handling of Japanese characters, and comparison of <TransactionInformation> vs. TI Details XML fields.
- [\*\*Appendix C, XML Processing\*\*](#)—Provides members with information about parsing options, validation procedures, and XML usage.
- [\*\*Appendix D, Sample Code\*\*](#)—Provides sample code associated with image handling and HTTP headers used with RTSI.
- [\*\*Appendix E, Web Service Clients\*\*](#)—Describes how to write the communications components of a Java-based Web services client.
- [\*\*Appendix F, Sample Project Plan\*\*](#)—Provides a sample project plan.
- [\*\*Appendix G, System Diagnostic Report\*\*](#)—Describes the System Diagnostic Report feature.

- [\*\*Appendix H, Alternative Web Services\*\*](#)—Provides the alternative Web services for existing functionalities.
- [\*\*Appendix J, RTSI Optimization Points\*\*](#)—Describes the additional important points to consider when implementing RTSI as well as the recommended best practices for optimizing RTSI.
- [\*\*Glossary\*\*](#)

## Related Publications

VROL documentation can be obtained from Visa. To acquire these documents, contact your Visa representative.

For information regarding other VROL features, refer to the following publications:

- *Real Time System Interface for VROL: Interface Element Specification (IES)*
- *Visa Resolve Online Documentation Roadmap*

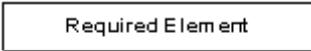
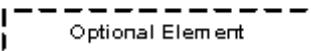
Copies of the forms required to sign-up for RTSI may be obtained as follows:

- The Visa Resolve Online Enrollment Form can be obtained from Visa OnLine (U.S. only).
- Visa Online IDs can be obtained at [www.volenvroll.com](http://www.volenvroll.com) (U.S. only). Simple ID is also acceptable.

## Document Conventions

[\*\*Table A\*\*](#) shows the document conventions used in this publication.

**Table A: Document Conventions**

Convention	Purpose
<b>bold</b>	Calls out menu names or choices; also highlights programming elements.
<i>italics</i>	Used for: <ul style="list-style-type: none"> <li>• Document titles; emphasis.</li> <li>• Explaining an unusual term the first time it appears.</li> </ul>
<XMLElement> in brackets	Indicates XML elements.
Arial typeface	Indicates code examples.
<b>Note:</b>	Provides more information about the preceding topic.
<b>Important:</b>	Highlights important information in the text.
text in quote marks	Used to refer to section names in a chapter.
Grayed-out code	In the code examples, code that is not particularly relevant to the example being discussed.
 	Required element (solid line) and optional element (dashed line) in Schema Overview.

**Table A: Document Conventions (cont'd.)**

Convention	Purpose
n/a	Stands for <i>not applicable</i> . Also used to indicate that there is not any information.
Courier New	Used for email addresses and for URLs.

## Website Links

This publication contains hyperlinks to websites. While at the time of publication all these links were current, the dynamic nature of the Internet implies that, over time, some of these websites may cease operation or change their URLs.

THIS PAGE INTENTIONALLY LEFT BLANK.

# Service Overview

1

Visa Resolve Online (VROL) is a Web-based service that enables Visa members to expedite the dispute resolution process. Member uses VROL to retrieve transaction information online, receive dispute financial liability, exchange information and documentation electronically, submit pre-filing and case filings electronically, receive financial messages, report fraud, and manage Exception File and Stop Payment Listings.

**Note:** *Each region may offer a different complement of functions and features. Contact your representative to determine what is available in your region.*

## Systems Interfaces

VROL systems interfaces include two integration points to VROL. Each uses Extensible Markup Language (XML) to structure the content of messages that are transferred between members and VROL.

- A Real-Time Systems Interface (RTSI) for submission of disputes and dispute questionnaires, as well as retrieval of dispute responses and dispute financials (dispute financial details, dispute financial reversals). Transaction Inquiry (TI), Request for Copy (RFC), exception file listings, fraud report information, and so forth.  
For more information on the VROL functions supported by RTSI, see [Table 1–1](#).
- A Bulk Systems Interface (Bulk SI) for bulk transfer of files related to dispute resolution.

## VROL Organization Types

VROL distinguishes organizations as:

- Issuer
- Acquirer
- Issuer and acquirer (dual organization <sup>1</sup>)

The organization may be a Visa member, a processor, or any one of a variety of Visa groups that require access to VROL.

---

1. The term *dual organization* is sometimes used in VROL documentation to mean an organization that does both issuing and acquiring business.

VROL uses member type to define the member's organization. The value for member type (issuer, acquirer, or issuer and acquirer) is used by VROL to determine the roles and actions that the member can perform.

VROL uses the <MemberRole> element with RTSI requests as follows.

- If the submitting member's organization is not issuer and acquirer, <MemberRole> is not required.
- If the submitting member's organization is issuer and acquirer, this value is required. The value should contain 'I' when the submitting member is acting as an issuer and 'A' when acting as an acquirer.

VROL returns the memberRole attribute in RTSI responses. The value is 'I' when the receiving organization is receiving the case as an issuer and 'A' when the receiving organization is acting as an acquirer.

## VROL Organization Setup

VROL provides a set of UI screens that are used to configure how a member wants VROL to process transactions and information about users of the system. Some processing information can only be entered and updated by the Visa Administrator. Other information can be updated by member staff with the role needed to perform the administration functions.

Administration functions generally include access entitlement information, activation dates, contact information, and transaction processing settings.

Information that can be updated by the member's staff is called *member preferences*. All fields on the member preferences screens are also on the organization configuration screen and can be maintained by the Visa administrator.

Information about users of the system includes the user's name, contact information, types of cases that a user can work, permission regarding the submission of certain transaction types, queues that the user can view, and custom status values that a user can assign to a case. This information is maintained by the Visa administrator.

## Supported Transaction Types

[Table 1–1](#) provides a list of transaction types supported by VROL.

**Note:** "y" indicates that the transaction type is newly supported in this release.

**Table 1–1: Transaction Types Supported by VROL**

<b>Transaction Type</b>	<b>BASE II Members</b>			<b>SMS Members</b>		
	<b>Bulk SI</b>	<b>UI</b>	<b>RTSI</b>	<b>Bulk SI</b>	<b>UI</b>	<b>RTSI</b>
Dispute questionnaire	x <sup>1</sup>	x	x	x <sup>1</sup>	x	x
Dispute financial, Dispute financial reversal	x <sup>1, 8</sup>	x <sup>8</sup>	x <sup>8</sup>	x <sup>1, 8</sup>	x <sup>8</sup>	x <sup>8</sup>
Dispute response	x <sup>1</sup>	x	x	x <sup>1</sup>	x	x
Request for Copy (RFC)	x	x	x	x	x	x
Adjustment				x	x	x
Fraud	x	x	x	x	x	x

**Table 1–1: Transaction Types Supported by VROL (cont'd.)**

<b>Transaction Type</b>	<b>BASE II Members</b>			<b>SMS Members</b>		
	<b>Bulk SI</b>	<b>UI</b>	<b>RTSI</b>	<b>Bulk SI</b>	<b>UI</b>	<b>RTSI</b>
Exception File Listing (EFL)	x <sup>2</sup>	x	x	x <sup>2</sup>	x	x
Stop Payment (Also known as PPCS—Preauthorized Payment Cancellation Service)		x <sup>3</sup>	x <sup>3</sup>		x	x
Transaction inquiry (TI)		x	x		x	x
Fee collection, funds disbursement	x	x	x	x	x	x
Dispute Pre-filing (Pre-Arbitration, Pre-Compliance)	x <sup>1</sup>	x	x	x <sup>1</sup>	x	x
Dispute Case filing (Arbitration, Compliance)	x <sup>1</sup>	x	x	x <sup>1</sup>	x	x
VROL reports	x <sup>4</sup>	x	x <sup>12</sup>	x <sup>4</sup>	x	x <sup>12</sup>
Accounting Entry Memo (AEM)		x	x		x	x
Transaction history inquiry (fraud, EFL <sup>2</sup> , PPCS)		x			x	
Create (empty) case	x	x	x	x	x	x
RFC advices (TC 38)	x	x	x	x	x	x
UMF settlement advice (UMF equivalent of BASE II TC 33); Dispute financial and Dispute financial reversal status advice (SMS 0480) <sup>9</sup>	x <sup>1</sup>	x	x	x <sup>1</sup>	x	x
UMF settlement advice (UMF equivalent of BASE II TC 33); Dispute response financial and Dispute response financial reversal status advice (SMS 0282) <sup>9</sup>	x <sup>1</sup>	x	x	x <sup>1</sup>	x	x
SMS approval status				x	x	x
Quick forms	x <sup>6</sup>	x <sup>6</sup>	x <sup>10</sup>	x <sup>6</sup>	x <sup>6</sup>	x <sup>10</sup>
Queues, queue actions, batch queues		x	x <sup>7</sup>		x	x <sup>7</sup>
Request Proof of Posting (RPP) Messages	x	x	x	x	x	x
VROL Merchant Contact		x			x	
Dashboard (Queue Monitor)		x			x	
Visa Fraud Monitoring Program (VFMP) for Disputes	x <sup>13</sup>	x	x	x <sup>13</sup>	x	x
Purchase Inquiry (PI)		x	x		x	x
Collaboration (Inquiry, Good Faith, ASRR, and Card Recovery)	x	x	x <sup>5</sup>	x	x	x <sup>5</sup>
Cardholder Residence Files <sup>11</sup>		x			x	

- 
- <sup>1</sup> VROL does not allow initiation of these transactions through Bulk SI; however, VROL supports receipt/delivery of these transactions through Bulk SI if user is configured to receive them in their download file. Bulk SI upload is not supported for this transaction type.
  - <sup>2</sup> EFL inquiry is not available in Bulk SI; only add/delete/update functions are available.
  - <sup>3</sup> For the BASE II member, an additional BIN setting is required; no alternate routing required.
  - <sup>4</sup> Only a subset of reports is available in Bulk SI.
  - <sup>5</sup> Card Recovery collaboration type is not available in RTSI; only Inquiry, Good Faith, and ASRR collaboration types are available.
  - <sup>6</sup> RFCs, adjustments, and fraud only.
  - <sup>7</sup> Only a subset of queues is available in RTSI.
  - <sup>8</sup> VROL does not support creation of dispute financials/financial reversals. Dispute financials are either VROL-generated or created outside of VROL.
  - <sup>9</sup> UMF settlement advice (internal advices which contain all fields in the TC 33 advice plus some additional fields) is sent to VROL to indicate that the dispute financial/reversal or dispute response financial/reversal was settled.
  - <sup>10</sup> Supported for dispute questionnaires only.
  - <sup>11</sup> This functionality is available only for issuers with licensed branches and cardholders in the state of Nebraska.
  - <sup>12</sup> RTSI members can access the details of their scheduled report or request a report data download of their completed scheduled reports in CSV format through the RTSI Report Disrtribution functionality.
  - <sup>13</sup> Available in Bulk SI scheduled reports for issuers.

## Dispute Resolution Process

The dispute process is invoked when user decides to initiate a valid dispute for a particular transaction(s) and chooses one of the following dispute categories:

- **Fraud**—Applicable when the cardholder did not participate or authorize the transaction.
- **Authorization**—Applicable when the required authorization was not processed for the transaction.
- **Processing Error**—Applicable when the transaction was not appropriately processed per *Visa Rules*.
- **Consumer**—Applicable when the cardholder has a dispute with a merchant over the quality or receipt of goods or services.

**Note:** Although not a dispute category, VROL provides a "Fraud Bundle" function that allows initiators to dispute multiple fraudulent low-amount eligible transactions as a group. See [Submit and Retrieve Fraud Bundle Disputes](#) for details.

When a dispute is successfully submitted, VROL derives the dispute condition for the selected dispute category. The category and its derived condition align with the conditions described in the *Visa Rules*. For example, "10.3 – Fraud: Card Present Environment". In this example, "10" is the value that represents the dispute category "Fraud", while "3" is the value that represents the dispute condition (within Fraud).

For certain scenarios where a dispute condition has variations, VROL assigns a unique value to each dispute condition variation and stores that value in the database. VROL uses these unique values to access descriptions and record the specific dispute condition variation. However, when displaying the dispute category/condition, VROL drops the decimalized portion of the dispute condition value. For example, in the case of Processing Error disputes (12.6):

- "Processing Error – Duplicate" dispute category/condition is stored as dispute category = 12 and dispute condition = 6.1; however, its dispute category/condition value is displayed as "12.6 Processing Error – Duplicate Processing".
  - "Processing Error – Paid by Other Means" dispute category/condition is stored as dispute category = 12 and dispute condition = 6.2; however, its dispute category/condition value is displayed as "12.6 Processing Error – Paid by Other Means".

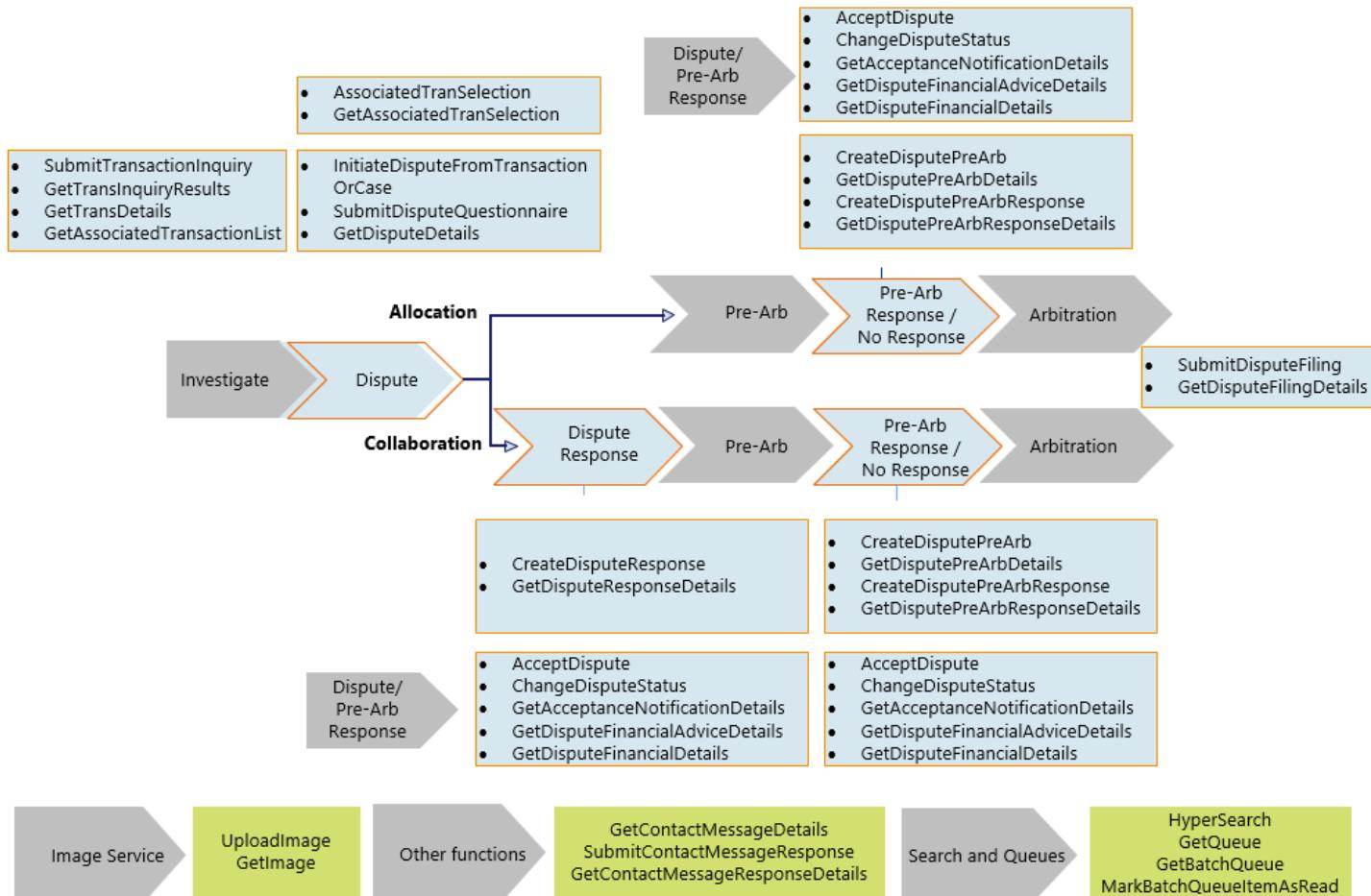
VROL routes valid disputes to one of the following two processes:

- **Allocation**—A workflow where VROL automatically determines liability based on transaction information provided, *Visa Rules*, and dispute category selected. Disputes processed with a Fraud or Authorization dispute category follow the allocation process.
  - **Collaboration**—A workflow where VROL does not determine and assign liability and an exchange of information between issuer and acquirer is required. Disputes processed with a Consumer or Processing Error dispute category follow this workflow.

## RTSI Flow—Overview

The following diagram illustrates an overview of the dispute resolution process in RTSI.

**Figure 1–1: RTSI Flow—Overview**

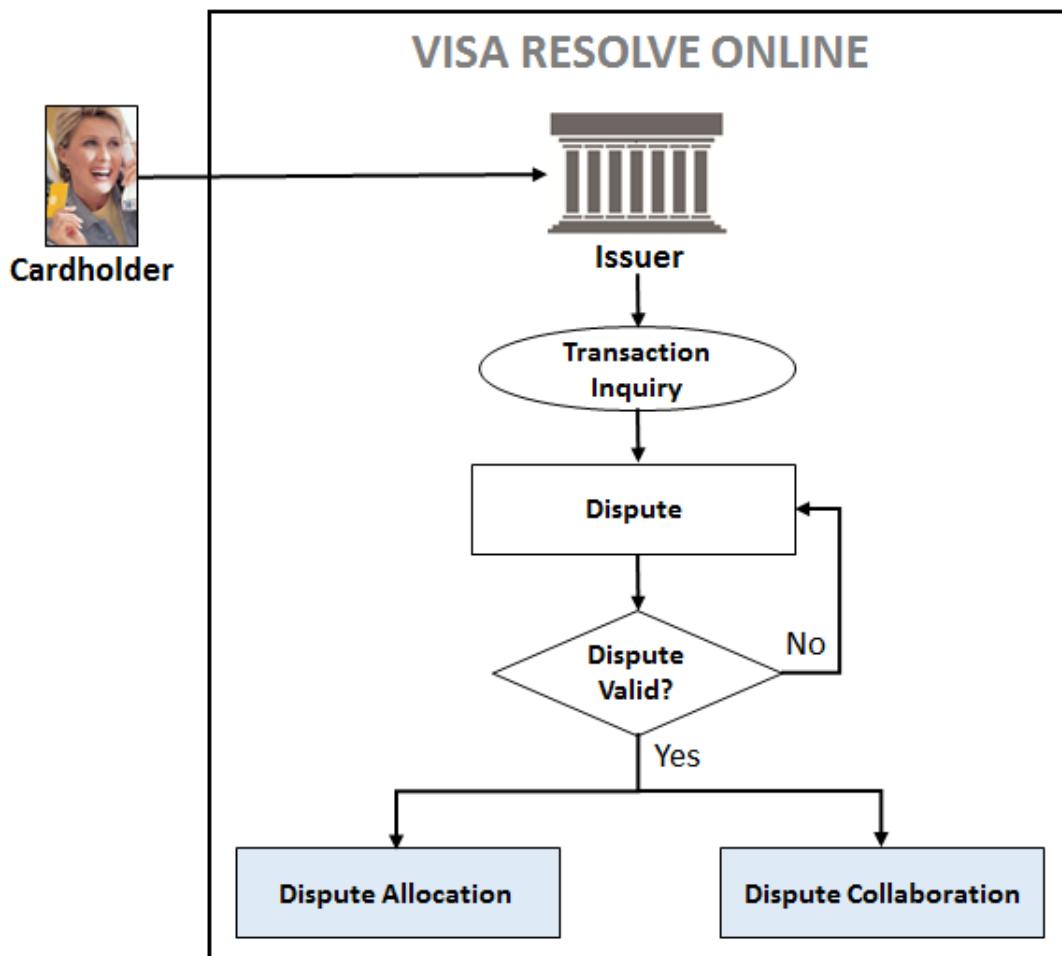


## Dispute Process Flow

The user who initiates the dispute is called the *initiator* and the recipient of dispute is called the *opposer*. Disputes are always initiated by issuers. Acquirers can initiate dispute-related Pre-Compliance as needed. For more information about dispute-related Pre-Compliance, see [Pre-Compliance Process](#).

The following figure shows the overall dispute flow:

**Figure 1-2: Dispute Process Flow**



**Note:** For more information on the dispute process, see [Chapter 7, Dispute Questionnaires](#).

In this case, issuer is the initiator of dispute, while acquirer is the opposer.

1. The process starts when a cardholder contacts the card issuer disputing a transaction posted against the cardholder's account.
2. The issuer searches for the transaction and associated transaction details from within the MHS. The MHS retrieves transaction information through VROL Transaction Inquiry.
3. The issuer initiates a dispute and selects a dispute category on an appropriate selected TI response.

Visa applies VROL system edits to validate if the dispute is valid.

- If the dispute is invalid, VROL edits block such requests and issuer can make appropriate corrections to resubmit the dispute request.
- If the dispute is valid, VROL generates a dispute financial or financial fields, depending on the organization's member configuration for dispute financials. Once the dispute financial advice is received, VROL makes the dispute questionnaire visible to the opposer either through an allocation or collaboration workflow.

Users can retrieve information about inbound dispute transactions by using the SIGetQueue or SIGetBatchQueue operation. VROL does not allow initiation of dispute transactions through Bulk SI; however, VROL supports receipt/delivery of these transactions through Bulk SI if user is configured to receive them in their download file.

## Allocation Dispute Process Workflow

**Note:** *The following workflow assumes that both issuer and acquirer are configured to have VROL generate financials on their behalf.*

1. The process starts when VROL determines that a submitted Authorization or Fraud dispute is valid.

VROL calculates and assigns financial liability, and systematically generates either the dispute financial message or financial fields depending on the member's configuration for dispute financials. VROL delivers the dispute questionnaire once a matching advice is received. At this point, the dispute is resolved. If the member objects to VROL's liability decision, they can proceed to the next step.

2. The member that received the financial liability may initiate a Pre-Arbitration if there is a valid reason to challenge the system's allocation decision.

VROL determines if the Pre-Arbitration is valid and if valid, then the Pre-Arbitration is made available to the opposing member in their queue for further action. No financial or financial fields are generated at this point.

3. The side receiving the Pre-Arbitration questionnaire can initiate a Pre-Arbitration response with a full or partial acceptance or a decline response.
  - a. If the Pre-Arbitration response is accept full or partial liability, VROL generates a dispute financial reversal or financial fields to adjust liability. VROL generates the financial reversal or financial fields depending on whether VROL or the member generated the original dispute financial. When the reversal financial clears, VROL delivers the Pre-Arbitration response to the opposing side.
  - b. If the Pre-Arbitration response is decline, then this response is made available to the other member in their queue for further action. No dispute financial reversal or financial fields are generated at this point.
4. For an incoming Pre-Arbitration response, the opposing member can either accept the Pre-Arbitration response and the dispute ends or continue to pursue the dispute by submitting an arbitration case filing.
5. An arbitration filed will be further adjudicated by Visa Dispute Resolution Management (DRM) team to give a final ruling.

## Collaboration Dispute Process Workflow

1. The process starts when VROL determines that a submitted Processing Error or Consumer dispute is valid.

VROL generates a dispute financial or financial fields. Once the matching advice is received, VROL makes the dispute questionnaire available to the acquirer.

2. The acquirer can respond by fully/partially accepting or declining the liability.
  - a. If the acquirer accepts partial liability or declines, VROL generates a dispute response financial message or financial fields to adjust liability. Once a matching advice is received, VROL will deliver the dispute response questionnaire to notify the issuer of the acquirer's response.
  - b. If the acquirer fully accepts the liability, the dispute response questionnaire is made available to the issuer in their queue for further action. No dispute response financial message or financial fields are generated.
3. The issuer can either accept the acquirer's response, or continue the dispute.
  - a. If the issuer accepts the acquirer's dispute response, the dispute ends.
  - b. If the issuer wishes to continue the dispute, they will submit a Pre-Arbitration questionnaire. Note that the issuer can change the dispute category as part of the Pre-Arbitration questionnaire. No financial or financial fields are generated with the Pre-Arbitration questionnaire.
4. The acquirer receives the Pre-Arbitration questionnaire and can respond by fully/partially accepting or declining the liability.
  - a. If the acquirer fully/partially accepts the liability, VROL generates a dispute response reversal financial message or financial fields to adjust liability. Once a matching advice is received, VROL will deliver the Pre-Arbitration Response questionnaire to notify the issuer of the acquirer's response.
  - b. If the acquirer declines the liability, the Pre-Arbitration Response questionnaire is made available to the issuer in their queue. No dispute response reversal financial message or financial fields are generated.
5. The issuer can accept the acquirer's Pre-Arbitration response. Alternatively, they can continue to pursue the dispute by submitting an arbitration case filing.
6. An arbitration filed will be further adjudicated by Visa Dispute Resolution Management (DRM) team to give a final ruling.

# RTSI Overview

2

The Real-Time Systems Interface (RTSI) provides for the exchange of dispute data between the member host system (MHS) and VROL using Web services or REST standards and supports the transaction types detailed in [Table 1–1](#) of [Chapter 1, Service Overview](#). As its name suggests, RTSI happens in real time.

Members looking to establish a system interface connection to VROL using RTSI should review the RTSI Implementation and Testing Checklist. The checklist can be found in [Appendix I, RTSI Implementation and Testing Checklist](#) to assess their needs and prepare for the development and implementation effort.

## Types of Web Services Supported

RTSI supports the following types of web services:

- HTTP/SOAP web services
- HTTP/Representational State Transfer (RESTful) web services.

## High-Level Characteristics

High-level characteristics of RTSI include:

### **SOAP-based Services**

- Use of a Simple Object Access Protocol (SOAP)-based request/response message exchange pattern. Member applications initiate the process by sending an RTSI request. The RTSI server receives the request and passes the dispute data to VROL. After VROL processing, the RTSI server returns a response to the client. For each request received, one response is returned.
- Sending of requests and responses using SOAP over HTTP.
- Use of the SOAP 'document' message style. All dispute data is passed as a literal XML document that is validated against the VROL schema. Note, however, that these messages are sent and received using the Remote Procedure Call (RPC) model where the XML document is passed as a single string parameter of a Web Service method.
- Support for the MIME multipart message structure for transmitting attached images (jpeg, tiff, or pdf). For details, see [Chapter 3, RTSI Request Operations](#).

- Use of XOP (XML-Binary Optimized Packaging) for image Web Services. The XOP specification defines an implementation for optimizing XML messages and image data.

### REST-based Services

- REST (Representational State Transfer) is an architectural style that uses HTTP protocol to exchange the request/response messages. Member applications initiate the process by sending an RTSI request. The RTSI server receives the request and passes the dispute data to VisaNet for processing. After VisaNet processing, the RTSI server returns a response to the client. For each request received, one response is returned.
- Sending of requests and responses using REST over HTTP protocol, mainly uses HTTP methods such as **POST** to manipulate the data.
- RESTful HTTP Web services emphasize the representation of resources in the form of URLs and JSON (JavaScript Object Notation) message format is used in the request/response to exchange the data.
- All dispute data are passed in JSON data format in the request/response that is validated against the VROL RTSI schema.
- Support for the MIME multipart message structure for transmitting attached images (jpeg, tiff, or pdf). For details, see [Chapter 3, RTSI Request Operations](#).

## Supported Standards

RTSI supports the following standards.

### SOAP-based Services

- **WS-I Basic Profile 1.0**

RTSI conforms to the basic Web services profile as documented in the WS-I Basic Profile 1.0 issued by the Web services Interoperability Organization. This profile is located at

<http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html>.

- **HTTP (Hypertext Transfer Protocol)**

RTSI SOAP messages are transmitted using HTTP.

- **Simple Object Access Protocol (SOAP) 1.1**

The interchange of information between VROL and the MHS is implemented as a request/response model using SOAP over HTTP.

- **XML (Extensible Markup Language) / XML Schema 1.0**

Dispute data exchanged between VROL and the MHS are sent as 'literal' XML documents that are validated against the VROL SI schema.

- **Web Service Description Language (WSDL) 1.2**

RTSI interfaces are formally described using the WSDL standard. The WSDL files are available from your Visa representative or MTE.

- **MIME (Multipurpose Internet Mail Extensions)**

RTSI supports the MIME standard for encapsulating non-XML attachments in SOAP messages.

Microsoft.net clients may need to consider using a third-party tool to convert attachments to and from MIME when using RTSI to upload or download files to VROL.

- **MTOM**

The MTOM specification conceptually defines a method for optimizing SOAP messages by separating out binary data, which is otherwise base64-encoded, and sending it in separate binary attachments using a MIME Multipart/Related message. This type of MIME message is called an MTOM message. Sending the data in binary format significantly reduces its size, thus optimizing the transmission of the SOAP message.

- **XOP**

The XOP specification defines an implementation for optimizing XML messages using binary attachments in a packaging format that includes but is not limited to MIME messages.

**Note:** *RTSI does not support Universal Description, Discovery and Integration (UDDI) or publish its services in a public Web services registry. A description of RTSI cannot be obtained through a Web service broker. RTSI WSDL files can be obtained directly from a VROL representative or MTE.*

### REST-based Services

- **HTTP (Hypertext Transfer Protocol)**

RTSI messages are transmitted using HTTP protocol. Mainly uses HTTP methods such as **POST** to manipulate the data.

- **JSON (JavaScript Object Notation) XML Schema (Extensible Markup Language)**

Dispute data exchanged between VROL and the MHS are sent as JSON documents that are validated against the VROL SI schema.

- **Web Application Description Language (WADL)**

WADL is an XML description of a deployed RESTful Web application. It contains model of the deployed resources, their structure, supported media types, and HTTP methods.

The RTSI RESTful services are described using the WADL standard. The WADL files are available from your Visa representative or MTE.

**Note:** *RTSI does not support Universal Description, Discovery and Integration (UDDI) or publish its services in a public Web services registry. A description of RTSI cannot be obtained through a Web service broker. RTSI WSDL files can be obtained directly from a VROL representative or MTE*

## XML Data Validation

RTSI request and response messages are validated against the VROL schema. The Real-Time SI schema is named ROLSIRT.XSD. This schema file validates RTSI XML documents.

## JSON Data validation

RTSI JSON format request and response messages are validated against the VROL schema. The Real-Time SI schema is named ROLSIRT.XSD. This schema file validates RTSI JSON format documents.

## Image Requirements

Images submitted to VROL must be TIFF, JPEG, or PDF images. VROL stores the image in a particular format and the image may be rejected if VROL is unable to convert to this format.

PDFs can be viewed provided the workstation is configured with a suitable PDF reader.

**Note:** *The member can choose to receive questionnaires as images in their preferred format (PDF or TIFF). This requires a change to the configuration of the member's organization. The member should contact their Visa Representative to request this setting.*

**Note:** *Image requirements are the same for both SOAP and REST-based services.*

- Valid baseline TIFF file according to the TIFF 6.0 specification or a valid JPEG file. JPEG files are converted by VROL to the TIFF format.

**Note:** *VROL only uses TIFFs for responses.*

- Limitations for TIFF, JPEG, and PDF requests:
  - Maximum individual file size of 10 MB (TIFF/JPEG/PDF) for requests (responses from Visa may be larger as a result of the conversion from JPEG format to TIFF format).
  - Maximum page size is 8.64 inches x 14 inches (for TIFF and JPEG).
- Maximum resolution is 200 x 200 dots per inch (dpi) for images sent through Visa Copy Request Fulfillment Service (VCRFS) and the VROL systems interface to members that elect to receive images in VIG-compatible format.

**Note:** *If the image resolution is greater than 200 dpi, VROL attempts to increase the dimensions of the image. If the expanded image exceeds the maximum size allowed by the VIG specification (8.64 inches x 11.04 inches), VROL "samples" the image accordingly so that it fits within the limitation.*

- Specific requirements for TIFF files:
  - All four baseline image formats accepted: bi-level (two colors, black and white), grayscale, palette color, and RGB color.
  - All TIFF fields that are required according to specifications and that do not have a default value must be shown.

**Table 2-1: TIFF Field Specifications**

Tag	Field
256	Image width
257	Image length
258	Bits per sample (non-bi-level images only)
259	Compression
273	Strip offsets
277	Samples per pixel (RGB color images only)
279	Strip byte counts
282	X resolution
283	Y resolution

**Table 2-1: TIFF Field Specifications (cont'd.)**

Tag	Field
296	Resolution unit <sup>1</sup>
320	Color map (palette color images only)

<sup>1</sup> The Resolution unit field (tag 296) can have a value of '2' (inches) or '3' (centimeters), but not '1' (no absolute unit of measurement)

- Uploading, downloading, or retrieving images
  - When uploading images, the Media Type setting must be specified as "multipart/form-data".
  - When downloading or retrieving images, the Media Type setting must be specified as "application/json".

[Table 2-2](#) outlines the graphics file formats supported by VROL and under what conditions they are supported.

**Table 2-2: VROL Image Support**

Uploaded using RTSI	Conversion and Storage Format	View Format		Bulk SI Download Option
Supported Image Types		Based on Member Configuration	Based on VIG Configuration	Based on Member Configuration
JPEG (Black & White or Color)	TIFF (converted)	TIFF	TIFF	TIFF
TIFF (Black & White, Grayscale, or Color)	TIFF	TIFF	TIFF	TIFF
PDF (Black & White or Color)	PDF	PDF	TIFF	PDF

## Attaching Images to Questionnaires or Existing Cases

RTSI users can add an image to an existing case or attach an image to an existing questionnaire.

1. Attach images to an existing questionnaire

An RTSI user can attach one or more images to a questionnaire that is in a Saved status by providing the image details along with the QnId. After attaching the images to the questionnaire, the user can either save the questionnaire again or submit it for processing.

**Note:** <QnId> is used to add an image to a questionnaire; otherwise, adds an image to a case. The <QnId> uses the value of the <DisputelId>.

2. Add images to an existing case

An RTSI user can attach one or more images to a case by providing the image details along with the VisaCaseNumber.

## Hybrid Image Uploads

As part of the dispute Hybrid SI approach, members can upload images to a central image repository. Members can also directly add an image to an existing case or attach an image to an existing dispute questionnaire. Members can upload a single image or upload images in bulk.

1. Upload images to a central image repository

A Bulk SI member can upload an image or multiple images to a central image repository. After a successful upload to the image repository, a unique image identification number (DocId) is provided for each image. The member can use the DocId(s) to add image(s) to cases or attach them to dispute questionnaires.

**Note:** *Multiple DocIds will be generated if multiple images are uploaded.*

2. Attach images to an existing dispute questionnaire

A Bulk SI member can attach one or more images to a dispute questionnaire that is in a Saved status by providing the image details along with the DisputeItemID.

After attaching the images to the dispute questionnaire, the member can either save the dispute questionnaire again or submit it for processing.

3. Add images to an existing case

A Bulk SI member can attach one or more images to a case by providing the image details along with the VisaCaseNumber.

See the *Visa Resolve Online Bulk Systems Interface Development Guide* for details.

## Service Description Files

### SOAP-based Services

RTSI provides separate classes for application and security services. These are formally defined in the WSDL files described in this section. Each WSDL file specifies the mechanics of interacting with RTSI, including the required message formats, data types, transport protocols and serialization formats, and the network address where each Web service operation must be invoked.

#### Application Services

The WSDL file, \*.wsdl, specifies operations, bindings, types, service names, and location of Web services providing transaction inquiries (TIs), copy requests and fulfillments, hypersearches, and other functions. VROL provides Web services that use MTOM encoding/WS-SOAP for image upload/download and WS-SOAP-based Web services in the WSDL file (VROL-RTSIService.wsdl).

The WSDL files are part of RTSI technical documentation available to all members in MTE or from a VROL representative.

**Note:** *There are separate WSDL files for production and one for member test (in MTE2 folder in wsdl.zip file).*

**Note:** *Contact your Visa representative for more information.*

The WSDL groups the Web services under a small number of URLs rather than one per Web service. Refer to [Appendix A, Systems Interface URLs](#) for more details on the URL formats for different environments.

## REST-based Services

RTSI provides separate classes for application and security services. These are formally defined in the WADL files described in this section. Each WADL file specifies the mechanics of interacting with RTSI. It includes deployed resources, their structure, supported media types, HTTP methods, data types, transport protocols and the network address where each Web service resource must be invoked.

### Application Services

The WADL file, \*.wadl, specifies resources, method, representation, media types, resource path, and location of Web services providing transaction inquiries (TIs), copy requests and fulfillments, hypersearches, and other functions.

The WADL files are part of RTSI technical documentation available to all members in MTE or from a VROL representative.

**Note:** *There are separate WADL files for production and one for member test (in MTE folder in wadls.zip file).*

**Note:** *Contact your Visa representative for more information.*

## Security Services

The RTSI Security Service allows members to integrate authentication, and password administration into client applications. For details about RTSI security, see [Chapter 26, Security](#).

**Note:** *The same security mechanism is used for both SOAP and REST-based services.*

## RTSI Clients

Members may implement a range of RTSI client types, including browsers, console applications, GUIs, and others. Because RTSI supports Web services standards, client applications can be developed using the platform, language, programming model, and development tool of choice. This section provides general information and a number of suggestions regarding client development.

### Client Requirements

#### SOAP-based Services

RTSI client applications are required to:

1. Generate and send SOAP requests over HTTPS. As stated previously, VROL dispute data is sent as literal XML documents in the SOAP message body.
2. Receive and unmarshal the SOAP responses from VROL.
3. Obtain and parse the XML-formatted response dispute data from the SOAP message and process the data according to local business logic.

#### REST-based Services

RTSI client applications are required to:

1. Generate and send REST requests over HTTPS. As stated previously, VROL dispute data is sent as literal JSON format documents in the message body.

2. Receive and parse the REST responses from VROL.
3. Obtain and parse the JSON-formatted response dispute data from the REST message using Jackson parser and process the data according to local business logic.

## Generating Proxy Classes

Web services development tools are readily available from major vendors. Visa recommends that members implement an RTSI client by generating proxy classes that can be invoked locally to communicate with RTSI over the network.

Generating proxy classes encapsulates the complexity of web service calls and hides lower-level implementation details. It allows developers to use the familiar development model of making method calls on objects and permits integration with RTSI as if it were a local component.

## Authentication and Authorization

The Member Host System (MHS) sends requests to VROL (RTSI) with the user ID and password. The user ID is tagged under <User id>. While the schema identifies this as an optional field, in order to satisfy information security and audit requirements, this field is only optional if the MHS connection is established using an end-user VROL ID.

**Note:** *If the request does not contain <User id>, the response will be based on the system ID (present in the HTTP headers).*

Refer to [Chapter 26, Security](#) for more information.

The user name must also have an appropriate role with privileges to perform an operation. The Universal User role is usually used.

Following is the basic request flow:

1. MHS sends VROL an RTSI request message and includes the System ID (in the HTTP header) and password, certificate, and user ID (not mandatory) with the specific end point URL.

**Note:** *This step is performed for every request sent from the MHS.*

2. WSI Datapower intercepts the message based on the particular end point URL and verifies the credentials against the policy for successful authentication and authorization.
3. If the authorization is successful, WSI Datapower directs the request to RTSI and the response is displayed to the user. If the authorization is not successful, appropriate error message is displayed to the user.

**Note:** *This is basically a two-factor Security scheme using basic authentication and mutual certificate authentication.*

## Internet Access

Access to VROL requires that each user have a valid VOL System ID that has been entitled to access VROL. To successfully implement RTSI, members must consider the network capacity requirements for their Internet connection or for their secure services connection.

Members must review their current Internet access bandwidth to determine whether the bandwidth needs to be augmented to accommodate increased user traffic.

## RTSI Requests

### SOAP-based Services

The following shows a sample of the required structure of an RTSI SOAP request message.

```
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:rtsi="http://www.visa.com/rtsi">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <rtsi:HyperSearchRequestElement>
      <![CDATA[
        <RealTimeSIRequest xmlns="http://www.visa.com/ROLSI">
          xmlns:vsi="http://www.visa.com/ROLSI"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <User id="xxxxxxxxxx" type="loginId" />
            <RequestName>HyperSearchRequest</RequestName>
            <MemberRole>I</MemberRole>
            <HyperSearchRequestBody>
              <CaseMatchingInfo>
                <VisaCaseNumber>XXXXXXXXXXXX</VisaCaseNumber>
                <Transaction></Transaction>
              </CaseMatchingInfo>
            </HyperSearchRequestBody>
          </RealTimeSIRequest>]]>
        </rtsi:HyperSearchRequestElement>
      </SOAP-ENV:Body>
    </SOAP-ENV:Envelope>
```

The following show samples of the required structure for an RTSI SOAP request message.

### Transaction Inquiry Web Service

```
<SOAP-ENV:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  xmlns:rol="http://www.visa.com/ROLSI">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <rol:SISubmitTranInquiryRequest>
      <rol:RequestHeader>
        <rol:User id="xxxxxx" type="internalId"/>
        <rol:MemberRole>I</rol:MemberRole>
      </rol:RequestHeader>
      <rol:requestData>
        <rol:CardNumber>xxxxxxxxxxxxxxxx</rol:CardNumber>
        <rol:StartDate>2013-09-02</rol:StartDate>
        <rol:EndDate>2013-09-02</rol:EndDate>
      </rol:requestData>
    </rol:SISubmitTranInquiryRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## Imaging Web Service

```
<SOAP-ENV:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:rol="http://www.visa.com/ROLSI">  
  <SOAP-ENV:Header/>  
  <SOAP-ENV:Body>  
    <rol:SIUploadImageRequest>  
      <rol:RequestHeader>  
        <rol:User id="xxxxxx" type="internalId"/>  
        <rol:MemberRole>I</rol:MemberRole>  
      </rol:RequestHeader>  
      <rol:RequestData>  
        <rol:VisaCaseNumber>1002029293</rol:VisaCaseNumber>  
        <rol:QnId>1335451</rol:QnId>  
        <rol:DisputeItemInfo>  
          <rol:DisputeItemID>0</rol:DisputeItemID>  
          <rol:DisputeItemType>String</rol:DisputeItemType>  
        </rol:DisputeItemInfo>  
        <rol:AttachmentDescriptor>  
          <rol:Attachment id="1">  
            <rol:ContentType>image/tiff</rol:ContentType>  
            <rol:Comment>test</rol:Comment>  
            <rol:ImageData>cid:556406701357</rol:ImageData>  
          </rol:Attachment>  
        </rol:AttachmentDescriptor>  
        <rol:DocType>CALET</rol:DocType>  
      </rol:RequestData>  
    </rol:SIUploadImageRequest>  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

## REST-based Services

The following show a sample of the required structure of an RTSI REST request message.

```
{  
    "RequestHeader": "...",  
    "requestData":{  
        "CaseMatchingInfo":{  
            "MemberCaseNumber": "a",  
            "VisaCaseNumber": 1000000000,  
            "AccountNumber": "100000000000",  
            "ARN": "41000000000000000000000000000000",  
            "Transaction":{  
                "id": "a",  
                "TranDate": "2016-01-01",  
                "Amount":{  
                    "currency": "840",  
                    "Text": 0  
                },  
                },  
                "RetrievalRequestID": "aaaaaaaaaaaa",  
                "AcquirerBIN": "100000",  
                "RetrievalReferenceNumber": "1212121212",  
                "SystemTraceAuditNumber": "111111",  
            },  
            "Token": "10000000000000",  
            "PageNum": 1,  
        }  
    }  
}
```

The following show samples of the required structure for an RTSI REST request message.

### Transaction Inquiry Web Service

```
{  
    "RequestHeader": "...",  
    "requestData":{  
        "CardNumber": "000000",  
        "Token": "00000000000000",  
        "TerminalId": "a",  
        "TransactionID": "a",  
        "ARN": "000000000000000000000000",  
        "StartDate": "2016-01-01",  
        "EndDate": "2016-01-01",  
        "IncludeInd": "Both",  
        "IncludeCreditsOrReversalsOrAdjustmentsInd": true,  
        "TranStatus": "Approved",  
        "TransactionTypeOption": "Standard",  
        "TransactionType": "String",  
        "ShowDuplicateAuthWithSameTranIDInd": true,  
        "AuthCode": "a",  
        "TranAmountRange1": 0,  
        "TranAmountRange2": 0,  
        "RetrievalReferenceNumber": "a",  
        "NetworkID": "aaa",  
        "MCC": "0000",  
        "Repository": "VTRS",  
        "AsynchronousRequestInd": true  
    }  
}
```

## Imaging Web Service

```
{
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": 1000000000,
        "MessageID": 0,
        "DisputeItemInfo": {
            "DisputeItemID": 0,
            "DisputeItemType": "String"
        },
        "AttachmentDescriptor": {
            "AttachType": "Mime",
            "Attachment": {
                "id": "1",
                "ContentType": "image/jpeg",
                "Comment": "a",
                "ImageData": "UjBsR09EbGhjZ0dTQUxNQUBUUNBRU1tQ1p0dU1GUUhEUzhi"
            }
        },
        "DocType": "CACHK"
    }
}
```

## HTTP Header

When a request is sent to RTSI, the header in the request contains the System ID and password. The header will also include SSL mutual authentication.

Http Basic Auth scheme adds the HTTP header (specified below)

```
oXMLHTTP.setRequestHeader "Authorization", "Basic " & Base64Encode("uid:password")
Header name: Authorization
Header value: Basic <base64encoded output of "vol_userid:vol_password" string>
```

For example: Basic x2351235sr2352352153 where x2351235sr2352352153 is the base64 encoded value of <username>:<password>

## SOAP-based Services

### SOAP Envelope

The Envelope element is required by SOAP standards and must be namespace-qualified. The Envelope element is the top SOAP element in the message.

### SOAP Body

The SOAP Body element is required by SOAP standards and must be the immediate child of the SOAP Envelope element.

### RTSI SOAP Operation (Method)

The RTSI operation name is required. It must be namespace-qualified and located as the immediate child of the SOAP Body element. In the previous example, the operation is a transaction inquiry request.

### SOAP Action (<soapaction>)

As described under Service Description Files, Application Services above, the WSDL groups the Web services under a small number of URLs rather than one per Web service. The <soapaction> element is required to identify the particular Web service.

## REST-based Services

REST services request/response messages will adhere to the JSON data format. Resource path/method is required to invoke the service end point.

## RTSI Responses

### SOAP-based Services

The following example shows the structure of an RTSI SOAP response message. Note that the structure of the response message and its components is similar to the request message, but with different content.

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
    <soapenv:isDps>true</soapenv:isDps>
</soapenv:Header>
<soapenv:Body>
    <tns:HyperSearchResponseElement xmlns:tns="http://www.visa.com/rtsi">
<![CDATA[
    <?xml version="1.0" encoding="UTF-8"?>
    <RealTimeSIResponse xmlns="http://www.visa.com/ROLSI"
        <xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <Status>
            <Code>I-300000000</Code>
            <Message>Successfully completed Operation</Message>
        </Status>
        <HyperSearchResponseBody>
            <Case>
                <VisaCaseNumber>XXXXXXXXXXXX</VisaCaseNumber>
                <MemberCaseNumber>a</MemberCaseNumber>
                <AccountNumber>xxxxxxxxxxxxxxxxxxxx</AccountNumber>
                <Token>xxxxxxxxxxxxxxxxxxxx</Token> [for tokenized transactions only]
                <TokenAssuranceLevel>00</TokenAssuranceLevel>
[issuer only field for tokenized transactions]
                <PurchaseInquiryEligibleInd>true</PurchaseInquiryEligibleInd>
                <ARN>xxxxxxxxxxxxxxxxxxxxxxxx</ARN>
                <Transaction id="xxxxxxxxxxxxxx">
                    <TranDate>YYYY-MM-DDD</TranDate>
                    <Amount currency="840">XX.XX</Amount>
                </Transaction>
                <ReasonCode>XX</ReasonCode>
                <MerchantName>xxxxxxxxxxxxxxxxxxxx</MerchantName>
                <DisputeAmount currency="840">XX.XX</DisputeAmount>
                <StageStateDesc>XXXXXXXXXXXX</StageStateDesc>
                <RetrievalReferenceNumber>xxxxxxxxxxxx</RetrievalReferenceNumber>
            </Case>
        </HyperSearchResponseBody>
    </RealTimeSIResponse>
</![CDATA]>
</tns:HyperSearchResponseElement>
</soapenv:Body>
</soapenv:Envelope>
```

```

<SystemTraceAuditNumber>xxxxx</SystemTraceAuditNumber>
<AcquirerBIN>xxxxxxxx</AcquirerBIN>
<NetworkID>xxx</NetworkID>
<CaseResolutionStatus>xxxx</CaseResolutionStatus>
<FraudBundleCaseNumber>1000000000</FraudBundleCaseNumber>
<LastActionDate>YYYY-MM-DD HH:MM:SS</LastActionDate>
<CaseViewIndicators>
    <RfiIndicators>
        <RfiEventId LastactionDate="2001-12-17T09:30:47Z">
            XXXXXXXX</RfiEventId>
        <RfiComplete>true</RfiComplete>
    </RfiIndicators>
    ...
</CaseViewIndicators>
...
</Case>
<PageInfo>
    <PageNum>1</PageNum>
    <TotalPages>1</TotalPages>
</PageInfo>
</HyperSearchResponseBody>
</RealTimeSIResponse>
]]>
</tns:HyperSearchResponseElement>
</soapenv:Body>
</soapenv:Envelope>

```

The following show samples for the RTSI SOAP response message for the two groups of Web services.

### Transaction Inquiry Web Service

```

<SOAP-ENV:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <SISubmitTranInquiryResponse xmlns="http://www.visa.com/ROLSI">
            <Status>
                <Code>I-300000000</Code>
                <Message>Successfully completed Operation</Message>
            </Status>
            <responseData>
                <TIEventID>1283543</TIEventID>
                <TransactionSummary>
                    ...
                </TransactionSummary>
                <TransactionSummary>
                    ...
                </TransactionSummary>
            </responseData>
        </SISubmitTranInquiryResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## Imaging Web Service

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <rol:SIUploadImageResponse xmlns:ns2="http://www.visa.com/ROLSI">
      <rol:Status>
        <rol:Code>I-300000000</rol:Code>
        <rol:Message>Successfully completed Operation</rol:Message>
      </rol:Status>
      <rol:responseData>
        <rol:VisaCaseNumber>1000000003</rol:VisaCaseNumber>
        <rol:QnId>1335451</rol:QnId>
        <rol:DisputeItemInfo>
          <rol:DisputeItemID>0</DisputeItemID>
          <rol:DisputeItemType>String</DisputeItemType>
        </rol:DisputeItemInfo>
        <rol:DocId>4567918</rol:DocId>
      </rol:responseData>
    </rol:SIUploadImageResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## REST-based Services

The following shows a sample of the structure of an RTSI REST response message.

```
{
  "Status": {
    "Code": "a",
    "Message": "a",
    "ErrorIn": "String"
  },
  "responseData": {
    "Case": {
      "VisaCaseNumber": 1000000000,
      "MemberCaseNumber": "a",
      "AccountNumber": "000000000000",
      "Token": "0000000000000000",
      "TokenAssuranceLevel": "a",
      "PurchaseInquiryEligibleInd": true,
      "ARN": "00000000000000000000000000000000",
      "Transaction": {
        "id": "a",
        "TranDate": "2016-01-01",
        "Amount": {
          "currency": "000",
          "Text": 0
        },
      },
      "ReasonCode": "String",
      "MerchantName": "a",
      "DisputeAmount": {
        "currency": "000",
        "Text": 0
      },
      "StageStateDesc": "String",
    }
  }
}
```

```
"RetrievalReferenceNumber": "a",
"SystemTraceAuditNumber": "a",
"AcquirerBIN": "000000",
"NetworkID": "aaa",
"CaseResolutionStatus": "String",
"LastActionDate": "2001-12-17T09:30:47Z",
"CaseViewIndicators": {
    "RfiIndicators": {
        "RfiEventID": {
            "LastActionDate": "2001-12-17T09:30:47Z",
            "Text": 0
        },
        "RfiComplete": true
    },
    "RfcIndicators": {
        "RFCID": {
            "LastActionDate": "2001-12-17T09:30:47Z",
            "Text": 0
        },
        "HasRfcFulfillment": true,
        "HasRfcNonFulfillment": true,
        "HasRfcAdvice": true
    },
    "FraudReportIndicators": {
        "FraudReport": {
            "FraudReportID": {
                "LastActionDate": "2001-12-17T09:30:47Z",
                "Text": 0
            }
        }
    },
    "ExceptionFileIndicators": {
        "ExceptionFileListing": {
            "ExceptionFileID": {
                "LastActionDate": "2001-12-17T09:30:47Z",
                "Text": 0
            },
            "ExceptionFileType": "EFNEG"
        }
    },
    "AdjustmentIndicators": {
        "AdjustmentID": {
            "LastActionDate": "2001-12-17T09:30:47Z",
            "Text": 0
        }
    },
    "StopPaymentIndicators": {
        "StopPaymentID": {
            "LastActionDate": "2001-12-17T09:30:47Z",
            "Text": 0
        }
    },
    "MiscFeeIndicators": {
        "MiscFeeID": {
            "LastActionDate": "2001-12-17T09:30:47Z",
            "Text": 0
        }
    }
},
```

```
        "MiscFeeType": "FundDisbursement",
        "Direction": "Outgoing"
    },
    "AEMIndicators": {
        "AccountingEntryID": {
            "LastActionDate": "2001-12-17T09:30:47Z",
            "Text": 0
        }
    },
    "DocumentIndicators": {
        "DocId": 0
    },
    "MessageIndicators": {
        "ActiveMessageID": {
            "LastActionDate": "2001-12-17T09:30:47Z",
            "Text": 0
        },
        "MessageType": "SDC",
        "MessageIndicatorType": "MessageResp"
    },
    "PurchaseInquiryIndicators": {
        "PurchaseInquiryId": {
            "LastActionDate": "2001-12-17T09:30:47Z",
            "PurchaseInquiryResponseStatus": "String",
            "Text": 0
        }
    },
    "CollaborationIndicators": {
        "CollaborationId": {
            "LastActionDate": "2001-12-17T09:30:47Z",
            "Text": 0
        },
        "SubmittedBy": "Issuer",
        "CollaborationType": "IQ",
        "CollaborationActiveInd": true
    },
    "MerchantNotificationIndicators": {
        "MerchantNotificationId": {
            "LastActionDate": "2001-12-17T09:30:47Z",
            "NotificationResponseStatus": "String",
            "Text": 0
        }
    },
    "DisputeIndicators": {
        "DisputeId": {
            "LastActionDate": "2001-12-17T09:30:47Z",
            "Text": 0
        },
        "AssociatedTransGroupId": {
            "LastActionDate": "2001-12-17T09:30:47Z",
            "Text": 0
        },
        "DisputeResponseId": {
            "LastActionDate": "2001-12-17T09:30:47Z",
            "Text": 0
        },
        "DisputePreArbId": {
```

```
        "LastActionDate": "2001-12-17T09:30:47Z",
        "Text": 0
    },
    "DisputePreArbResponseId":{
        "LastActionDate": "2001-12-17T09:30:47Z",
        "Text": 0
    },
    "DisputeFilingItemId":{
        "LastActionDate": "2001-12-17T09:30:47Z",
        "DisputeFilingItemType": "DecisionLetter",
        "Text": 0
    },
    "DisputePreCompId":{
        "LastActionDate": "2001-12-17T09:30:47Z",
        "Text": 0
    },
    "DisputePreCompResponseId":{
        "LastActionDate": "2001-12-17T09:30:47Z",
        "Text": 0
    },
    "ContactMessages":{
        "ContactMessageId":{
            "LastActionDate": "2001-12-17T09:30:47Z",
            "Text": 0
        },
        "ContactMessageResponseId":{
            "LastActionDate": "2001-12-17T09:30:47Z",
            "Text": 0
        }
    },
    "VROLFinancialID":{
        "LastActionDate": "2001-12-17T09:30:47Z",
        "Text": "String"
    },
    "DisputeFinId":{
        "LastActionDate": "2001-12-17T09:30:47Z",
        "Text": 0
    },
    "DisputeAdviceId":{
        "LastActionDate": "2001-12-17T09:30:47Z",
        "Text": 0
    },
    "QuickFormID":{
        "LastActionDate": "2001-12-17T09:30:47Z",
        "QuickFormType": "Dispute",
        "Text": 0
    },
    "DisputeCategory": 11,
    "DisputeCategoryCondition": "String",
    "RetrievalRequestID": "aaaaaaaaaaaaaa",
    "CaseStatus": "String",
    "MemberStatus": "String",
    "DisputeJurisdiction":{
        "@jurisdictionCd": "REG",
        "Jurisdiction": "String",
        "IssuerRegion": "LAC",
        "AcquirerRegion": "US",
```

```

        "CountryCd": "AA"
    },
    "AssignedAnalyst": "String",
    "CaseType": "DSPT",
    "FollowUpDate": "2017-08-13",
    "FraudClassification": "F",
    "ATMCaseInd": "true",
    "VROLProductType": "String",
    "PriorityCaseInd": "true",
    "AccountType": "a",
    "DaysToAct": 0,
    "LastDateToAct": "2017-08-13"
},
}
},
"PageInfo":{
    "PageNum": 1,
    "TotalPages": 0
}
}
}
}
```

The following show samples for the RTSI REST response message for the two groups of Web services.

### Transaction Inquiry Web Service

```
{
    "Status": {
        "Code": "a",
        "Message": "a",
        "ErrorIn": "String"
    },
    "ResponseData": {
        "TIEventID": 0,
        "Repository": "VTRS",
        "TransactionSummary": {
            "RolTransactionId": 0,
            "IdentifiedTransaction": true,
            "TransactionType": "String",
            "TransactionTypeShortDescription": "String",
            "TransactionDateTime": "2001-12-17T09:30:47Z",
            "CPD": "2016-01-01",
            "BusinessAppId": "aa",
            "BusinessAppIdDesc": "String",
            "SourceOfFunds": "a",
            "TransactionAmount": {
                "currency": "000",
                "Text": 0
            },
            "TranAmountDesc": "String",
            "MerchantName": "a",
            "MerchantCity": "a",
            "MerchantCountrySubEntityCode": "a",
            "MerchantCountryCode": "AA",
            "MoToEciCode": "String",
            "MoToEciDescription": "String",
            "NetworkID": "aaa",
        }
    }
}
```

```
"AuthorizationResponseCode": "String",
"AuthorizationResponseDescription": "String",
"AuthorizationCode": "String",
" MCC": "0000",
" MCCDescription": "String",
" TransactionID": "String",
" RetrievalReferenceNumber": "a",
" ARN": "00000000000000000000000000",
" CardNumber": "000000000000",
" Token": "String",
" CVV2Result": "String",
" CVV2ResultDesc": "String",
" PurchaseInquiryEligibleInd": true,
" AssociatedTransGroupId": 0
},
"TIErrorResponse": {
"CardNumber": "000000",
"Token": "000000000000",
"TerminalId": "a",
"TransactionID": "a",
"ARN": "000000000000000000000000",
"StartDate": "2016-01-01",
"EndDate": "2016-01-01",
"IncludeInd": "Both",
"IncludeCreditsOrReversalsOrAdjustmentsInd": true,
"TranStatus": "Denied Or Referred",
"TransactionTypeOption": "Single",
"TransactionType": "String",
>ShowDuplicateAuthWithSameTranIDInd": true,
"AuthCode": "a",
"TranAmountRange1": 0,
"TranAmountRange2": 0,
"RetrievalReferenceNumber": "a",
"NetworkID": "aaa",
" MCC": "0000",
"Errors": {
"Description": "String"
}
}
}
}
```

## Imaging Web Service

```
{  
    "Status": {  
        "Code": "a",  
        "Message": "a",  
        "ErrorIn": "String"  
    },  
    "ResponseData": {  
        "VisaCaseNumber": 1000000000,  
        "MessageID": 0,  
        "DisputeItemInfo": {  
            "DisputeItemID": 0,  
            "DisputeItemType": "String"  
        },  
        "DocId": 0  
    }  
}
```

THIS PAGE INTENTIONALLY LEFT BLANK.

# RTSI Request Operations

3

The following sections describe the basic format of RTSI calls and operations available in RTSI request messages and VROL-generated error messages to support VROL functions.

## RTSI Calls

### SOAP-based Services

The generic format for RTSI calls is as follows:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
    xmlns:rol="http://www.visa.com/ROLSI">  
    <soapenv:Header/>  
    <soapenv:Body>  
        <rol:SISubmitTranInquiryRequest> - name of request  
        <rol:RequestHeader>  
            <rol:User id="xxxxxx" type="internalId"/>  
            <rol:MemberRole>I</rol:MemberRole> -  
                required for dual member  
        </rol:RequestHeader>  
        <rol:RequestData> - body of request  
        ...  
        </rol:RequestData>  
    </rol:SISubmitTranInquiryRequest>  
    </soapenv:Body>  
</soapenv:Envelope>
```

The name of the request equates to the operations shown in the schema. Specific examples are shown in the various chapters of this document.

## REST-based Services

```
{
    "RequestHeader": {
        "User": {
            "id": "a",
            "type": "internalId"
        },
        "MemberRole": "A" - required for dual member
    },
    "requestData": { - body of request
        ...
        "CardNumber": "1000000000000000",
        "Token": "10000000000000",
        "TerminalId": "a",
        "TransactionID": "a",
        "ARN": "10000000000000000000000000000000",
        "StartDate": "2016-01-01",
        "EndDate": "2016-01-01",
        ...
    }
}
```

## Transaction Operations and Actions

[Table 3-1](#) describes the important attributes associated with <QuestionnaireDescriptor>.

**Table 3-1: Important Attributes**

Attribute	Description
operation	See <a href="#">Table 3-2</a> .
action	See <a href="#">Table 3-3</a> .
ProcessingPlatform	The ProcessingPlatform attribute on the <QuestionnaireDescriptor>, <RequestForCopy>, and <MiscellaneousFee> is used only when a member's organization is set up to support both SMS and BASE II transactions (financials, RFC, or miscellaneous fees).  It is required if VROL cannot determine if the transaction should be sent as SMS or BASE II. If this value is not provided when required, VROL will return an error. If it is provided and the value conflicts with the information in VROL, the transaction will be submitted correctly and a warning is returned in the RTSI response.

**Table 3–2: Transaction Operations**

<b>Operation</b>	<b>Function</b>	<b>Case must be in the following state:</b>
Add	Adds a new questionnaire of the specified type to the case. The system will create a new case even if an identical case exists in the system.	The case can be in any state, but the type of questionnaire must be appropriate for the state.
Update	Updates an existing saved questionnaire of the same type to the case.	Questionnaire saved state:  The saved questionnaire on the case must be of the same type as that specified in the upload request. The questionnaire may have been explicitly saved by the user.
Delete	Deletes a previously saved questionnaire from a case, when combined with the QuestionnaireDescriptor "Delete" action.	Questionnaire saved state:  The saved questionnaire must have been created by the member attempting to perform the delete operation.  The questionnaire may have been explicitly saved by the user.

**Table 3–3: Transaction Actions**

<b>Action</b>	<b>Function</b>
Submit	The Submit action is applicable to scenarios where an existing saved questionnaire exists for a case or one has just been created by the current upload request. It submits the questionnaire to the opposite party in the dispute through VROL. See example after the table.
Save	The Save action saves the existing or newly created questionnaire upon completion of any questionnaire data updates or image attach operations. A saved questionnaire will not be sent to the opposing side until it is submitted.  VROL allows users to change the state (for example, from Saved to Submit) of an existing questionnaire.
Delete	The Delete action is only valid for a QuestionnaireDescriptor "Delete" operation. It deletes an existing saved questionnaire from the case. The questionnaire may have been explicitly saved by the user.

## Important XML/JSON Fields in Transactions

Field	Description
ARN	VROL accepts either 23- or 24-digit Acquirer Reference Numbers (ARNs). VROL automatically converts all ARNs to 23 digits upon Save or Submit and sends the 23-digit ARN in the response.  If the user provides a 23-digit value for ARN, VROL provides an edit check. A warning is displayed if the ARN does not pass the edit. No edit check will be performed on 24-digit ARNs.  Response may contain ARN.  For SMS members, ARN is 23 digits.
RRN STAN Acquirer BIN	RRN, STAN, and Acquirer BIN are alternatives to ARN. SMS members can use these fields instead of the ARN.  VROL sends the Retrieval Reference Number (RRN), System Trace Audit Number (STAN), and acquirer BIN in the response, upon which the user should rely.  Expect to receive these fields in response.  If required, the RRN, STAN, and acquirer BIN can be used to re-create a 24-digit ARN.
Card/Account Number	The Card/Account Number is a required field for most transactions to identify the card or account performing the transaction.  For tokenized transactions using the Visa Token Service (VTS), an acquirer may put a Token value in either Card/Account Number or Token field in an RTSI request. The issuer must always use the Token field for a Token value. For both issuers and acquirer, in an RTSI response, the Card/Account Number field will contain the Card/Account Number value and the Token field will contain the Token value.
Token Token Assurance Level (for issuers only)	These fields are used for tokenized transactions (Visa Token Service or VTS).

## Image Handling

RTSI supports the transfer of images using the MIME multipart message structure in which the SOAP message and one or more image files are encapsulated and sent in a SOAP message as attachments. Users must use this message structure to send images to VROL and must be able to process this message structure in order to receive images from VROL.

**Note:** *Image handling is the same for both SOAP and REST-based services.*

## SOAP Message Package

The SOAP Message Package is built using the MIME Multipart/Related Content type with each body part separated from another by a MIME-boundary. For transferring fulfillments, the body parts are referred to here as the Message Body and the Image Body.

In addition, this message structure requires that Content-Type information be added to the HTTP header.

## Content-Type Information

Content-Type information within the HTTP header must be set to "multipart/related" and include parameters as described in the following table.

Boundary	This parameter specifies the MIME-boundary used to demarcate the Message Body from the Image Body.
Start	This parameter is required and specifies the Content-ID header of the Message Body. This ensures that the Message Body is processed first.
Type	This parameter is required and must be set to "text/xml."

## SOAP Part

The message body is the "root" element of the message. It begins with the boundary indicator. It includes header information, the request/response message, and an HREF reference to each image file sent in the message package. Header information and the HREF reference are described in the following table.

Content-Type	Specifies the content type of the message body and must be set to "text/xml; charset=UTF-8."
Content-Id	This parameter is required and must be set to the value specified in the Start parameter in the HTTP header.
Content-Transfer-Encoding	This parameter is required and must be set to "binary."
Attachment	This element is required for each attached image file. The ID attribute of this element is used to specify the value of the Content ID header of an attached image file.
Image	<p>This element is required within the Message Body and uses an HREF reference to specify the Image Body.</p> <p>The HREF value must contain the "cid" prefix, for example:</p> <p>&lt;image href="cid:1234567"/&gt;</p> <p>The HREF value must match the Content-Id header of the attachment part.</p>

## Attachment Part

The Image body contains a header and one or more image files. Each image file also contains a header. The Attachment part is of the type Multipart as specified in RTSIServices.wsdl. (See [Multipart Message](#) following.)

Boundary	Specifies the boundary indicator used to separate image files in the Image body.
Content-Transfer-Encoding	This parameter is required and must be set to "binary."

Content-Id	This element is required and specifies the Image body identifier. This identifier is used in the Message Body's Image element to reference the image files.
Attachment header	<p>Each Attachment header specifies:</p> <p>Content-Type: This value must be "image/tiff", "image/jpeg", or "application/pdf".</p> <p>Content-ID: This value specifies the image file identifier and must be taken from an Attachment ID value in the Message body.</p>

## Multipart Message

The following is a sample of a request submitting an RFC fulfillment, including the HTTP header.

**Note:** *For readability, the Content-Type header is shown on multiple lines. In actual messages, this header must be one continuous line.*

The following is a sample of a request submitting an RFC fulfillment, including the HTTP header:

## Transaction Inquiry Web Service

```
POST https://address:port/websrv_rolti/RTSIService HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: ""
Authorization: Basic YXV0b3Rlc3Q1cjpwAQNhMTQwMg==
Content-Length: 656
Host: address:port
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

<SOAP-ENV:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<ns1:Header>
<ns1:Body>
<ns1:SISubmitTranInquiryRequest>
<ns1:RequestHeader>
<ns1:User id="xxxxxx" type="internalId"/>
<ns1:MemberRole>I</ns1:MemberRole>
</ns1:RequestHeader>
<ns1:RequestData>
<ns1:CardNumber>xxxxxxxxxxxxxxxx</ns1:CardNumber>
<ns1:StartDate>2013-09-02</ns1:StartDate>
<ns1:EndDate>2013-09-02</ns1:EndDate>
</ns1:RequestData>
</ns1:SISubmitTranInquiryRequest>
</ns1:Body>
</SOAP-ENV:Envelope>
```

## Imaging Web Service

```
POST https://address:port/websrv_rolsi/RTSIService HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: multipart/related; type="application/xop+xml"; start=<rootpart@soapui.org>;
start-info="text/xml"; boundary="----=_Part_0_10714568.1392074245778"
```

```
SOAPAction: "urn:#UploadImageOperation"
Authorization: Basic YXV0b3Rlc3Q1cjpWaXNhMTQwMg==
MIME-Version: 1.0
Content-Length: 379579
Host: address:port
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

-----=_Part_0_10714568.1392074245778
Content-Type: application/xop+xml; charset=UTF-8; type="text/xml"
Content-Transfer-Encoding: 8bit
Content-ID: <rootpart@soapui.org>

<SOAP-ENV:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<ns1:rol="http://www.visa.com/ROLSI">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<rol:SIUploadImageRequest>
<rol:RequestHeader>
<rol:User id="xxxxxx" type="internalId"/>
<rol:MemberRole>I</rol:MemberRole>
</rol:RequestHeader>
<rol:RequestData>
<rol:VisaCaseNumber>1000000003</rol:VisaCaseNumber>
<rol:QnId>1335451</rol:QnId>
<rol:DisputeItemInfo>
<rol:DisputeItemID>0</DisputeItemID>
<rol:DisputeItemType>String</DisputeItemType>
<rol:/DisputeItemInfo>
<rol:AttachmentDescriptor>
<rol:Attachment id="1">
<rol:ContentType>image/tiff</rol:ContentType>
<rol:Comment>test</rol:Comment>
<rol:ImageData><inc:Include href="cid:556406701357"
xmlns:inc="http://www.w3.org/2004/08/xop/include"/></rol:ImageData>
</rol:Attachment>
</rol:AttachmentDescriptor>
<rol:DocType>CALET</rol:DocType>
</rol:RequestData>
</rol:SIUploadImageRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

-----=_Part_0_10714568.1392074245778
Content-Type: image/tiff
Content-Transfer-Encoding: binary
Content-ID: <556406701357>
Content-Disposition: attachment; name="TUA50001.tif"
```

## Error Handling

Members may encounter error conditions during message validation or when RTSI processes the request.

For example, if an RFC request message fails VROL validation, VROL returns a message specifying the errors in the SOAP Fault element. No XML payload is provided in the message. A sample message follows.

## SOAP-based Services

```
<soapenv:Body>
<soapenv:Fault>
<faultcode xmlns="">Client</faultcode>
<faultstring xmlns="">
<RealTimeSIResponse xmlns="http://www.visa.com/
ROLSI" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance">
<Status>
<Code>E-300400005</Code>
<Message>No case found for case number: 1000382198
and RetrievalRequestID: 011211084321.</Message>
</Status>
</RealTimeSIResponse>
</faultstring>
<detail xmlns="">
</soapenv:Fault>
</soapenv:Body>
```

## REST-based Services

```
{
  "soapenv:Envelope": {
    "soapenv:Body": {
      "soapenv:Fault": {
        "faultcode": "soapenv:Server",
        "faultstring": "One or more errors occurred while processing
the request.",
        "detail": {
          "SIErrors": {
            "Status": {
              "Code": "E-300400005",
              "Message": "No case found for case number: 1000382198
and RetrievalRequestID: 011211084321."
            }
          }
        }
      }
    }
  }
}
```

The error handling has changed and a sample message is given below.

## SOAP-based Services

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server</faultcode>
      <faultstring>One or more errors occurred while processing the
request.</faultstring>
      <detail>
        <SIErrors xmlns="http://www.visa.com/ROLSI">
          <Status>
            <Code>E-300600010</Code>
            <Message>As an Issuer, you requested to access a queue that
is available only to Acquirers.</Message>
          </Status>
        </SIErrors>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

## REST-based Services

```
{
  "soapenv:Envelope": {
    "soapenv:Body": {
      "soapenv:Fault": {
        "faultcode": "soapenv:Server",
        "faultstring": "One or more errors occurred while processing
the request.",
        "detail": {
          "SIErrors": {
            "Status": {
              "Code": "E-300600010",
              "Message": "As an Issuer, you requested to access a queue that
is available only to Acquirers."
            }
          }
        }
      }
    }
  }
}
```

## Reject Messages

VROL passes on reject and return codes and messages from other external systems with which it interfaces. The BASE II, V.I.P., and FRS manuals explain these codes.

VROL creates reject messages in certain cases. Some messages that are commonly used are given in the table below:

Message	Description
Expired Awaiting Network Response (EANR)	Message displayed when a VROL request times out while waiting for a response. The member should re-send the request later.

## General Workflow Considerations

### Case Matching

When receiving a transaction in an RTSI request, VROL performs case matching to match a transaction in an RTSI request to an existing case.

There are two choices:

- VROL uses the <VisaCaseNumber> in the RTSI request to uniquely identify the case to be updated. If there is no match, VROL sends an error message (E-120000004: Case could not be updated because no matching case was found).
   
If <VisaCaseNumber> is provided, the other case matching fields will be ignored *in the case matching step*. These fields may be used in the later locked fields check.
- If <VisaCaseNumber> is not provided, VROL will use the selected fields in the <TransactionInfo> (or the equivalent transaction fields for transaction types with no <TransactionInfo>) as described in [Table 3-4](#).
  - If required fields are missing, VROL sends an error message.
  - If all required fields are available and there is no match, VROL creates a new case. This is a “manual transaction creation” where a new case is created by sending a transaction.
  - If there are multiple matches, VROL sends an error message (E-120000005: Case could not be updated because multiple matching cases were found).

**Recommendation:** If a new case is needed, a member should use create case, or create case from TI, similar to other transactions. For case matching, <VisaCaseNumber> should always be used. The case matching with <TransactionInfo> fields should not be used to create a new case.

**Note:** RTSI transaction unique IDs, such as MessageID, are not used in case matching.

### Stage-State check

If a match to an existing case is found, VROL checks that the state and stage of the existing case allows the incoming transaction. A state-stage violation can result to a warning or error.

**Table 3-4: Case matching fields when <VisaCaseNumber> is not used**

Field	Matching Rule	Other Considerations
<VisaCaseNumber>	Must match exactly.	The recommended approach, and should be used as a single key.
<MemberCaseNumber>	Must match exactly.	Assuming these values are unique on the member's internal application, this could be used as a single key. This implies that the MemberCaseNumber being used is from the member's organization.
<RetrievalRequestID>	Must match exactly.	Can be used as a single key for matching to RFC cases, such as when an acquirer updates an RFC request with a fulfillment.

**Table 3–4: Case matching fields when <VisaCaseNumber> is not used (cont'd.)**

Field	Matching Rule	Other Considerations
<AccountNumber>	Must match exactly.	This is the card/account number. For tokenized transactions (Visa Token Service), acquirers should put token in this field for matching.
<ARN>	Must match exactly.	n/a
<DisputeCategory>	Must match exactly.	
<DisputeCategoryCondition>	Must match exactly.	
<AcquirerBIN>	Must match exactly.	n/a
<RetrievalReferenceNumber>	Must match exactly.	n/a
<SystemTraceAuditNumber>	Must match exactly.	n/a
Transaction ID - XML Element: "id" attribute of <Transaction>	This is considered a match if either the new questionnaire or existing case has a zero or blank value. If a value other than zero or blank is present on both the new questionnaire and the existing case, they must match exactly.	It should be noted that some processors use Transaction IDs that are not the Visa Transaction ID. The transaction ID supplied to VROL must be the Visa Transaction ID.
Multiple Clearing Sequence Number (MCSN) - XML Element: final two digits of "id" attribute of <Transaction> element	This is considered a match if either the new questionnaire or existing case has a zero or blank value. If a value other than zero or blank is present on both new questionnaire and the existing case, they must match exactly.	n/a
<TranDate>	Must match exactly.	n/a
Transaction Amount (<Amount> element within <Transaction> element)	This is considered a match if either the new questionnaire or existing case has a zero or blank value. If a value other than zero or blank is present on both new questionnaire and the existing case, they must match exactly.	n/a

## Locked Fields

When receiving a transaction in an RTSI request, once an existing matched case is found, VROL checks if the key transaction fields have been established in a case, that is, are the fields “locked.” For example, VROL checks if there is a selected TI, or transaction fields have been sent to opposing member.

### Selected TI Check

Prior to Locked fields check, VROL checks if there is selected TI in the case. If this occurs, there is no locked field test and the incoming fields used for locked fields are ignored.

### Locked Fields Check

The locked fields and the triggers for locking these fields, are given in [Figure B-2](#). VROL compares the fields in <TransactionInfo> (or the equivalent transaction fields for transaction types with no <TransactionInfo>) with the case folder information in the existing case to determine if there is a "locked field violation."

If the values in the RTSI request and existing case differ:

- If the field is "locked": VROL returns a hard stop error (usually E-1200000027: One or more key fields entered in the TransactionInfo section do not match the case, so the case cannot be updated).
- If the field is not "locked": VROL updates the existing field(s) in the case, such as case header and most other transaction headers.

No warning is given in either scenario.

**Recommendation:** To avoid unexpected behavior due to mismatched fields, once transaction information is established in a case, there is no need to provide it in <TransactionInfo> with subsequent actions or transactions. Also, if a case is created from a TI result, <TransactionInfo> is no longer required.

### Other Transaction Specific Checks

After the locked field check is done, VROL performs additional checking depending on the incoming transaction in the RTSI request.

Additional checks include:

- Check to determine if this is an update of an existing transaction such as a repair of a rejected transaction, or matching of related transactions.
- Check for additional required field(s) in <TransactionInfo>.
- Check of the <MessageID> required for distinguishing between different message threads.
- Check if there are any matching associated transactions. If no transactions match, VROL sends an error message (E-300300051: No matching transactions found).

### Other Miscellaneous Checks

VROL performs additional checking such as validation, access/permissions, or business rule checks.

Examples of these checking scenarios are as follows:

- Access/permissions check: For cases where user or user's organization is not configured to access RTSI services, VROL sends an error message (E-300100030: Member is not configured for RTSI services).
- Validation check: For cases such as invalid resource name or incorrect client request, VROL sends any of the following messages:
  - E-900000500: Resource Name is invalid.
  - E-300199909: Request Sent by client is incorrect. Please try again with correct request.
- Business rule check: For cases where the request does not pass the defined business rules, VROL sends an error message (E-300300057: An internal system error occurred while processing your request. If the problem persists, please contact your regional ROL Help Desk).

# Transaction Inquiries

4

VROL enables users to create a transaction inquiry (TI) to search Visa's databases for original authorization and settlement transaction information. A TI is also used for cardholder research and as the initial transaction in a dispute. Dispute transactions are not included in transaction inquiry results.

**Note:** *Transaction inquiry is a standalone activity in VROL; that is, a case is not created when the transaction inquiry results are returned. If the user takes action on the transaction inquiry results, a case is created. If no action is taken, the transaction inquiry results are deleted when the user navigates to a different VROL function or a system time-out occurs.*

The information returned in a response depends on the parameters specified in the inquiry request.

## Synchronous vs. Asynchronous Processing

Transaction inquiries can be processed synchronously or asynchronously.

- Synchronous

VROL attempts to return the data in real-time to the member host system (MHS). The MHS waits for VROL to return the inquiry response.

VROL uses existing rules and functionality to determine when to switch to asynchronous processing (either the request takes too long to process or it is determined from the parameters that it will likely take too long).

When an inquiry cannot be processed synchronously, VROL returns a case number and TI identifier that can be used to obtain the inquiry details when processing is complete.

**Note:** *Even if a member requests synchronous processing, the MHS needs to be prepared to handle an asynchronous result.*

- Asynchronous

VROL returns an acknowledgment of receipt (for inquiries that pass validation) and proceeds with asynchronous processing without waiting to see if synchronous process is possible.

Member host system may specifically request asynchronous processing by setting the AsynchronousRequestInd element in the TI request to "true."

## TI Process

The following sections describe the TI process and XML operations used by both issuers and acquirers to perform the task.

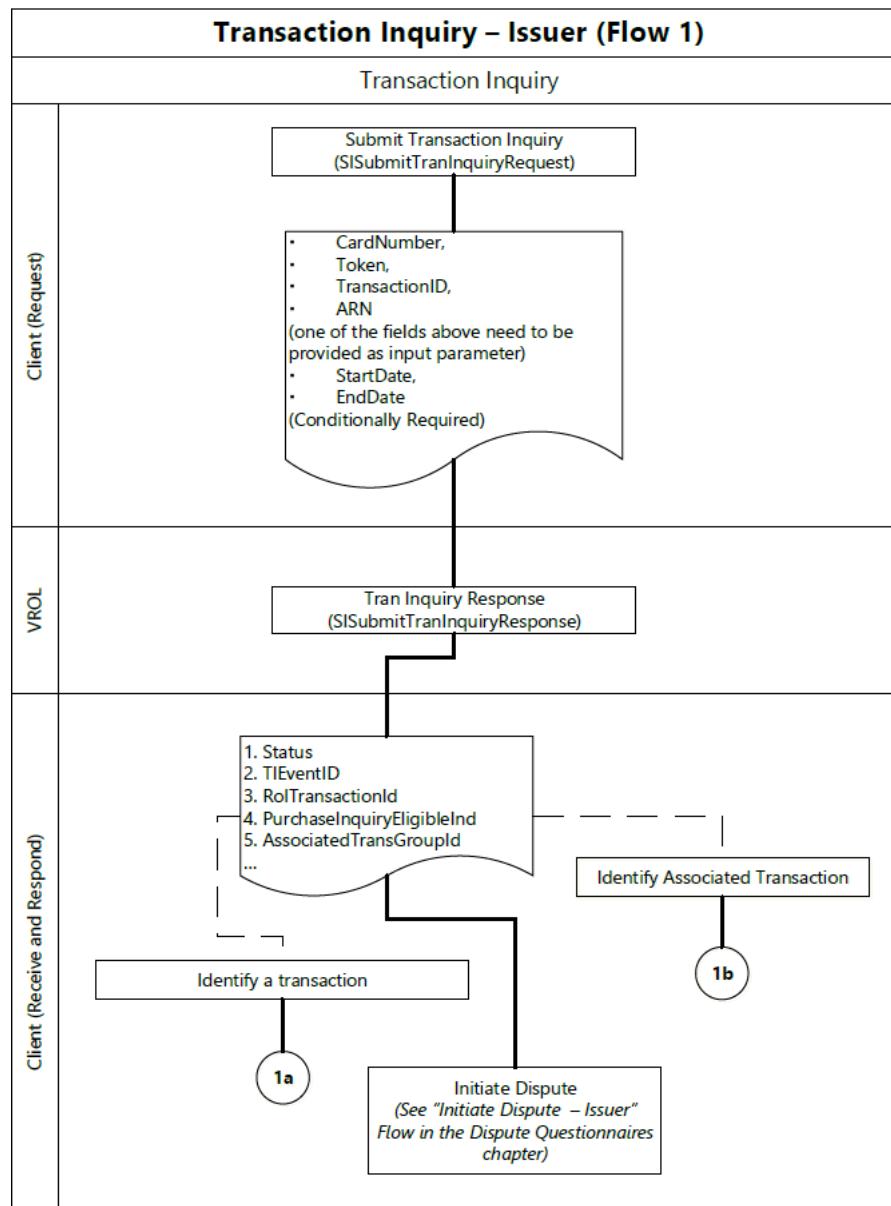
### Transaction Inquiry Flows

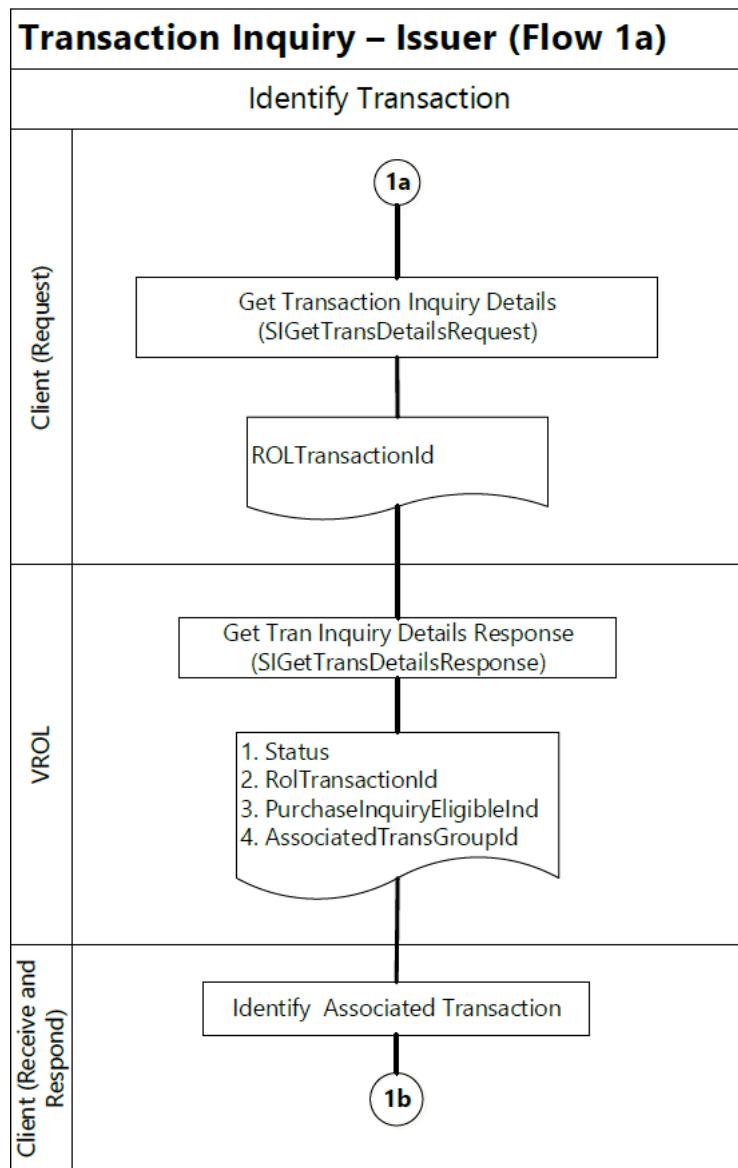
The following diagrams illustrate the transaction inquiry process.

**Note:**

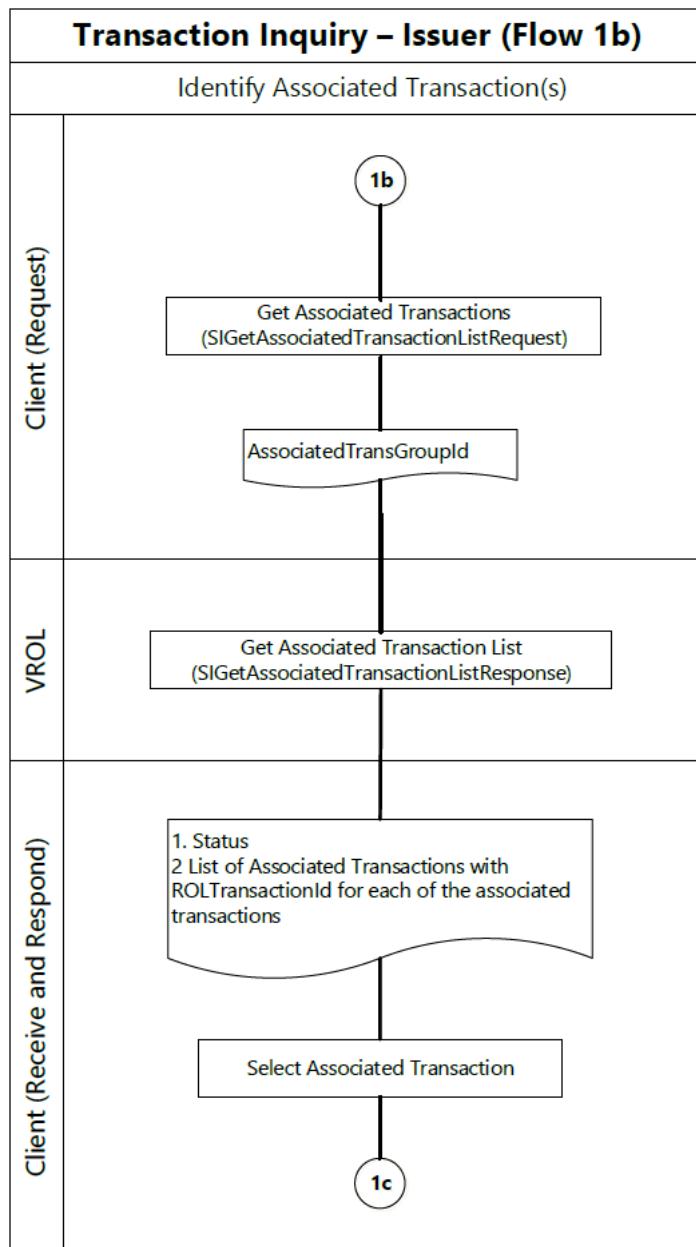
- *The diagrams list only the important elements. For the complete elements, refer to the Interface Element Specification (IES).*
- *The dashed lines in the flow refer to optional functions that the user can use, depending on the business process that the user wants to implement.*

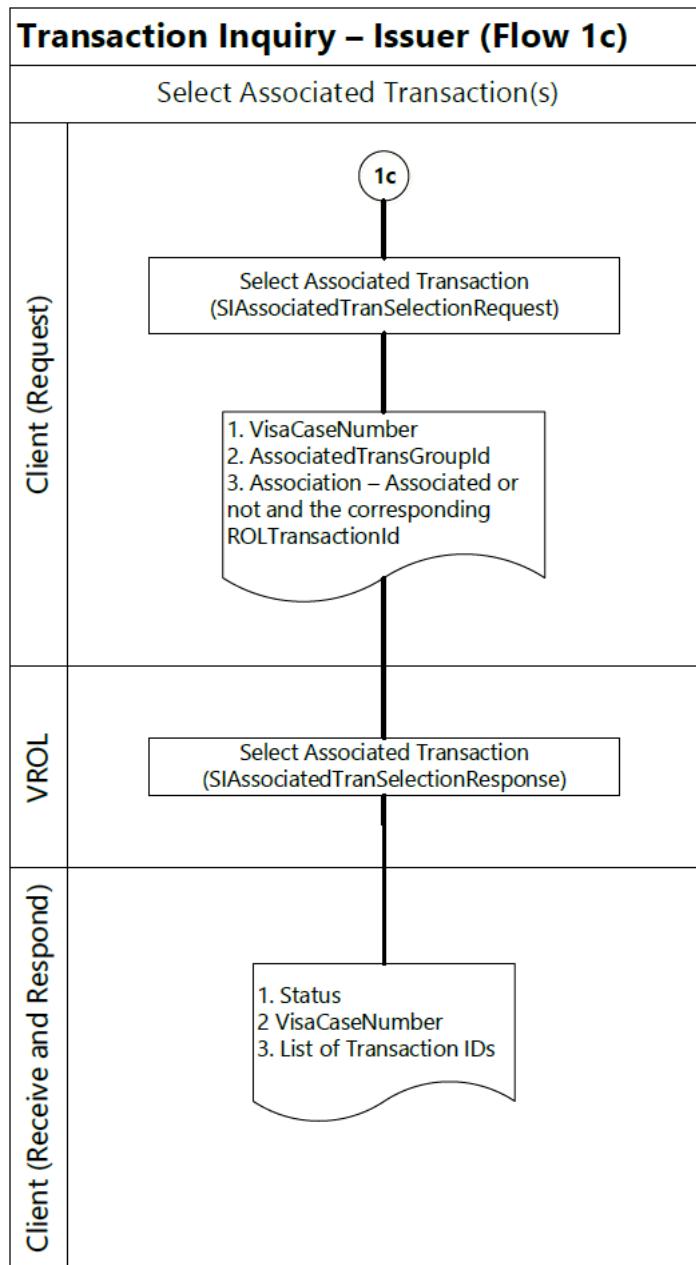
**Figure 4–1: Transaction Inquiry—Flow 1**



**Figure 4–2: Transaction Inquiry—Flow 1a**

**Figure 4–3: Transaction Inquiry—Flow 1b**



**Figure 4-4: Transaction Inquiry—Flow 1c**

### Request TI

1. User sends request SISubmitTranInquiryRequest with search criteria that identifies one or more transactions to retrieve. User can specify that the response is always asynchronous.
2. VROL sends response SISubmitTranInquiryResponse with success status.
  - Synchronous.
    - If VROL determines the results immediately, the TI results return the TI summary with a few TI fields and a unique identifier (RolTransactionID)

for each TI returned with the response. The RolTransactionId is not the same as the Transaction ID used with CPS transactions. A maximum of 500 transactions is returned, except for searches done within a single day range using a full card number.

- If VROL cannot immediately return results, only the TI Event ID is returned (similar to the asynchronous case).
- Asynchronous. The TI Event ID is returned in the response. The TI Event ID returned is used to request the TI results later. Use SIGetTransInquiryResultsRequest to return the full set of matching transactions.

A maximum of 500 transactions is returned, except for searches done within a single day range using a full card number. If more than 500 are returned, VROL returns warning message: W-121005205 (The number of matching transactions returned has exceeded the maximum limit in VROL. To see a complete set of matching transactions, revise your search criteria and try again.). If user searches within a single day range using a full card number, a maximum of 1000 transactions is returned to the user.

**Note:** *If the user gets a partial results warning, they should consider getting the results using SIGetTransInquiryResultsRequest rather than repeatedly using SISubmitTranInquiryRequest. Subsequent calls to SISubmitTranInquiryRequest will repeat the previously returned results.*

If VROL determines that the request involves a time consuming search but is able to report interim results, the partial results are returned in the response with a warning message that indicates that the query has not finished processing and additional results may be found later. The warning message is W-121005204 (A partial set of matching transactions has been returned. Use SIGetTransInquiryResultsRequest to return the full set of matching transactions.)

If a BIN search is performed and the VROL search times out, VROL sends message: E-100010003 (VROL timed out while processing your Transaction Inquiry request. Please modify your TI Request parameters and/ or try again later.)

If no transactions are found, VROL sends message: E-3003000051 (No transactions found).

3. For an asynchronous TI response or if the synchronous response could not be completed by VROL, the RTSI user sends a TI summary request SIGetTransInquiryResultsRequest using the TI Event ID to determine if the TI request has been processed.
  - If VROL has the results, it sends a response SIGetTransInquiryResultsResponse with success status. The TI results returns the TI summary and a few TI fields (equivalent to TI results grid in UI) and a unique identifier (RolTransactionID) for each transaction in the TI results. The RolTransactionId is not the same as the Transaction ID used with CPS transactions.

If VROL is only able to retrieve partial results, the warning message W-121005204 will be returned, similar to SISubmitTranInquiryResponse.

**Note:** *A subsequent call will repeat the previously returned results in addition to any new results.*

- If VROL does not yet have the results, it returns the I-300300000 message.
4. To retrieve the TI details, user sends request SIGetTransDetailsRequest using the RolTransactionId.

VROL sends response SIGetTransDetailsResponse with details of TI results.

5. User can elect to create a case SICreateCaseFromTransactionRequest using the RolTransactionId.  
VROL sends response SICreateCaseFromTransactionResponse with the VROL Case Number. Refer to [Chapter 5, Case Creation](#), for details.

## Request Original Transaction Information

1. To get original transaction information for an existing case, user sends request SIGetTransactionsFromCaseRequest.
2. VROL sends response SIGetTransactionsFromCaseResponse of full details of selected transaction and summary of other matching TIs.

## Request Associated Transactions

VROL returns a list of TI transactions that are associated with a selected transaction.

These associated transactions can include the following transaction types:

- Credits (excluding Visa Direct OCTs)
- Credit Authorizations (not displayed if the associated transaction has a matching Credit)
- Reversals (additionally includes reversals of all the exception financials)
- Authorizations
- Adjustments
- Chargebacks
- Representments
- Dispute Financials
- Dispute Response Financials

VROL also allows members to associate a group of transactions to a selected transaction.

### Note:

- *Credit Authorizations are not included if the associated transaction has a matching Credit.*
- *The selected (disputed) transaction is not included in the list of associated transactions.*
- *The list of associated transactions can include one or more Authorizations that match on the Transaction ID.*
- *Visa Direct Original Credit Transaction (OCT) is not included in the list of associated transactions.*
- *A maximum of 500 associated transactions is returned to the user.*

Any credit, reversal, or adjustment transaction in the list of associated transactions that has "Yes" as Associated value is filtered out from the list of associated transactions for subsequent disputes. However, if any Credit, Reversal, Adjustment (CRA) was submitted or saved in an Associated Transaction selection as part of a prior dispute then it is not displayed in subsequent associated transactions lists.

Before a dispute can be created, VROL performs validation using the original transaction data, transactions associated with the original transaction, and the selected dispute category. VROL uses these associated transactions during the dispute evaluation and liability assignment processes. Some associated transactions support the dispute evaluation process, while others resolve the need for a dispute to be initiated. VROL

displays an error message if it determines that the dispute is invalid. See [Chapter 7, Dispute Questionnaires](#) for more information on disputes.

1. To retrieve the RolTransactionId, the member host system triggers a SISubmitTranInquiryRequest message.

VROL returns a SISubmitTranInquiryResponse message with a RolTransactionId for every transaction.

2. To retrieve the AssociatedTransGroupId, the member host system triggers a SIGetTransDetailsRequest supplying the RolTransactionId of the selected original transaction.

**Note:** *AssociatedTransGroupId identifies a group of transactions associated to the selected transaction.*

VROL returns a SIGetTransDetailsResponse message with details of the selected transaction, including the AssociatedTransGroupId.

3. To retrieve the list of associated transactions, the member host system triggers a SIGetAssociatedTransactionListRequest message supplying the AssociatedTransGroupId.

VROL returns a SIGetAssociatedTransactionListResponse message with the list of associated transactions associated with a selected transaction. Each listed associated transaction will have its own RolTransactionId.

There is only one record for each transaction. If a duplicate is found, VROL removes this duplicate. VROL also applies upgrade or downgrade scenarios to filter duplicate authorizations.

VROL filters out and discards transactions that have a source transaction amount that is greater than the source transaction amount present in the selected transaction. For transactions that are declined, rejected, or discarded, VROL filters out these transactions from the list of associated transactions.

**Note:** *VROL returns a message informing the user if there are no associated transactions found.*

4. To retrieve the transaction details of the associated transaction, the member host system triggers a SIGetTransDetailsRequest message supplying the RolTransactionId of the associated transaction.

VROL returns a SIGetTransDetailsResponse message with details of the associated transaction.

5. To associate a group of transactions in a case, the member host system triggers a SIAssociatedTranSelectionRequest message supplying the VisaCaseNumber, RolTransactionId (of the associated transaction), AssociatedTransGroupId, and Associated value (whether to associate the transaction or not).

VROL returns a SIAssociatedTranSelectionResponse message with the VisaCaseNumber, RolTransactionId (of the associated transaction), and AssociatedTransGroupId.

RolTransactionId identifies the individual transaction that has been associated in the request, AssociatedTransGroupId identifies the group that contains all the RolTransactionIds after the association, while VisaCaseNumber provides the case to which the Ids have been associated. These three elements returned in the SIAssociatedTranSelectionResponse message help to confirm that these transactions are now tied to each other.

6. To retrieve the association status of the associated transaction in a case, the member host system triggers a SIGetAssociatedTranSelectionRequest message supplying the VisaCaseNumber and AssociatedTransGroupId.

VROL returns a SIGetAssociatedTranSelectionResponse message with the VisaCaseNumber, RolTransactionId (of the associated transaction), AssociatedTransGroupId, and Associated value (whether the transaction is associated or not) for each of the associated transactions.

**Note:** *The user can update the associated transaction of the selected transaction. For more information, see [Update Associated Transactions Selection Data](#).*

## Update Associated Transactions Selection Data

Before submitting a Dispute (or Pre-Arbitration) questionnaire, VROL allows the user to update the data of the associated transactions selection if there is a related saved (not submitted) Dispute or Pre-Arbitration questionnaire. For Pre-Arbitration, only the Associated values for new associated transactions that occurred since the Dispute questionnaire submission are editable.

1. To retrieve the list of any new net associated transactions since the last associated selection was saved or submitted, the member host system triggers a SIGetAssociatedTransactionListRequest message supplying the AssociatedTransGroupId.
- VROL returns a SIGetAssociatedTransactionListResponse message with the list of associated transactions for subsequent disputes. Each listed associated transaction will have its own RolTransactionId.
2. To retrieve the transaction details of the associated transaction, the member host system triggers a SIGetTransDetailsRequest message supplying the RolTransactionId of the associated transaction.
- VROL returns a SIGetTransDetailsResponse message with details of the associated transaction.
3. To update the data associated transactions selection, the member host system triggers a SIAssociatedTranSelectionRequest message supplying the VisaCaseNumber, RolTransactionId (of the associated transaction), AssociatedTransGroupId, and Associated value (whether to associate the transaction or not).
  4. VROL checks if the update passes the condition for updating the associated transaction selection.

**Note:**

- VROL returns an error if it determines that the update for the associated transaction selection is not allowed.
  - VROL returns a warning message if there's an attempt to update a specific transaction that is not editable.
5. VROL saves the associated transaction selection as a new event and returns a SIAssociatedTranSelectionResponse message with the VisaCaseNumber, RolTransactionId (of the associated transaction), and AssociatedTransGroupId.

## Request More Merchant Information (MMI)

This feature is enabled for all authorization and settlement transactions. It is used after a TI request.

1. User sends MMI request SIGetMMIRequest using RolTransactionId that has been returned from a TI request.
2. VROL sends response SIGetMMIResponse with the MMI data.

## TI Operations

RTSI exposes the following transaction inquiry (TI) operations.

**SISubmitTranInquiryRequest**—Used to search for transactions using a set of search criteria.

**SIGetTransInquiryResultsRequest**—Used to request the results of a previous search.

**SIGetTransDetailsRequest**—Used to request the details of a transaction from a previous search response using RolTransactionId.

**SICreateCaseFromTransactionRequest**—Used to create a case using RolTransactionId. For details, see [Chapter 5, Case Creation](#).

**SIGetTransactionsFromCaseRequest**—Used to request details of the TI transactions of a case based on the Visa Case Number.

**SIGetAssociatedTransactionListRequest**—Used to retrieve a list of associated transactions.

**SIAssociatedTranSelectionRequest**—Used to confirm or refute the associations of some or all of the associated transactions within a group to a selected transaction in a case.

**SIGetAssociatedTranSelectionRequest**—Used to retrieve the association status of the associated transaction in a case.

**SIGetMMIRequest**—Used to get merchant information for all authorization and settlement transactions.

## SISubmitTranInquiryRequest

SISubmitTranInquiryRequest accepts parameters identifying one or more transactions and returns information about the transactions found.

The following table provides information about some XML elements that should be considered when submitting a TI request or retrieving a response.

### SISubmitTranInquiryRequest

Element	Required	Description
CardNumber, Token, TransactionID, ARN, SenderAccountNumber	Yes	<p>The user is required to enter one of these fields for a TI search.</p> <p>CardNumber: Used for the card account number. It can also be used for 8-digit BIN or extended BIN, and Sender Account Number searches.</p> <p>For Visa Direct transactions such as Money Transfer (credit) and Merchant Payment (mVisa), the originating or sending account is entered in the SenderAccountNumber field and not in the CardNumber field. The CardNumber is the recipient's account number. However, for Issuer and Issuer/Acquirer organizations that are enabled for Visa Direct dispute processing, VROL will search both CardNumber and SenderAccountNumber with the value entered in the CardNumber field.</p> <p>For Visa Direct Merchant Payments, the Merchant's PAN is the CardNumber in the transaction. For a Visa Direct push payment, enter the Merchant's PAN in the CardNumber field and enter the Consumer's card account number in the SenderAccountNumber field.</p> <p><b>Note:</b> <i>Visa Direct original transactions use Original Credit Transactions (OCTs) that are sent from the originating member using an acquirer BIN to the receiving member's issuer BIN. See "Visa Direct Original Credit Transaction (OCT) – Global Implementation Guide" and "mVisa Mobile Push Payment Program Implementation Guide" on Visa Online for more details about Visa Direct.</i></p>
CardNumber, Token, TransactionID, ARN, SenderAccountNumber (continued)	Yes	<p>For tokenized transactions (Visa Token Service):</p> <ul style="list-style-type: none"> <li>For TI search, users can put token or token BIN in the CardNumber or Token field. In the TI responses, the card/account number will be returned in CardNumber and the Token in Token fields, respectively.</li> <li>Issuers should put token in the Token field. When receiving tokenized transactions in the response, the Token field will contain the token value for issuers.</li> </ul> <p><b>Note:</b> <i>Token BIN searches are supported in the CardNumber field but not in the Token field.</i></p> <p>For associated transactions:</p> <ul style="list-style-type: none"> <li>Authorization transaction type</li> </ul> <p>In TI search, the Card/Account Number and Central Processing Date (CPD) of the pinned transaction are used. The date range is calculated as +3 and - [Span] days from the CPD.</p> <p><b>Note:</b> <i>Each transaction type has a default [Span] value, which is a configurable property.</i></p> <ul style="list-style-type: none"> <li>Credits, Credit Authorizations, Reversals, Adjustments, Chargebacks transaction types</li> </ul> <p>In TI search, the Card/Account Number and CPD of the pinned transaction are used. The date range is calculated where the From Date is -3 from the CPD and the To Date is the CPD + 90 days with a maximum of the current date</p> <p>TransactionID: If a Multiple Clearing Sequence Number (MCSN) exists for the transaction, the two-digit MCSN can be added to the end of the transaction ID in the request and will be returned if available in the response. A transaction ID is typically 15 digits; however, the transaction ID will be 17 digits with an additional MCSN.</p> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li><i>If a 17-digit transaction ID has "00" as its last two digits, VROL will use the first 15 digits as the transaction ID instead.</i></li> <li><i>VROL returns an error if all zeroes are entered for TransactionID.</i></li> </ul>

Element	Required	Description
StartDate, EndDate	Conditional (see Description)	<p>StartDate and EndDate are both required unless searching by TransactionID or ARN.</p> <p>For the BIN and extended BIN searches, the start and end dates must be the same. Credit searches will not be allowed if any part of the date range is older than 24 months.</p> <p>If either Transaction ID or ARN is supplied and a date range is not provided, VROL calculates the date based on the Transaction ID or ARN (+[days forward]/- 5 days).</p>
IncludeCreditsOrReversals OrAdjustmentsInd	No	<p>This element is used to include credit/reversals or adjustments with the original transaction. The search should include &lt;CardNumber&gt;, &lt;StartDate&gt; and &lt;EndDate&gt; to capture the original transaction. Other elements such as &lt;TransactionID&gt; or &lt;ARN&gt; can also be used in the TI search.</p> <p>If &lt;IncludeCreditsOrReversalsOrAdjustmentsInd&gt;=true, VROL performs two searches to provide the transactions returned in the response. The first search returns the original transactions matching all the elements provided in the request. The second search returns any credit/reversals or adjustments matching &lt;CardNumber&gt; and &lt;StartDate&gt;, that is, VROL is ignoring &lt;TransactionID&gt; and all other elements except &lt;CardNumber&gt; and &lt;StartDate&gt;, when returning these additional transactions. The &lt;EndDate&gt; on the second search is replaced with the current date.</p>
TransactionTypeOption	No	<p>Domain values:</p> <ul style="list-style-type: none"> <li>• Standard = Standard List</li> <li>• Expanded = Expanded List</li> <li>• Single = Single Transaction Type</li> <li>• PPCS = PPCS - Eligible Transactions</li> <li>• CardOnFile = Stored Credential Transactions (Card on File)</li> </ul> <p>This field is used to restrict the types of transactions returned by a TI request.</p> <p>See &lt;TransactionType&gt; field in IES to distinguish which Transaction Types are included in the Standard versus Extended List.</p>
TransactionType	Conditional (see Description)	This field is required when TransactionTypeOption=Single.
AsynchronousRequestInd	No	If the value of this element in the request is "true," VROL will process the transaction inquiry request asynchronously.
ShowDuplicateAuthWith SameTranIDInd	No	If the value of this element in the request is "true," VROL will show the duplicate authorizations with the same Transaction ID.
IncludeInd	No	<p>Domain values: Authorizations/Inquiries, Financials, Both</p> <p>If &lt;TransactionTypeOption&gt;=Single, VROL defaults the value to "Both" and this field will be ignored.</p> <p>This field is used to select whether to return authorizations, financials, or both in the TI response.</p>
TranStatus	No	<p>Domain values: Approved, Denied or Referred</p> <p><b>Note:</b> All BASE II transactions are considered "Approved."</p>

Element	Required	Description
AuthCode	No	Text input for this field is not case sensitive, which means that entering '123ABC' or '123abc' will return the same TI search results.
TranAmountRange1, TranAmountRange2	No	These fields are used to specify the transaction amount range in the TI request.
RetrievalReferenceNumber	No	Text input for this field is not case sensitive, which means that entering '123ABC' or '123abc' will return the same TI search results.  This field can be shorter than 12 digits and can contain non-numeric alpha characters.
NetworkID	No	This field is used to specify the applicable V.I.P. or BASE II numeric network ID.  <b>Note:</b> For a full list of network IDs, refer to the IES.
MCC	No	This field is used to enter a valid 4-digit Merchant Category Code.

**SISubmitTranInquiryResponse**

Element	Required	Description
TIEventID		Unique internal ID returned in response, which can be used for subsequent operations for a TI request.
RollTransactionId		Uniquely identifies a transaction. Member can use this to access the TI transaction details.
PurchaseInquiryEligibleInd		Indicates whether the TI transaction is eligible or not for Purchase Inquiry (PI).  If <PurchaseInquiryEligibleInd>=true is returned in the response, this indicates that the TI transaction is currently eligible for PI. The PI eligibility indicator may change if a merchant is activated or deactivated.
AssociatedTransGroupId		Uniquely identifies a list of associated transactions.  Members can use this to retrieve a list of associated transactions or to associate a group of transactions to a case.

The SISubmitTranInquiryRequest operation contains request and response messages as shown in the following sample messages.

**Request Message—SOAP-based Services**

The following example shows a sample SubmitTranInquiryRequest message used to submit a basic transaction inquiry request.

```
<SISubmitTranInquiryRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <CardNumber>xxxxxxxxxxxxxxxx</CardNumber>
    <Token>xxxxxxxxxxxxxxxx</Token> [for tokenized transactions only]
    ...
    <StartDate>2014-05-05</StartDate>
    <EndDate>2014-05-05</EndDate>
    <IncludeInd>Both</IncludeInd>
    <IncludeCreditsOrReversalsOrAdjustmentsInd>true
    </IncludeCreditsOrReversalsOrAdjustmentsInd>
    <TranStatus>Denied Or Referred</TranStatus>
    <TransactionTypeOption>CardOnFile</TransactionTypeOption>
    <TransactionType>String</TransactionType>
    <ShowDuplicateAuthWithSameTranIDInd>true
    </ShowDuplicateAuthWithSameTranIDInd>
    <AuthCode>a</AuthCode>
    <TranAmountRange1>0</TranAmountRange1>
    <TranAmountRange2>0</TranAmountRange2>
    <RetrievalReferenceNumber>a</RetrievalReferenceNumber>
    <NetworkID>aaa</NetworkID>
    <MCC>0000</MCC>
    <AsynchronousRequestInd>true</AsynchronousRequestInd>
    <BusinessAppId>aa</BusinessAppId>
  </requestData>
</SISubmitTranInquiryRequest>
```

### Response Message—Synchronous Processing—SOAP-based Services

The inquiry data returned in the response is based on the data requested. The field IdentifiedTransaction indicates whether this is the selected transaction in a case. This transaction is the equivalent of the Pinned (selected) Transaction in the UI case detail events list.

```
<SISubmitTranInquiryResponse>
    <Status>...</Status>
    <responseData>
        <TIEventID>0000000</TIEventID>
        <TransactionSummary>
            <RolTransactionId>0000001</RolTransactionId>
            <IdentifiedTransaction>true</IdentifiedTransaction>
            <TransactionType>Trantype</TransactionType>
            <TransactionTypeShortDescription>Auth
            </TransactionTypeShortDescription>
            <TransactionDateTime>2014-05-05T23:43:48+00:00
            </TransactionDateTime>
            TI summary data (many fields - see IES)
            <CardNumber>xxxxxxxxxxxxxxxxxx</CardNumber>
            <Token>xxxxxxxxxxxxxxxxxx</Token> [for tokenized
                transactions only]
        </TransactionSummary>
        <TransactionSummary>
            <RolTransactionId>0000002</RolTransactionId>
            TI summary data (many fields - see IES)
        </TransactionSummary>
        <TransactionSummary>
            <RolTransactionId>0000003</RolTransactionId>
            TI summary data (many fields - see IES)
        </TransactionSummary>
    </responseData>
</SISubmitTranInquiryResponse>
```

### Response Message—Asynchronous Processing—SOAP-based Services

The inquiry data returned in the response is based on the data requested.

```
<SISubmitTranInquiryResponse>
    <Status>...</Status>
    <responseData>
        <TIEventID>0000000</TIEventID>
    </responseData>
</SISubmitTranInquiryResponse>
```

### Request Message—REST-based Services

The following example shows a sample SubmitTranInquiryRequest message used to submit a basic transaction inquiry request.

```
{  
    "RequestHeader": "...",  
    "requestData": {  
        "CardNumber": "xxxxxxxxxxxxxxxxxx",  
        "Token": "xxxxxxxxxxxxxxxxxx", [for tokenized  
        transactions only]  
        ...  
        "StartDate": "2016-01-01",  
        "EndDate": "2016-01-01",  
        "IncludeInd": "Financials",  
        "IncludeCreditsOrReversalsOrAdjustmentsInd": true,  
        "TranStatus": "Denied Or Referred",  
        "TransactionTypeOption": "Expanded",  
        "TransactionType": "A",  
        "ShowDuplicateAuthWithSameTranIDInd": true,  
        "AuthCode": "a",  
        "TranAmountRange1": 1,  
        "TranAmountRange2": 1,  
        "RetrievalReferenceNumber": "a",  
        "NetworkID": "aaa",  
        "MCC": "1111",  
        "AsynchronousRequestInd": true  
        "BusinessAppId": "aa"  
    }  
}
```

### Response Message—Synchronous Processing—REST-based Services

The inquiry data returned in the response is based on the data requested. The field IdentifiedTransaction indicates whether this is the selected transaction in a case. This transaction is the equivalent of the Pinned (selected) Transaction in the UI case detail events list.

```
{
  "Status": "...",
  "responseData":{
    "TIEventID": 1,
    "Repository": "VTRS",
    "TransactionSummary":{
      "R01TransactionId": 1,
      "IdentifiedTransaction": true,
      "TransactionType": "A",
      "TransactionTypeShortDescription": "A",
      "TransactionDateTime": "2016-01-01T19:31:47Z",
      "CPD": "2016-01-01",
      ...
    },
    "TIErrorResponse":{
      "CardNumber": "xxxxxxxxxxxxxxxxxxxx",
      "Token": "xxxxxxxxxxxxxxxxxxxx",
      "TransactionID": "a",
      "ARN": "1111111111111111111111111111",
      "StartDate": "2016-01-01",
      "EndDate": "2016-01-01",
      ...
      "Errors":{
        "Description": "A"
      }
    }
  }
}
```

### Response Message—Asynchronous Processing—REST-based Services

The inquiry data returned in the response is based on the data requested.

```
{
  "Status": "...",
  "responseData":{
    "TIEventID": 1,
  }
}
```

### **SIGetTransInquiryResultsRequest**

SIGetTransInquiryResultsRequest accepts a TI Event ID and returns transaction summary information. This operation contains request and response messages as shown in the following sample messages.

Element	Description
R01TransactionId	Uniquely identifies a transaction. Member can use this to access the TI transaction details.

PurchaseInquiryEligibleInd	Indicates whether the TI transaction is eligible or not for Purchase Inquiry (PI).  If <PurchaseInquiryEligibleInd>=true is returned in the response, this indicates that the TI transaction is currently eligible for PI. The PI eligibility indicator may change if a merchant is activated or deactivated.
AssociatedTransGroupId	Uniquely identifies a list of associated transactions.  Members can use this to retrieve a list of associated transactions or to associate a group of transactions to a case.

### Request Message—SOAP-based Services

The following example shows a sample SIGetTransInquiryResultsRequest message.

```
<SIGetTransInquiryResultsRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <TIEventID>0000000</TIEventID>
  </requestData>
</SIGetTransInquiryResultsRequest>
```

### Response Message—SOAP-based Services

The following example shows the transaction inquiry summary data returned in the SIGetTransInquiryResultsResponse. Note that the response data depends on the data available in VROL.

```
<SIGetTransInquiryResultsResponse>
  <Status>...</Status>
  <responseData>
    <TIEventID>0000000</TIEventID>
    <TransactionSummary>
      <R0lTransactionId>0000001</R0lTransactionId>
      <IdentifiedTransaction>true</IdentifiedTransaction>
      <TransactionType>0100</TransactionType>
      <TransactionTypeShortDescription>Auth
      </TransactionTypeShortDescription>
      <TransactionDateTime>2014-05-05T23:43:48+00:00
      </TransactionDateTime>
      TI summary data (many fields - see IES)
    </TransactionSummary>
    <TransactionSummary>
      <R0lTransactionId>0000002</R0lTransactionId>
      TI summary data (many fields - see IES)
    </TransactionSummary>
    <TransactionSummary>
      <R0lTransactionId>0000003</R0lTransactionId>
      TI summary data (many fields - see IES)
    </TransactionSummary>
  </responseData>
</SIGetTransInquiryResultsResponse>
```

## Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData": {
    "TIEventID": "0000000"
  }
}
```

## Response Message—REST-based Services

```
{
  "Status": "...",
  "responseData": [
    {
      "TIEventID": 1,
      "TransactionSummary": [
        {
          "R01TransactionId": 1,
          "IdentifiedTransaction": true,
          "TransactionType": "A",
          "TransactionTypeShortDescription": "A",
          "TransactionDateTime": "2016-01-01T19:31:47Z",
          "CPD": "2016-01-01",
        },
        {
          "R01TransactionId": "0000002"
        },
        {
          "R01TransactionId": "0000003"
        }
      ]
    }
  ]
}
```

## **SIGetTransDetailsRequest**

SIGetTransDetailsRequest accepts R01TransactionId and returns detail-level data for the returned transaction. This operation contains request and response messages, as shown in the following sample messages.

The following XML elements should be considered when submitting a TI request or retrieving a response.

### **SIGetTransDetailsRequest**

Element	Required	Description
IncludeDescriptionsInd	No	If the value of this optional element in the request is "true", the response message will contain both codes and the associated descriptions. If not included or set to "false", the response message will contain all codes, but only a subset of descriptions.
R01TransactionId		Uniquely identifies a transaction. Member can use this to access the TI transaction details.
FieldsToInclude	No	Optional field used to restrict fields returned by a response to those needed for a subdraft. blank = all fields, "SD" = subdraft

ProcessingCode should be considered when retrieving SIGetTransDetailsResponse. The element specifies the type of transaction being disputed.

### SIGetTransDetailsResponse

Repository	TransactionType	TransactionMessageType	ProcessingCode
SMS	0200/00	0200	00
BASE II	TC 05	TC 05	(missing)

**Note:** The values in the above table are only examples.

Element	Description
CardNumber	Card/Account Number  For tokenized transactions (Visa Token Service): For acquirers receiving tokenized transactions, this field will always contain the card/account number, even if Token was used in the TI request
Token	Returned with tokenized transactions (Visa Token Service)
Token Assurance Level (for issuers only)	
PurchaseInquiryEligibleInd	Indicates whether the TI transaction is eligible or not for Purchase Inquiry (PI).  If <PurchaseInquiryEligibleInd>=true is returned in the response, this indicates that the TI transaction is currently eligible for PI. The PI eligibility indicator may change if a merchant is activated or deactivated.
AssociatedTransGroupId	Uniquely identifies a list of associated transactions.  Members can use this to retrieve a list of associated transactions or to associate a group of transactions to a case.
PPCSEligibleInd	Indicates that the TI transaction is eligible for Preauthorized Payment Cancellation Service (PPCS).

### Request Message—SOAP-based Services

```
<SIGetTransDetailsRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <rolTransactionId>0000000</rolTransactionId>
    <includeDescriptionsInd>true</includeDescriptionsInd>
    <fieldsToInclude>SD</fieldsToInclude>
  </requestData>
</SIGetTransDetailsRequest>
```

## Response Message—SOAP-based Services

```

<SIGetTransDetailsResponse>
    <Status>...</Status>
    <responseData>
        <TransactionDetails>
            <RolTransactionId>0000000</RolTransactionId>
            <IdentifiedTransaction>true</IdentifiedTransaction>
            <TransactionType>0100</TransactionType>
            <TransactionTypeShortDescription>Auth
            </TransactionTypeShortDescription>
            <TransactionDateTime>2014-05-05T23:43:48+00:00
            </TransactionDateTime>
            <CPD>2016-08-13</CPD>
            <BusinessAppId>aa</BusinessAppId>
            <BusinessAppIdDesc>String</BusinessAppIdDesc>
            <SourceOfFunds>a</SourceOfFunds>
            <TransactionAmount currency="840">31.55</TransactionAmount>
TI details data (many fields - see IES)
            <CardNumber>xxxxxxxxxxxxxxxx</CardNumber>
            <Token>xxxxxxxxxxxxxxxx</Token> [for tokenized
                transactions only]
            <CVV2Result>result</CVV2Result>
            <CVV2ResultDesc>result description</CVV2ResultDesc>
            <PurchaseInquiryEligibleInd>true</PurchaseInquiryEligibleInd>
            <RdrEligibilityInd>true</RdrEligibilityInd>
            <AssociatedTransGroupId>0001</AssociatedTransGroupId>
TI details data (many fields - see IES)
            <TokenAssuranceLevel>00</TokenAssuranceLevel>
                [issuer only field for tokenized transactions]
            <TokenRequestorId>xxxxxxxxxxxx</TokenRequestorId>
                [issuer only field for tokenized transactions]
            <TokenExpirationDate>2018-01-31</TokenExpirationDate>
                [issuer only field for tokenized transactions]
            <TI details data> (many fields - see IES)
            <TransactionMessageType>0100</TransactionMessageType>
            <ProcessingCode>00</ProcessingCode>
            <ProcessingCodeDescription>Goods/Services
            </ProcessingCodeDescription>
TI details data (many fields - see IES)
        </TransactionDetails>
    </responseData>
</SIGetTransDetailsResponse>

```

## Request Message—REST-based Services

```

{
    "RequestHeader": "...",
    "requestData": {
        "RolTransactionId": "0000000",
        "IncludeDescriptionsInd": "true",
        "FieldsToInclude": "SD"
    }
}

```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData":{  
        "transactionDetails":{  
            "rolTransactionId": 1,  
            "identifiedTransaction": true,  
            "transactionType": "A",  
            "transactionTypeShortDescription": "A",  
            "transactionDateTime": "2016-01-01T19:31:47Z",  
            "CPD": "2016-01-01",  
            "businessAppId": "aa",  
            "businessAppIdDesc": "Short description",  
            "sourceOfFunds": "a",  
            "transactionAmount":{  
                "@currency": "000",  
                "$": 0  
            },  
            TI details data (many fields - see IES)  
            "cardNumber": "000000000000",  
            "token": "xxxxxxxxxx", [for tokenized transactions only]  
            "CVV2Result": "result",  
            "CVV2ResultDesc": "result description",  
            "PurchaseInquiryEligibleInd": "true",  
            "RdrEligibilityInd": "true"  
            "AssociatedTransGroupId": 0001,  
            TI details data (many fields - see IES)  
            "TokenAssuranceLevel": "00",  
            [issuer only field for tokenized transactions]  
            "TokenRequestorId": "xxxx",  
            [issuer only field for tokenized transactions]  
            "TokenExpirationDate": "2018-08-13",  
            [issuer only field for tokenized transactions]  
            TI details data (many fields - see IES)  
            "transactionMessageType": "0100",  
            "processingCode": "00",  
            "processingCodeDescription": "description",  
            TI details data (many fields - see IES)  
        }  
    }  
}
```

## SIGetTransactionsFromCaseRequest

SIGetTransactionsFromCaseRequest returns the TI details for the "selected" transaction in the case; that is, the original transaction being disputed. Also, it returns summaries for the "selected" and "non-selected" transactions returned from the TI search that are also stored with the case.

The following XML element should be considered when submitting a TI request or retrieving a response.

Element	Description
RolTransactionId	Uniquely identifies a transaction. Member can use this to access the TI transaction details.
PurchaseInquiryEligibleInd	Indicates whether the TI transaction is eligible or not for Purchase Inquiry (PI).  If <PurchaseInquiryEligibleInd>=true is returned in the response, this indicates that the TI transaction is currently eligible for PI. The PI eligibility indicator may change if a merchant is activated or deactivated.
AssociatedTransGroupId	Identifies a group of transactions associated to the selected transaction, and is used to access the associated transaction detail information.

### Request Message—SOAP-based Services

```
<SIGetTransactionsFromCaseRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
    </requestData>
</SIGetTransactionsFromCaseRequest>
```

### Response Message—SOAP-based Services

```
<SIGetTransactionsFromCaseResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <TransactionSummary>
            <RolTransactionId>0000001</RolTransactionId>
            TI summary data (many fields - see IES)
        </TransactionSummary>
        <TransactionSummary>
            <RolTransactionId>0000002</RolTransactionId>
            TI summary data (many fields - see IES)
        </TransactionSummary>
        <TransactionSummary>
            <RolTransactionId>0000003</RolTransactionId>
            TI summary data (many fields - see IES)
        </TransactionSummary>
        <PinnedTransactionDetails>
            <RolTransactionId>0000003</RolTransactionId>
            TI details data (many fields - see IES)
        </PinnedTransactionDetails>
    </responseData>
</SIGetTransactionsFromCaseResponse>
```

### Request Message—REST-based Services

```
{
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": "0000000000"
    }
}
```

## Response Message—REST-based Services

```
{  
    "Status": "..."  
    "responseData": {  
        "VisaCaseNumber": "0000000000",  
        "TransactionSummary": [  
            {  
                "R0lTransactionId": "0000000",  
                TI summary data (many fields - see IES)  
            },  
            {  
                "R0lTransactionId": "0000000",  
                TI summary data (many fields - see IES)  
            },  
            {  
                "R0lTransactionId": "0000000",  
                TI summary data (many fields - see IES)  
            },  
            "PinnedTransactionDetails": {  
                "R0lTransactionId": "0000000",  
                TI details data (many fields - see IES)  
            }  
        ]  
    }  
}
```

## SIGetAssociatedTransactionListRequest

SIGetAssociatedTransactionListRequest enables users to retrieve a list of associated transactions by supplying the AssociatedTransGroupId.

The following element should be considered when submitting a request or receiving a response from this service.

Element	Description
AssociatedTransGroupId	Identifies a group of transactions associated to the selected transaction, and is used to access the associated transaction detail information.
R0lTransactionId	Uniquely identifies a transaction. User can use this to access all associated transaction details.
ATRTranType	Determines the type of associated transaction. Domain values: Dispute, PreArb, RDR

SIGetAssociatedTransactionList contains request and response messages as shown in the following sample messages.

**Request Message—SOAP-based Services**

```
<SIGetAssociatedTransactionListRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <AssociatedTransGroupId>123456</AssociatedTransGroupId>
    <R0lTransactionId>000000000001</R0lTransactionId>
  </requestData>
</SIGetAssociatedTransactionListRequest>
```

## Response Message—SOAP-based Services

```
<SIGetAssociatedTransactionListResponse>
  <Status>...</Status>
  <responseData>
    <AssociatedTransactionList>
      <AssociatedTransGroupId>123456</AssociatedTransGroupId>
      <RolTransactionId>111111111111</RolTransactionId>
      <TransactionType>String</TransactionType>
      <TransactionAmount currency="840">100
      </TransactionAmount>
      <TranDate>2016-01-01</TranDate>
      <MerchantName>Merchant 1</MerchantName>
      <TransactionID>123456789</TransactionID>
      <AuthCode>222222</AuthCode>
      <ARN>10100000000000000000000000</ARN>
      <RetrievalReferenceNumber>22222222
      </RetrievalReferenceNumber>
      <MatchingPass>String</MatchingPass>
      <MatchingScore>String</MatchingScore>
    </AssociatedTransactionList>
    <AssociatedTransactionList>
      <AssociatedTransGroupId>123456</AssociatedTransGroupId>
      <RolTransactionId>0</RolTransactionId>
      <TransactionType>String</TransactionType>
      <TransactionAmount currency="840">100
      </TransactionAmount>
      <TranDate>2016-01-01</TranDate>
      <MerchantName>Merchant 2</MerchantName>
      <TransactionID>123456789</TransactionID>
      <AuthCode>222222</AuthCode>
      <ARN>10100000000000000000000000</ARN>
      <RetrievalReferenceNumber>22222222
      </RetrievalReferenceNumber>
      <MatchingPass>String</MatchingPass>
      <MatchingScore>String</MatchingScore>
    </AssociatedTransactionList>
    <AssociatedTransactionList>
      <AssociatedTransGroupId>123456</AssociatedTransGroupId>
      <RolTransactionId>111111111111</RolTransactionId>
      <TransactionType>String</TransactionType>
      <TransactionAmount currency="840">100
      </TransactionAmount>
      <TranDate>2016-01-01</TranDate>
      <MerchantName>Merchant 3</MerchantName>
      <TransactionID>123456789</TransactionID>
      <AuthCode>222222</AuthCode>
      <ARN>10100000000000000000000000</ARN>
      <RetrievalReferenceNumber>22222222
      </RetrievalReferenceNumber>
      <MatchingPass>String</MatchingPass>
      <MatchingScore>String</MatchingScore>
    </AssociatedTransactionList>
  </responseData>
</SIGetAssociatedTransactionListResponse>
```

**Request Message—REST-based Services**

```
{  
  "RequestHeader": "...",  
  "requestData":{  
    "AssociatedTransGroupId": 123456,  
    "RolTransactionId": 000000000001  
  }  
}
```

**Response Message—REST-based Services**

```
{  
  "Status": "...",  
  "responseData":{  
    "AssociatedTransactionList": [  
      {  
        "AssociatedTransGroupId": 12346,  
        "RolTransactionId": 111111111111,  
        "TransactionType": "TC05",  
        "TransactionAmount":{  
          "value": 59.97,  
          "currency": "554"  
        },  
        "TranDate": "2017-07-19",  
        "MerchantName": "Merchant 1",  
        "TransactionID": "123456789",  
        "AuthCode": "111111",  
        "ARN": "10100000000000000000000000",  
        "RetrievalReferenceNumber": "22222222",  
        "MatchingPass": "1",  
        "MatchingScore": "100"  
      },  
      {"AssociatedTransGroupId": 12346,  
       "RolTransactionId": 111111111111,  
       "TransactionType": "TC05",  
       "TransactionAmount":{  
         "value": 59.97,  
         "currency": "554"  
       },  
       "TranDate": "2017-07-19",  
       "MerchantName": "Merchant 1",  
       "TransactionID": "123456789",  
       "AuthCode": "111111",  
       "ARN": "10100000000000000000000000",  
       "RetrievalReferenceNumber": "22222222",  
       "MatchingPass": "1",  
       "MatchingScore": "100"  
     },  
      {"AssociatedTransGroupId": 12346,  
       "RolTransactionId": 111111111111,  
       "TransactionType": "TC05",  
       "TransactionAmount":{  
         "value": 59.97,  
         "currency": "554"  
       },  
       "TranDate": "2017-07-19",  
     ]  
  }  
}
```

```
        "MerchantName": "Merchant 1",
        "TransactionID": "123456789",
        "AuthCode": "111111",
        "ARN": " 10100000000000000000000000000000",
        "RetrievalReferenceNumber": "22222222",
        "MatchingPass": "1",
        "MatchingScore": "100"
    }
]
}
```

## SIAssociatedTranSelectionRequest

### Important:

The Associated Transaction call functionality that allows members to provide the "change reason" for the default selection of the Associated Transaction will not be available until a future release (TBD). The "change reason" will still be supported in the questionnaire request.

The following XML elements should be considered when submitting a request or receiving a response from this service.

Element	Description
Associated	Indicates whether or not to associate a group of transactions to the selected transaction.  Domain values: <ul style="list-style-type: none"><li>• Y=Associate the group of transactions to the selected transaction.</li><li>• N=Do not associate the group of transactions to the selected transaction.</li></ul>
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.  <b>Note:</b> In the schema, the VROL Case Number is <VisaCaseNumber>.
RolTransactionId (of the associated transaction)	Uniquely identifies a transaction.  Member can use this to access the TI transaction details. In the context of this Web service, this ID refers to an associated transaction.
AssociatedTransGroupId	Identifies a group of transactions associated to the selected transaction, and is used to access the associated transaction detail information.

SIAssociatedTranSelection contains request and response messages as shown in the following sample messages.

**Request Message—SOAP-based Services**

```
<SIAssociatedTranSelectionRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <AssociatedTransGroupId>0000</AssociatedTransGroupId>
    <Association>
      <R0lTransactionId>123456783</R0lTransactionId>
      <Associated>N</Associated>
    </Association>
    <Association>
      <R0lTransactionId>123456782</R0lTransactionId>
      <Associated>Y</Associated>
    </Association>
    <Association>
      <R0lTransactionId>123456789</R0lTransactionId>
      <Associated>Y</Associated>
    </Association>
  </requestData>
</SIAssociatedTranSelectionRequest>
```

**Response Message—SOAP-based Services**

```
<SIAssociatedTranSelectionResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <AssociatedTransGroupId>0000</AssociatedTransGroupId>
    <R0lTransactionIdsList>
      <R0lTransactionId>123456783</R0lTransactionId>
      <R0lTransactionId>123456782</R0lTransactionId>
      <R0lTransactionId>123456789</R0lTransactionId>
    </R0lTransactionIdsList>
  </responseData>
</SIAssociatedTranSelectionResponse>
```

## Request Message—REST-based Services

```
{  
    "RequestHeader": "...",  
    "requestData": {  
        "VisaCaseNumber": 1000000000,  
        "AssociatedTransGroupId": 1234,  
        "Association": [  
            {  
                "RolTransactionId": 111,  
                "Associated": "N"  
            },  
            {  
                "RolTransactionId": 222,  
                "Associated": "Y"  
            },  
            {  
                "RolTransactionId": 333,  
                "Associated": "N"  
            }  
        ]  
    }  
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "VisaCaseNumber": 1000000000,  
        "AssociatedTransGroupId": 1234,  
        "RolTransactionIdsList": {  
            "RolTransactionId": [  
                111,  
                222,  
                333  
            ]  
        }  
    }  
}
```

## SIGetAssociatedTranSelectionRequest

The following XML element should be considered when submitting this request service.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL. <b>Note:</b> In the schema, the VROL Case Number is <VisaCaseNumber>.
AssociatedTransGroupId	Used to identify the status of each associated transaction (whether the transaction is associated or not).

**Note:** The response elements of this Web service are VisaCaseNumber, RolTransactionId (of the associated transaction), AssociatedTransGroupId, and Associated.

SIGetAssociatedTranSelection contains request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SIGetAssociatedTranSelectionRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <AssociatedTransGroupId>0000</AssociatedTransGroupId>
  </requestData>
</SIGetAssociatedTranSelectionRequest>
```

### Response Message—SOAP-based Services

```
<SIGetAssociatedTranSelectionResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <AssociatedTransGroupId>0000</AssociatedTransGroupId>
    <Association>
      <rolTransactionId>0000001</rolTransactionId>
      <Associated>N</Associated>
    </Association>
    <Association>
      <rolTransactionId>0000001</rolTransactionId>
      <Associated>Y</Associated>
    </Association>
    <Association>
      <rolTransactionId>0000001</rolTransactionId>
      <Associated>Y</Associated>
    </Association>
  </responseData>
</SIGetAssociatedTranSelectionResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData":{
    "VisaCaseNumber": 1000000000
    "AssociatedTransGroupId": 00000,
  },
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "VisaCaseNumber": 1000000000,  
        "AssociatedTransGroupId": 00000,  
        "Association": [  
            {  
                "rolTransactionId": 0000001,  
                "Associated": "N"  
            },  
            {  
                "rolTransactionId": 0000002,  
                "Associated": "Y"  
            },  
            {  
                "rolTransactionId": 0000003,  
                "Associated": "Y"  
            }  
        ]  
    }  
}
```

## SIGetMMIRequest

The following XML element should be considered when submitting a TI request.

Element	Description
rolTransactionId	Uniquely identifies a transaction. Member can use this to access the TI transaction details.

## Request Message—SOAP-based Services

```
<SIGetMMIRequest>  
    <RequestHeader>...</RequestHeader>  
    <requestData>  
        <rolTransactionId>0000000</rolTransactionId>  
    </requestData>  
</SIGetMMIRequest>
```

## Response Message—SOAP-based Services

```
<SIGetMMIResponse>  
    <status>...</status>  
    <responseData>  
        <MMIDetails>  
            <MerchantAlternateName>name</MerchantAlternateName>  
            <MerchantStreetAddress>street</MerchantStreetAddress>  
            <MerchantCity>city</MerchantCity>  
            <MerchantStateProvince>CA</MerchantStateProvince>  
            <MerchantZipPostalCode>94070</MerchantZipPostalCode>  
            <MerchantPhone>1234567890</MerchantPhone>  
        </MMIDetails>  
    </responseData>  
</SIGetMMIResponse>
```

### Request Message—REST-based Services

```
{  
  "RequestHeader": "...",  
  "requestData": {  
    "R01TransactionId": "0000000"  
  }  
}
```

### Response Message—REST-based Services

```
{  
  "Status": "...",  
  "responseData": {  
    "MMIDetails": {  
      "MerchantAlternateName": "name",  
      "MerchantStreetAddress": "street",  
      "MerchantCity": "city",  
      "MerchantStateProvince": "CA",  
      "MerchantZipPostalCode": 94070,  
      "MerchantPhone": 1234567890  
    }  
  }  
}
```

## Inquiry Data Lifetime

A TI response is deleted if no action is taken on the TI results within 10 days of the request. If action is taken on the TI results, a case is created. If no further action is taken on the case, the newly created case is archived and purged after 90 days. Comments and images are not archived. Once TI results are purged, they are no longer available for viewing.

**Note:** *Transaction inquiry is a stand-alone activity in VROL; that is, a case is not created when the transaction inquiry results are returned. If the user takes action on the TI results, a case is created. If no action is taken, the TI results are deleted.*

## Request/Response Tracking

All transaction inquiry requests and responses are audited for reporting purposes. Both users who log on to the system and those who actually initiate the action are audited.

# Case Creation

5

This chapter describes the RTSI operations required to create a case from a transaction inquiry or manually, get case details, and add image to cases.

## Case Creation Process

The following steps describe the case creation process and XML operations used by both issuers and acquirers to perform the task. Issuers open most cases, with the exception of adjustments. The acquirer is expected to make every attempt to perform related transactions (responses, and so on) in the same case as the original dispute action.

### Create, Update Case

1. To create a case from a TI, user sends request SICreateCaseFromTransactionRequest using the R0lTransactionId.

**Note:** *This is the recommended operation.*

VROL returns response SICreateCaseFromTransactionResponse with the VROL Case Number. If no transactions are found, VROL sends message: E-3003000051 (No transactions found). User includes member case number if configured to require this field.

OR

To create a case manually, user sends request SISubmitManualCaseCreationRequest with case and transaction data. The stage of the new manual case (with or without an image and always without a questionnaire) is "Correspondence." User includes member case number if configured to require this field.

VROL returns response with the VROL Case Number of the new case.

#### Important:

Once transaction information is established in a case, there is no need to provide it in subsequent actions. In addition, <TransactionInfo> is no longer required if a case is created from a TI result.

2. To update the case, user sends request SISubmitManualCaseCreationRequest with case and transaction data.

VROL sends response with success status.

3. To request details of a specific VROL case, including transaction information, dispute jurisdiction, case type, and other case-related fields, RTSI user sends request **SIGetCaseDetailsRequest** using the VROL Case Number and Member Role.  
VROL sends response with success status.
4. To add images to the case, user sends request **SIUploadImageRequest** with the image information using the VROL Case Number. The image is sent as encoded data with the request. See [Image Handling in Chapter 3, RTSI Request Operations](#), for more information.  
VROL sends response with success status.

**Important:**

Images added to a case, rather than to a questionnaire for example, cannot be attached later to the questionnaire using any SI request. So, for an SI user, this document will only be visible to their side of the dispute.

5. To update the case resolution status of a case, user sends request **SIUpdateCaseResolutionStatusRequest** with VROL Case Number and case resolution status.  
VROL sends response with success status.

## Create Case Operations

RTSI exposes the following operations to create a case.

**SICreateCaseFromTransactionRequest**—Enables a user to create a case from a TI request.

**SISubmitManualCaseCreationRequest**—Enables a user to create a case without any transactions.

### SICreateCaseFromTransactionRequest

The SICreateCaseFromTransactionRequest operation enables a user to create a case from a TI request. (The SISubmitTranInquiryRequest response returns the RolTransactionId that is needed for the SICreateCaseFromTransactionRequest). See [Chapter 4, Transaction Inquiries](#)). This is the recommended operation for creating a case.

**Important:**

Once transaction information is established in a case, there is no need to provide it in subsequent actions. In addition, <TransactionInfo> is no longer required if a case is created from a TI result.

SICreateCaseFromTransactionRequest is used in request and response messages as shown in the following sample messages.

**Note:** *If any Purchase Inquiry (PI) has been created for the TI prior to case creation, the last PI associated with the TI will be added to the case with SICreateCaseFromTransactionRequest.*

### Request Message—SOAP-based Services

```
<SICreateCaseFromTransactionRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <rolTransactionId>0000000</rolTransactionId>
  </requestData>
</SICreateCaseFromTransactionRequest>
```

### Response Message—SOAP-based Services

```
<SICreateCaseFromTransactionResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
  </responseData>
</SICreateCaseFromTransactionResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData": {
    "rolTransactionId": "0000000"
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": "...",
  "responseData": {
    "VisaCaseNumber": "0000000000"
  }
}
```

## SISubmitManualCaseCreationRequest

The SISubmitManualCaseCreationRequest operation is used in request and response messages as shown in the following sample messages.

**Note:** Using *SISubmitManualCaseCreationRequest* requires entering a large number of transaction-related fields that cannot all be entered in a later financial. Using *SICreateCaseFromTransactionRequest* is recommended to ensure all required fields are entered and for ease of use.

The following table provides information about one of the XML elements that should be considered when creating a case or retrieving a response. Refer to the table in [Chapter 3, RTSI Request Operations](#) for additional XML elements to consider.

Element	Description
VisaCaseNumber	<p>Case tracking number assigned by VROL to uniquely identify a case in VROL.</p> <p><b>Note:</b> In the schema, the VROL Case Number is &lt;VisaCaseNumber&gt;.</p>

### Request Message—SOAP-based Services

```

<SISubmitManualCaseCreationRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <TransactionInfo>
            <AccountNumber>xxxxxxxxxxxxxxxx</AccountNumber>
            <Transaction id="xxxxxxxxxxxxxx">
                <TranDate>2014-01-01</TranDate>
                <Amount currency="840">118.53</Amount>
            </Transaction>
            <MerchantName>name</MerchantName>
            <MerchantCity>city</MerchantCity>
            <MerchantCountrySubEntityCode>CA
            </MerchantCountrySubEntityCode>
            <MerchantCountryCode>US</MerchantCountryCode>
            <MerchantPostalCode>94404</MerchantPostalCode>
            <MCC>3000</MCC>
            <ARN>xxxxxxxxxxxxxxxxxxxxxxxx</ARN>
            <ReimbursementAttribute>A</ReimbursementAttribute>
            <CPD>2014-01-01</CPD>
            <TransactionCategory>D</TransactionCategory>
            <NetworkID>0002</NetworkID>
        </TransactionInfo>
        <CaseInfo>
            <CaseType>DSPT</CaseType>
            <FraudClassification>F</FraudClassification>
            <MemberCaseNumber>test</MemberCaseNumber>
        </CaseInfo>
        <CardholderInfo>
            ...
        </CardholderInfo>
        <AccountOwnerInfo>
            ...
        </AccountOwnerInfo>
        <Note>note</Note>
        <PrivilegeInd>true</PrivilegeInd>
    </requestData>
</SISubmitManualCaseCreationRequest>

```

### Response Message—SOAP-based Services

```

<SISubmitManualCaseCreationResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
    </responseData>
</SISubmitManualCaseCreationResponse>

```

### Request Message—REST-based Services

```
{  
    "RequestHeader": "...",  
    "requestData": {  
        "TransactionInfo": "...",  
        "CaseInfo": {  
            "CaseType": "AJMT",  
            "FraudClassification": "N",  
            ...  
        },  
        "CardholderInfo": "...",  
        "AccountOwnerInfo": "...",  
        "Note": "note",  
        "PrivilegeInd": true,  
    },  
}
```

### Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "VisaCaseNumber": "0000000000"  
    }  
}
```

## Update Existing Case

The SISubmitManualCaseCreationRequest operation is also used to update an existing case. The request must include the Visa Case Number and the information to be changed as shown in the following request message. The user can update any information that is not locked.

**Request Message—SOAP-based Services**

```
<SISubmitManualCaseCreationRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <TransactionInfo>
      ...
    </TransactionInfo>
    <CaseInfo>
      ...
    </CaseInfo>
    <CardholderInfo>
      <CardholderAddress>Second address</CardholderAddress>
      <CardholderCity>Second city</CardholderCity>
      ...
    </CardholderInfo>
    <Note>note</Note>
    <PrivilegeInd>true</PrivilegeInd>
  </requestData>
</SISubmitManualCaseCreationRequest>
```

**Response Message—SOAP-based Services**

```
<SISubmitManualCaseCreationResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
  </responseData>
</SISubmitManualCaseCreationResponse>
```

**Request Message—REST-based Services**

```
{
  "RequestHeader": "...",
  "requestData": {
    "VisaCaseNumber": "0000000000",
    "TransactionInfo": "...",
    "CaseInfo": "...",
    "CardholderInfo": {
      "CardholderName": "a",
      "CardholderAddress": "a",
      ...
    },
    "Note": "a",
    "PrivilegeInd": true,
  },
}
```

### Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "VisaCaseNumber": "0000000000"  
    }  
}
```

## Get Case Details

The SIGetCaseDetailsRequest operation enables a user to request details of a specific VROL case, including transaction information, dispute jurisdiction, case type, and other case-related fields, when supplying the VisaCaseNumber and MemberRole. The information returned in the response depends on the role of the user making the request, and what stage and state the case is currently in.

This operation contains request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SIGetCaseDetailsRequest>  
    <RequestHeader>...</RequestHeader>  
    <requestData>  
        <VisaCaseNumber>0000000000</VisaCaseNumber>  
    </requestData>  
</SIGetCaseDetailsRequest>
```

### Response Message—SOAP-based Services

The following table provides information about XML elements to consider.

Element	Description
CaselInformation	Group containing case-level data
TransactionInformation	Group containing transaction data, mostly fields from the original transaction
PurchaselInquiryEligibleInd	Indicates whether the selected transaction in a case is eligible or not for Purchase Inquiry (PI).  If <PurchaselInquiryEligibleInd>=true is returned in the response, this indicates that the selected transaction in a case is currently eligible for PI. The PI eligibility indicator may change if a merchant is activated or deactivated.

Refer to [Comparison of <TransactionInformation> vs. TI Details XML Fields](#) for more details.

```
<SIGetCaseDetailsResponse>  
    <Status>...</Status>  
    <responseData>  
        <CaseInformation>...</CaseInformation>  
        <TransactionInformation>...</TransactionInformation>  
    </responseData>  
</SIGetCaseDetailsResponse>
```

### Request Message—REST-based Services

```
{  
    "RequestHeader": "...",  
    "requestData": {  
        "VisaCaseNumber": "0000000000"  
    }  
}
```

### Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "CaseInformation": "...",  
        "TransactionInformation": "..."  
    }  
}
```

## Add Image to Case

### Important:

Images added to a case, rather than to a questionnaire for example, cannot be attached later to the questionnaire using any SI request. So, for an SI user, this document will only be visible to their side of the dispute.

Users can add one or multiple documents to a case from a local or network drive by scanning a physical document or by inputting text. An added document will only be visible to the member organization that owns the document. Only after the document is attached to a questionnaire and delivered to the opposite party will it become visible to the opposite party.

Images added at the master case level can be copied to the subordinate cases as duplicate images.

This permits different edits to be made in different subordinate cases, such that different pages can be removed in different subordinate cases. Once the document is added at the Master Case level after possible edits and pushed down to all subordinate cases, any subsequent edits at the master case level are not reflected in the subordinate cases, as the image will have already been copied. Similarly, further edits in the subordinate cases are not reflected in the master case or in other subordinate cases.

When an image is pushed down to all subordinate cases, it is not counted as an additional image for reporting purposes. For example, report 200 will count it as only 1 image even if further edits at the subordinate case level modify the contents.

Supporting documentation can be added and retrieved as needed using the following operations:

**SIUploadImageRequest**—Used to add an image to a case.

**SIGetImageRequest**—Used to retrieve an image from a case.

## SIUploadImageRequest

In order to attach an image to a newly created case, SIUploadImageRequest requires the VROL Case Number and the image data.

**Important:**

SIUploadImageRequest in schema allows multiple images to be uploaded at once but the API implementation has a limitation of uploading only one image at a time.

The following table provides information about some XML elements that should be considered when attaching an image to a case or retrieving a response.

Element	Description
MemberRole	If the submitting member's organization type is issuer and acquirer (dual organization), then this value is required. The value should contain 'I' when submitting member is acting as an issuer and 'A' when acting as an acquirer.  If submitting member is not a dual organization, then <MemberRole> is not required.
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.  <b>Note:</b> In the schema, the VROL Case Number is <VisaCaseNumber>.
DocType	A field to more conveniently name the image file that is being uploaded.

SIUploadImageRequest is used in request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

**Note:** <QnId> is used to add an image to a questionnaire; otherwise, adds an image to a case. The <QnId> uses the value of the <DisputeId>.

**Note:** <DisputeItemID> and <DisputeItemType> are used to add an image to a dispute questionnaire; otherwise, adds an image to a case.

```
<SIUploadImageRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <AttachmentDescriptor>
            <Attachment id="1">
                <Comment>String</Comment>
                <ImageData contentType="image/tiff">binary data
                </ImageData>
            </Attachment>
        </AttachmentDescriptor>
        <DocType>CALET</DocType>
    </requestData>
</SIUploadImageRequest>
```

### Response Message—SOAP-based Services

**Note:** <QnId> is returned if image is added to questionnaire.

**Note:** <DisputeItemID> and <DisputeItemType> are returned if an image is added to a dispute questionnaire.

```
<SIUploadImageResponse>
  <Status>...</Status>
  <ResponseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <DocId>000000</DocId>
  </ResponseData>
</SIUploadImageResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData": {
    "VisaCaseNumber": "0000000000",
    "AttachmentDescriptor": {
      "AttachType": "Mime",
      "Attachment": [
        {
          "ContentType": "image/tiff",
          "Comment": "a",
          "ImageData": "image1"
          "id": "123456",
        }
      ]
    },
    "DocType": "TEFOL",
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": "...",
  "responseData": {
    "VisaCaseNumber": "0000000000",
    "DocId": "123456"
  }
}
```

## SIGetImageRequest

In order to retrieve an image added to a newly created case, SIGetImageRequest requires the VROL Case Number and the Doc ID.

The following table provides information about some XML elements that should be considered when submitting a get image request or retrieving a response.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.  <b>Note:</b> In the schema, the VROL Case Number is <VisaCaseNumber>.
DocId	Uniquely identifies a document (image) added to a case. The ID can be used to access the image information in a subsequent request.

SIGetImageRequest is used in request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SIGetImageRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <DocId>000000</DocId>
  </requestData>
</SIGetImageRequest>
```

### Response Message—SOAP-based Services

**Note:** If image was added specifically at the case level and not at the questionnaire level, the Qnld will not be returned in the response.

```
<SIGetImageResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <DocId>00000000</DocId>
    <AttachmentDescriptor>
      <Attachment id="1">
        <ContentType>image/tiff</ContentType>
        <Comment>String</Comment>
        <ImageData>binary data</ImageData>
      </Attachment>
    </AttachmentDescriptor>
    <DocType>CALET</DocType>
  </responseData>
</SIGetImageResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData": {
    "VisaCaseNumber": "0000000000",
    "DocId": "123456"
  }
}
```

## Response Message—REST-based Services

```
{
  "Status": "...",
  "responseData": {
    "VisaCaseNumber": "0000000000",
    "DocId": "123456",
    "AttachmentDescriptor": {
      "Attachment": [
        {
          "ContentType": "image/tiff",
          "ImageData": "api/downloadImage/111111111111/
123456?memberRole=I&user=test",
          "id": 123456
        }
      ]
    },
    "DocType": "TEFOL"
  }
}
```

## Update Case Resolution Status

### SIUpdateCaseResolutionStatusRequest

The SIUpdateCaseResolutionStatusRequest operation is used to update the case resolution status of a case. This may close the case, depending on the status change made. The request must include the VROL Case Number and case resolution status.

The following table provides information about some XML elements that should be considered when submitting an update case resolution status request or retrieving a response.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.  <b>Note:</b> In the schema, the VROL Case Number is <VisaCaseNumber>.
CaseResolutionStatus	Used to indicate what has happened in the dispute.  Domain values: INPROC, UNASGN, PNDCLS, RESLVD, DENIED, CANCEL, and PENDED.

### Request Message—SOAP-based Services

```
<SIUpdateCaseResolutionStatusRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <CaseResolutionStatus>RESLVD</CaseResolutionStatus>
  </requestData>
</SIUpdateCaseResolutionStatusRequest>
```

### Response Message—SOAP-based Services

```
<SIUpdateCaseResolutionStatusResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <CaseResolutionStatus>RESLVD</CaseResolutionStatus>
  </responseData>
</SIUpdateCaseResolutionStatusResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData":{
    "VisaCaseNumber": "0000000000",
    "CaseResolutionStatus": "RESLVD"
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": "...",
  "responseData":{
    "VisaCaseNumber": "0000000000",
    "CaseResolutionStatus": "RESLVD"
  }
}
```

THIS PAGE INTENTIONALLY LEFT BLANK.

# Request for Copies

6

A request for copy (RFC) is used by an issuer to request a copy of the merchant's sales draft from an acquirer. Issuers can create and submit an RFC from their member host system (MHS) or VROL.

**Note:** *RFCs cannot be created for ATM transactions, or for PIN-Authenticated Visa Debit transactions, or if the transaction is on a network other than Visa.*

**Note:** *For issuers, a PIN-Authenticated Visa Debit transaction has a Reimbursement Attribute of Y, Z, or 8 on network 0002. For acquirers, a PIN-Authenticated Visa Debit transaction has a Reimbursement Attribute of 8 on network 0002. These transactions are available only for US domestic.*

The following functions are available through RTSI.

- Issuers may request copies of sales drafts from acquirers through an RFC.
- Acquirers and issuers may obtain RFC advices. Advices are received for RFCs on Visa network and may also be received for Interlink and PLUS depending on the member's BIN settings.
- Acquirers may submit fulfillments and nonfulfillments.
- Issuers may receive acquirer fulfillments and nonfulfillments that have been processed and received back from RTSI.

**Note:** *RFC fulfillments include images. For more information, see [Image Handling in Chapter 3, RTSI Request Operations](#)*

Members must be set up to process RFCs as follows:

- If an issuer is sending RFCs or an acquirer is receiving RFCs in VROL (in addition to RFC advices), the member's financial BINs must be set up in CORE and Config, and the VROL organization setting must be enabled.
- (Visa network only) The member's financial BINs must be set up in Config to receive RFC advices.
- (Optional) An auto-fulfillment option is available to acquirers in the VROL organization settings. This option allows VROL to automatically generate nonfulfillment messages when the RFC fulfillment time frame limits are about to expire.

**Important:**

VisaNet Technical Specifications and Interchange Format manuals should be consulted when coding for dispute RFCs to determine which fields are required for transactions processed by other Visa systems. For instance, AcquirerBID is required for other VisaNet systems but not in VROL. For many fields, VROL defaults the field from TI, if the VROL case contains a TI. Refer to the IES to identify fields supported by VROL.

VROL provides an error message for RFCs submitted for a U.S. domestic card transaction with a CPD older than 12 months, except for Reason code 27, which does not have a time frame. For other transactions and jurisdictions, no message is provided. Refer to the Interface Element Spreadsheet (IES) RFC Reason Codes tab for a description of the various reason codes.

VROL identifies potential duplicate RFCs against the same original and provides a warning. For all transactions, the RFC will be considered a duplicate if another RFC (that is not rejected or returned) exists with the same reason code.

## RFC Process

Each RTSI copy request and fulfillment process consists of a series of request/response transmissions managed through VROL. This section describes the process and XML operations used by issuers and acquirers to perform the task.

### Issuer Steps

#### Create RFC

1. Prior to sending an RFC, issuer may perform a TI search and create a case from the TI. See [Chapter 5, Case Creation](#), for the information to create a case from a TI and retrieve the associated VROL Case Number.
2. To create an RFC, issuer sends request SISubmitRFCRequest with RFC data. User must provide the cardholder's amount and currency in the request. This information is also found in TI results (see [Chapter 4, Transaction Inquiries](#)). To update an existing case, the VROL Case Number is required.  
VROL sends a response with VROL Case Number, RFC ID, and success status.
3. Once the RFC is processed by BASE II or SMS and EM (CPS transaction), the user can determine the success as follows:
  - Send request SIGetRFCAdviceRequest using the VROL Case Number and RFC ID returned previously.  
VROL sends response with details of the RFC advice. If no advice is found, VROL returns: "E-300400006 RFC with given identifier is not found."
  - If SMS, check the approval status as described in [Retrieve RFC Network Approval Status](#).
  - Check for the rejected RFC as described in [Check for RFC Rejects](#).

## Retrieve RFC Network Approval Status

### (SMS Only)

Network approval is not returned in the response when sending an RFC from VROL. The issuer can check the reject queue, but may prefer confirmation that an RFC was processed successfully by V.I.P. (and EM when applicable for CPS).

SMS issuers that send RFCs from VROL can determine if the transactions sent have been approved or rejected by V.I.P. by sending the detail request each day until they receive a status in the response that the approval or rejection has occurred. This process typically takes 1-2 days. If the transaction is still waiting for a response from V.I.P., it will be indicated in the response.

1. To retrieve RFC network approval status, issuer sends request SIGetRFCDetailsRequest using VROL case number, RFC ID, and QN type. The <ReturnDetails> field is used to indicate that the user wants to receive the network approval status message.
2. VROL sends response SIGetRFCDetailsResponse including the network approval status.

## Check for RFC Rejects

1. If applicable, check for RFCs rejected by BASE II or SMS by sending one of the following:
  - RFC details request SIGetRFCDetailsRequest. This returns RFCRejectsReturn if the RFC is rejected.
  - Queue request SIGetQueueRequest for MY\_REJECTS or ALL\_REJECTS queue. VROL sends response.
2. To ignore RFC rejects when using the reject queue, send request SIgnoreRejectRequest using VROL Case Number, Reject ID, and RejectedTransactionType=Request For Copy. This action prevents the transaction from being returned again in the next queue request.  
VROL sends response with success status.
3. To repair an RFC reject, user sends request SISubmitRFCRequest for the reject to be repaired along with RFC data.
  - If the case contains a single unrepairsd reject of the same type as the request, VROL marks the reject as repaired.
  - If the case contains multiple unrepairsd rejects of the same type as the request, VROL marks the most recent reject as repaired.
  - If the case does not contain a reject of the same type as the request, VROL does not mark any transaction as repaired.  
VROL adds the incoming transaction to the case and sends response with success status.

## Retrieve RFC Responses

1. To retrieve RFC responses, there are a number of alternatives including the following:
  - a. From MY\_RFC\_RESPONSES or ALL\_RFC\_RESPONSES queue, issuer sends request SIGetQueueRequest. See [Chapter 17, Queues](#), for more information.

VROL returns a summary of the RFC responses in the queue including their VROL Case Numbers.

- b. From INCOMING\_BQ\_RFC\_RESPONSES queue, issuer sends batch request SIGetBatchQueueRequest. See [Chapter 18, RTSI Batch Queues](#), for more information, including how to remove the responses from the queue once they are read.
- c. Some members opt to receive responses using Bulk SI Download files.
2. To retrieve RFC ID, issuer sends a Hypersearch request. See [Chapter 19, Hypersearch Requests](#) for more information.
3. To retrieve details of RFC responses:
  - For each fulfillment, issuer sends RFC fulfillment request SIGetRFCFulfillmentRequest.
    - For each image id returned in the response, the issuer sends request SIGetImageRequest using the <Attachment> id attribute in the <DocId> field.

**Important:**

Multiple images may be sent in an RFC fulfillment response.

- For each nonfulfillment, issuer sends RFC nonfulfillment request SIGetRFCNonFulfillmentRequest.

VROL sends response with details of RFC response.

4. To close the RFC and remove it from the queue, issuer sends close RFC request SICloseRFCRequest or SICloseTransactionRequest with CloseItem Type as RFC. This step prevents the transaction from being returned again in the next queue request (see [Chapter 17, Queues](#)).

**Note:** *SICloseTransactionRequest closes the transaction and removes it from its respective queue but does not close the case. For more information, see [SICloseTransactionRequest](#).*

VROL sends response with success status.

## Acquirer Steps

### Retrieve RFC Advices and RFCs, Create RFC Responses

1. To retrieve the RFC advices from (for example) MY\_INCOMING\_RFC\_REQUESTS or ALL\_INCOMING\_RFC\_REQUESTS queue, acquirer sends request SIGetQueueRequest. See [Chapter 17, Queues](#), for more information.

VROL returns a summary of the RFC advices in the queue including their VROL Case Numbers.

2. To retrieve RFC ID, acquirer sends a Hypersearch request. See [Chapter 19, Hypersearch Requests](#) for more information.
3. For each RFC advice in queue, acquirer sends request SIGetRFCAdviceRequest using VROL Case Number and RFC ID returned previously.

VROL sends response with details of the RFC advice.

4. If acquirer receives RFC (equivalent to TC 52 or 0600) in VROL, acquirer sends request SIGetRFCDetailsRequest to retrieve RFC details.

VROL sends response with details of RFC.

5. To create RFC responses, acquirer sends request SISubmitRFCFulfillmentRequest or SISubmitRFCNonFulfillmentRequest with RFC response information. For the fulfillment, the image is sent as encoded data with the request.

VROL sends response with success status.

6. To close the RFC advice and remove it from the acquirer queue, acquirer sends close RFC request SICloseRFCRequest or SICloseTransactionRequest with CloseItem Type as RFC. This step prevents the transaction from being returned again in the next queue request. The VROL Case Number and RFC ID are sent with the request (see [Chapter 17, Queues](#)).

**Note:** *SICloseTransactionRequest closes the transaction and removes it from its respective queue but does not close the case. For more information, see [SICloseTransactionRequest](#).*

VROL sends response with success status.

## RFC Operations

RTSI exposes the following copy request and fulfillment operations:

**SISubmitRFCRequest**—Used by issuer to request a copy of the merchant's sales draft from the acquirer.

**SIGetRFCDetailsRequest**—Used by issuer to request detailed information regarding an RFC transaction, including transaction header information such as merchant and cardholder data, when supplying Visa Case Number and RFCID. When the RFC is successful, this request returns SIGetRFCDetailsResponse. When the RFC is rejected, it returns an RFCRejectsReturn.

**SIGetRFCAdviceRequest**—Used by acquirer or issuer to obtain an RFC advice stored in VROL.

**SISubmitRFCFulfillmentRequest**—Used by acquirer to respond to an RFC request with a fulfillment.

**SISubmitRFCNonFulfillmentRequest**—Used by acquirer to respond to an RFC request with a nonfulfillment.

**SIGetRFCFulfillmentRequest**—Used by issuer to obtain an RFC fulfillment from the acquirer stored in VROL.

**SIGetRFCNonFulfillmentRequest**—Used by issuer to obtain an RFC nonfulfillment from the acquirer stored in VROL.

**SICloseRFCRequest**—Used by issuer or acquirer to close request items in a queue (see [Chapter 17, Queues](#)).

**SIIgnoreRejectRequest**—Used by issuer to ignore a rejected RFC. See [Chapter 17, Queues](#), for details.

**SIGetQueueRequest**—Used by issuer to access inbound RFC responses, or by acquirer to access inbound RFC requests. See [Chapter 17, Queues](#) and [Chapter 18, RTSI Batch Queues](#).

**SIGetBatchQueueRequest**—Used by issuer to access inbound RFC responses, or by acquirer to access inbound RFC requests. See [Chapter 17, Queues](#) and [Chapter 18, RTSI Batch Queues](#).

## SISubmitRFCRequest

Issuers use SISubmitRFCRequest to create an original request for sales draft to VROL, which is forwarded to the acquirer.

SISubmitRFCRequest is used in request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SISubmitRFCRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <CaseMatchingInfo>
      <VisaCaseNumber>0000000000</VisaCaseNumber>
    </CaseMatchingInfo>
    <RequestForCopy ProcessingPlatform="BASE2">
      <RFCReasonCode>28</RFCReasonCode>
      <QueueForFulfillment>RtqMyQ</QueueForFulfillment>
    </RequestForCopy>
  </requestData>
</SISubmitRFCRequest>
```

### Response Message—SOAP-based Services

The actual elements returned in a message depend on the RFC data available in VROL. For a full set of possible fields, refer to the XML schema.

```
<SISubmitRFCResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <RFCID>0000000</RFCID>
  </responseData>
</SISubmitRFCResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData": {
    "CaseMatchingInfo": {
      "VisaCaseNumber": "0000000000"
    },
    "RequestForCopy": {
      "RFCReasonCode": 28,
      "QueueForFulfillment": "RtqMyQ"
    }
  }
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "VisaCaseNumber": "0000000000",  
        "RFCID": "0000000"  
    }  
}
```

## SIGetRFCDetailsRequest

SIGetRFCDetailsRequest is used by issuers to determine the success status of the RFC they send. Acquirers can request RFC details if their BINs are set up in CORE, Config, and VROL. SIGetRFCDetailsRequest requires the VROL Case Number, RFC identifier, and RFCID, and returns a possible response from VROL.

The following table provides information about some XML elements that should be considered when submitting an RFC detail request or retrieving a response.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL. <b>Note:</b> In the schema, the VROL Case Number is <VisaCaseNumber>.
RFCID	VROL-generated identifier that uniquely associates an RFCAdvice with a VROL case. If the RFCAdvice is matched to more than one VROL case, each case will carry a unique RFCID. The RFCAdvice itself is identified through the RetrievalRequestID.
ReturnDetails <b>Note:</b> SMS only.	The domain value of <ReturnDetails> indicates if the user wants to receive the SMS approval status message as follows: <ul style="list-style-type: none"><li>• (not supplied) —Existing details only (no SMS status information).</li><li>• FULL —Existing details and SMS status information.</li><li>• STATUS_ONLY —SMS status only (no financial details).</li></ul>

## Request Message—SOAP-based Services

```
<SIGetRFCDetailsRequest>  
    <RequestHeader>...</RequestHeader>  
    <requestData>  
        <RFCID>0000000</RFCID>  
        <VisaCaseNumber>0000000000</VisaCaseNumber>  
    </requestData>  
</SIGetRFCDetailsRequest>
```

## Response Message—SOAP-based Services

```

<SIGetRFCDetailsResponse>
    <Status>...</Status>
    <responseData>
        <RFCDetailResponse>
            <AccountNumber>xxxxxxxxxxxxxxxx</AccountNumber>
            <Transaction id="xxxxxxxxxxxxxx">
                <TranDate>2013-05-09</TranDate>
                <Amount currency="840">129.00</Amount>
            </Transaction>
            <MerchantName>Any merchant</MerchantName>
            <MerchantCity>Any city</MerchantCity>
            <MerchantCountrySubEntityCode>OK
            </MerchantCountrySubEntityCode>
            <MerchantCountryCode>US</MerchantCountryCode>
            <MerchantPostalCode>53207</MerchantPostalCode>
            <MCC>3132</MCC>
            <ARN>xxxxxxxxxxxxxxxxxxxxxx</ARN>
            <RFCReasonCode>28</RFCReasonCode>
            ...
            <QueueForFulfillment>RtqMyQ</QueueForFulfillment>
            <ReimbursementAttribute>A</ReimbursementAttribute>
            ...
            <VisaCaseNumber>0000000000</VisaCaseNumber>
            <RFCID>0000000</RFCID>
        </RFCDetailResponse>
    </responseData>
</SIGetRFCDetailsResponse>

```

Following is the response if rejected.

```

<SIGetRFCDetailsResponse>
    <Status>...</Status>
    <responseData>
        <RFCRejectReturn>
        ...
        </RFCRejectReturn>
    </responseData>
</SIGetRFCDetailsResponse>

```

## Request Message—REST-based Services

```

{
    "RequestHeader": "...",
    "requestData": {
        "RFCID": "0000000",
        "VisaCaseNumber": "0000000000"
    }
}

```

## Response Message—REST-based Services

```
{  
    "responseData": {  
        "RFCDetailResponse": {  
            "AccountNumber": "111111111111",  
            "Token": "111111111111",  
            "TokenAssuranceLevel": "a",  
            "Transaction": {  
                "id": "a",  
                "TranDate": "2016-01-01",  
                "Amount": {  
                    "currency": "111",  
                    "Text": 1  
                },  
                ...  
            },  
            "MerchantName": "Any merchant",  
            "MerchantCity": "Any city",  
            "MerchantCountrySubEntityCode": "OK",  
            "MerchantCountryCode": "US",  
            "MerchantPostalCode": 53207,  
            "MCC": 3132,  
            "ARN": "111111111111111111111111",  
            "RFCReasonCode": 28,  
            "QueueForFulfillment": "RtqMyQ",  
            "ReimbursementAttribute": "A",  
            ...  
            "VisaCaseNumber": "0000000000",  
            "RFCID": "0000000"  
        },  
        ...  
    }  
}
```

Following is the response if rejected.

```
{  
    "responseData": {  
        "RFCRejectReturn": "..."  
    }  
}
```

## Network Approval Status Messages

### SMS Only

For SMS members only, the value for ReturnDetails in the SIGetRFCDetailsRequest (FULL or STATUS\_ONLY) determines the request and response messages as shown in the following examples. One of two messages appear in the StatusMessage of a successful response: Acknowledged or Pending Network Response.

**Note:** *ReturnDetails is not applicable for SIGetRFCNonFulfillment function, though it is available in the schema due to commonality with Get RFC details and Get RFC advice functions.*

If a response has not yet been received by VROL from the network at the time the RTSI user sends the request, an error is returned (E-140000010). If the RTSI user receives Pending Network Response or this error, they should repeat the request later.

### Request Message—SOAP-based Services

```
<SIGetRFCDetailsRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <RFCID>0000000</RFCID>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <ReturnDetails>...</ReturnDetails>
    </requestData>
</SIGetRFCDetailsRequest>
```

### Response Message—ReturnDetails=FULL—SOAP-based Services

Following is response if successful or no advice returned as yet from VisaNet.

```
<SIGetRFCDetailsResponse>
    <Status>...</Status>
    <responseData>
        <RFCDetailResponse>
            <AccountNumber>xxxxxxxxxxxxxxxxxxxx</AccountNumber>
            <Transaction id="xxxxxxxxxxxxxx">
                <TranDate>2013-05-09</TranDate>
                <Amount currency="840">129.00</Amount>
            </Transaction>
            <MerchantName>Any merchant</MerchantName>
            <MerchantCity>Any city</MerchantCity>
            <MerchantCountrySubEntityCode>OK
            </MerchantCountrySubEntityCode>
            <MerchantCountryCode>US</MerchantCountryCode>
            <MerchantPostalCode>53207</MerchantPostalCode>
            <MCC>5621</MCC>
            <ARN>xxxxxxxxxxxxxxxxxxxxxxxxx</ARN>
            <RFCReasonCode>28</RFCReasonCode>
            ...
            <QueueForFulfillment>RtqMyQ</QueueForFulfillment>
            <ReimbursementAttribute>A</ReimbursementAttribute>
            ...
            <SwitchingSystemStatus>
                <StatusMessage>Acknowledged</StatusMessage>
            </SwitchingSystemStatus>
            <VisaCaseNumber>0000000000</VisaCaseNumber>
            <RFCID>0000000</RFCID>
        </RFCDetailResponse>
    </responseData>
</SIGetRFCDetailsResponse>
```

Following is the response if request is rejected. Note that <StatusCode> and <StatusMessage> repeat up to six times if multiple rejects/returns.

```
<SIGetRFCDetailsResponse>
  <Status>...</Status>
  <responseData>
    <RFCDetailResponse>
      <AccountNumber>xxxxxxxxxxxxxxxx</AccountNumber>
      <Transaction id="xxxxxxxxxxxxxx">
        <TranDate>2013-11-04</TranDate>
        <Amount currency="554">109.90</Amount>
      </Transaction>
      <MerchantName>Any merchant</MerchantName>
      <MerchantCity>Any city</MerchantCity>
      <MerchantCountryCode>NZ</MerchantCountryCode>
      <MerchantPostalCode>53207</MerchantPostalCode>
      <MCC>4814</MCC>
      <ARN>xxxxxxxxxxxxxxxxxxxxxx</ARN>
      <RFCReasonCode>34</RFCReasonCode>
      ...
      <QueueForFulfillment>RtqMyQ</QueueForFulfillment>
      <ReimbursementAttribute>0</ReimbursementAttribute>
      ...
      <SwitchingSystemStatus>
        <StatusCode>EANR</StatusCode>
        <StatusMessage>Expired Awaiting Network Response
        </StatusMessage>
      </SwitchingSystemStatus>
      <VisaCaseNumber>0000000000</VisaCaseNumber>
      <RFCID>0000000</RFCID>
    </RFCDetailResponse>
  </responseData>
</SIGetRFCDetailsResponse>
```

### Response Message—ReturnDetails=STATUS\_ONLY—SOAP-based Services

#### Important:

The successful and reject response differ only in the value of the StatusCode and description. Care must be taken to ensure the reject response is corrected and resent.

Following is response if request is successful or no advice returned as yet from VisaNet.

```
<SIGetRFCDetailsResponse>
  <Status>...</Status>
  <responseData>
    <RFCswitchStatusResponse>
      <VisaCaseNumber>0000000000</VisaCaseNumber>
      <RFCID>0000000</RFCID>
      <SwitchingSystemStatus>
        <StatusMessage>Acknowledged</StatusMessage>
      </SwitchingSystemStatus>
    </RFCswitchStatusResponse>
  </responseData>
</SIGetRFCDetailsResponse>
```

Following is the response if request is rejected. Note that <StatusCode> and <StatusMessage> repeat up to six times if multiple rejects/returns.

```
<SIGetRFCDetailsResponse>
  <Status>...</Status>
  <responseData>
    <RFCSwitchStatusResponse>
      <VisaCaseNumber>0000000000</VisaCaseNumber>
      <RFCID>0000000</RFCID>
      <SwitchingSystemStatus>
        <StatusCode>EANR</StatusCode>
        <StatusMessage>Expired Awaiting Network Response
        </StatusMessage>
      </SwitchingSystemStatus>
    </RFCSwitchStatusResponse>
  </responseData>
</SIGetRFCDetailsResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData": {
    "RFCID": "0000000",
    "VisaCaseNumber": "0000000000",
    "ReturnDetails": "..."
  }
}
```

### Response Message—ReturnDetails=FULL—REST-based Services

```
{  
    "responseData": {  
        "RFCDetailResponse": {  
            "AccountNumber": "111111111111",  
            "Token": "111111111111",  
            "TokenAssuranceLevel": "a",  
            "Transaction": {  
                "id": "a",  
                "TranDate": "2016-01-01",  
                "Amount": {  
                    "currency": "111",  
                    "Text": 1  
                },  
                },  
            "MerchantName": "Any merchant",  
            "MerchantCity": "Any city",  
            "MerchantCountrySubEntityCode": "OK",  
            "MerchantCountryCode": "US",  
            "MerchantPostalCode": 53207,  
            "MCC": 5621,  
            "ARN": "111111111111111111111111",  
            "RFCReasonCode": 28,  
            "QueueForFulfillment": "RtqMyQ",  
            "ReimbursementAttribute": "A",  
            "SwitchingSystemStatus": {  
                "StatusMessage": "Acknowledged"  
            },  
            "VisaCaseNumber": "0000000000",  
            "RFCID": "00000000"  
        }  
    }  
}
```

### Response Message—ReturnDetails=STATUS\_ONLY—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "RFCSwitchStatusResponse": {  
            "VisaCaseNumber": "0000000000",  
            "RFCID": "00000000",  
            "SwitchingSystemStatus": {  
                "StatusCode": "RC14",  
                "StatusMessage": "Invalid account number..."  
            },  
        }  
    }  
}
```

## SIGetRFCAdviceRequest

Acquirers and issuers use SIGetRFCAdviceRequest to obtain an RFC advice stored in VROL. This operation requires an RFC identifier (RFCID) and case number as parameters and returns the RFC advice. The RFCID can be obtained using the SIGetQueue operation described in [Chapter 17, Queues](#).

### Request Message—SOAP-based Services

```
<SIGetRFCAdviceRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <RFCID>0000000</RFCID>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
    </requestData>
</SIGetRFCAdviceRequest>
```

### Response Message—SOAP-based Services

The following example shows a response with the RFC advice data. The actual elements returned in a message depend on the RFC data available in VROL.

```
<SIGetRFCAdviceResponse>
    <Status>...</Status>
    <responseData>
        <AccountNumber>xxxxxxxxxxxxxxxxxxxx</AccountNumber>
        <Transaction id="xxxxxxxxxxxxxx">
            <TranDate>2013-05-09</TranDate>
            <Amount currency="840">129.00</Amount>
        </Transaction>
        <MerchantName>Any Merchant</MerchantName>
        <MerchantCity>Any City</MerchantCity>
        <MerchantCountrySubEntityCode>OK
        </MerchantCountrySubEntityCode>
        <MerchantCountryCode>US</MerchantCountryCode>
        <MerchantPostalCode>53207</MerchantPostalCode>
        <MCC>3132</MCC>
        <ARN>xxxxxxxxxxxxxxxxxxxxxxxx</ARN>
        <RFCReasonCode>28</RFCReasonCode>
        <IssuerControlNumber>0000000</IssuerControlNumber>
        ...
        <ReimbursementAttribute>0</ReimbursementAttribute>
        <SettlementFlag>0</SettlementFlag>
        <CRSPprocessingCode>Y</CRSPprocessingCode>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
    </responseData>
</SIGetRFCAdviceResponse>
```

## Request Message—REST-based Services

```
{  
    "RequestHeader": "...",  
    "requestData": {  
        "RFCID": "0000000",  
        "VisaCaseNumber": "0000000000"  
  
    }  
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "AccountNumber": "111111111111",  
        "Token": "111111111111",  
        "Transaction": {  
            "id": "a",  
            "TranDate": "2016-01-01",  
            "Amount": {  
                "currency": "111",  
                "Text": 1  
            },  
            ...  
        },  
        "MerchantName": "a",  
        "MerchantCity": "a",  
        "MerchantCountrySubEntityCode": "a",  
        "MerchantCountryCode": "AA",  
        "MerchantPostalCode": "a",  
        "MCC": "1111",  
        "ARN": "111111111111111111111111",  
        "RFCReasonCode": "1143",  
        "IssuerControlNumber": "0000000",  
        ...  
        "ReimbursementAttribute": 1,  
        "SettlementFlag": "a",  
        "CRSPprocessingCode": "a",  
        "VisaCaseNumber": 1111111111  
    }  
}
```

## SISubmitRFCFulfillmentRequest

Acquirers use SISubmitRFCFulfillmentRequest to submit an RFC fulfillment to VROL. SISubmitRFCFulfillmentRequest accepts parameters identifying the RFC and associated images. It returns the case number and RFC identifier, RFCID.

The following table provides information about some XML elements that should be considered when submitting an RFC fulfillment request or retrieving a response.

Element	Description
VisaCaseNumber, RFCID, Attachment Descriptor	All elements are required in the request. Only VisaCaseNumber and RFCID are provided in the response.
RFCID	VROL does not support the receipt of subsequent fulfillments for the same RFCID. An error message is logged.
RetrievalRequestID	The value is obtained from the RFC advice.
AttachType	This element is required in the request and must be set to "Mime."
Attachment	A separate Attachment element is required in the request for each image file sent. The ID attribute specifies the Content-Id of the attached image file.
ContentType	This element is required in the request for each attached image file. The accepted value is "image/tiff", "image/jpeg", or "application/pdf".

The SISubmitRFCFulfillmentRequest operation is used in request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

For a description of the complete MIME multipart message structure, see [Image Handling in Chapter 3, RTSI Request Operations](#).

```
<SISubmitRFCFulfillmentRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <RetrievalRequestID>000000000000</RetrievalRequestID>
        <RFCID>000000</RFCID>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <AttachmentDescriptor>
            <Attachment id="1">
                <Comment>String</Comment>
                <ImageData contentType="image/tiff">binary data</ImageData>
            </Attachment>
        </AttachmentDescriptor>
    </requestData>
</SISubmitRFCFulfillmentRequest>
```

### Response Message—SOAP-based Services

VROL uses the RFCID to match the fulfillment response to an existing case folder, extracts the fulfillment's image, and updates the case stage and state as necessary. It makes the fulfillment visible to the issuer and marks the matching case unread.

After successful message validation, RTSI sends a return message as shown in the following example.

```
<SISubmitRFCFulfillmentResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <RFCID>00000000</RFCID>
    </responseData>
</SISubmitRFCFulfillmentResponse>
```

## Request Message—REST-based Services

```
{  
    "RequestHeader": "...",  
    "requestData": {  
        "RetrievalRequestID": "000000000000",  
        "RFCID": "000000",  
        "VisaCaseNumber": "0000000000"  
        "AttachmentDescriptor": {  
            "AttachType": "Mime",  
            "Attachment": [  
                {  
                    "ContentType": "image/tiff",  
                    "Comment": "a",  
                    "ImageData": "image1"  
                    "id": "123456",  
                }  
            ]  
        },  
        "DocType": "CALET"  
    }  
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "VisaCaseNumber": "0000000000",  
        "RFCID": "0000000"  
    }  
}
```

## SISubmitRFCNonFulfillmentRequest

Acquirers use SISubmitRFCNonFulfillmentRequest to submit an RFC nonfulfillment to VROL. This operation accepts parameters specifying the RFC request and nonfulfillment data. It returns the same information to verify that the request was received.

The following table provides information about some XML elements that should be considered when submitting an RFC nonfulfillment request or retrieving a response.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL. <b>Note:</b> In the schema, the VROL Case Number is <VisaCaseNumber>.
RFCID	VROL-generated identifier that uniquely associates an RFCAdvice with a VROL case. If the RFCAdvice is matched to more than one VROL case, each case will carry a unique RFCID. The RFCAdvice itself is identified through the RetrievalRequestID.
RetrievalRequestID	The value is obtained from the RFC advice.

Element	Description
NonFulfillmentReason	See Interface Element Spreadsheet (IES) for reason codes.
Comment	Provides an explanation associated with the reason code. This element is optional in the request. The response contains this element only when it is provided in the request.

### Request Message—SOAP-based Services

The following example shows a SubmitRFCNonFulfillmentRequest message providing an RFC nonfulfillment.

```
<SISubmitRFCNonFulfillmentRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <RetrievalRequestID>000000000000</RetrievalRequestID>
        <NonFulfillmentReason>01</NonFulfillmentReason>
        <Comment>test</Comment>
        <RFCID>0000000</RFCID>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
    </requestData>
</SISubmitRFCNonFulfillmentRequest>
```

### Response Message—SOAP-based Services

After successful message validation, RTSI sends a return message, as shown in the following example.

At the same time, the nonfulfillment is sent to RTSI for processing. When returned from RTSI, VROL uses the Retrieval Request ID to match the nonfulfillment response to an existing case folder and updates the case stage and state as necessary. It makes the nonfulfillment visible to the issuer and marks the matching case unread.

```
<SISubmitRFCNonFulfillmentResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <RFCID>0000000</RFCID>
    </responseData>
</SISubmitRFCNonFulfillmentResponse>
```

## Request Message—REST-based Services

```
{  
    "RequestHeader": "...",  
    "requestData":{  
        "RetrievalRequestID": "aaaaaaaaaaaa",  
        "NonFulfillmentReason": "B",  
        "Comment": "A",  
        "RFCID": 1,  
        "VisaCaseNumber": "0000000000",  
        "AttachmentDescriptor":{  
            "AttachType": "Mime",  
            "Attachment": [  
                {  
                    "ContentType": "image/tiff",  
                    "Comment": "a",  
                    "ImageData": "image1"  
                    "id": "123456",  
                }  
            ]  
        },  
        "DocType": "CALET"  
    }  
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData":{  
        "VisaCaseNumber": "0000000000",  
        "RFCID": 1,  
        "DocumentIndicators":{  
            "DocId": 123456  
        }  
    }  
}
```

## SIGetRFCFulfillmentRequest

An issuer uses SIGetRFCFulfillmentRequest to obtain the acquirer-supplied RFC fulfillment. This operation takes an RFC identifier (RFCID) and Visa Case Number and returns the RFC fulfillment.

## Request Message—SOAP-based Services

```
<SIGetRFCFulfillmentRequest>  
    <RequestHeader>...</RequestHeader>  
    <requestData>  
        <RFCID>0000000</RFCID>  
        <VisaCaseNumber>0000000000</VisaCaseNumber>  
    </requestData>  
</SIGetRFCFulfillmentRequest>
```

## Response Message—SOAP-based Services

```
<SIGetRFCFulfillmentResponse>
  <Status>...</Status>
  <ResponseData>
    <RetrievalRequestID>000000000000</RetrievalRequestID>
    <RFCCPD>2013-12-10</RFCCPD>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <RFCID>00000000</RFCID>
    <AttachmentDescriptor>
      <Attachment id="1">
        <ContentType>image/tiff</ContentType>
        <Comment>String</Comment>
        <ImageData>binary data</ImageData>
      </Attachment>
    </AttachmentDescriptor>
  </ResponseData>
</SIGetRFCFulfillmentResponse>
```

## Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData": {
    "RFCID": "00000000",
    "VisaCaseNumber": "0000000000"
  }
}
```

## Response Message—REST-based Services

```
{
  "Status": "...",
  "responseData": {
    "RetrievalRequestID": "000000000000",
    "RFCCPD": "2016-01-01",
    "VisaCaseNumber": "0000000000",
    "RFCID": 123456,
    "AttachmentDescriptor": {
      "Attachment": [
        {
          "ContentType": "image/tiff",
          "ImageData": "api/getFulfillment/111111111111/123456?memberRole=I&user=test",
          "id": 123456
        }
      ]
    },
    "DocType": "CALET"
  }
}
```

## SIGetRFCNonFulfillmentRequest

An issuer uses SIGetRFCNonFulfillmentRequest to obtain the RFC nonfulfillment provided by the acquirer. This operation takes an RFC identifier and Visa Case Number and returns the RFC nonfulfillment data.

This operation is used in request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SIGetRFCNonFulfillmentRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <RFCID>0000000</RFCID>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
    </requestData>
</SIGetRFCNonFulfillmentRequest>
```

### Response Message—SOAP-based Services

The nonfulfillment data returned includes the nonfulfillment reason code and an optional explanation. For RFC reason codes, see the Interface Element Spreadsheet.

```
<SIGetRFCNonFulfillmentResponse>
    <Status>...</Status>
    <responseData>
        <RetrievalRequestID>000000000000</RetrievalRequestID>
        <NonFulfillmentReason>04</NonFulfillmentReason>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <RFCID>0000000</RFCID>
    </responseData>
</SIGetRFCNonFulfillmentResponse>
```

### Request Message—REST-based Services

```
{
    "RequestHeader": "...",
    "requestData": {
        "RFCID": "0000000",
        "VisaCaseNumber": "0000000000"
    }
}
```

**Response Message—REST-based Services**

```
{  
  "Status": "...",  
  "responseData": {  
    "RetrievalRequestID": "aaaaaaaaaaaa",  
    "NonFulfillmentReason": "15",  
    "Comment": "A",  
    "RFCCPD": "2016-01-01",  
    "VisaCaseNumber": "0000000000",  
    "RFCID": "0000000",  
    "AttachmentDescriptor": {  
      "Attachment": [  
        {  
          "ContentType": "image/tiff",  
          "ImageData": "api/getNonFulfillment/1111111111/  
          123456?memberRole=I&user=test",  
          "id": 123456  
        }  
      ]  
    },  
    "DocType": "CALET"  
  }  
}
```

# Dispute Questionnaires

7

Dispute is a process for initiators to submit a single or multiple single-dispute transactions for the correction of liability as per *Visa Core Rules* and *Visa Product and Service Rules*. The dispute process is invoked when a member decides to initiate a dispute for one or more selected transactions and chooses a dispute category.

The member that initiates the dispute is called the *initiator* and the receiving member is called the *opposer*. Only issuers can initiate disputes for Fraud, Authorization, Processing Error, and Consumer categories. For more information about dispute categories, refer to [Dispute Resolution Process](#).

During the dispute process, the initiator must specify the dispute category along with other details. VROL determines if disputes are valid or invalid by validating the transaction data, dispute category selected, and inputs to the dispute questionnaire against *Visa Rules*. VROL blocks invalid disputes. VROL routes valid disputes through one of two workflows—allocation or collaboration, based on the dispute category.

Users can identify incoming dispute transactions from opposing members in the regular and batch dispute queues and take the required action. In addition, these transactions can be identified using Hypersearch.

Multi-dispute is not supported. Member host systems that need multi-dispute capabilities can be designed to trigger multiple single-dispute Web service calls. Disputes can be initiated for Visa, Plus, and Interlink networks only.

## Dispute Workflows

VROL routes valid disputes to one of the following two processes:

- **Allocation**—A workflow where VROL automatically determines liability based on transaction information provided, *Visa Rules*, and dispute category selected. VROL generates dispute financial messages and assigns liability. VROL generates dispute financial messages only if the organization is configured to have VROL generate the financials; otherwise, VROL generates financial fields for the member host system to use when creating the financial. Disputes processed with a Fraud or Authorization dispute category follow the allocation process.
- **Collaboration**—A workflow where VROL does not determine and assign liability and an exchange of information between issuer and acquirer is required. Disputes processed with a Consumer or Processing Error dispute category follow this workflow.

## Dispute Process

Dispute transactions can be created in a new case, or in an existing case as long as the case contains a selected original transaction.

The possible actions for dispute transactions are as follows:

- **Initiate and Submit**—The member that initiates the dispute is considered the initiator and the receiving member is considered the opposer. A dispute will qualify for the allocation or collaboration workflow based on the dispute category selected.
- **Respond**—Members can use different response Web services based on the dispute phase that they are in.
  - Response to an incoming allocation dispute will be a "Pre-Arbitration".
  - Response to an incoming collaboration dispute will be a "Dispute Response".
  - Response to an incoming collaboration Dispute Response will be a "Pre-Arbitration".
  - Response to an incoming allocation/collaboration Pre-Arbitration will be a "Pre-Arbitration Response".
- **Recall**—Initiators of a dispute or dispute response who changed their mind and want to recall the submitted dispute or dispute response can do so if the following conditions are met:

**Note:** *Initiators can only recall a dispute response with "Decline" value.*

- The dispute or dispute response has not been previously recalled.
- The current date is within 3 days of the dispute or dispute response's submission date.
- The opposer has not yet responded to the incoming dispute or accepted the received dispute.
- The time frame for responding to the dispute has not yet expired.

VROL also allows the users to recall a dispute-related questionnaire that is in "Pending Advice" status, as well as those that have been delivered to the other side. When this is recalled, VROL checks if the associated financial, if any, has been received by VROL and validated.

Recalls can only be submitted. They cannot be saved.

[Table 7-1](#) shows the available request and response Web services for disputes.

**Table 7-1: Available Requests and Responses**

Dispute workflow	Request	Response
Allocation	Dispute (using SIIInitiateDisputeFromTransactionOrCaseRequest and SISubmitDisputeQuestionnaireRequest)	Pre-Arbitration (using SICreateDisputePreArbRequest)
Collaboration	Dispute (using SIIInitiateDisputeFromTransactionOrCaseRequest and SISubmitDisputeQuestionnaireRequest)	Dispute Response (using SICreateDisputeResponseRequest)

**Table 7–1: Available Requests and Responses (cont'd.)**

Dispute workflow	Request	Response
Collaboration	Dispute Response (using SICreateDisputeResponseRequest)	Pre-Arbitration (using SICreateDisputePreArbRequest)
Allocation or Collaboration	Pre-Arbitration (using SICreateDisputePreArbRequest)	Pre-Arbitration Response (using SICreateDisputePreArbResponseRequest)

See [Chapter 10, Dispute Pre-Filings](#) for details on Pre-Arbitration. If the dispute remains unresolved at the end of Pre-Arbitration, the initiator can file the case with Visa. See [Chapter 11, Dispute Case Filings](#) for details.

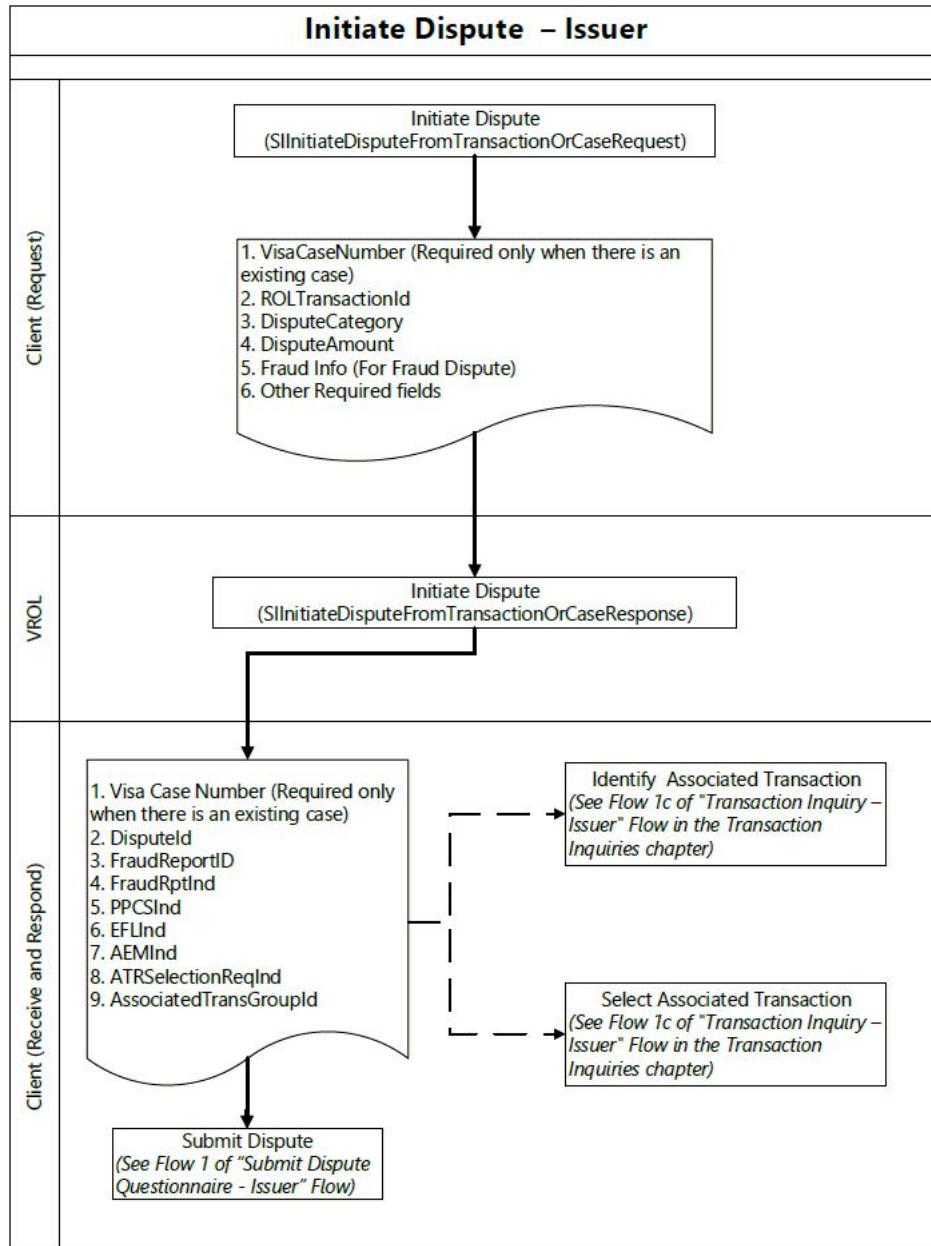
## Dispute Process Flows

The following diagrams illustrate the initiate and submit dispute process flows.

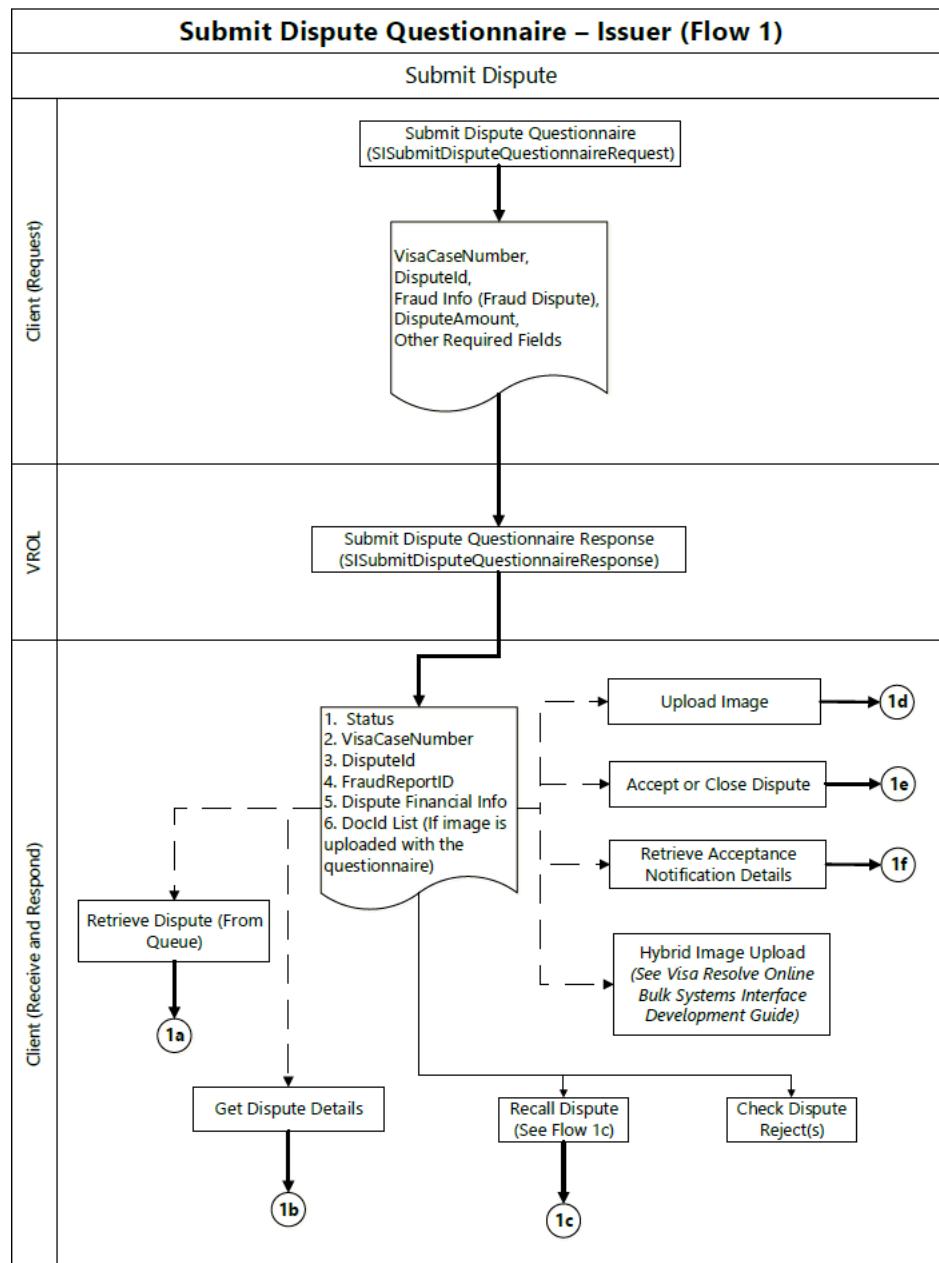
**Note:**

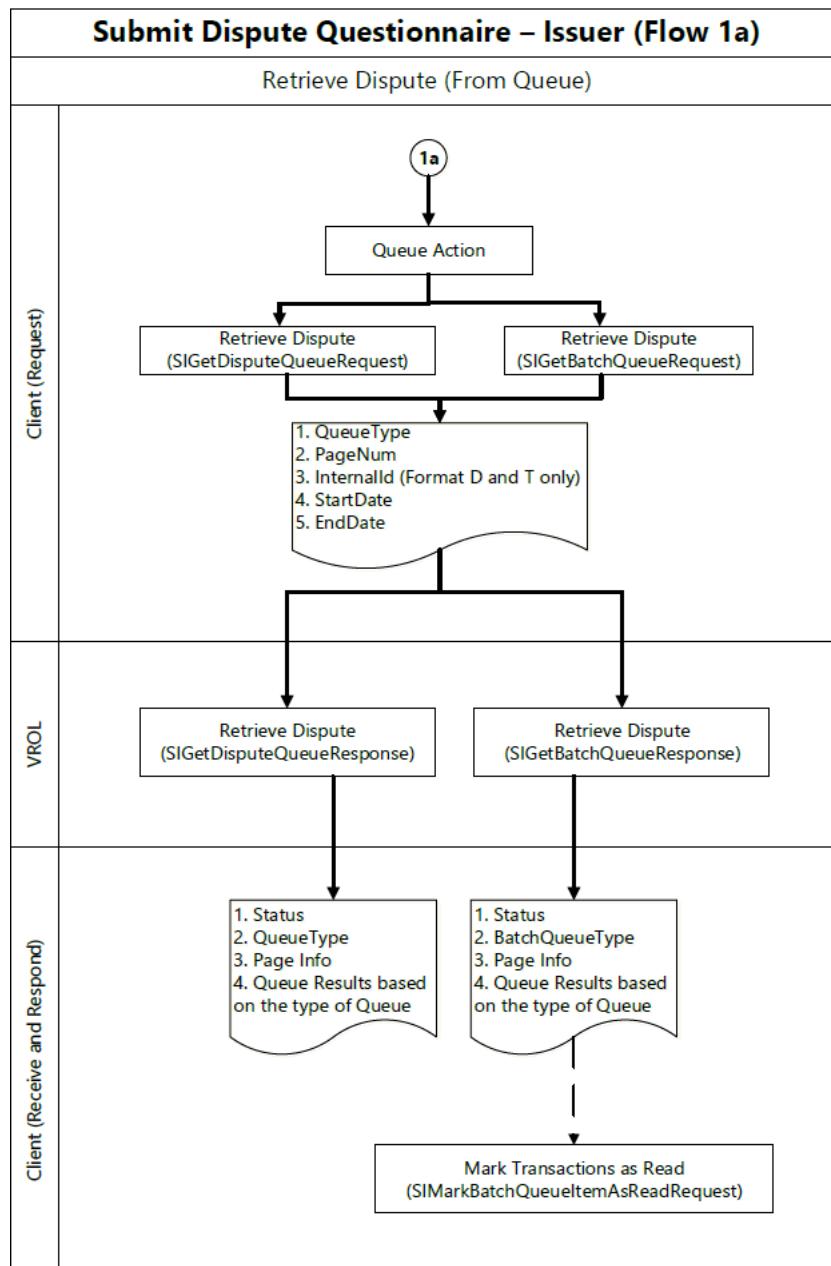
- *The diagrams list only the important elements. For the complete elements, refer to the Dispute tab of the Interface Element Specification (IES).*
- *The dashed lines in the flow refer to optional functions that the user can use, depending on the business process that the user wants to implement.*

**Figure 7–1: Initiate Dispute Flow**

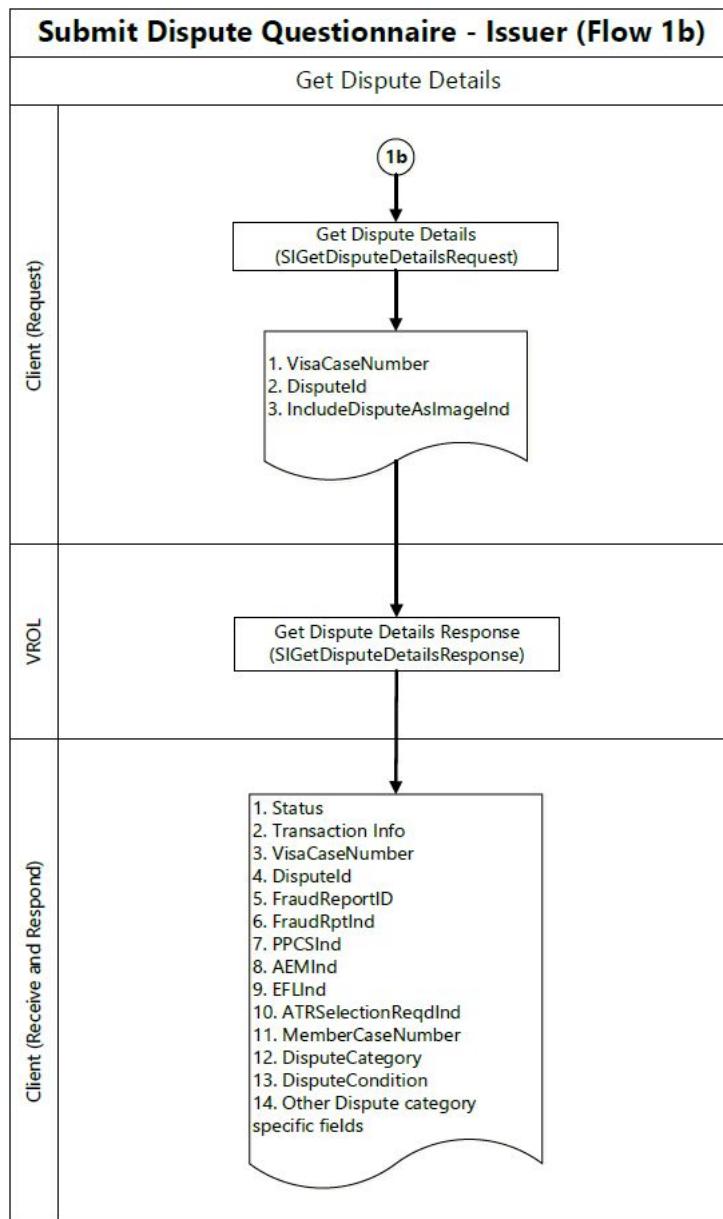


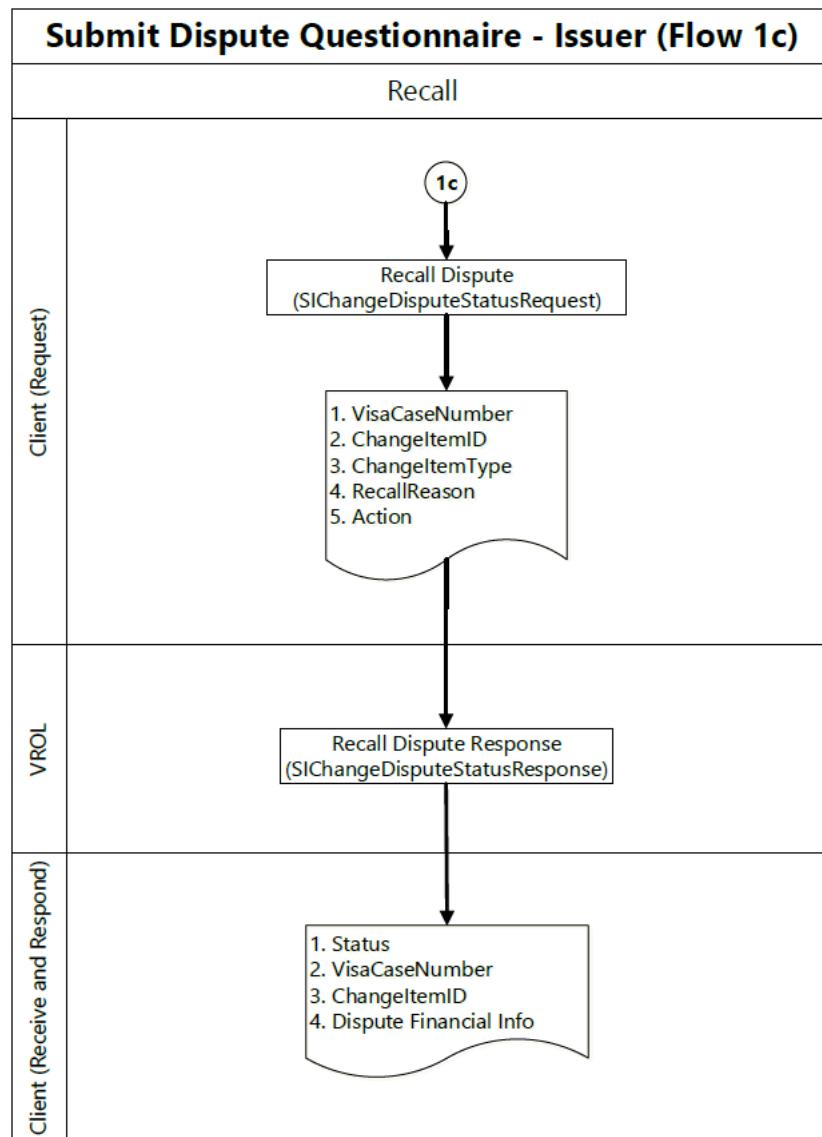
**Figure 7–2: Submit Dispute—Flow 1**

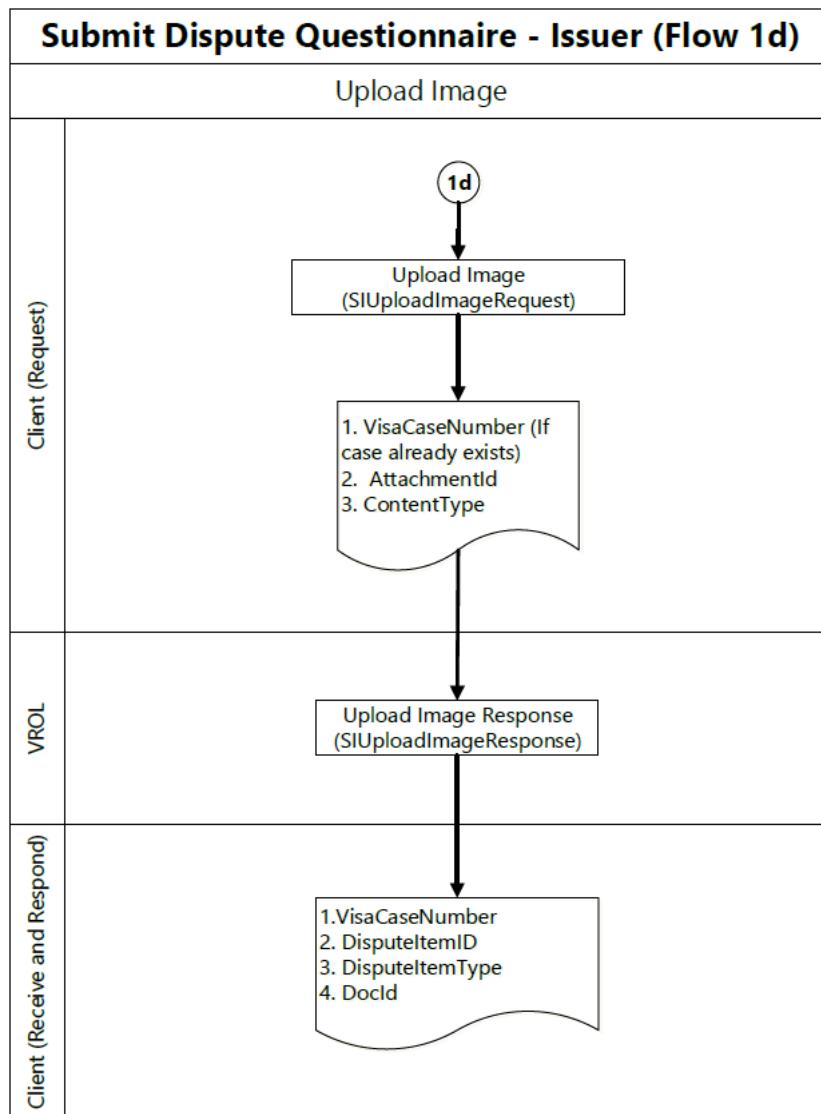


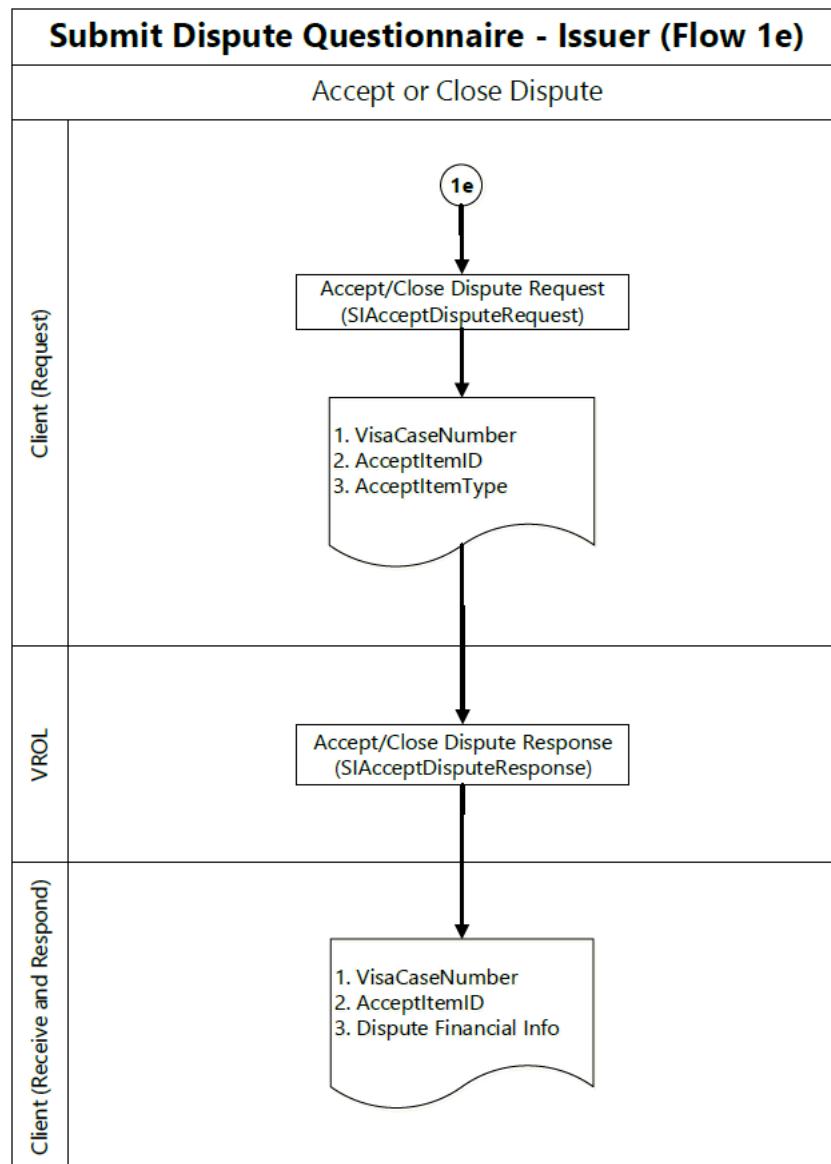
**Figure 7-3: Submit Dispute—Flow 1a**

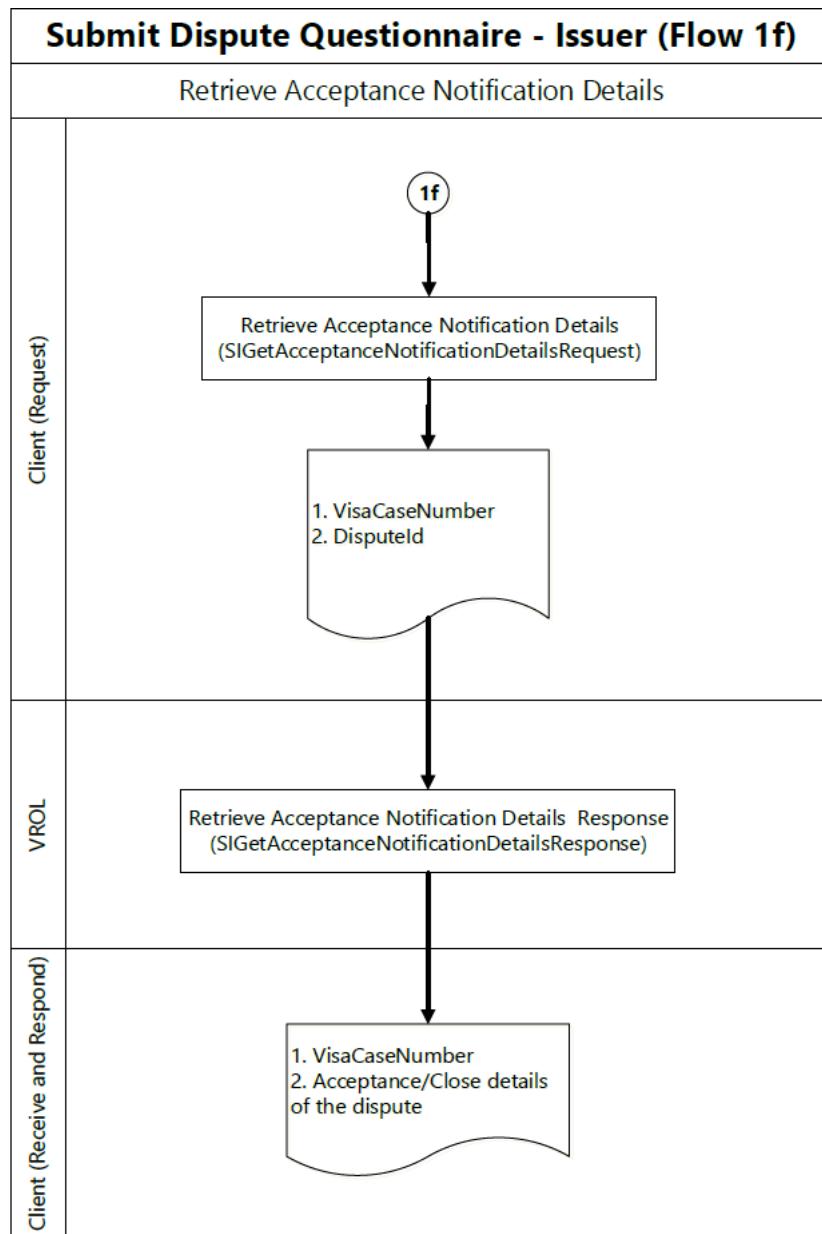
**Figure 7–4: Submit Dispute—Flow 1b**



**Figure 7-5: Submit Dispute—Flow 1c**

**Figure 7–6: Submit Dispute—Flow 1d**

**Figure 7-7: Submit Dispute—Flow 1e**

**Figure 7–8: Submit Dispute—Flow 1f**

## Initiate and Submit Dispute

This process of initiating and submitting disputes is applicable for the following issuer-initiated disputes. For Pre-Compliance, see [Chapter 10, Dispute Pre-Filings](#) for details.

- Fraud
- Authorization
- Processing Error
- Consumer

**Note:** Although not a dispute category, VROL provides a "Fraud Bundle" function that allows initiators to dispute multiple fraudulent low-amount eligible transactions as a group. See [Submit and Retrieve Fraud Bundle Disputes](#) for details.

Creating a dispute is a two-step process:

1. Initiate a dispute

This first step involves the following actions:

- a. Member initiates a dispute.
- b. VROL validates the supplied dispute category.
- c. VROL searches for associated transactions.
- d. VROL searches for matching Fraud Reports, Exception File Listings, Stop Payment Orders, and Provisional Credits as appropriate.
- e. VROL returns the validation status/search status results to the member host system.

2. Submit the dispute

This second step involves the following actions:

- a. Member submits or saves the dispute.
- b. VROL accepts the dispute questionnaire elements.
- c. VROL applies dispute validation edits.
- d. VROL saves or submits the dispute.

The two-step process allows VROL to notify the member host system of steps that must be taken by the user before the dispute questionnaire can be successfully submitted. Those intermediate steps include retrieving and changing/confirming associated transactions, and in some scenarios, submitting a Fraud Report, Exception File Listing, Stop Payment Order, and/or Provisional Credit.

**Note:** If the user does not perform the mandatory step of retrieving and changing/confirming associated transactions before submitting the dispute, VROL returns a hard stop error.

## Initiate Dispute

Users can initiate a dispute from an original transaction or from an existing case that contains a selected original transaction. The original transaction can be retrieved using the [SIGetTransactionsFromCaseRequest](#) Web service. See [Chapter 4, Transaction Inquiries](#).

Users cannot create disputes with manually-supplied transaction data.

1. To initiate a dispute, the member host system triggers a [SIIInitiateDisputeFromTransactionOrCaseRequest](#) message supplying the `R01TransactionId`, `VisaCaseNumber`, `DisputeCategory`, and `DisputeAmount` as required fields.

For a dispute with Fraud as the supplied dispute category, the member host system can optionally provide Fraud and Exception File Listing fields as necessary.

When initiating a dispute from an existing case, provide only the `VisaCaseNumber`. A new case will be created if `VisaCaseNumber` is not provided.

2. VROL applies initial dispute validation edits based on transaction data elements and the supplied dispute category.
  - If edits pass, proceed to step 3.

- If edits fail, VROL returns an error response which includes a full list of all edits that failed the initial dispute validation.
3. VROL searches for associated transaction information and checks for the following conditions:

Authorization disputes	If an Exception File Listing is optional and already exists or not.
Fraud disputes	<p><b>Note:</b> VROL always searches for an existing Fraud Report (within VROL or Fraud History Inquiry) or Exception File Listing.</p> <ul style="list-style-type: none"> <li>• Fraud Report <ul style="list-style-type: none"> <li>- If Fraud details were provided in the SIIInitiateDisputeFromTransactionOrCaseRequest message and VROL determines that there is an existing Fraud Report within the allowed time frame, then the FraudRptInd value is set to "Exist" and no Fraud Report is submitted. If VROL determines that there is no existing Fraud Report, then one will be submitted using the information provided by the user. The FraudRptInd value is set to "Exist" and the FraudReportID is generated in the response. If the submission process is beyond the allowed time frame, then VROL still sets the FraudRptInd as "Exist" and continues to asynchronously process Fraud Report.</li> <li>- If Fraud details were not provided in the SIIInitiateDisputeFromTransactionOrCaseRequest message and VROL determines that a Fraud Report is missing, then the FraudRptInd value is set to "NotExist". If a Fraud Report already exists, then the FraudRptInd value is set to "Exist". If VROL could not determine the submission within the allowed time frame, then FraudRptInd value is set as "Unknown".</li> </ul> </li> <li>• Exception File Listing (EFL) <ul style="list-style-type: none"> <li>- If EFL details were provided in the SIIInitiateDisputeFromTransactionOrCaseRequest message and VROL determines that there is an existing EFL within the allowed time frame, then the EFLInd value is set to "Exist" and no EFL is submitted. If VROL determines that there is no existing EFL, then one will be submitted using the information provided by the user. The EFLInd value is set to "Exist". If the submission process is beyond the allowed time frame, then VROL still sets the EFLInd as "Exist" and continues to asynchronously process EFL.</li> <li>- If EFL details were not provided in the SIIInitiateDisputeFromTransactionOrCaseRequest message and VROL determines that an EFL is missing, then the EFLInd value is set to "NotExist". If an EFL already exists, then the EFLInd value is set to "Exist". If VROL could not determine the submission within the allowed time frame, then EFLInd value is set as "Unknown".</li> </ul> </li> </ul>
Consumer disputes	If a Stop Payment Order is optional and already exists or not.
All dispute types	If a Provisional Credit based on VROL Member Configuration is required/optional and already exists or not.

VROL returns a SIIInitiateDisputeFromTransactionOrCaseResponse message with the VisaCaseNumber, DisputId, FraudReportID (if applicable), and AssociatedTransGroupId along with the following elements:

- ATRSelectionReqdInd (whether ATR selection is required or not)
- FraudRptInd (whether a Fraud Report is required to be submitted or not)
- PP CSI Ind (whether a Stop Payment Order is required to be submitted or not)
- EFLInd (whether an Exception File Listing is required to be submitted or not)
- AEMInd (whether a Provisional Credit is required to be submitted or not)

**Note:** *These elements are displayed as "Unknown" in circumstances where VROL exceeds the allowed response time of 3 seconds to fetch ATR and/or determine the statuses of ancillary actions.*

**Note:** *After SISubmitDisputeQuestionnaireRequest is received, VROL processes the dispute initiation for the specified combination of VisaCaseNumber and Disputelid. If the user initiates SISubmitDisputeQuestionnaireRequest while the dispute initiation is still in progress, VROL returns an error message that indicates that the dispute initiation is still in progress and to try again after sometime.*

The member host system can retrieve the response by triggering a SIGetDisputeDetailsRequest message using the VisaCaseNumber and Disputelid. See [Get Dispute Details](#).

## Submit Dispute

This step assumes that the user already performed the following actions as required and/or appropriate:

- a. Retrieving and viewing the list of associated transactions (if any exists) by triggering a SIGetAssociatedTransactionListRequest message. This requires the AssociatedTransGroupId in the request message which will then return a list of transactions that are associated with a selected original transaction.
- b. Confirming or updating the associated transaction selection (if any exists) by triggering a SIAssociatedTranSelectionRequest message. This requires the VisaCaseNumber, R0lTransactionId (of the associated transaction), AssociatedTransGroupId, and Associated value (whether or not to associate a group of transactions to a selected transaction) in the request message.
- c. Required/optional supplementary actions (submission of Fraud Report, Exception File Listing, Stop Payment Order, and/or Provisional Credit as appropriate) using the corresponding legacy RTSI Web services.

**Note:**

- *If these supplementary actions are marked as required but are not submitted before submitting a dispute questionnaire request, then VROL returns a hard stop error.*
- *Existing partial AEMs are conveyed to end users through the SIGetDisputeDetailsResponse status message. Unless user submits the remaining AEM, continuing with the dispute questionnaire request will result in a hard stop error if the user's organization is configured for mandatory Provisional Credit with disputes.*
- *If the user does not perform the mandatory step of retrieving and changing/confirming associated transactions before submitting the dispute, VROL returns a hard stop error.*

See [Chapter 4, Transaction Inquiries](#) for associated transactions, and the corresponding chapters for legacy RTSI Web services ([Chapter 13, Fraud Reporting](#), [Chapter 14, Exception File Listings](#), [Chapter 15, Stop Payments](#), and [Chapter 16, Accounting Entry Memos](#)).

1. Submitting a non-Fraud dispute questionnaire or a Fraud dispute questionnaire:
  - To submit (or save) a non-Fraud dispute questionnaire, the member host system triggers a SISubmitDisputeQuestionnaireRequest message supplying the VisaCaseNumber, dispute questionnaire data elements, and an Action value=Save or Submit. This request can include image(s).
  - To submit (or save) a Fraud dispute questionnaire, the member host system takes the same steps as above, and provides optional Fraud or EFL fields.

**Note:** *Perform the same step to submit a previously-saved dispute and change the Action value to Submit.*

All disputes must be submitted beyond the allowed time frame.

- Fraud disputes and Consumer disputes must be submitted by 9 AM GMT on the 120th calendar day from the Transaction Processing Date (time starts on the day following the Transaction Processing Date). Some exceptions may apply to Consumer disputes and time frames of such exceptions as per the dispute validation edits.

**Note:** *For members who generate their own financials, the time limit is set to 9AM GMT on the 119th calendar day.*

- Authorization disputes and Processing Error disputes with 'Invalid Data' condition must be submitted by 9 AM GMT on the 75th calendar day from the Transaction Processing Date (time starts on the day following the Transaction Processing Date).

**Note:** *For members who generate their own financials, the time limit is set to 9 AM GMT on the 74th calendar day.*

2. When a dispute is saved, VROL checks for the following items:

- Potential duplicates of dispute in other cases. If a duplicate is found, VROL returns a warning message that indicates that one or more potential duplicates have been found.

**Note:** *VROL performs the duplicate check only when saving the dispute in the case for the first time.*

- Legacy VDAS advice (chargeback or representment) was already received for the same dispute. If a legacy VDAS advice is found, this advice is considered as an evidence of the prior duplicate dispute.

3. To attach one or more images to the questionnaire, include the DisputeAttachmentDescriptor group element in the request message.

Members can also use the Bulk SI Hybrid Image upload functionality to generate DocIds in bulk which can be supplied in the SISubmitDisputeQuestionnaireRequest against the DocId element. For details, see [Hybrid Image Uploads](#) and *Visa Resolve Online Bulk Systems Interface Development Guide*.

4. For a Submit action, VROL applies dispute validation edits. The Save action does not trigger any validations.

VROL displays a message indicating which workflow the dispute is going to follow.

5. If the user wants to use the Bulk SI Hybrid Image upload functionality to supply the documents, including image details, to a saved dispute questionnaire, set the <SupplyDocInd>=true.

- If the <SupplyDocInd>=true, then the dispute questionnaire will always get saved even if the user sets the Action value to Submit.

- If the <SupplyDocInd>=false, then the dispute questionnaire will get submitted when the user submits the dispute questionnaire.
- If the user fails to attach the image to a saved questionnaire through the Bulk SI Hybrid Image upload functionality, the user must supply the missing elements through RTSI, save the questionnaire, and try to attach the image again using the same Bulk SI service.
6. If the user has submitted a Fraud dispute, VROL will search again to determine if an EFL or a matching Fraud Report exists.

Fraud Report	<ul style="list-style-type: none"> <li>• If one exists and Fraud Report information is supplied with the questionnaire although it is not required, then the Fraud Report information supplied is discarded.</li> <li>• If one does not exist, VROL uses any Fraud Report fields provided in the request message to create and submit a new Fraud Report.</li> <li>• If one does not exist and the request message does not include any Fraud Report fields, VROL returns a hard stop error and the dispute is not submitted.</li> </ul>
EFL	<ul style="list-style-type: none"> <li>• If one exists and EFL information is supplied with the questionnaire although it is not required, then the EFL Report information supplied is discarded.</li> <li>• If one does not exist, then one will be submitted using the EFL fields provided in the request message.</li> </ul>

VROL returns a SISubmitDisputeQuestionnaireResponse message with the success status, FraudReportID (if a Fraud Report is submitted along with the dispute), DocIds for any attached images, and financial fields.

See [Financial Fields Returned in Immediate Response](#) for a list of financial fields returned in the response. See [Chapter 8, Dispute Financials](#) for an explanation of what these fields represent and how to use them.

7. If the dispute fails one or more edits, then the dispute submission is unsuccessful and VROL returns a hard stop error. VROL returns an error response that includes a full list of all edits that failed the dispute validation. The user has to submit the dispute request again.

## Submit and Retrieve Fraud Bundle Disputes

Initiators can dispute multiple fraudulent low-amount eligible transactions as a group by using the Fraud Bundle function. All transactions that are disputed using this function will use the Fraud dispute category.

Transactions are eligible for fraud bundling if all the following conditions are met:

- Each selected transaction qualifies for the Fraud dispute category.
- Each selected transaction is for a digital goods merchant category code (MCC).
- Each selected transaction has the same card/account number, merchant, acquirer, and merchant location.
- Each selected transaction has a transaction amount or an equivalent settlement transaction amount that is less than 25 USD.
- The user has selected a minimum of two transactions up to a maximum total of 25 transactions.
- The total transaction amount or equivalent settlement transaction amount of all selected transactions is equal to or less than 250 USD.

**Note:** Transactions where the USD equivalent settlement transaction amount is zero are not eligible for fraud bundling.

- VROL creates an individual case containing a Fraud Dispute Questionnaire for each transaction being disputed as part of the bundle.
- During Fraud Bundling, VROL adds a Fraud Dispute Questionnaire to each case and creates an additional 'parent' case with a unique VROL case number. This additional case is linked to each case within the bundle, and each case within the bundle is linked to the additional case. This allows users to identify which Fraud disputes are part of a specific Fraud bundle.

For Fraud bundled cases, VROL generates the dispute financials or returns financial information depending on the organization's member configuration. For example, if the bundle contains five individual cases plus the umbrella case, and the organization's member configuration is set to have VROL generate the dispute financials, then VROL generates five dispute financials once the dispute is successfully submitted and liability is allocated.

**Note:** Transactions or cases cannot be added to, or removed from, an existing Fraud bundle.

## Submit Fraud Bundle

1. To submit a Fraud Bundle dispute, the member host system triggers a SISubmitFraudBundleDisputeQuestionnaireRequest message supplying the different RolTransactionIds and applicable dispute questionnaire data elements.

Fraud Bundle disputes can also be initiated from existing cases. When initiating a Fraud Bundle dispute from existing cases, the VisaCaseNumber of all the cases must be provided.

In either cases, a FraudBundleID is created.

SISubmitFraudBundleDisputeQuestionnaireRequest can be saved or submitted.

- Save

A SISubmitFraudBundleDisputeQuestionnaireRequest message can be saved by supplying the required fields above and an Action value=Save. This request can include image(s).

- Submit

A SISubmitFraudBundleDisputeQuestionnaireRequest message can be submitted by supplying the required fields above and an Action value=Submit. This request can include image(s).

**Note:** Perform the same step to submit a previously-saved dispute and change the Action value to Submit (using the list of RolTransactionIds).

VROL returns a SISubmitFraudBundleDisputeQuestionnaireResponse message with the FraudBundleID.

2. VROL applies applicable dispute validation edits based on transaction data elements.
  - For transactions in the bundle that pass all edits, proceed to step 4.
  - For transactions in the bundle that fail one or more edits, VROL returns an error response which includes a full list of all edits that failed the initial dispute validation.

In the response message, VROL indicates the status for each transaction or case in the bundle. For example, if 5 transactions were submitted and 2 failed the

edits, the response message will contain a success status for the 3 transactions as well as a failure status and transaction details for the 2 transactions that failed the edits.

**Note:** For a Submit action, VROL applies dispute validation edits. The Save action does not trigger any validations.

**Note:** If only one of the transactions or cases in the bundle fails the edit validation, VROL returns a hard-stop error only for that transaction or case that failed the edit validation.

3. To attach one or more images to the questionnaire, include the DisputeAttachmentDescriptor group element in the request message.

Members can also use the Bulk SI Hybrid Image upload functionality to generate DocIds in bulk which can be supplied in the SISubmitDisputeQuestionnaireRequest against the DocId element. For details, see [Hybrid Image Uploads](#) and [Visa Resolve Online Bulk Systems Interface Development Guide](#).

**Note:** The attached image gets associated with the Fraud Dispute questionnaire in each transaction or case in the bundle.

4. VROL searches for associated transaction information and checks for the following conditions in each of the associated transaction or case:

- Fraud Report

If Fraud details for each transaction or case in the bundle were provided in the SISubmitFraudBundleDisputeQuestionnaireRequest message, then VROL submits the Fraud Report with the supplied details for each transaction or case in the bundle.

If Fraud details were not provided in the SISubmitFraudBundleDisputeQuestionnaireRequest message, VROL searches for an existing Fraud Report for each of the transactions. When an existing Fraud Report is available for a transaction or case in the bundle, VROL processes the transaction or case successfully. Otherwise, if the Fraud Report is not available for a transaction, VROL returns a hard stop error for that transaction or case.

- Exception File Listing is optional and already exists or not

If Exception File Listing details were provided in the SISubmitFraudBundleDisputeQuestionnaireRequest message, then VROL submits a single Exception File Listing with the supplied details. Otherwise, if Exception File Listing details were not provided in the SISubmitFraudBundleDisputeQuestionnaireRequest message, then VROL searches for an existing EFL.

- Provisional Credit based on VROL Member Configuration is required/optional and already exists or not

If Provisional Credit details were provided in the SISubmitFraudBundleDisputeQuestionnaireRequest message, then VROL submits the Provisional Credit for each transaction or case in the bundle with the supplied details if one does not exist. Otherwise, if Provisional Credit details were not provided in the SISubmitFraudBundleDisputeQuestionnaireRequest message, then VROL searches for an existing Provisional Credit.

If Provisional Credit is not available, based on the member configuration settings, VROL either ignores it or returns a hard stop error for that transaction or case.

## Retrieve Fraud Bundle Dispute Details

To retrieve fraud bundle dispute details, the member host system triggers a SIGetFraudBundleDisputeQuestionnaireStatusRequest message supplying the FraudBundleID.

VROL returns a SIGetFraudBundleDisputeQuestionnaireStatusResponse message with the details of the fraud bundle dispute, which includes all VisaCaseNumbers, RolTransactionIds, and Disputelids along with the complete or in progress response that includes the list of errors and edit errors, if any.

**Note:** *If the user intends to get the details of the individual cases in a bundle, the member host system can retrieve the response by triggering a SIGetDisputeDetailsRequest message supplying the individual VisaCaseNumbers, corresponding RolTransactionIds, and Disputelids from SIGetFraudBundleDisputeQuestionnaireStatusResponse. See [Get Dispute Details](#) for more information.*

**Note:** *The edit rejected case in the steps above will still be part of the bundle and once it has been corrected, the behavior will be similar to the other cases in the same bundle. If the case is not created, then the transaction will not be part of the bundle.*

## Retrieve Dispute from Queue

1. To retrieve dispute questionnaires from the corresponding dispute queues, the member host system triggers a SiGetQueueRequest message or a SIGetBatchQueueRequest message. See [Chapter 17, Queues](#) and [Chapter 18, RTSI Batch Queues](#).

VROL returns a list of the dispute cases in the queue including the VisaCaseNumber and InternalId for each case. The InternalId element contains the Disputelid value, and can be used to retrieve details of the dispute.

2. To close the dispute or dispute response and remove it from the queue, user sends close dispute request SICloseTransactionRequest. See [Chapter 17, Queues](#) for more information.

**Note:** *SICloseTransactionRequest closes the transaction and removes it from its respective queue but does not close the case. For more information, see [SICloseTransactionRequest](#).*

VROL sends response with success status.

## Get Dispute Details

1. To retrieve dispute details, the member host system triggers a SIGetDisputeDetailsRequest message supplying the VisaCaseNumber and Disputelid.

VROL returns a SIGetDisputeDetailsResponse message with the dispute transaction and image details along with the ATRSelectionReqdInd, FraudRptlnd, FraudReportID, PPCSInd, EFLInd, and/or AEMInd. See the table in [SIIInitiateDisputeFromTransactionOrCaseRequest](#) for descriptions and domain values of these response indicators.

**Note:** *FraudReportID is displayed in the SIGetDisputeDetailsResponse message if a Fraud Report is submitted along with the dispute. Likewise, the other response indicators are displayed only if they are applicable.*

**Note:** The dispute questionnaire may be returned as an image (TIFF/PDF), in addition to XML, as part of this response message (if optional <IncludeDisputeAsImageInd>=true is included in the request). However, any images attached to the questionnaire must be retrieved separately (step 2 below).

**Note:** Any image attached to the dispute questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.

**Note:** For an existing partial Provisional Credit, the SIGetDisputeDetailsResponse status message conveys the Accounting Entry Memo amount and its currency.

2. To retrieve the attached images, the member host system triggers a SIGetImageRequest message supplying the VisaCaseNumber and DocId.

**Note:** DocIds are generated in bulk when the member uses the Bulk SI Hybrid Image upload functionality to attach image(s) in the dispute questionnaire. For details, see [Hybrid Image Uploads](#) and [Visa Resolve Online Bulk Systems Interface Development Guide](#).

VROL returns a SIGetImageResponse message with image data.

## Respond to a Received Dispute

There must be a submitted dispute in a case in order to do a Dispute Response. Prior to submitting a response, an RTSI user should retrieve the dispute details in the case.

Depending on the dispute workflow, VROL provides different options for responding to a received dispute:

- If VROL determines that the dispute follows the allocation workflow, the opposer can respond to the dispute by initiating a Pre-Arbitration. See [Chapter 10, Dispute Pre-Filings](#) for more information.
- If VROL determines that the dispute follows the collaboration workflow, the opposer can send a Dispute Response to the initiator indicating that he accepts full or partial liability, or declines the liability. The opposer performs the steps below to send a Dispute Response in the collaboration workflow.

If the time frame for responding to the received dispute has expired, VROL generates and submits a Dispute Response (Accept Full) on behalf of the non-responding side. This response is generated on the calendar day after the response time frame has expired.

The detailed flow for the process of responding to a received dispute is illustrated in [Figure 10-1 of Chapter 10, Dispute Pre-Filings](#).

1. To create a Dispute Response, the member host system triggers a SICreateDisputeResponseRequest message supplying the VisaCaseNumber, DisputeResponse, and Dispute Response data.
- SICreateDisputeResponseRequest can be saved or submitted. VROL returns a SICreateDisputeResponseResponse message with the DisputeResponseId, DocIds for any attached images, financial fields, and success status.

See [Financial Fields Returned in Immediate Response](#) for a list of financial fields returned in the response. See [Chapter 8, Dispute Financials](#) for an explanation of what these fields represent and how to use them.

- Save

A SICreateDisputeResponseRequest can be saved. When it is saved, it generates a DisputeResponseId.

- Submit  
If a SICreateDisputeResponseRequest message is submitted for a new dispute response, a DisputeResponseId is generated in the response.  
If a SICreateDisputeResponseRequest message is submitted for a previously-saved dispute response, there is no new DisputeResponseId generated in the response.
- 2. To attach one or more images to the questionnaire, include the DisputeAttachmentDescriptor group element in the request message.  
Members can also use the Bulk SI Hybrid Image upload functionality to generate DocIds in bulk which can be supplied in the SICreateDisputeResponseRequest against the DocId element. For details, see [Hybrid Image Uploads](#) and [Visa Resolve Online Bulk Systems Interface Development Guide](#).
- 3. If the user wants to use the Bulk SI Hybrid Image upload functionality to supply the documents, including image details, to a saved dispute response questionnaire, set the <SupplyDocInd>=true.
  - If the <SupplyDocInd>=true, then the dispute response questionnaire will always get saved even if the user sets the Action value to Submit.
  - If the <SupplyDocInd>=false, then the dispute response questionnaire will get submitted when the user submits the dispute response questionnaire.If the user fails to attach the image to a saved questionnaire through the Bulk SI Hybrid Image upload functionality, the user must supply the missing elements through RTSI, save the questionnaire, and try to attach the image again using the same Bulk SI service.

## Get Dispute Response Details

The detailed flow for the process of retrieving the details of a dispute response is illustrated in [Figure 10–4 of Chapter 10, Dispute Pre-Filings](#).

1. To retrieve the details of a submitted or received Dispute Response, the member host system triggers a SIGetDisputeResponseDetailsRequest message supplying the VisaCaseNumber and DisputeResponseId.  
VROL returns a SIGetDisputeResponseDetailsResponse message with the details of the Dispute Response.

**Note:** *The Dispute Response questionnaire may be returned as an image (TIFF/PDF), in addition to XML, as part of this response message (if optional <IncludeDisputeAsImageInd>=true is included in the request). However, any images attached to the questionnaire must be retrieved separately (step 2 below).*

**Note:** *Any image attached to the Dispute Response questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.*

2. To retrieve the attached images, the member host system triggers a SIGetImageRequest message supplying the VisaCaseNumber and DocId.  
**Note:** *DocIds are generated in bulk when the member uses the Bulk SI Hybrid Image upload functionality to attach image(s) in the Dispute Response questionnaire. For details, see [Hybrid Image Uploads](#) and [Visa Resolve Online Bulk Systems Interface Development Guide](#).*

VROL returns a SIGetImageResponse message with image data.

## Retrieve Acceptance Notification Details

The RTSI user can retrieve the acceptance notification details for disputes or dispute responses that were accepted and closed.

The user can accept and close a dispute or dispute response by triggering the [SIAcceptDisputeRequest](#) message. For more information, see [SIAcceptDisputeRequest](#). Optionally, the user can close the dispute or dispute response by triggering the [SICloseTransactionRequest](#) message. For more information, see [SICloseTransactionRequest](#).

1. To retrieve accepted dispute responses from the dispute acceptance queue, the member host system triggers a [SiGetQueueRequest](#) message. See [Chapter 17, Queues](#).  
VROL returns a list of the accepted dispute response cases in the queue including the VisaCaseNumber for each case.
2. To retrieve details of the acceptance notification for a dispute or dispute response, the member host system triggers a [SiGetAcceptanceNotificationDetailsRequest](#) message supplying the VisaCaseNumber, Disputelid, or DisputeResponseId.  
VROL returns a [SiGetAcceptanceNotificationDetailsResponse](#) message with the VisaCaseNumber along with the acceptance and close details of the dispute or dispute response item.

## Recall Dispute

The initiator can recall a submitted dispute or dispute response if the following conditions are met:

**Note:** *The initiator can only recall a dispute response with "Decline" value.*

- The dispute or dispute response has been submitted.
  - The dispute or dispute response has not been previously recalled.
  - The current date is within 3 calendar days of the dispute or dispute response's submission date (1 day if the pinned original transaction in the dispute case is an Original Credit Transaction).
  - The opposer has not yet responded to the incoming dispute, or accepted or partially accepted the received dispute.
  - The time frame for responding to the dispute has not yet expired.
1. To retrieve recalled disputes from the recall queue, the member host system triggers a [SiGetQueueRequest](#) message. See [Chapter 17, Queues](#).  
VROL returns a list of the recalled dispute cases in the queue including the VisaCaseNumber for each case.

2. To recall an initiated dispute, the member host system triggers a [SIChangeDisputeStatusRequest](#) message supplying the VisaCaseNumber, ChangeItemID, ChangeItemType, and Action value=Recall.

**Note:** *ChangeItemID refers to the Disputelid value of the submitted dispute.*

VROL returns a [SIChangeDisputeStatusResponse](#) message with the success status.

VROL returns a hard-stop error if a recall request is submitted into a case that does not qualify for recall.

## Delete Dispute

The initiator can delete saved collaboration or allocation disputes.

1. To delete a dispute, the member host system triggers a SIChangeDisputeStatusRequest message supplying the VisaCaseNumber, ChangeItemID, ChangeItemType, and Action value=Delete.

**Note:** *ChangeItemID includes the unique ID value for the corresponding dispute (for example, DisputeId for dispute and DisputeResponseId for Dispute Response).*

VROL returns a SIChangeDisputeStatusResponse message with the success status.

## Check for Dispute Rejects

If the dispute questionnaire or dispute response questionnaire fails one or more validation edits, VROL blocks the questionnaire and places it in the MY\_REJECTS or ALL\_REJECTS queue.

1. To check for rejected dispute questionnaires, the member host system triggers a SIGetQueueRequest message for MY\_REJECTS or ALL\_REJECTS queue. See [Chapter 17, Queues](#), for more information.

VROL returns a SIGetQueueResponse message with the VisaCaseNumber and RejectID.

2. To ignore a rejected dispute questionnaire when using the reject queue, the member host system triggers a SIgnoreRejectRequest message supplying the VisaCaseNumber, RejectID, and RejectedTransactionType="Dispute Request" or "Dispute Response". This step prevents the transaction from being returned again in the next queue request. See [Chapter 17, Queues](#), for more information.

VROL returns a SIgnoreRejectResponse message with the success status.

3. To repair a rejected dispute questionnaire, the member host system triggers a SISubmitDisputeQuestionnaireRequest message along with the updated dispute questionnaire information.

- If the case contains a single unrepaired reject of the same type as the request, VROL marks the reject as repaired.
- If the case contains multiple unrepaired rejects of the same type as the request, VROL marks the most recent reject as repaired.
- If the case does not contain a reject of the same type as the request, VROL does not mark any transaction as repaired. VROL adds the incoming transaction to the case and sends response with success status.

VROL returns a SISubmitDisputeQuestionnaireResponse message with the success status.

## Check for Disputes Eligible for Rapid Dispute Resolution

With Purchase Inquiry, VROL can request a merchant to provide purchase details and additional information. In response to VROL's Purchase Inquiry request, the merchant can choose to perform the following actions:

- Return the requested purchase details back to VROL. The purchase information becomes available to the issuer.
- Express an "intent to credit" the full transaction amount to the cardholder through the Rapid Dispute Resolution (RDR) process.

**Important:**

This document provides information regarding RDR for reference purposes. Contact your Visa representative to know if the RDR functionality is already implemented and available in your region.

**Rapid Dispute Resolution Process**

In the first purchase inquiry request, VROL queries the merchant for purchase information. Merchant returns a response with the requested purchase information.

If the issuer decides to submit a dispute including a TC 40 (where applicable), VROL performs another request to the merchant. This time however, VROL sends a dispute notification alert along with an indicator specifying whether the merchant is eligible to participate in the RDR process or not.

The merchant indicates in their response if they intend to credit the cardholder. VROL temporarily blocks the dispute and places it in the All Submitted Rapid Dispute Resolution queue. The merchant can credit the cardholder within the configurable window to avoid further dispute submission. VROL automatically searches VisaNet to determine if a TC 06 (Credit Voucher) that matches the dispute in the All Submitted Rapid Dispute Resolution queue was sent to the acquirer. If the merchant does not credit the cardholder within the configurable window, VROL automatically proceeds with the dispute submission.

If the merchant provides CreditTranID, then it is saved and used during ATR search. VROL checks if the CreditTranID exists or not. If it does not exist, then VROL returns to the merchant for credit inquiry.

**Note:** *VROL places all active disputes, regardless on whether they are resolved or still on hold, in the All Submitted Rapid Dispute Resolution queue.*

A transaction is eligible for RDR only if both the merchant (or in other cases, a 3rd party service provider) and issuer are configured to participate in RDR. Only the Visa Administrator can configure the merchant or 3rd party service provider's participation in the RDR process.

The RTSI issuer can check the details of dispute transactions eligible for RDR.

1. To check if RDR information is available or not for participating merchants, the member host system triggers a SISubmitDisputeQuestionnaireRequest message supplying the VisaCaseNumber and Disputelid.

VROL returns a SISubmitDisputeQuestionnaireResponse message with <IsRdrInfoAvailable>, indicating whether RDR information is available or not for participating merchants.

2. To retrieve active RDR-eligible disputes from the All Submitted Rapid Dispute Resolution queue, the member host system triggers a SiGetQueueRequest message. See [Chapter 17, Queues](#).

VROL returns a list of active RDR-eligible dispute cases (resolved or on hold) in the queue including the VisaCaseNumber for each case.

3. To retrieve details of the RDR-eligible dispute, the member host system triggers a SIGetRdrDetailsRequest message supplying the Disputelid.

VROL returns a SIGetRdrDetailsResponse message with the Disputelid along with the details of the RDR-eligible dispute.

## Submit Disputes Using Fraud on Authorization Transactions

Visa Risk Manager (VRM) sends the following Fraud on Authorization transactions to VROL:

- Transactions that are marked as Fraud by the VRM system and are confirmed to be fraudulent by the cardholder.
- The above transactions for which the cardholder has opted for "Send Confirmed Fraud Info to Visa". Fraud reports are created only for these transactions.

### Important:

VRM to VROL Fraud reports and subsequent functionalities are dependent on the new VRM User Interface (UI), with migrations to the new UI occurring over the next several months. Please contact [VAA\\_VRM@visa.com](mailto:VAA_VRM@visa.com) if you have questions about migration to the new VRM UI.

If the Fraud on Authorization transaction from VRM qualifies for dispute processing in VROL, VROL creates the fraud report and either creates a dispute automatically or allows the issuer to submit a dispute on these transactions, depending on the issuing member configuration. VROL creates cases for Fraud on Authorization transactions depending on the member configuration.

The Visa Administrator or Member Site Administrator can configure the necessary 'Fraud-on-Auth Disputes' options when setting member preferences for the issuer's organization.

For more information about submitting disputes using Fraud on Authorization transactions from VRM, see *Visa Resolve Online User's Guide*.

## Dispute Operations

RTSI exposes the following dispute operations:

**SISubmitDisputeFromTransactionOrCaseRequest**—Used to initiate a dispute from a Transaction Inquiry (TI) or from an existing case.

**SISubmitDisputeQuestionnaireRequest**—Used to submit the dispute questionnaire.

**SIGetDisputeDetailsRequest**—Used to retrieve details of a submitted or received dispute.

**SISubmitFraudBundleDisputeQuestionnaireRequest**—Used to submit the fraud bundle dispute questionnaire.

**SIGetFraudBundleDisputeQuestionnaireStatusRequest**—Used to retrieve details of a submitted fraud bundle dispute.

**SICreateDisputeResponseRequest**—Used to respond to a received collaboration dispute.

**SIGetDisputeResponseDetailsRequest**—Used to retrieve details of a submitted or received Dispute Response.

**SIGetAcceptanceNotificationDetailsRequest**—Used to retrieve the acceptance notification details for disputes or dispute responses that were accepted and closed.

**SICChangeDisputeStatusRequest**—Used to recall a submitted dispute item or delete a saved dispute item. The dispute item can either be a dispute or a dispute response.

**SIGetRdrDetailsRequest**—Used to retrieve details of dispute transactions eligible for RDR.

## SIIInitiateDisputeFromTransactionOrCaseRequest

SIIInitiateDisputeFromTransactionOrCaseRequest allows RTSI users to initiate a dispute from a Transaction Inquiry (TI) or from an existing case.

The following elements should be considered when initiating a dispute or receiving a response from this service.

Element	Description
VisaCaseNumber	<p>Case tracking number assigned by VROL to uniquely identify a case in VROL.</p> <p>When initiating a dispute from an existing case, provide only the VisaCaseNumber. A new case is created if VisaCaseNumber is not supplied in the request.</p>
RoITransactionId	<p>Uniquely identifies an original transaction retrieved from VROL's Transaction Inquiry data repository.</p> <p>This element is required when initiating a dispute from a selected original transaction.</p>
DisputeCategory	<p>Indicates the category of the dispute.</p> <p>Domain values:</p> <ul style="list-style-type: none"> <li>• 10=Fraud</li> <li>• 11=Authorization</li> <li>• 12=Processing Error</li> <li>• 13=Consumer</li> </ul>
DisputeId	A unique VROL-generated ID for each initiated dispute.
AssociatedTransGroupId	Identifies a group of transactions associated to the selected transaction, and is used to access the associated transaction detail information.
FraudReportID	Uniquely identifies a fraud report. Member can use this to access the fraud report information in a subsequent request.
ATRSelectionReqdInd	<p>Indicates in the response if the user is required to select associated transactions.</p> <p>Domain values:</p> <ul style="list-style-type: none"> <li>• Required=Indicates that ATR selection is required.</li> <li>• NotRequired=Indicates that ATR selection is not required.</li> <li>• Unknown=Indicates that VROL was not able to determine the correct status yet as the query is still in progress. (SIGetDisputeDetailsRequest can be submitted to check the status.)</li> </ul>

Element	Description
FraudRptInd	<p>Indicates in the response if a Fraud Report is required, and already exists or not.</p> <p>Domain values:</p> <ul style="list-style-type: none"> <li>• Exist=Indicates that a Fraud Report exists.</li> <li>• NotExist=Indicates that a Fraud Report does not exist.</li> <li>• Unknown=Indicates that VROL was not able to determine the correct status yet as the query is still in progress. (SIGetDisputeDetailsRequest can be submitted to check the status.)</li> </ul> <p>This element is applicable only for the Fraud dispute category and will not be present in the response if other dispute categories are supplied.</p>
PPCSInd	<p>Indicates in the response if a Stop Payment Order is optional, and already exists or not.</p> <p>Domain values:</p> <ul style="list-style-type: none"> <li>• Exist=Indicates that a Stop Payment Order exists.</li> <li>• NotExist=Indicates that a Stop Payment Order does not exist.</li> <li>• Unknown=Indicates that VROL was not able to determine the correct status yet as the query is still in progress. (SIGetDisputeDetailsRequest can be submitted to check the status.)</li> </ul> <p>This element is applicable only for the Consumer dispute category and will not be present in the response if other dispute categories are supplied.</p>

Element	Description
EFLInd	<p>Indicates in the response if an Exception File Listing is optional, and already exists or not.</p> <p>Domain values:</p> <ul style="list-style-type: none"> <li>• Exist=Indicates that an Exception File Listing exists.</li> <li>•NotExist=Indicates that an Exception File Listing does not exist.</li> <li>• Unknown=Indicates that VROL was not able to determine the correct status yet as the query is still in progress. (SIGetDisputeDetailsRequest can be submitted to check the status.)</li> </ul> <p>This element is applicable only for the Fraud dispute category and will not be present in the response if other dispute categories are supplied.</p>
AEMInd	<p>Indicates in the response if a Provisional Credit based on VROL Member Configuration is required/optional, and already exists or not.</p> <p>Domain values:</p> <ul style="list-style-type: none"> <li>• ExistPartialOptional=Indicates that a partial Provisional Credit exists and it is optional for the member to submit one.</li> <li>• ExistPartialRequired=Indicates that a partial Provisional Credit exists and it is required for the member to submit one.</li> <li>• Exist=Indicates that an exact Provisional Credit exists.</li> <li>• NotExistOptional=Indicates that a Provisional Credit does not exist and it is optional for the member to submit one.</li> <li>• NotExistRequired=Indicates that a Provisional Credit does not exist and it is required for the member to submit one.</li> <li>• NotConfigured=Indicates that the member is not configured to submit a Provisional Credit with their dispute.</li> <li>• Unknown=Indicates that VROL was not able to determine the correct status yet as the query is still in progress. (SIGetDisputeDetailsRequest can be submitted to check the status.)</li> </ul> <p>This element is applicable for all the dispute categories.</p>

SIIInitiateDisputeFromTransactionOrCase contains request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```

<SIIInitiateDisputeFromTransactionOrCaseRequest>
    <RequestHeader>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <MemberCaseNumber>555555555555</MemberCaseNumber>
        <DisputeCategory>13</DisputeCategory>
        <DisputeAmount currency="840">100</DisputeAmount>
    </requestData>
</SIIInitiateDisputeFromTransactionOrCaseRequest>

```

### Response Message—SOAP-based Services

```
<SIInitiateDisputeFromTransactionOrCaseResponse>
  <Status>
    <Code>I-300000000</Code>
    <Message>Successfully completed Operation.</Message>
  </Status>
  <responseData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <DisputeId>123456</DisputeId>
    <FraudReportID>111111</FraudReportID>
    <FraudRptInd>Exist</FraudRptInd>
    <PPCSIInd>NotExist</PPCSIInd>
    <AEMIInd>NotConfigured</AEMIInd>
    <ATRSelectionReqdInd>NotRequired</ATRSelectionReqdInd>
    <AssociatedTransGroupId>123456</AssociatedTransGroupId>
  </responseData>
</SIInitiateDisputeFromTransactionOrCaseResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": {
    "MemberRole": "I"
  },
  "requestData": {
    "VisaCaseNumber": 1000000000,
    "MemberCaseNumber": 555555555555,
    "DisputeCategory": {
      "value": "13"
    }
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": {
    "Code": "I-300000000",
    "Message": "Successfully completed Operation."
  },
  "responseData": {
    "VisaCaseNumber": 1000000000,
    "DisputeId": 123456,
    "FraudReportID": 111111,
    "FraudRptInd": "Exist",
    "PPCSIInd": "NotExist",
    "AEMIInd": "NotConfigured",
    "ATRSelectionReqdInd": "NotRequired",
    "AssociatedTransGroupId": 123456
  }
}
```

## SISubmitDisputeQuestionnaireRequest

SISubmitDisputeQuestionnaireRequest allows RTSI users to submit or save the dispute questionnaire. This request can include image(s).

The following elements should be considered when submitting a dispute questionnaire or receiving a response from this service.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.
DisputeId	A unique VROL-generated ID for each saved or submitted dispute.
FraudReportID	Uniquely identifies a fraud report. Member can use this to access the fraud report information in a subsequent request.
IsRdrInfoAvailable	Conveys to the issuer whether Rapid Dispute Resolution (RDR) information is available or not for merchants participating in RDR.

Questionnaire actions are Save and Submit. See [Chapter 3, RTSI Request Operations](#) for a description of these operations and actions.

SISubmitDisputeQuestionnaire contains request and response messages as shown in the following sample messages.

### Request Message—Fraud—SOAP-based Services

```

<SISubmitDisputeQuestionnaireRequest>
    <RequestHeader>
        <User id="abcde1234f" type="internalId"/>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <DisputeId>123456</DisputeId>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <MemberCaseNumber>555555555555</MemberCaseNumber>
        <DisputeAmount currency="840">.50</DisputeAmount>
        <ExceptionCPD>2019-06-13</ExceptionCPD>
        <Note>note</Note>
        <DisputeAmountChangeReason>reason
        </DisputeAmountChangeReason>
        <CreateDisputeFinancial>Y</CreateDisputeFinancial>
        <TransactionMessageFormat>B</TransactionMessageFormat>
        <Fraud>
            <CardholderDeniesAuthorizingInd>true
            </CardholderDeniesAuthorizingInd>
        </Fraud>
        <IsParcelado>N</IsParcelado> [Valid for Brazil domestic jurisdiction only]
        <Action>Submit</Action>
    </requestData>
</SISubmitDisputeQuestionnaireRequest>
```

### Response Message—Fraud—SOAP-based Services

```
<SISubmitDisputeQuestionnaireResponse>
    <Status>
        <Code>I-920000000</Code>
        <Message>The dispute has been allocated at 100.00% acquirer liability.</Message>
    </Status>
    <responseData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <DisputeId>123456</DisputeId>
        <FraudReportID>1000000</FraudReportID>
        <DisputeFinancialInfo>
            <VROLFinancialSentInd>true</VROLFinancialSentInd>
            <VROLFinancialID>10.4-9111</VROLFinancialID>
            <FinancialTransactionType>DISPUTE_FINANCIAL</FinancialTransactionType>
            <FinancialAmt currency="840">16706.50</FinancialAmt>
            <DisputeCategory>10</DisputeCategory>
            <DisputeCondition>4</DisputeCondition>
            <DisputeId>111111</DisputeId>
            <TransactionID>123456789012345</TransactionID>
            <IssuerCaseNumber>1234567</IssuerCaseNumber>
            <OpposerExpectedRespDate>2019-06-13</OpposerExpectedRespDate>
        </DisputeFinancialInfo>
    </responseData>
</SISubmitDisputeQuestionnaireResponse>
```

### Request Message—Authorization—SOAP-based Services

```
<SISubmitDisputeQuestionnaireRequest>
    <RequestHeader>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <DisputeId>123456</DisputeId>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <DisputeAmount currency="840">.50</DisputeAmount>
        <DisputeCardholderInfo/>
        <CreateDisputeFinancial>Y</CreateDisputeFinancial>
        <ExplanationOfCreditPresented>Explain why Credit presented does or does not apply</ExplanationOfCreditPresented>
        <Authorization>
            <ExplanationOfAuthorizationsPresented>explanation</ExplanationOfAuthorizationsPresented>
        </Authorization>
        <IssuerAcquirerContactInfo>
            <ContactName>John Smith</ContactName>
            <ContactPhone>1-100-200-3111</ContactPhone>
        </IssuerAcquirerContactInfo>
        <Action>Submit</Action>
        <SupplyDocInd>false</SupplyDocInd>
    </requestData>
</SISubmitDisputeQuestionnaireRequest>
```

**Response Message—Authorization—SOAP-based Services**

```
<SISubmitDisputeQuestionnaireResponse>
  <Status>
    <Code>I-920000000</Code>
    <Message>The dispute has been allocated at 100.00%
      acquirer liability.</Message>
  </Status>
  <ResponseData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <DisputeId>123456</DisputeId>
    <DisputeFinancialInfo>
      <VROLFinancialSentInd>true</VROLFinancialSentInd>
      <VROLFinancialID>11.3-9111</VROLFinancialID>
      <FinancialTransactionType>DISPUTE_FINANCIAL
      </FinancialTransactionType>
      <FinancialAmt currency="840">16717.50</FinancialAmt>
      <DisputeCategory>11</DisputeCategory>
      <DisputeCondition>3</DisputeCondition>
      <DisputeId>111111</DisputeId>
      <TransactionID>123456789012345</TransactionID>
      <IssuerCaseNumber>1234567</IssuerCaseNumber>
      <OpposerExpectedRespDate>2019-06-03
      </OpposerExpectedRespDate>
    </DisputeFinancialInfo>
  </ResponseData>
</SISubmitDisputeQuestionnaireResponse>
```

## Request Message—Processing Error—SOAP-based Services

```
<SISubmitDisputeQuestionnaireRequest>
  <RequestHeader>
    <MemberRole>I</MemberRole>
  </RequestHeader>
  <requestData>
    <DisputeId>123456</DisputeId>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <MemberCaseNumber>55555555555</MemberCaseNumber>
    <DisputeAmount currency="344">300</DisputeAmount>
    <ExceptionCPD>2019-06-13</ExceptionCPD>
    <DisputeAmountChangeReason>test</DisputeAmountChangeReason>
    <CreateDisputeFinancial>Y</CreateDisputeFinancial>
    <TransactionMessageFormat>B</TransactionMessageFormat>
    <ProcessingError>
      <WhatIncorrectAboutTransaction>ID
      </WhatIncorrectAboutTransaction>
      <AuthorizationRequestDeclinedValidDataInd>true
      </AuthorizationRequestDeclinedValidDataInd>
      <ExplainWhyValidDataInclusionReason>explanation
      </ExplainWhyValidDataInclusionReason>
    </ProcessingError>
    <DisputeAttachmentDescriptor>
      <Attachment id="1">
        <ContentType>application/pdf</ContentType>
        <Comment>any comment</Comment>
        <ImageData>cid:100000000003</ImageData>
        <DocType>CALET</DocType>
      </Attachment>
    </DisputeAttachmentDescriptor>
    <Action>Submit</Action>
  </requestData>
</SISubmitDisputeQuestionnaireRequest>
```

**Response Message—Processing Error—SOAP-based Services**

```
<SISubmitDisputeQuestionnaireResponse>
    <Status>
        <Code>I-3000000000</Code>
        <Message>Successfully completed Operation.</Message>
    </Status>
    <responseData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <DisputeId>123456</DisputeId>
        <DisputeFinancialInfo>
            <VROLFInancialSentInd>true</VROLFInancialSentInd>
            <VROLFIncialID>12.7-9001</VROLFIncialID>
            <FinancialTransactionType>DISPUTE_FINANCIAL
            </FinancialTransactionType>
            <FinancialAmt currency="344">300</FinancialAmt>
            <DisputeCategory>12</DisputeCategory>
            <DisputeCondition>7</DisputeCondition>
            <DisputeId>111111</DisputeId>
            <TransactionID>123456789012345</TransactionID>
            <IssuerCaseNumber>1234567</IssuerCaseNumber>
            <OpposerExpectedRespDate>2019-03-29
            </OpposerExpectedRespDate>
        </DisputeFinancialInfo>
    </responseData>
</SISubmitDisputeQuestionnaireResponse>
```

### Request Message—Consumer—SOAP-based Services

```
<SISubmitDisputeQuestionnaireRequest>
  <RequestHeader>
    <MemberRole>I</MemberRole>
  </RequestHeader>
  <requestData>
    <DisputeId>123456</DisputeId>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <MemberCaseNumber>55555555555</MemberCaseNumber>
    <DisputeAmount currency="840">.50</DisputeAmount>
    <Note>any note</Note>
    <DisputeAmountChangeReason>reason
    </DisputeAmountChangeReason>
    <CreateDisputeFinancial>Y</CreateDisputeFinancial>
    <ExplanationOfCreditPresented>Explain why Credit presented
    does or does not apply</ExplanationOfCreditPresented>
    <ExceptionCPD>2019-06-13</ExceptionCPD>
    <Consumer>
      <DisputeDueTo>NR</DisputeDueTo>
      <CardholderAttemptToResolve>Y</CardholderAttemptToResolve>
      <WhatWasNotReceived>M</WhatWasNotReceived>
      <PurchasedInfo>shoes</PurchasedInfo>
      <ExpectedReceiptDateTime>2019-03-01T23:59:00-06:00
      </ExpectedReceiptDateTime>
      <DidCardholderCancelPriorExpectedDate>N
      </DidCardholderCancelPriorExpectedDate>
      <MerchandiseDeliveredWrongLocation>N
      </MerchandiseDeliveredWrongLocation>
    </Consumer>
    <Action>Submit</Action>
  </requestData>
</SISubmitDisputeQuestionnaireRequest>
```

**Response Message—Consumer—SOAP-based Services**

```
<SISubmitDisputeQuestionnaireResponse>
    <Status>
        <Code>I-920000001</Code>
        <Message>No associated transactions found.</Message>
    </Status>
    <ResponseData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <DisputeId>123456</DisputeId>
        <DisputeFinancialInfo>
            <VROLFinancialSentInd>true</VROLFinancialSentInd>
            <VROLFinancialID>13.1-9111</VROLFinancialID>
            <FinancialTransactionType>DISPUTE_FINANCIAL
            </FinancialTransactionType>
            <FinancialAmt currency="840">59.65</FinancialAmt>
            <DisputeCategory>13</DisputeCategory>
            <DisputeCondition>1</DisputeCondition>
            <DisputeId>111111</DisputeId>
            <TransactionID>123456789012345</TransactionID>
            <IssuerCaseNumber>3001</IssuerCaseNumber>
            <OpposerExpectedRespDate>2019-07-28
            </OpposerExpectedRespDate>
            <IsRdrInfoAvailable>Y<IsRdrInfoAvailable>
        </DisputeFinancialInfo>
    </ResponseData>
</SISubmitDisputeQuestionnaireResponse>
```

**Request Message—Fraud—REST-based Services**

```
{
    "RequestHeader": {
        "MemberRole": "I"
    },
    "requestData": {
        "DisputeId": "123456",
        "VisaCaseNumber": "1000000000",
        "MemberCaseNumber": "555555555555",
        "DisputeFraudInfo": {
            "FraudType": "0"
        },
        "DisputeAmountChangeReason": "Test",
        "CreateDisputeFinancial": "N",
        "ExceptionCPD": "2019-06-13",
        "Fraud": {
            "CardholderDeniesAuthorizingInd": "true"
        },
        "Action": "Submit"
    }
}
```

### Response Message—Fraud—REST-based Services

```
{  
    "Status": [ {  
        "Code": "I-3000000000",  
        "Message": "Successfully completed Operation."  
    } ],  
    "ResponseData": {  
        "VisaCaseNumber": 1000000000,  
        "DisputeId": 12345,  
        "FraudReportID": 1600000,  
        "DisputeFinancialInfo": {  
            "VROLFinancialSentInd": "false",  
            "VROLFinancialID": "12.7.7146",  
            "FinancialTransactionType": "SALES",  
            "FinancialAmt": {  
                "value": 23.44,  
                "currency": "344"  
            },  
            "DisputeCategory": { "value": "10" },  
            "DisputeCondition": 2,  
            "DisputeId": 11111,  
            "TransactionID": 1000000000000000,  
            "IssuerCaseNumber": "1234567",  
            "OpposerExpectedRespDate": "2019-06-26"  
            "IsRdrInfoAvailable": "Y"  
        }  
    }  
}
```

**Request Message—Authorization—REST-based Services**

```
{  
  "requestData": {  
    "Action": "Submit",  
    "VisaCaseNumber": "1000000000",  
    "DisputeAmount": {  
      "currency": "344",  
      "value": "10"  
    },  
    "DisputeAmountChangeReason": "TEST",  
    "ExceptionCPD": "2019-06-13",  
    "CreateDisputeFinancial": "Y",  
    "DisputeAttachmentDescriptor": {  
      "Attachment": [  
        {  
          "ContentType": "image/tiff",  
          "Comment": "image is proof of fraud",  
          "ImageData": "image1",  
          "id": "1",  
          "DocType": "CALET"  
        }  
      ]  
    },  
    "Authorization": {}  
  },  
  "DisputeId": "12345",  
  "MemberCaseNumber": "555555555555",  
}  
}
```

### Response Message—Authorization—REST-based Services

```
{  
    "Status": [ {  
        "Code": "I-3000000000",  
        "Message": "Successfully completed Operation."  
    } ],  
    "responseData": {  
        "VisaCaseNumber": 1000000000,  
        "DisputeId": 12345,  
        "DisputeFinancialInfo": {  
            "VROLFinancialSentInd": true,  
            "VROLFinancialID": "11.3-9197",  
            "FinancialTransactionType": "DISPUTE_FINANCIAL",  
            "FinancialAmt": {  
                "value": 16706.5,  
                "currency": "840"  
            },  
            "DisputeCategory": { "value": "11" },  
            "DisputeCondition": "3",  
            "DisputeId": 11111,  
            "TransactionID": "1000000000000000",  
            "IssuerCaseNumber": "MCN-Test",  
            "OpposerExpectedRespDate": "2019-06-26"  
            "IsRdrInfoAvailable": "Y"  
        }  
    }  
}
```

**Request Message—Processing Error—REST-based Services**

```
{  
  "RequestHeader": {  
    "MemberRole": "I"  
  },  
  "requestData": {  
    "Action": "Submit"  
    "VisaCaseNumber": "123456789",  
    "DisputeAmount": {  
      "currency": "344",  
      "value": "23.44"  
    },  
    "DisputeAmountChangeReason": "Test",  
    "ProcessingError": {  
      "WhatIncorrectAboutTransaction": "D",  
      "BothTransactionsOnSameAcctNumberInd": "N",  
      "IssuerOrCardholderOtherAcctWithDiffVisaCardInd": "N",  
      "IssuerEvidenceMerchantPassedOnFundsInd": "N",  
      "ProvidingPaymentProofOtherMeans": "V",  
      "CardholderAttemptToResolve": "N",  
      "AttemptToResolveProhLocalLaw": "Y",  
      "OtherTranSameMerchantInd": "N"  
    },  
    "DisputeId": "12345",  
    "MemberCaseNumber": "DSPT-Test",  
    "CreateDisputeFinancial": "Y"  
  }  
}
```

### Response Message—Processing Error—REST-based Services

```
{  
    "Status": [ {  
        "Code": "I-3000000000",  
        "Message": "Successfully completed Operation."  
    } ],  
    "ResponseData": {  
        "VisaCaseNumber": 1000000000,  
        "DisputeId": 12345,  
        "DisputeFinancialInfo": {  
            "VROLFinancialSentInd": true,  
            "VROLFinancialID": "12.6.1-9092",  
            "FinancialTransactionType": "DISPUTE_FINANCIAL",  
            "FinancialAmt": {  
                "value": 23.44,  
                "currency": "344"  
            },  
            "DisputeCategory": { "value": "12" },  
            "DisputeCondition": "6.1",  
            "DisputeId": 65131,  
            "TransactionID": "1000000000000000",  
            "IssuerCaseNumber": "DSPT-Test",  
            "OpposerExpectedRespDate": "2019-06-26"  
            "IsRdrInfoAvailable": "Y"  
        }  
    }  
}
```

### Request Message—Consumer—REST-based Services

```
{  
    "RequestHeader": {  
        "MemberRole": "I"  
    },  
    "requestData": {  
        "DisputeId": "12345",  
        "VisaCaseNumber": "1000000000",  
        "DisputeAmount": {  
            "Value": "479.90",  
            "Currency": "344"  
        },  
        "CreateDisputeFinancial": "Y",  
        "Consumer": {  
            "DisputeDueTo": "CS",  
            "MerchandiseOrServices": "Services",  
            "DidCardholderCancel": "N",  
            "ServiceType": "T",  
            "TimeshareDate": "2019-02-01"  
        },  
        "Action": "Submit"  
    }  
}
```

## Response Message—Consumer—REST-based Services

```
{  
    "Status": [{  
        "Code": "I-920000001",  
        "Message": "No associated transactions found."  
    }],  
    "ResponseData": {  
        "VisaCaseNumber": 1000000000,  
        "DisputeId": 12345,  
        "DisputeFinancialInfo": {  
            "VROLFinancialSentInd": true,  
            "VROLFinancialID": "13.7-9089",  
            "FinancialTransactionType": "DISPUTE_FINANCIAL",  
            "FinancialAmt": {  
                "value": 479.9,  
                "currency": "344"  
            },  
            "DisputeCategory": {"value": "13"},  
            "DisputeCondition": "7",  
            "DisputeId": 12345,  
            "TransactionID": 1000000000000000,  
            "OpposerExpectedRespDate": "2019-06-30"  
            "IsRdrInfoAvailable": "Y"  
        }  
    }  
}
```

## SIGetDisputeDetailsRequest

SIGetDisputeDetailsRequest allows RTSI users to request for the dispute questionnaire data to be returned in image format, as well as in XML. The returned image will be in TIFF or PDF format. To retrieve the dispute as an image, the user must set <IncludeDisputeAsImageInd>=true. The image data will be returned in the <Image> group in the SIGetDisputeDetailsResponse.

**Note:** Only the dispute questionnaire will be returned as an image. Any attached documents must be retrieved using the AttachmentId value(s) in this response as input to the DocId field(s) of SIGetImageRequest.

**Note:** Any image attached to the dispute questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.

SIGetDisputeDetails contains request and response messages as shown in the following sample messages.

**Note:** See the table in [SIInitiateDisputeFromTransactionOrCaseRequest](#) for descriptions and domain values of indicators returned in the SIGetDisputeDetailsResponse.

## Request Message—Fraud—SOAP-based Services

```
<SIGetDisputeDetailsRequest>
  <RequestHeader>
    <MemberRole>I</MemberRole>
  </RequestHeader>
  <requestData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <DisputeId>1234</DisputeId>
  </requestData>
</SIGetDisputeDetailsRequest>
```

## Response Message—Fraud—SOAP-based Services

```
<SIGetDisputeDetailsResponse>
  <Status>
    <Code>I-300000000</Code>
    <Message>Successfully completed Operation.</Message>
  </Status>
  <responseData>
    <TransactionInfo>
      <AccountNumber>41000000000000000000</AccountNumber>
      <Transaction id="0000000000000000">
        <TranDate>2019-05-24</TranDate>
        <Amount currency="840">3000.00</Amount>
      </Transaction>
      <MerchantName>MERCHANT NAME</MerchantName>
      <MerchantCity>MERCHANT CITY</MerchantCity>
      <MerchantCountrySubEntityCode>CA
      </MerchantCountrySubEntityCode>
      <MerchantCountryCode>MU</MerchantCountryCode>
      <MerchantPostalCode>94402</MerchantPostalCode>
      <MCC>4111</MCC>
      <ARN>00000000000000000000000000000001</ARN>
      <ECMOTO>1</ECMOTO>
      <ReimbursementAttribute>A</ReimbursementAttribute>
      <CPD>2019-05-24</CPD>
      <TransactionCategory>D</TransactionCategory>
      <AcquirerBID>00000001</AcquirerBID>
      <NetworkID>0002</NetworkID>
      <AcquirerBIN>000002</AcquirerBIN>
      <RetrievalReferenceNumber>000000000001
      </RetrievalReferenceNumber>
      <SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
      <CardholderName>John</CardholderName>
      <TranDestAmt currency="840">3000.00</TranDestAmt>
      <ProcessingCode>00</ProcessingCode>
      <IssuerName>issuer name</IssuerName>
      <AcquirerName>acquirer name</AcquirerName>
      <AccountType>0</AccountType>
      <DisputeJurisdiction jurisdictionCd="DOM">
        <Jurisdiction>DOMESTIC-MU</Jurisdiction>
        <IssuerRegion>CEMEA</IssuerRegion>
        <AcquirerRegion>CEMEA</AcquirerRegion>
        <CountryCd>MU</CountryCd>
      </DisputeJurisdiction>
    </TransactionInfo>
  </responseData>
</SIGetDisputeDetailsResponse>
```

```
<VisaCaseNumber>1000000000</VisaCaseNumber>
<DisputeId>1234</DisputeId>
<FraudReportID>1234567</FraudReportID>
<FraudRptInd>Exist</FraudRptInd>
<EFLInd>NotExist</EFLInd>
<AEMInd>NotExistOptional</AEMInd>
<DisputeCategory>10</DisputeCategory>
<DisputeCondition>4</DisputeCondition>
<DisputeCategoryCondition>10.4 - Fraud: Card Absent Environment
</DisputeCategoryCondition>
<DisputeAmount currency="840">3000.00</DisputeAmount>
<ExceptionCPD>2019-07-30</ExceptionCPD>
<DisputeCardholderInfo>
    <CardholderName>John</CardholderName>
    <CardholderAddress>Test Address</CardholderAddress>
    <CardholderCity>City</CardholderCity>
    <CardholderCountrySubEntityCode>CA
    </CardholderCountrySubEntityCode>
    <NameType>PN</NameType>
    <CardholderPostalCode>12345</CardholderPostalCode>
    <CardholderCountryCode>US</CardholderCountryCode>
    <CardholderPhoneNum>155130312316</CardholderPhoneNum>
    <CardholderFaxNum>020-222220000</CardholderFaxNum>
    <CardholderEmail>John@visa.com</CardholderEmail>
    <CardholderBestTimeCall>20190302 150010
    </CardholderBestTimeCall>
    <CardholderOptInInd>true</CardholderOptInInd>
    <CardholderPrefix>Mr</CardholderPrefix>
    <CardholderLastName>Name</CardholderLastName>
    <CardholderMiddleInitial>Z</CardholderMiddleInitial>
    <Address2>Test Address</Address2>
    <CardholderSuffix>jd</CardholderSuffix>
</DisputeCardholderInfo>
<CreateDisputeFinancial>Y</CreateDisputeFinancial>
<FraudReportedDate>2019-06-25</FraudReportedDate>
<FullMagStripeNotRead>true</FullMagStripeNotRead>
<CVVPresent>false</CVVPresent>
<Fraud>
    <CardholderDeniesAuthorizingInd>true
    </CardholderDeniesAuthorizingInd>
</Fraud>
<FraudType>0</FraudType>
<IssuerAcquirerContactInfo>
    <ContactName>name</ContactName>
</IssuerAcquirerContactInfo>
<AssociatedTransGroupId>123456</AssociatedTransGroupId>
<RecallInd>false</RecallInd>
<IsParcelado>N</IsParcelado> [Valid for Brazil domestic jurisdiction only]
</ResponseData>
</SIGetDisputeDetailsResponse>
```

### Request Message—Authorization—SOAP-based Services

```
<SIGetDisputeDetailsRequest>
  <RequestHeader>
    <MemberRole>I</MemberRole>
  </RequestHeader>
  <requestData>
    <VisaCaseNumber>1234567890</VisaCaseNumber>
    <DisputeId>12345</DisputeId>
    <IncludeDisputeAsImageInd>false</IncludeDisputeAsImageInd>
    <DownloadImageInd>false</DownloadImageInd>
  </requestData>
</SIGetDisputeDetailsRequest>
```

### Response Message—Authorization—SOAP-based Services

```
<SIGetDisputeDetailsResponse>
  <Status>
    <Code>I-300000000</Code>
    <Message>Successfully completed Operation.</Message>
  </Status>
  <responseData>
    <TransactionInfo>
      <AccountNumber>4100000000000000</AccountNumber>
      <Transaction id="0000000000000000">
        <TranDate>2019-05-24</TranDate>
        <Amount currency="840">16717.50</Amount>
      </Transaction>
      <MerchantName>MERCHANT NAME</MerchantName>
      <MerchantCity>MERCHANT CITY</MerchantCity>
      <MerchantCountrySubEntityCode>CA
      </MerchantCountrySubEntityCode>
      <MerchantCountryCode>US</MerchantCountryCode>
      <MerchantPostalCode>94407</MerchantPostalCode>
      <MCC>5411</MCC>
      <ARN>00000000000000000001</ARN>
      <ECMOTO>1</ECMOTO>
      <ReimbursementAttribute>A</ReimbursementAttribute>
      <CPD>2019-05-24</CPD>
      <TransactionCategory>D</TransactionCategory>
      <AcquirerBID>00000001</AcquirerBID>
      <NetworkID>0002</NetworkID>
      <AcquirerBIN>000002</AcquirerBIN>
      <RetrievalReferenceNumber>000000000001
      </RetrievalReferenceNumber>
      <SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
      <TranDestAmt currency="840">16717.50</TranDestAmt>
      <ProcessingCode>00</ProcessingCode>
      <IssuerName>issuer name</IssuerName>
      <AcquirerName>acquirer name</AcquirerName>
      <AccountType>0</AccountType>
      <DisputeJurisdiction jurisdictionCd="DOM">
        <Jurisdiction>DOMESTIC-US</Jurisdiction>
        <IssuerRegion>US</IssuerRegion>
        <AcquirerRegion>US</AcquirerRegion>
        <CountryCd>US</CountryCd>
      </DisputeJurisdiction>
    </TransactionInfo>
  </responseData>
</SIGetDisputeDetailsResponse>
```

```
</TransactionInfo>
<VisaCaseNumber>1000000000</VisaCaseNumber>
<DisputeId>1234</DisputeId>
<EFLInd>NotExist</EFLInd>
<AEMInd>NotConfigured</AEMInd>
<MemberCaseNumber>MCN-DSPT RTSI</MemberCaseNumber>
<DisputeCategory>11</DisputeCategory>
<DisputeCondition>3</DisputeCondition>
<DisputeCategoryCondition>11.3 - Authorization:
No Authorization</DisputeCategoryCondition>
<DisputeAmount currency="840">16717.50</DisputeAmount>
<DisputeCardholderInfo>
    <CardholderCountryCode>US</CardholderCountryCode>
    <CardholderOptInInd>false</CardholderOptInInd>
</DisputeCardholderInfo>
<Note>test</Note>
<CreateDisputeFinancial>Y</CreateDisputeFinancial>
<Authorization>
    <AuthorizationNotObtained>false</AuthorizationNotObtained>
    <FullAuthorizationNotObtained>false
    </FullAuthorizationNotObtained>
    <AuthorizationNotObtainedForCardType>false
    </AuthorizationNotObtainedForCardType>
</Authorization>
<IssuerAcquirerContactInfo>
    <ContactName>name</ContactName>
</IssuerAcquirerContactInfo>
<AssociatedTransGroupId>123456</AssociatedTransGroupId>
<RecallInd>false</RecallInd>
<IsParcelado>N</IsParcelado> [Valid for Brazil domestic jurisdiction only]
</ResponseData>
</SIGetDisputeDetailsResponse>
```

### Request Message—Processing Error-SOAP-based Services

```
<SIGetDisputeDetailsRequest>
    <RequestHeader>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1234567890</VisaCaseNumber>
        <DisputeId>12345</DisputeId>
        <IncludeDisputeAsImageInd>false</IncludeDisputeAsImageInd>
        <DownloadImageInd>false</DownloadImageInd>
    </requestData>
</SIGetDisputeDetailsRequest>
```

### Response Message—Processing Error—SOAP-based Services

```
<SIGetDisputeDetailsResponse>
    <Status>
        <Code>I-3000000000</Code>
        <Message>Successfully completed Operation.</Message>
    </Status>
    <responseData>
        <TransactionInfo>
```

```
<AccountNumber>4100000000000000</AccountNumber>
<Transaction id="0000000000000000">
    <TranDate>2019-05-24</TranDate>
    <Amount currency="840">22.31</Amount>
</Transaction>
<MerchantName>MERCHANT NAME</MerchantName>
<MerchantCity>MERCHANT CITY</MerchantCity>
<MerchantCountrySubEntityCode>CA
</MerchantCountrySubEntityCode>
<MerchantCountryCode>US</MerchantCountryCode>
<MerchantPostalCode>94419</MerchantPostalCode>
<MCC>5411</MCC>
<ARN>00000000000000000001</ARN>
<ECMOTO>1</ECMOTO>
<ReimbursementAttribute>A</ReimbursementAttribute>
<CPD>2019-05-24</CPD>
<TransactionCategory>D</TransactionCategory>
<AcquirerBID>00000001</AcquirerBID>
<NetworkID>0002</NetworkID>
<AcquirerBIN>000002</AcquirerBIN>
<RetrievalReferenceNumber>000000000001
</RetrievalReferenceNumber>
<SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
<TranDestAmt currency="840">22.31</TranDestAmt>
<ProcessingCode>00</ProcessingCode>
<IssuerName>issuer name</IssuerName>
<AcquirerName>acquirer name</AcquirerName>
<AccountType>0</AccountType>
<DisputeJurisdiction jurisdictionCd="DOM">
    <Jurisdiction>DOMESTIC-US</Jurisdiction>
    <IssuerRegion>US</IssuerRegion>
    <AcquirerRegion>US</AcquirerRegion>
    <CountryCd>US</CountryCd>
</DisputeJurisdiction>
</TransactionInfo>
<VisaCaseNumber>1000000000</VisaCaseNumber>
<DisputeId>1234</DisputeId>
<AEMInd>NotConfigured</AEMInd>
<DisputeCategory>12</DisputeCategory>
<DisputeCondition>6.1</DisputeCondition>
<DisputeCategoryCondition>12.6 - Processing Error: Duplicate
    Processing</DisputeCategoryCondition>
<DisputeAmount currency="840">22.31</DisputeAmount>
<ExceptionCPD>2019-06-13</ExceptionCPD>
<ProcessingError>
    <WhatIncorrectAboutTransaction>D</WhatIncorrectAboutTransaction>
    <BothTransactionsOnSameAcctNumberInd>N
    </BothTransactionsOnSameAcctNumberInd>
    <IssuerOrCardholderOtherAcctWithDiffVisaCardInd>N
    </IssuerOrCardholderOtherAcctWithDiffVisaCardInd>
    <ProvidingPaymentProofOtherMeans>C
    </ProvidingPaymentProofOtherMeans>
    <OtherTransSameMerchantInd>Y</OtherTransSameMerchantInd>
</ProcessingError>
<IssuerAcquirerContactInfo>
    <ContactName>name</ContactName>
</IssuerAcquirerContactInfo>
```

```
<AssociatedTransGroupId>123456</AssociatedTransGroupId>
<RecallInd>false</RecallInd>
<IsParcelado>N</IsParcelado> [Valid for Brazil domestic jurisdiction only]
<DisputeImageAttachment>
    <Attachment id="1234567">
        <ContentType>image/tiff</ContentType>
        <Comment>any comment</Comment>
        <DocType>CALET</DocType>
    </Attachment>
</DisputeImageAttachment>
</ResponseData>
</SIGetDisputeDetailsResponse>
```

### Request Message—Consumer-SOAP-based Services

```
<SIGetDisputeDetailsRequest>
    <RequestHeader>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1234567890</VisaCaseNumber>
        <DisputeId>1234</DisputeId>
        <IncludeDisputeAsImageInd>false</IncludeDisputeAsImageInd>
        <DownloadImageInd>false</DownloadImageInd>
    </requestData>
</SIGetDisputeDetailsRequest>
```

### Response Message—Consumer—SOAP-based Services

```
<SIGetDisputeDetailsResponse>
    <Status>
        <Code>I-300000000</Code>
        <Message>Successfully completed Operation.</Message>
    </Status>
    <responseData>
        <TransactionInfo>
            <AccountNumber>4100000000000000</AccountNumber>
            <Transaction id="0000000000000000">
                <TranDate>2019-05-24</TranDate>
                <Amount currency="840">16706.50</Amount>
            </Transaction>
            <MerchantName>MERCHANT NAME</MerchantName>
            <MerchantCity>MERCHANT CITY</MerchantCity>
            <MerchantCountrySubEntityCode>CA
            </MerchantCountrySubEntityCode>
            <MerchantCountryCode>US</MerchantCountryCode>
            <MerchantPostalCode>94407</MerchantPostalCode>
            <MCC>5411</MCC>
            <ARN>000000000000000000000001</ARN>
            <ECMOTO>1</ECMOTO>
            <ReimbursementAttribute>A</ReimbursementAttribute>
            <CPD>2019-05-24</CPD>
            <TransactionCategory>D</TransactionCategory>
            <AcquirerBID>00000001</AcquirerBID>
            <NetworkID>0002</NetworkID>
        </TransactionInfo>
    </responseData>
</SIGetDisputeDetailsResponse>
```

```
<AcquirerBIN>000002</AcquirerBIN>
<RetrievalReferenceNumber>000000000001
</RetrievalReferenceNumber>
<SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
<TranDestAmt currency="840">16706.50</TranDestAmt>
<ProcessingCode>00</ProcessingCode>
<IssuerName>issuer name</IssuerName>
<AcquirerName>acquirer name</AcquirerName>
<AccountType>0</AccountType>
<DisputeJurisdiction jurisdictionCd="DOM">
    <Jurisdiction>DOMESTIC-US</Jurisdiction>
    <IssuerRegion>US</IssuerRegion>
    <AcquirerRegion>US</AcquirerRegion>
    <CountryCd>US</CountryCd>
</DisputeJurisdiction>
</TransactionInfo>
<VisaCaseNumber>1000000000</VisaCaseNumber>
<DisputeId>1234</DisputeId>
<PPCSInd>NotExist</PPCSInd>
<AEMInd>NotExistOptional</AEMInd>
<MemberCaseNumber>a</MemberCaseNumber>
<DisputeCategory>13</DisputeCategory>
<DisputeCondition>1</DisputeCondition>
<DisputeCategoryCondition>13.1 - Consumer: Merchandise/Services Not Received</DisputeCategoryCondition>
<DisputeAmount currency="840">59.65</DisputeAmount>
<DisputeCardholderInfo>
    <CardholderCountryCode>US</CardholderCountryCode>
    <CardholderOptInInd>false</CardholderOptInInd>
</DisputeCardholderInfo>
<Note>notes</Note>
<DisputeAmountChangeReason>dispute amount partial</DisputeAmountChangeReason>
<CreateDisputeFinancial>Y</CreateDisputeFinancial>
<Consumer>
    <DisputeDueTo>NR</DisputeDueTo>
    <CardholderAttemptToResolve>Y</CardholderAttemptToResolve>
    <WhatWasNotReceived>M</WhatWasNotReceived>
    <PurchasedInfo>shoes</PurchasedInfo>
    <ExpectedReceiptDateTime>2019-03-02T05:59:00.000</ExpectedReceiptDateTime>
    <DidCardholderCancelPriorExpectedDate>N</DidCardholderCancelPriorExpectedDate>
    <MerchandiseDeliveredWrongLocation>N</MerchandiseDeliveredWrongLocation>
</Consumer>
<IssuerAcquirerContactInfo>
    <ContactName>name</ContactName>
</IssuerAcquirerContactInfo>
<AssociatedTransGroupId>123456</AssociatedTransGroupId>
<RecallInd>false</RecallInd>
<IsParcelado>N</IsParcelado> [Valid for Brazil domestic jurisdiction only]
</ResponseData>
</SIGetDisputeDetailsResponse>
```

**Request Message—REST-based Services**

```
{  
    "RequestHeader": {  
        "MemberRole": "I"  
    },  
    "requestData": {  
        "DisputeId": 1234  
        "VisaCaseNumber": 1000000000,  
        "IncludeDisputeAsImageInd": "false"  
    }  
}
```

**Response Message—Fraud—REST-based Services**

```
{  
    "Status": [{  
        "Code": "I-3000000000",  
        "Message": "Successfully completed Operation."  
    }],  
    "responseData": {  
        "TransactionInfo": {  
            "AccountNumber": "4100000000000000",  
            "Transaction": {  
                "TranDate": "2019-05-16",  
                "Amount": {  
                    "value": 3000,  
                    "currency": "840"  
                },  
                "id": "8111111111111111"  
            },  
            "MerchantName": "MERCHANT NAME",  
            "MerchantCity": "MERCHANT CITY",  
            "MerchantCountrySubEntityCode": "CA",  
            "MerchantCountryCode": "MU",  
            "MerchantPostalCode": 94402,  
            "MCC": 4111,  
            "ARN": 00000000000000000000000000000001,  
            "ECMOTO": "1",  
            "ReimbursementAttribute": "A",  
            "CPD": "2019-05-24",  
            "TransactionCategory": "D",  
            "AcquirerBID": 00000001,  
            "NetworkID": "0002",  
            "AcquirerBIN": 000002,  
            "RetrievalReferenceNumber": 000000000001,  
            "SystemTraceAuditNumber": "000001",  
            "CardholderName": "John",  
            "TranDestAmt": {  
                "value": 3000,  
                "currency": "840"  
            },  
            "ProcessingCode": "00",  
            "IssuerName": "Issuer name",  
            "AcquirerName": "Acquirer name",  
            "AccountType": "0",  
            "DisputeJurisdiction": {  
                "value": 00000000000000000000000000000000,  
                "currency": "840"  
            }  
        }  
    }  
}
```

```
"Jurisdiction": "DOMESTIC-MU",
"IssuerRegion": "CEMEA",
"AcquirerRegion": "CEMEA",
"CountryCd": "MU",
"jurisdictionCd": "DOM"
},
{
"VisaCaseNumber": 1000000000,
"DisputeId": 1234,
"FraudReportID": 12345,
"FraudRptInd": "Exist",
"EFLInd": "NotExist",
"AEMInd": "NotConfigured",
"DisputeCategory": {"value": "10"}, 
"DisputeCondition": "4",
"DisputeCategoryCondition": "10.4 - Fraud: Card Absent Environment",
"DisputeAmount": {
    "value": 3000,
    "currency": "840"
},
"ExceptionCPD": "2019-07-30",
"DisputeCardholderInfo": {
    "CardholderPrefix": "Mr",
    "CardholderLastName": "John",
    "CardholderMiddleInitial": "Z",
    "Address2": "Address",
    "CardholderSuffix": "xiao",
    "CardholderName": "Name",
    "CardholderAddress": "Test Address",
    "CardholderCity": "City",
    "CardholderCountrySubEntityCode": "CA",
    "NameType": "PN",
    "CardholderPostalCode": "12345",
    "CardholderCountryCode": "US",
    "CardholderPhoneNum": "155130312316",
    "CardholderFaxNum": "020-222000444",
    "CardholderEmail": "John@visa.com",
    "CardholderBestTimeCall": "20190302 150010",
    "CardholderOptInInd": true
},
"CreateDisputeFinancial": "Y",
"FraudReportedDate": "2019-06-25",
"FullMagStripeNotRead": true,
"CVVPresent": false,
"Fraud": {
    "CardholderDeniesAuthorizingInd": true
},
"FraudType": "0",
"IssuerAcquirerContactInfo": {"ContactName": "Name"},
"AssociatedTransGroupId": 123456,
"RecallInd": false,
"IsParcelado": "N"
}
}
```

**Response Message—Authorization—REST-based Services**

```
{  
    "Status": [{  
        "Code": "I-3000000000",  
        "Message": "Successfully completed Operation."  
    }],  
    "responseData": {  
        "TransactionInfo": {  
            "AccountNumber": 4100000000000000,  
            "Transaction": {  
                "TranDate": "2019-05-16",  
                "Amount": {  
                    "value": 22.22,  
                    "currency": "840"  
                },  
                "id": "8111111111111111"  
            },  
            "MerchantName": "MERCHANT NAME",  
            "MerchantCity": "MERCHANT CITY",  
            "MerchantCountrySubEntityCode": "CA",  
            "MerchantCountryCode": "MU",  
            "MerchantPostalCode": "94404",  
            "MCC": "4111",  
            "ARN": 00000000000000000001,  
            "ECMOTO": "1",  
            "ReimbursementAttribute": "A",  
            "CPD": "2019-05-24",  
            "TransactionCategory": "D",  
            "AcquirerBID": 00000001,  
            "NetworkID": "0002",  
            "AcquirerBIN": 000002,  
            "RetrievalReferenceNumber": 00000000001,  
            "SystemTraceAuditNumber": "000001",  
            "TranDestAmt": {  
                "value": 22.22,  
                "currency": "840"  
            },  
            "ProcessingCode": "00",  
            "IssuerName": "Issuer name",  
            "AcquirerName": "Acquirer name",  
            "AccountType": "0",  
            "DisputeJurisdiction": {  
                "Jurisdiction": "DOMESTIC-MU",  
                "IssuerRegion": "CEMEA",  
                "AcquirerRegion": "CEMEA",  
                "CountryCd": "MU",  
                "jurisdictionCd": "DOM"  
            }  
        },  
        "VisaCaseNumber": 1000000000,  
        "DisputeId": 12345,  
        "EFLInd": "NotExist",  
        "AEMIInd": "NotConfigured",  
        "DisputeCategory": {"value": "11"},  
        "DisputeCondition": "3",  
        "DisputeCategoryCondition": "11.3 - Authorization: No Authorization",  
    }},  
}
```

```
"DisputeAmount":{  
    "value": 22.22,  
    "currency": "840"  
},  
"ExceptionCPD": "2019-06-13",  
"Authorization":{},  
"AuthorizationNotObtained": false,  
"FullAuthorizationNotObtained": false,  
"AuthorizationNotObtainedForCardType": false,  
"IssuerAcquirerContactInfo": {"ContactName": "Name"},  
"AssociatedTransGroupId": 123456,  
"RecallInd": false,  
"IsParcelado": "N"  
}  
}
```

## Response Message—Processing Error—REST-based Services

```

    "AccountType": "0",
    "DisputeJurisdiction": {
        "Jurisdiction": "DOMESTIC-US",
        "IssuerRegion": "US",
        "AcquirerRegion": "US",
        "CountryCd": "US"
        "jurisdictionCd": "DOM"
    },
    "VisaCaseNumber": 1000000000,
    "DisputeId": 1234,
    "AEMInd": "NotConfigured",
    "DisputeCategory": {"value": "12"},
    "DisputeCondition": "6.1",
    "DisputeCategoryCondition": "12.6 - Processing Error: Duplicate Processing",
    "DisputeAmount": {
        "value": 22.31,
        "currency": "840"
    },
    "ExceptionCPD": "2019-06-13",
    "ProcessingError": {
        "WhatIncorrectAboutTransaction": "D",
        "BothTransactionsOnSameAcctNumberInd": "N",
        "IssuerOrCardholderOtherAcctWithDiffVisaCardInd": "N",
        "ProvidingPaymentProofOtherMeans": "C",
        "OtherTransSameMerchantInd": "Y"
    },
    "IssuerAcquirerContactInfo": {"ContactName": "Name"},
    "AssociatedTransGroupId": 123456,
    "RecallInd": false,
    "IsParcelado": "N"
    "DisputeImageAttachment": {"Attachment": [
        {
            "ContentType": "image/tiff",
            "Comment": "image is proof of fraud",
            "ImageData": "data",
            "DocType": "CALET",
            "id": "1234567"
        }
    ]}
}
}

```

### Response Message—Consumer—REST-based Services

```

{
    "Status": [
        {
            "Code": "I-300000000",
            "Message": "Successfully completed Operation."
        }
    ],
    "responseData": {
        "TransactionInfo": {
            "AccountNumber": 4100000000000000,
            "Transaction": {
                "TranDate": "2019-05-16",
                "Amount": {
                    "value": 3000,
                    "currency": "840"
                }
            }
        }
    }
}

```

```
        "id": "8111111111111111",
    },
    "MerchantName": "MERCHANT NAME",
    "MerchantCity": "MERCHANT CITY",
    "MerchantCountrySubEntityCode": "CA",
    "MerchantCountryCode": "MU",
    "MerchantPostalCode": "94402",
    "MCC": "4111",
    "ARN": 0000000000000000000000000001,
    "ECMOTO": "1",
    "ReimbursementAttribute": "A",
    "CPD": "2019-05-24",
    "TransactionCategory": "D",
    "AcquirerBID": 00000001,
    "NetworkID": "0002",
    "AcquirerBIN": 000002,
    "RetrievalReferenceNumber": 000000000001,
    "SystemTraceAuditNumber": "000001",
    "CardholderName": "John",
    "TranDestAmt":{
        "value": 22.22,
        "currency": "840"
    },
    "ProcessingCode": "00",
    "IssuerName": "Issuer name",
    "AcquirerName": "Acquirer name",
    "AccountType": "0",
    "DisputeJurisdiction":{
        "Jurisdiction": "DOMESTIC-MU",
        "IssuerRegion": "CEMEA",
        "AcquirerRegion": "CEMEA",
        "CountryCd": "MU",
        "jurisdictionCd": "DOM"
    }
},
"VisaCaseNumber": 1000000000,
"DisputeId": 12345,
"PPCSIInd": "NotExist",
"AEMInd": "NotConfigured",
"DisputeCategory": {"value": "13"},
"DisputeCondition": "3",
"DisputeCategoryCondition": "13.3 - Consumer: Not as Described",
"DisputeAmount": {
    "value": 3000,
    "currency": "840"
},
"ExceptionCPD": "2019-07-13",
"DisputeCardholderInfo": {
    "CardholderPrefix": "Mr",
    "CardholderLastName": "John",
    "CardholderMiddleInitial": "Z",
    "Address2": "Address",
    "CardholderSuffix": "xiaor",
    "CardholderName": "Name",
    "CardholderAddress": "Test Address",
    "CardholderCity": "City",
    "CardholderCountrySubEntityCode": "CA",
```

```

    "NameType": "PN",
    "CardholderPostalCode": "12345",
    "CardholderCountryCode": "US",
    "CardholderPhoneNum": "155130312316",
    "CardholderFaxNum": "020-222000444",
    "CardholderEmail": "John@visa.com",
    "CardholderBestTimeCall": "20190302 150010",
    "CardholderOptInInd": true
},
"CreateDisputeFinancial": "Y",
"Consumer": {
    "DisputeDueTo": "Q",
    "MerchandiseOrServices": "Services",
    "DidCardholderCancel": "N",
    "CardholderFirstNotifiedIssuerDisputeDate": "2019-05-26",
    "CardholderAttemptToResolve": "Y",
    "DidCardholderPayWorkRedone": "N"
},
"IssuerAcquirerContactInfo": {"ContactName": "Name"},
"AssociatedTransGroupId": 123456,
"RecallInd": false,
"IsParcelado": "N"
}
}

```

## SISubmitFraudBundleDisputeQuestionnaireRequest

SISubmitFraudBundleDisputeQuestionnaireRequest allows RTSI users to submit or save the fraud bundle dispute questionnaire. This request can include image(s).

The following XML elements should be considered when submitting a fraud bundle dispute questionnaire or receiving a response from this service.

Element	Description
DisputeCategory	Indicates the category of the dispute. Domain value: 10=Fraud
FraudBundleID	Uniquely identifies a fraud bundle dispute.
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.  The user can provide a minimum of 2 and Maximum of 25 cases.  <b>Note:</b> <i>The user must provide either &lt;RolTransactionId&gt; or &lt;VisaCaseNumber&gt;.</i>
RolTransactionId	Uniquely identifies an original transaction retrieved from VROL's Transaction Inquiry data repository.  The user can provide a minimum of 2 and Maximum of 25 transactions.  <b>Note:</b> <i>The user must provide either &lt;RolTransactionId&gt; or &lt;VisaCaseNumber&gt;.</i>

SISubmitFraudBundleDisputeQuestionnaire contains request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SISubmitFraudBundleDisputeQuestionnaireRequest>
  <RequestHeader>
    <User id="a" type="loginId"/>
    <MemberRole>A</MemberRole>
  </RequestHeader>
  <requestData>
    <R01TransactionId>7000001</R01TransactionId>
    <R01TransactionId>7000002</R01TransactionId>
    <CaseInformation>
      <MemberCaseNumber>a</MemberCaseNumber>
    </CaseInformation>
    <Fraud>
      <FraudType>1</FraudType>
      <CardholderDeniesAuthorizingInd>true
      </CardholderDeniesAuthorizingInd>
    </Fraud>
    <TransmittalInfo>
      <DisputeFraudInfo>
        <FraudType>1</FraudType>
      </DisputeFraudInfo>
    </TransmittalInfo>
    <IssuerDisputeReqInfo>
      <CreateDisputeFinancial>Y</CreateDisputeFinancial>
    </IssuerDisputeReqInfo>
    <Action>Submit</Action>
  </requestData>
</SISubmitFraudBundleDisputeQuestionnaireRequest>
```

### Response Message—SOAP-based Services

```
<SISubmitFraudBundleDisputeQuestionnaireResponse>
  <Status>
    <Code>a</Code>
    <Message>a</Message>
    <ErrorIn>String</ErrorIn>
  </Status>
  <responseData>
    <FraudBundleID>1000000000</FraudBundleID>
  </responseData>
</SISubmitFraudBundleDisputeQuestionnaireResponse>
```

## Request Message—REST-based Services

```
{
  "RequestHeader": {
    "User": {
      "@id": "a",
      "@type": "loginId"
    },
    "MemberRole": "A"
  },
  "requestData": {
    "R01TransactionId": [
      7000001,
      7000002,
    ],
    "CaseInformation": {
      "MemberCaseNumber": "a",
      "FraudClassification": "B",
    },
    "Fraud": {
      "FraudType": 1,
    },
    "TransmittalInfo": {
      "DisputeFraudInfo": {
        "FraudType": "A"
      }
    },
    "IssuerDisputeReqInfo": {
      "CreateDisputeFinancial": "Y",
      "TransactionMessageFormat": "B"
    },
    "Action": "Submit"
  }
}
```

## Response Message—REST-based Services

```
{
  "Status": [
    {
      "Code": "a",
      "Message": "a",
      "ErrorIn": "String"
    },
  ],
  "responseData": {
    "FraudBundleID": 1000000000
  }
}
```

## **SIGetFraudBundleDisputeQuestionnaireStatusRequest**

SIGetFraudBundleDisputeQuestionnaireStatusRequest allows RTSI users to retrieve details of a submitted fraud bundle dispute.

The following XML elements should be considered when retrieving details of a fraud bundle dispute or receiving a response from this service.

Element	Description
FraudBundleID	Uniquely identifies a fraud bundle dispute.
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.
RolTransactionId	Uniquely identifies an original transaction retrieved from VROL's Transaction Inquiry data repository.
DisputeId	A unique VROL-generated ID for each initiated dispute.

SIGetFraudBundleDisputeQuestionnaireStatus contains request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SIGetFraudBundleDisputeQuestionnaireStatusRequest>
    <RequestHeader>
        <User id="a" type="loginId"/>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <FraudBundleID>1000000000</FraudBundleID>
    </requestData>
</SIGetFraudBundleDisputeQuestionnaireStatusRequest>
```

### Response Message—SOAP-based Services

```
<SIGetFraudBundleDisputeQuestionnaireStatusResponse>
    <status>
        <Code>a</Code>
        <Message>a</Message>
        <ErrorIn>String</ErrorIn>
    </status>
    <responseData>
        <FraudBundleDisputeDetails>
            <FraudBundle>
                <FraudBundleCase>
                    <status>
                        <Code>a</Code>
                        <Message>a</Message>
                        <ErrorIn>String</ErrorIn>
                    </status>
                    <VisaCaseNumber>1000000000</VisaCaseNumber>
                    <DisputeId>1234</DisputeId>
                    <RolTransactionId>0</RolTransactionId>
                </FraudBundleCase>
            </FraudBundle>
        </FraudBundleDisputeDetails>
    </responseData>
</SIGetFraudBundleDisputeQuestionnaireStatusResponse>
```

## Request Message—REST-based Services

```
{  
  "RequestHeader": {  
    "User": {  
      "@id": "a",  
      "@type": "loginId"  
    },  
    "MemberRole": "I"  
  },  
  "requestData": {  
    "FraudBundleID": 1000000000  
  }  
}
```

## Response Message—REST-based Services

```
{  
  "Status": [  
    {  
      "Code": "a",  
      "Message": "a",  
      "ErrorIn": "String"  
    },  
  ],  
  "responseData": {  
    "FraudBundleDisputeDetails": {  
      "FraudBundle": {  
        "FraudBundleCase": [  
          {  
            "Status": [  
              {  
                "Code": "a",  
                "Message": "a",  
                "ErrorIn": "String"  
              },  
            ],  
            "VisaCaseNumber": 1000000000,  
            "DisputeId": 1234,  
            "R0lTransactionId": 0  
          }  
        ]  
      }  
    }  
  }  
}
```

## SICreateDisputeResponseRequest

SICreateDisputeResponseRequest allows RTSI users to respond to a collaboration dispute.

The following XML elements should be considered when responding to received disputes or receiving a response from this service.

Element	Description
DisputeResponse	Indicates the user's response to a received dispute.  Domain values: <ul style="list-style-type: none"><li>• FULL=Accept Full</li><li>• PART=Accept Partial</li><li>• DECL=Decline</li></ul>
DisputeResponseId	A unique VROL-generated ID for each Dispute Response.

SICreateDisputeResponse request and response messages are shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SICreateDisputeResponseRequest>
    <RequestHeader>
        <MemberRole>A</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <CreateDisputeFinancial>Y</CreateDisputeFinancial>
        <TransactionMessageFormat>B</TransactionMessageFormat>
        <Note>[DECLINE]</Note>
        <DisputeResponse>DECL</DisputeResponse>
        <DisputeResponseReason>CP</DisputeResponseReason>
        <DisputeResponseReasonExplanation>explanation
        </DisputeResponseReasonExplanation>
        <ARNandDateCreditReversal>411111-000000000111-000111 2019-06-15
        </ARNandDateCreditReversal>
        <WhyCreditReversalNotApply>a</WhyCreditReversalNotApply>
        <Action>Submit</Action>
    </requestData>
</SICreateDisputeResponseRequest>
```

**Response Message—SOAP-based Services**

```
<SICreateDisputeResponseResponse>
  <Status>
    <Code>I-3000000000</Code>
    <Message>Successfully completed Operation.</Message>
  </Status>
  <responseData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <DisputeResponseId>1234</DisputeResponseId>
    <DisputeFinancialInfo>
      <VROLFinancialSentInd>true</VROLFinancialSentInd>
      <VROLFinancialID>12.7-9101</VROLFinancialID>
      <FinancialTransactionType>DISPUTE_RESPONSE_FINANCIAL
      </FinancialTransactionType>
      <FinancialAmt currency="344">2332.03</FinancialAmt>
      <DisputeCategory>12</DisputeCategory>
      <DisputeCondition>7</DisputeCondition>
      <DisputeId>1234</DisputeId>
      <TransactionID>123456</TransactionID>
      <OpposerExpectedRespDate>2019-01-01
      </OpposerExpectedRespDate>
    </DisputeFinancialInfo>
  </responseData>
</SICreateDisputeResponseResponse>
```

### Request Message—REST-based Services

```
{  
    "RequestHeader": {  
        "MemberRole": "A"  
    },  
    "requestData": {  
        "DisputeResponse": "DECL",  
        "DisputeResponseReason": "CP",  
        "DisputeResponseReasonExplanation": "explanation",  
        "MultipleCredits": {  
            "Credit": [  
                {  
                    "id": "1",  
                    "CreditDate": "2019-06-13",  
                    "CreditAmount": {  
                        "currency": "344",  
                        "value": 213.64  
                    },  
                    "MatchingCreditTranId": 1000000,  
                },  
                {  
                    "id": "2",  
                    "CreditDate": "2019-06-13",  
                    "CreditAmount": {  
                        "currency": "840",  
                        "value": 1.60  
                    },  
                    "MatchingCreditTranId": 1000001,  
                }  
            ]  
        },  
        "VisaCaseNumber": "1234567890",  
        "MemberCaseNumber": "1000000000",  
        "Action": "Submit"  
    }  
}
```

## Response Message—REST-based Services

```
{  
    "Status": [{  
        "Code": "I-3000000000",  
        "Message": "Successfully completed Operation."  
    }],  
    "ResponseData": {  
        "VisaCaseNumber": 1000000000,  
        "DisputeResponseId": 12345,  
        "DisputeFinancialInfo": {  
            "VROLFinancialSentInd": "false",  
            "VROLFinancialID": "12.5-9106",  
            "FinancialTransactionType": "DISPUTE_RESPONSE_FINANCIAL",  
            "FinancialAmt": {  
                "value": 113.30,  
                "currency": "978"  
            },  
            "DisputeCategory": {"value": "12"},  
            "DisputeCondition": "5",  
            "DisputeId": 12345,  
            "TransactionID": 1000000000000000,  
            "AcquirerCaseNumber": "Test567",  
            "OpposerExpectedRespDate": "2019-07-01"  
            "IsRdrInfoAvailable": "Y"  
        }  
    }  
}
```

## SIGetDisputeResponseDetailsRequest

SIGetDisputeResponseDetailsRequest allows RTSI users to request for the Dispute Response questionnaire data to be returned in image format, as well as in XML. The returned image will be in TIFF or PDF format. To retrieve the dispute as an image, the user must set <IncludeDisputeAsImageInd>=true. The image data will be returned in the <Image> group in the SIGetDisputeResponseDetailsResponse.

**Note:** Only the Dispute Response questionnaire will be returned as an image. Any attached documents must be retrieved using the AttachmentId value(s) in this response as input to the DocId field(s) of SIGetImageRequest.

**Note:** Any image attached to the Dispute Response questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.

SIGetDisputeResponseDetails request and response messages are shown in the following sample messages.

## Request Message—SOAP-based Services

```
<SIGetDisputeResponseDetailsRequest>
    <RequestHeader>
        <MemberRole>A</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <DisputeResponseId>1234</DisputeResponseId>
    </requestData>
</SIGetDisputeResponseDetailsRequest>
```

## Response Message—SOAP-based Services

**Note:** The *SIGetDisputeResponseDetailsResponse* message includes the dispute history, which consists of information from all previously-submitted disputes and dispute responses that are part of the same dispute thread.

```
<SIGetDisputeResponseDetailsResponse>
    <Status>
        <Code>I-300000000</Code>
        <Message>Successfully completed Operation.</Message>
    </Status>
    <responseData>
        <TransactionInfo>
            <AccountNumber>4100000000000000</AccountNumber>
            <Transaction id="0000000000000000">
                <TranDate>2019-05-24</TranDate>
                <Amount currency="840">15200.13</Amount>
            </Transaction>
            <MerchantName>MERCHANT NAME</MerchantName>
            <MerchantCity>MERCHANT CITY</MerchantCity>
            <MerchantCountrySubEntityCode>CA
            </MerchantCountrySubEntityCode>
            <MerchantCountryCode>US</MerchantCountryCode>
            <MerchantPostalCode>94422</MerchantPostalCode>
            <MCC>5411</MCC>
            <ARN>00000000000000000001</ARN>
            <ECMOTO>1</ECMOTO>
            <ReimbursementAttribute>A</ReimbursementAttribute>
            <CPD>2019-05-24</CPD>
            <TransactionCategory>D</TransactionCategory>
            <AcquirerBID>00000001</AcquirerBID>
            <NetworkID>0002</NetworkID>
            <AcquirerBIN>000002</AcquirerBIN>
            <RetrievalReferenceNumber>000000000001
            </RetrievalReferenceNumber>
            <SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
            <CardholderName>Name</CardholderName>
            <TranDestAmt currency="840">15200.13</TranDestAmt>
            <ProcessingCode>00</ProcessingCode>
            <IssuerName>issuer name</IssuerName>
            <AcquirerName>acquirer name</AcquirerName>
            <AccountType>0</AccountType>
            <DisputeJurisdiction jurisdictionCd="DOM">
                <Jurisdiction>DOMESTIC-US</Jurisdiction>
                <IssuerRegion>US</IssuerRegion>
                <AcquirerRegion>US</AcquirerRegion>
            </DisputeJurisdiction>
        </TransactionInfo>
    </responseData>
</SIGetDisputeResponseDetailsResponse>
```

```
<CountryCd>US</CountryCd>
</DisputeJurisdiction>
</TransactionInfo>
<VisaCaseNumber>1000000000</VisaCaseNumber>
<DisputeResponseId>1234</DisputeResponseId>
<MemberCaseNumber>555555555555</MemberCaseNumber>
<DisputeCategoryCondition>12.7 - Processing Error: Invalid
Data</DisputeCategoryCondition>
<DisputeCondition>7</DisputeCondition>
<DisputeHistorySummary>
    <DisputeHistory>
        <SubmittedBy>Acquirer</SubmittedBy>
        <DisputeCategory>12</DisputeCategory>
        <DisputeItemType>DISPRESP</DisputeItemType>
        <RecallInd>false</RecallInd>
        <ResponseCode>PART</ResponseCode>
        <EventDate>2019-06-17T09:30:47Z</EventDate>
        <UserOrOrgName>user or organization name
        </UserOrOrgName>
        <DisputeItemID>12345</DisputeItemID>
    </DisputeHistory>
    <DisputeHistory>
        <SubmittedBy>Issuer</SubmittedBy>
        <DisputeCategory>12</DisputeCategory>
        <DisputeItemType>DISPREQ</DisputeItemType>
        <RecallInd>false</RecallInd>
        <EventDate>2019-07-16T14:01:45.161</EventDate>
        <UserOrOrgName>user or organization name</UserOrOrgName>
        <DisputeItemID>12345</DisputeItemID>
    </DisputeHistory>
</DisputeHistorySummary>
<ExceptionCPD>2019-06-13</ExceptionCPD>
<DisputeResponse>PART</DisputeResponse>
<AcceptanceAmount currency="840">15199.13</AcceptanceAmount>
<DisputeResponseReason>CNLD</DisputeResponseReason>
<DisputeResponseReasonExplanation>reason
</DisputeResponseReasonExplanation>
<LiabilityNotAcceptedFully>Text</LiabilityNotAcceptedFully>
<IssuerAcquirerContactInfo>
    <ContactName>name</ContactName>
</IssuerAcquirerContactInfo>
<RecallInd>false</RecallInd>
<DisputeImageAttachment>
    <Attachment id="1234567">
        <ContentType>image/tiff</ContentType>
        <DocType>OTHER</DocType>
    </Attachment>
</DisputeImageAttachment>
</ResponseData>
</SIGetDisputeResponseDetailsResponse>
```

## Request Message—REST-based Services

```
{  
    "RequestHeader": {  
        "MemberRole": "A"  
    },  
    "requestData": {  
        "VisaCaseNumber": 1000000000,  
        "DisputeResponseId": 12345  
    }  
}
```

## Response Message—REST-based Services

```
{  
    "Status": [ {  
        "Code": "I-3000000000",  
        "Message": "Successfully completed Operation."  
    } ],  
    "responseData": {  
        "TransactionInfo": {  
            "AccountNumber": 4100000000000000,  
            "Transaction": {  
                "TranDate": "2019-01-13",  
                "Amount": {  
                    "value": 15200.13,  
                    "currency": "840"  
                }  
            },  
            "id": "811111111111110"  
        },  
        "MerchantName": "KITCHEN",  
        "MerchantCity": "GLENDALE",  
        "MerchantCountrySubEntityCode": "CA",  
        "MerchantCountryCode": "US",  
        "MerchantPostalCode": "94422",  
        "MCC": "5411",  
        "ARN": 00000000000000000001,  
        "ECMOTO": "1",  
        "ReimbursementAttribute": "A",  
        "CPD": "2019-01-14",  
        "TransactionCategory": "D",  
        "AcquirerBID": 00000001,  
        "NetworkID": "0002",  
        "AcquirerBIN": 000002,  
        "RetrievalReferenceNumber": 000000000001,  
        "SystemTraceAuditNumber": "000001",  
        "CardholderName": "Name",  
        "TranDestAmt": {  
            "value": 15200.13,  
            "currency": "840"  
        },  
        "ProcessingCode": "00",  
        "IssuerName": "Issuer name",  
        "AcquirerName": "Acquirer name",  
        "AccountType": "0",  
        "DisputeJurisdiction": {  
            "Jurisdiction": "DOMESTIC-US",  
        }  
    }  
}
```

```
"IssuerRegion": "US",
"AcquirerRegion": "US",
"CountryCd": "US"
"jurisdictionCd": "DOM"
}
},
"VisaCaseNumber": 1000000000,
"DisputeResponseId": 12345,
"MemberCaseNumber": "555555555555",
"DisputeCategoryCondition": "12.7 - Processing Error:  
Invalid Data",
"DisputeCondition": "7",
"DisputeHistorySummary":{
"DisputeHistory": [ {
"SubmittedBy": "Acquirer",
"DisputeCategory":{
"value": "12"
},
"DisputeItemType": "DISPRESP",
"RecallInd": false,
"ResponseCode": "PART",
"EventDate": "2019-07-16",
"UserOrOrgName": "Mbr B2",
"DisputeItemID": 123456
}, {
"SubmittedBy": "Issuer",
"DisputeCategory":{
"value": "12"
},
"DisputeItemType": "DISPREQ",
"RecallInd": false,
"EventDate": "2019-07-16",
"UserOrOrgName": "Test B2",
"DisputeItemID": 123456
} ]
},
"ExceptionCPD": "2019-06-13"
"DisputeResponse": "PART",
"AcceptanceAmount":{
"value": 15199.13,
"currency": "840"
},
"DisputeResponseReason": "CNLD",
"DisputeResponseReasonExplanation": "Test",
"LiabilityNotAcceptedFully": "Text",
"IssuerAcquirerContactInfo":{
>ContactName": "name"
},
"RecallInd": "false"
"DisputeImageAttachment":{
"Attachment": [ {
"ContentType": "image/tiff",
"DocType": "OTHER",
"id": "1234567"
} ]
},
}
```

}

## SIGetAcceptanceNotificationDetailsRequest

SIGetAcceptanceNotificationDetailsRequest allows RTSI users to retrieve the acceptance notification details for disputes or dispute responses that were accepted and closed.

The following elements should be considered when retrieving the acceptance notification details for disputes or dispute responses that were accepted and closed.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.
DisputeId	A unique VROL-generated ID for each saved or submitted dispute.
DisputeResponseId	A unique VROL-generated ID for each Dispute Response.

SIGetAcceptanceNotificationDetails request and response messages are shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SIGetAcceptanceNotificationDetailsRequest>
    <RequestHeader>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <DisputeId>12345</DisputeId>
    </requestData>
</SIGetAcceptanceNotificationDetailsRequest>
```

**Response Message—SOAP-based Services**

```
<SIGetAcceptanceNotificationDetailsResponse>
    <Status>
        <Code>I-300000000</Code>
        <Message>Successfully completed Operation.</Message>
    </Status>
    <responseData>
        <TransactionInfo>
            <AccountNumber>41000000000000000000</AccountNumber>
            <Transaction id="0000000000000000">
                <TranDate>2019-05-24</TranDate>
                <Amount currency="840">3000.00</Amount>
            </Transaction>
            <MerchantName>MERCHANT NAME</MerchantName>
            <MerchantCity>MERCHANT CITY</MerchantCity>
            <MerchantCountrySubEntityCode>CA
            </MerchantCountrySubEntityCode>
            <MerchantCountryCode>MU</MerchantCountryCode>
            <MerchantPostalCode>94402</MerchantPostalCode>
            <MCC>4111</MCC>
            <ARN>000000000000000000000001</ARN>
            <ECMOTO>1</ECMOTO>
            <ReimbursementAttribute>A</ReimbursementAttribute>
            <CPD>2019-05-24</CPD>
            <TransactionCategory>D</TransactionCategory>
            <AcquirerBID>00000001</AcquirerBID>
            <NetworkID>0002</NetworkID>
            <AcquirerBIN>000002</AcquirerBIN>
            <RetrievalReferenceNumber>000000000001
            </RetrievalReferenceNumber>
            <SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
            <CardholderName>John</CardholderName>
            <TranDestAmt currency="840">3000.00</TranDestAmt>
            <ProcessingCode>00</ProcessingCode>
            <IssuerName>issuer name</IssuerName>
            <AcquirerName>acquirer name</AcquirerName>
            <AccountType>0</AccountType>
            <DisputeJurisdiction jurisdictionCd="DOM">
                <Jurisdiction>DOMESTIC-MU</Jurisdiction>
                <IssuerRegion>CEMEA</IssuerRegion>
                <AcquirerRegion>CEMEA</AcquirerRegion>
                <CountryCd>MU</CountryCd>
            </DisputeJurisdiction>
        </TransactionInfo>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <MemberCaseNumber>555555555555</MemberCaseNumber>
        <AcceptItemType>DISPREQ</AcceptItemType>
        <DisputeCategory>13</DisputeCategory>
        <AcceptanceAmount currency="840">3000.00</AcceptanceAmount>
        <AcceptanceDateTime>2019-05-21T11:35:17.000Z
        </></AcceptanceDateTime>
    </responseData>
</SIGetAcceptanceNotificationDetailsResponse>
```

### Request Message—REST-based Services

```
{  
    "SIGetAcceptanceNotificationDetailsRequest":{  
        "RequestHeader":{  
            "MemberRole": "I"  
        },  
        "requestData":{  
            "DisputeId": 12345,  
            "VisaCaseNumber": 1000000000,  
        }  
    }  
}
```

### Response Message—REST-based Services

```
{  
    "SIGetAcceptanceNotificationDetailsResponse":{  
        "Status": [ {  
            "Code": "I-3000000000",  
            "Message": "Successfully completed Operation."  
        } ],  
        "responseData":{  
            "TransactionInfo":{  
                "AccountNumber": "4000000000011111",  
                "Transaction":{  
                    "TranDate": "2019-01-13",  
                    "Amount":{  
                        "value": 3000,  
                        "currency": "840"  
                    }  
                    "id": "8111111111111111"  
                },  
                "MerchantName": "KITCHEN",  
                "MerchantCity": "GLENDALE",  
                "MerchantCountrySubEntityCode": "CA",  
                "MerchantCountryCode": "MU",  
                "MerchantPostalCode": "94402",  
                "MCC": "4111",  
                "ARN": 0000000000000000000000000001,  
                "ECMOTO": "1",  
                "ReimbursementAttribute": "A",  
                "CPD": "2019-01-14",  
                "TransactionCategory": "D",  
                "AcquirerBID": 00000001,  
                "NetworkID": "0002",  
                "AcquirerBIN": 000002,  
                "RetrievalReferenceNumber": 000000000001,  
                "SystemTraceAuditNumber": "000001",  
                "CardholderName": "John",  
                "TranDestAmt":{  
                    "value": 3000,  
                    "currency": "840"  
                },  
                "ProcessingCode": "00",  
                "IssuerName": "Issuer name",  
                "AcquirerName": "Acquirer name",  
            }  
        }  
    }  
}
```

```

    "AccountType": "0",
    "DisputeJurisdiction": {
        "Jurisdiction": "DOMESTIC-MU",
        "IssuerRegion": "CEMEA",
        "AcquirerRegion": "CEMEA",
        "CountryCd": "MU"
        "jurisdictionCd": "DOM"
    },
    "VisaCaseNumber": 1000000000,
    "MemberCaseNumber": 555555555555,
    "AcceptItemType": "DISPREQ",
    "DisputeCategory": {"value": "13"},
    "AcceptanceAmount": {
        "value": 3000,
        "currency": "840"
    },
    "AcceptanceDateTime": "2019-05-21T11:35:17.000Z"
}
}

```

## SIChangeDisputeStatusRequest

SIChangeDisputeStatusRequest allows RTSI users to do the following actions:

- Delete a saved dispute or dispute response
- Recall a submitted dispute or dispute response

**Note:** *RTSI users can only recall a dispute response with "Decline" value.*

The following elements should be considered when deleting a saved dispute or when recalling a submitted dispute.

Element	Description
ChangelitemId	Uniquely identifies the dispute item being recalled or deleted.  ChangelitemId includes the following unique ID values for the corresponding dispute item: <ul style="list-style-type: none"> <li>– Disputeid (for Dispute)</li> <li>– DisputeResponseid (for Dispute Response)</li> </ul>
ChangelitemType	Indicates the type of dispute being recalled or deleted.  Domain values: <ul style="list-style-type: none"> <li>• DISPREQ=Dispute</li> <li>• DISPRESP=Dispute Response</li> </ul>
Action	Indicates the corresponding action to be applied to the dispute.  Domain values: Recall, Delete

SIChangeDisputeStatus request and response messages are shown in the following sample messages.

**Request Message—Recall action—SOAP-based Services**

```
<SIChangeDisputeStatusRequest>
  <RequestHeader>
    <MemberRole>I</MemberRole>
  </RequestHeader>
  <requestData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <ChangeItemID>1234</ChangeItemID>
    <ChangeItemType>DISPREQ</ChangeItemType>
    <RecallReason>reason</RecallReason>
    <Action>Recall</Action>
  </requestData>
</SIChangeDisputeStatusRequest>
```

**Note:** For Cardholder No Longer Disputes, add the keyword '[CLD]' at the beginning of the reason text in the <RecallReason> element to allow VROL to recall the dispute transaction even after 3 calendar days (or 1 calendar day if the pinned original transaction in the dispute case is an Original Credit Transaction) provided that the liability is not shifted to the issuer.

**Response Message—Recall action—SOAP-based Services**

```
<SIChangeDisputeStatusResponse>
  <Status>
    <Code>I-300000000</Code>
    <Message>Successfully completed Operation.</Message>
  </Status>
  <responseData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <ChangeItemID>1234</ChangeItemID>
    <DisputeFinancialInfo>
      <VROLFInFinancialSentInd>false</VROLFInFinancialSentInd>
      <VROLFInFinancialID>10.3-9183</VROLFInFinancialID>
      <FinancialTransactionType>DISPUTE_FINANCIAL_REVERSAL
      </FinancialTransactionType>
      <FinancialAmt currency="840">23.01</FinancialAmt>
      <DisputeCategory>10</DisputeCategory>
      <DisputeCondition>3</DisputeCondition>
      <DisputeId>447295</DisputeId>
      <TransactionID>8111111111110</TransactionID>
      <IssuerCaseNumber>Test</IssuerCaseNumber>
      <OpposerExpectedRespDate>2019-07-14</OpposerExpectedRespDate>
    </DisputeFinancialInfo>
  </responseData>
</SIChangeDisputeStatusResponse>
```

**Request Message—Recall action—REST-based Services**

```
{  
  "RequestHeader": {  
    "MemberRole": "I"  
  },  
  "requestData": {  
    "VisaCaseNumber": 1000000000,  
    "ChangeItemID": 1234,  
    "ChangeItemType": "DISPREQ",  
    "RecallReason": "reason",  
    "Action": "Recall"  
  }  
}
```

**Note:** For Cardholder No Longer Disputes, add the keyword '[CLD]' at the beginning of the reason text in the <RecallReason> element to allow VROL to recall the dispute transaction even after 3 calendar days (or 1 calendar day if the pinned original transaction in the dispute case is an Original Credit Transaction) provided that the liability is not shifted to the issuer.

**Response Message—Recall action—REST-based Services**

```
{  
  "Status": {  
    "Code": "I-300000000",  
    "Message": "Successfully completed Operation."  
  },  
  "responseData": {  
    "VisaCaseNumber": 1000000000,  
    "ChangeItemID": 1234  
    "DisputeFinancialInfo": {  
      "VROLFinancialSentInd": false,  
      "VROLFinancialID": "10.3-9183",  
      "FinancialTransactionType": "DISPUTE_FINANCIAL_REVERSAL",  
      "FinancialAmt": {  
        "value": 23.01,  
        "currency": "840"  
      },  
      "DisputeCategory": {"value": "10"},  
      "DisputeCondition": "3",  
      "DisputeId": 12345,  
      "TransactionID": "81111111111110",  
      "IssuerCaseNumber": "Test",  
      "OpposerExpectedRespDate": "2019-07-14"  
    }  
  }  
}
```

### Request Message—Delete action—SOAP-based Services

```
<SIChangeDisputeStatusRequest>
    <RequestHeader>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1234567890</VisaCaseNumber>
        <ChangeItemID>1234</ChangeItemID>
        <ChangeItemType>DISPREQ</ChangeItemType>
        <Action>Delete</Action>
    </requestData>
</SIChangeDisputeStatusRequest>
```

### Response Message—Delete action—SOAP-based Services

```
<SIChangeDisputeStatusResponse>
    <Status>
        <Code>I-300000000</Code>
        <Message>Successfully completed Operation.</Message>
    </Status>
    <responseData>
        <VisaCaseNumber>1234567890</VisaCaseNumber>
        <ChangeItemID>1234</ChangeItemID>
        <DisputeFinancialInfo/>
    </responseData>
</SIChangeDisputeStatusResponse>
```

### Request Message—Delete action—REST-based Services

```
{
    "RequestHeader": {
        "MemberRole": "I"
    },
    "requestData": {
        "VisaCaseNumber": 1234567890,
        "ChangeItemID": 1234,
        "ChangeItemType": "DISPREQ",
        "Action": "Delete"
    }
}
```

### Response Message—Delete action—REST-based Services

```
{
    "Status": {
        "Code": "I-300000000",
        "Message": "Successfully completed Operation."
    },
    "responseData": {
        "VisaCaseNumber": 1234567890,
        "ChangeItemID": 1234
    }
}
```

## SIGetRdrDetailsRequest

SIGetRdrDetailsRequest allows RTSI users to retrieve details of dispute transactions eligible for RDR.

SIGetRdrDetails request and response messages are shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SIGetRdrDetailsRequest>
  <RequestHeader>
    <User id="a"/>
    <MemberRole>A</MemberRole>
  </RequestHeader>
  <requestData>
    <DisputeId>446881</DisputeId>
  </requestData>
</SIGetRdrDetailsRequest>
```

### Response Message—SOAP-based Services

```
<SIGetRdrDetailsResponse>
  <status>
    <Code>a</Code>
    <Message>a</Message>
  </status>
  <responseData>
    <DisputeId>456789</DisputeId>
    <MerchantName>Walmart</MerchantName>
    <DisputeAmount>50.80</DisputeAmount>
    <RdrTimeframeInHours>48</RdrTimeframeInHours>
    <DisputeOnHoldSubmittedDate>2001-12-17T09:30:47Z
    <DisputeOnHoldSubmittedDate>
    <DisputeOnHoldExpiryDate>2001-12-17T09:30:47Z
    <DisputeOnHoldExpiryDate>
    <MatchCreditDetails>
      <CreditDetails>
        <MatchedCreditTranId>7667968</MatchedCreditTranId>
        <MatchedCreditAmount>50.8</MatchedCreditAmount>
        <MatchedCreditARN>74040000000000000000000000000000
        </MatchedCreditARN>
        <MatchedCreditRRN>834870849347</MatchedCreditRRN>
        <R0lTransactionId>1234567890</R0lTransactionId>
      </CreditDetails>
    </MatchCreditDetails>
  </responseData>
</SIGetRdrDetailsResponse>
```

### Request Message—REST-based Services

```
{  
    "RequestHeader": {  
        "MemberRole": "I"  
    },  
    "requestData": {  
        "DisputeId": 456789,  
    },  
    "Action": "Submit"  
}
```

### Response Message—REST-based Services

```
{  
    "Status": [ {  
        "Code": "I-3000000000",  
        "Message": "Successfully completed Operation."  
    } ],  
    "responseData": {  
        "DisputeId": 456789,  
        "MerchantName": "Walmart",  
        "DisputeAmount": {  
            "@currency": 344,  
            "$": 50.80  
        },  
        "RdrTimeframeInHours": 26,  
        "DisputeOnHoldSubmittedDate": "2001-12-17T09:30:47Z",  
        "DisputeOnHoldExpiryDate": "2001-12-17T09:30:47Z",  
        "MatchCreditDetails": {  
            "CreditDetails": {  
                "MatchedCreditTranId": 7667968,  
                "MatchedCreditAmount": {  
                    "@currency": 344,  
                    "$": 50.80  
                },  
                "MatchedCreditARN": 74047619081000158620786,  
                "MatchedCreditRRN": 834870849347,  
                "R0lTransactionId": 1234567890  
            }  
        }  
    }  
}
```

THIS PAGE INTENTIONALLY LEFT BLANK.

# Dispute Financials

8

Dispute Financials are messages used by VisaNet to transfer liability between parties throughout a disputes lifecycle. With the dispute process, VROL Member Configuration allows the member to choose whether to generate their own financial records outside of VROL, or have VROL generate the financials on the issuer/acquirer's behalf.

In the Allocation workflow, VROL automatically determines financial liability based on the dispute category, transaction data, and *Visa Rules*. Liability between members is assigned using the following VisaNet messages:

- Dispute Financial (0422 / TC15-17) for Dispute submission
- Dispute Reversal Financial (0422 / TC35-37) for Pre-Arbitration acceptance/expiration
- Dispute Reversal Financial (0422 / TC35-37) for recalled issuer-initiated dispute questionnaire

In the Collaboration workflow, liability is transferred through the following transactions:

- Dispute Financial (0422 / TC15-17) for Dispute submission
- Dispute Reversal Financial (0422 / TC35-37) for recalled issuer-initiated dispute questionnaire
- Dispute Response Financial (0220 / TC05-07) for Dispute Response submission
- Dispute Response Reversal Financial (0220 / TC25-27) for Pre-Arbitration acceptance/expiration
- Dispute Response Reversal Financial (0220 / TC25-27) for recalled acquirer-initiated dispute response questionnaire

When a Dispute, Dispute Response or Pre-Arbitration acceptance is successfully processed, VROL performs one of the following actions, depending on the BINs/organization's configuration:

1. Generate the financial records on the member's behalf.
2. Generate data elements required for the financial message, where the member can create the financial message and submit it directly into VisaNet.

See the Dispute Financial and Dispute Advice tabs of the Interface Element Specification (IES) for details on these fields.

**Important:**

Any member host system that is sending or receiving financial records is required to make changes to accommodate new data fields that are applied to the financial messages, as defined in the *Global Technical Letter and Implementation Guide*.

In addition to supplying the additional data, dispute financials must be processed in a timely manner to ensure the following:

- All the generated dispute financials have been authorized by the current and relevant information
- The validity of the Dispute, Dispute Response, or Pre-Arbitration response
- The financial must be received by VisaNet within the agreed timelines per VROL Dispute Resolution Rules and Regulation

Financials are also required when a Pre-Arbitration response timeline expires and, therefore, is automatically accepted. Since this is not an explicit acceptance from the member host system, RTSI connected members should perform member side monitoring of time frames to generate financials at the appropriate times.

**Note:** *Dispute amount on reversal has the same currency as dispute amount on prior direct transaction.*

**Note:** *Client-generated financials must be received within 48 hours. If no financial is received within the expected time frame, VROL invalidates the questionnaire and makes it unavailable for matching to the client-initiated financial. To revalidate the questionnaire, the user needs to review and submit the questionnaire again.*

The recall action triggers a Dispute Financial Reversal only if there is an initial financial already triggered in scenarios where the member is configured to submit financials through VROL. Otherwise, only the dispute questionnaire is recalled and the original questionnaire becomes null and void.

For recalled transactions, members are given two days from recall submission for the financial reversal to be created, sent to RSI/VIP, and then cleared or processed successfully. If VROL has not received and successfully validated a reversal financial after two days from recall submission, VROL automatically generates a reversal financial transaction on behalf of the side that failed to submit a financial reversal.

Penalties may apply in situations where members fail to generate financials in a timely manner and Visa is required to correct the dispute financial liability.

See [Chapter 7, Dispute Questionnaires](#) for more information on initiating disputes.

## Dispute Financial Process

Members cannot submit dispute financial in RTSI. They can only perform the following actions in RTSI for dispute financial:

- View a VROL-generated or submitted dispute financial
- View a received dispute financial
- View a received dispute financial advice

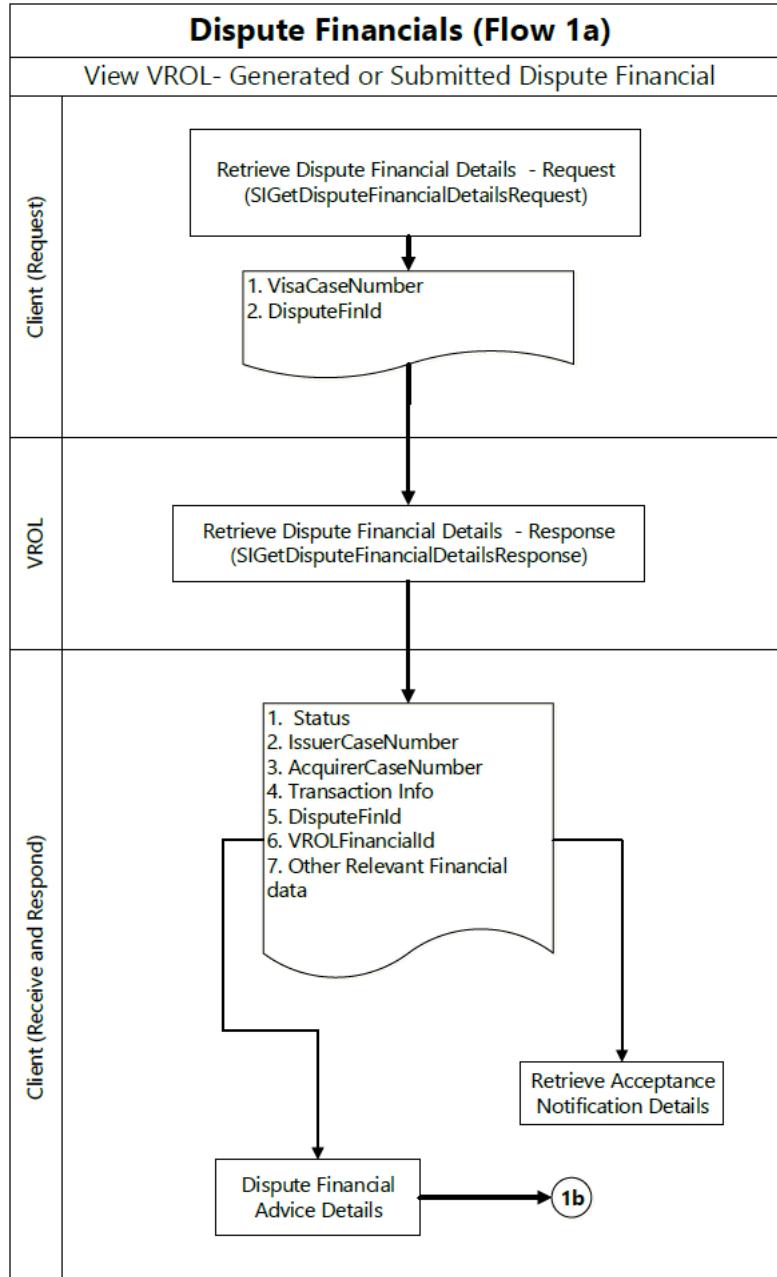
All types of the aforementioned dispute financials and their reversals are supported.

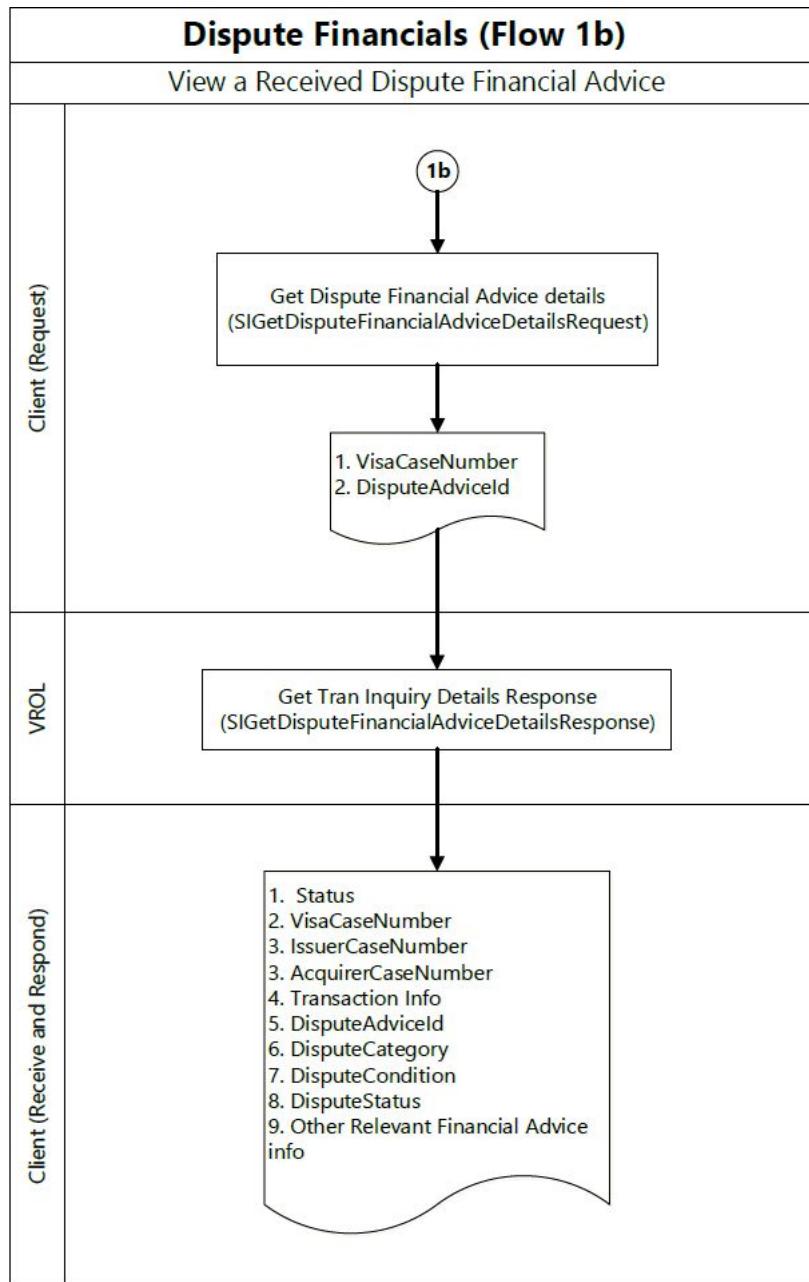
## Dispute Financial Flows

The following diagrams illustrate the dispute financial process.

**Note:**

- The diagrams list only the important elements. For the complete elements, refer to the Dispute Financial and Dispute Advice tabs of the Interface Element Specification (IES).
- The dashed lines in the flow refer to optional functions that the user can use, depending on the business process that the user wants to implement.

**Figure 8–1: Dispute Financials—Flow 1a**

**Figure 8–2: Dispute Financials—Flow 1b**

## Retrieve Dispute Financial from Queue

To retrieve dispute financials from the dispute financial queue, the member triggers a SIGetQueueRequest message. See [Chapter 17, Queues](#).

VROL returns a list of dispute financial cases in the queue including the VisaCaseNumber.

## Get Dispute Financial Details

To retrieve the dispute financial details in a case, the member host system triggers a **SIGetDisputeFinancialDetailsRequest** message supplying the VisaCaseNumber and DisputeFinId.

VROL returns a **SIGetDisputeFinancialDetailsResponse** message, which includes details of either the non-settled submitted financial (if requested by the issuer prior to receiving the advice) or the combined details of both submitted VROL financial and received settlement advice. This settlement advice updates the settlement amount, date, and status on the submitted financial; however, it does not return all the submitted fields. On the acquirer side, this method returns the dispute financial received on member's behalf, if the member is configured to receive dispute financials.

If member submits dispute financials directly to RSI or VIP, then no data will be returned by **SIGetDisputeFinancialDetailsResponse** message.

## Get Dispute Financial Advice Details

To retrieve the dispute financial advice details in a case, the member host system triggers a **SIGetDisputeFinancialAdviceDetailsRequest** message supplying the VisaCaseNumber and DisputeAdviceId.

VROL returns a **SIGetDisputeFinancialAdviceDetailsResponse** message, which includes details of the settlement advice where some fields (for example, dispute amount and currency) are different for issuer and acquirer.

If member submits dispute financials directly to RSI or VIP, then the settlement advice will still be returned by **SIGetDisputeFinancialAdviceDetailsResponse** message.

# Dispute Financial Operations

RTSI exposes the following operations:

**SIGetDisputeFinancialDetailsRequest**—Used by the acquirer or issuer to retrieve dispute financial details in a case.

**SIGetDisputeFinancialAdviceDetailsRequest**—Used by the acquirer or issuer to retrieve dispute financial advice details in a case.

## SIGetDisputeFinancialDetailsRequest

**SIGetDisputeFinancialDetailsRequest** allows RTSI users to retrieve dispute financial details in a case.

The following elements should be considered when retrieving the financial details.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.
DisputeFinId	A unique VROL-generated ID for each financial.

**SIGetDisputeFinancialDetails** contains request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SIGetDisputeFinancialDetailsRequest>
  <RequestHeader>
    <MemberRole>A</MemberRole>
  </RequestHeader>
  <requestData>
    <VisaCaseNumber>1234567890</VisaCaseNumber>
    <DisputeFinId>12345</DisputeFinId>
  </requestData>
</SIGetDisputeFinancialDetailsRequest>
```

### Response Message—SOAP-based Services

```
<SIGetDisputeFinancialDetailsResponse>
  <Status>
    <Code>a</Code>
    <Message>a</Message>
    <ErrorIn>String</ErrorIn>
  </Status>
  <responseData>
    <IssuerCaseNumber>a</IssuerCaseNumber>
    <AcquirerCaseNumber>a</AcquirerCaseNumber>
    <TransactionInfo>
      <AccountNumber>410000000000</AccountNumber>
      <Token>000000000000</Token>
      <TokenAssuranceLevel>a</TokenAssuranceLevel>
      <Transaction id="0000000000000000">
        <TranDate>2017-05-24</TranDate>
        <Amount currency="344">8.24</Amount>
      </Transaction>
      <MerchantName>MERCHANT NAME</MerchantName>
      <MerchantCity>MERCHANT CITY</MerchantCity>
      <MerchantCountrySubEntityCode>CA
      </MerchantCountrySubEntityCode>
      <MerchantCountryCode>HK</MerchantCountryCode>
      <MerchantPostalCode>94408</MerchantPostalCode>
      <MCC>5000</MCC>
      <ARN>00000000000000000000000000000001</ARN>
      <ECMOTO>a</ECMOTO>
      <ReimbursementAttribute>A</ReimbursementAttribute>
      <SpecialConditionInd>a</SpecialConditionInd>
      <CPD>2017-05-24</CPD>
      <TransactionCategory>D</TransactionCategory>
      <AcquirerBID>00000001</AcquirerBID>
      <NetworkID>0002</NetworkID>
      <AcquirerBIN>000002</AcquirerBIN>
      <RetrievalReferenceNumber>000000000001
      </RetrievalReferenceNumber>
      <SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
      <CardholderName>a</CardholderName>
      <TranDestAmt currency="344">8.24</TranDestAmt>
      <ProcessingCode>00</ProcessingCode>
      <IssuerName>issuer name</IssuerName>
      <AcquirerName>acquirer name</AcquirerName>
      <AccountType>00</AccountType>
    </responseData>
  </SIGetDisputeFinancialDetailsResponse>
```

```
<AccountType1Identification>a
</AccountType1Identification>
<DCCIInd>a</DCCIInd>
<DCCIIndDescription>a</DCCIIndDescription>
<DomesticProcessingInd>true</DomesticProcessingInd>
<DisputeJurisdiction jurisdictionCd="DOM">
    <Jurisdiction>DOMESTIC-HK</Jurisdiction>
    <IssuerRegion>AP</IssuerRegion>
    <AcquirerRegion>AP</AcquirerRegion>
    <CountryCd>HK</CountryCd>
</DisputeJurisdiction>
<BusinessAppId>aa</BusinessAppId>
</TransactionInfo>
<VisaCaseNumber>1234567890</VisaCaseNumber>
<DisputeFinId>123456</DisputeFinId>
<VROLFinancialID>123456789</VROLFinancialID>
<DisputeFinancialData>
    <DisputeCategory>11</DisputeCategory>
    <DisputeCategoryDesc>Authorization
    </DisputeCategoryDesc>
    <DisputeCondition>9</DisputeCondition>
    <DisputeConditionDesc>String</DisputeConditionDesc>
    <SrcAmt currency="000" SrcAmtDebitCreditInd="CR"
    >0</SrcAmt>
    <DestAmt currency="000" DestAmtDebitCreditInd="CR"
    >0</DestAmt>
    <SettlementAmount currency="840">100</SettlementAmount>
    <PartialInd>a</PartialInd>
    <CPD>2016-01-01</CPD>
    <SettlementDate>2016-01-01</SettlementDate>
    <ChargebackRefNum>000000</ChargebackRefNum>
    <DocumentationInd>String</DocumentationInd>
    <MemberMsgEditText>a</MemberMsgEditText>
    <ExcludedTranIdReason>a</ExcludedTranIdReason>
    <CRSProcessingCode>a</CRSProcessingCode>
    <RetrievalReferenceNumber>000000000001
    </RetrievalReferenceNumber>
    <SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
    <DisputeStatus>P1</DisputeStatus>
    <VROLBundleID>0</VROLBundleID>
    <UsageCode>0</UsageCode>
    <Code>a</Code>
    <Message>a</Message>
</DisputeFinancialData>
<DisputeMerchantData>
    <MerchantName>a</MerchantName>
    <MerchantNameKorea>a</MerchantNameKorea>
    <MerchantNameSweden>a</MerchantNameSweden>
    <JapanTCR2MerchantNameKatakana>a
    </JapanTCR2MerchantNameKatakana>
    <JapanTCR2MerchantNameKanji>a
    </JapanTCR2MerchantNameKanji>
    <JapanTCR2AcquiringBankMerchantNumber>a
    </JapanTCR2AcquiringBankMerchantNumber>
    <MerchantId>0</MerchantId>
    <MerchantType>0</MerchantType>
    <DepartmentId>a</DepartmentId>
```

```
<SpecialMerchantIdentifier>a</SpecialMerchantIdentifier>
<MerchantLocation>a</MerchantLocation>
<MerchantLocationKorea>a</MerchantLocationKorea>
<MerchantCitySweden>a</MerchantCitySweden>
<MerchContactPhone>a</MerchContactPhone>
<IrfCategory>0</IrfCategory>
<IRFTax>0</IRFTax>
<MexicoAccountType>a</MexicoAccountType>
<IRFType>01</IRFType>
<MCC>0000</MCC>
<POSCondCode>a</POSCondCode>
<MerchantVerificationValue>a</MerchantVerificationValue>
<CardAcceptorId>a</CardAcceptorId>
<AcquirerBID>00000000</AcquirerBID>
<TerminalId>a</TerminalId>
<ARN>000000000000000000000000</ARN>
</DisputeMerchantData>
<DisputePOSSInfoData>
    <DraftTransactionCode>String</DraftTransactionCode>
    <CatInd>a</CatInd>
    <PosEntryModeCd>a</PosEntryModeCd>
    <PosEnvironment>a</PosEnvironment>
    <PosTermCapability>a</PosTermCapability>
    <ECMOTO>a</ECMOTO>
    <RequestedPaymentServicesInd>a
    </RequestedPaymentServicesInd>
    <PrepaidCardInd>a</PrepaidCardInd>
    <PurchaseIdentifierFmt>String</PurchaseIdentifierFmt>
    <PurchaseIdentifier>String</PurchaseIdentifier>
    <PaymentInd>a</PaymentInd>
    <CardTypeLocal>a</CardTypeLocal>
    <ElectronicCommerceGoodsInd>a</ElectronicCommerceGoodsInd>
    <ServiceDevelopmentField>String</ServiceDevelopmentField>
    <ProductId>a</ProductId>
    <SpndQ1fdInd>a</SpndQ1fdInd>
    <ProgramId>a</ProgramId>
    <TerminalCapabilityProfile>a</TerminalCapabilityProfile>
    <RemoteTerminalInd>String</RemoteTerminalInd>
    <JapanTCR2NetSalesAmount>0</JapanTCR2NetSalesAmount>
    <JapanTCR2NetGrossFlag>a</JapanTCR2NetGrossFlag>
</DisputePOSSInfoData>
<DisputeAuthInfoData>
    <AuthRespCode>a</AuthRespCode>
    <AuthCode>a</AuthCode>
    <AuthAmt currency="000">0</AuthAmt>
    <TotalAuthAmount currency="000">0</TotalAuthAmount>
    <AuthSrcCode>String</AuthSrcCode>
    <FloorLimitInd>a</FloorLimitInd>
    <CardholderIdMethod>a</CardholderIdMethod>
    <AvsResponseCode>a</AvsResponseCode>
    <PcasInd>a</PcasInd>
    <CrbExceptionFileInd>a</CrbExceptionFileInd>
    <SpecialConditionInd>a</SpecialConditionInd>
    <RISInd>a</RISInd>
    <MerchantTranInd>a</MerchantTranInd>
</DisputeAuthInfoData>
<DisputeProcessingSettingsData>
```

```
<ReimbursementAttribute>0</ReimbursementAttribute>
<NetworkIdentificationCode>String
</NetworkIdentificationCode>
<ContactInfo>a</ContactInfo>
<AdjustmentProcessingInd>a</AdjustmentProcessingInd>
<NationalReimbursementFee currency="000">0
</NationalReimbursementFee>
<NationalTax>0</NationalTax>
<InterchangeReimbursementFeeMultiplier>00
</InterchangeReimbursementFeeMultiplier>
<SettlementType>0</SettlementType>
<AuthCharCd>a</AuthCharCd>
<MerchantVolumeInd>a</MerchantVolumeInd>
<FeeProgramInd>a</FeeProgramInd>
<ChargebackRightsInd>a</ChargebackRightsInd>
<SettlementFlag>a</SettlementFlag>
<InternationalFeeInd>a</InternationalFeeInd>
<IssuerInterchangeReimbursementFee currency="000"
IssuerInterchangeReimbursementFeeDebitCreditInd="CR">0
</IssuerInterchangeReimbursementFee>
<IssuerCashbackInterchangeReimbursementFee currency="000"
IssuerCashbackInterchangeReimbursementFeeDebitCreditInd=
"DR">0</IssuerCashbackInterchangeReimbursementFee>
<IssuerInterchangeReimbursementFeeSSBC currency="000">0
</IssuerInterchangeReimbursementFeeSSBC>
<IssuerInterchangeReimbursementFeeSSBCDebitCreditInd>CR
</IssuerInterchangeReimbursementFeeSSBCDebitCreditInd>
<IssuerCashbackInterchangeReimbursementFeeSSBC currency=
"000">0</IssuerCashbackInterchangeReimbursementFeeSSBC>
<IssuerCashbackInterchangeReimbursementFeeSSBCDebitCreditInd>CR
</IssuerCashbackInterchangeReimbursementFeeSSBCDebitCreditInd>
<AcquirerInterchangeReimbursementFee currency="000"
AcquirerInterchangeReimbursementFeeDebitCreditInd=
"CR">0</AcquirerInterchangeReimbursementFee>
<AcquirerCashbackInterchangeReimbursementFee currency="000"
AcquirerCashbackInterchangeReimbursementFeeDebitCreditInd=
"DR">0</AcquirerCashbackInterchangeReimbursementFee>
<AcquirerInterchangeReimbursementFeeSSBC currency="000">0
</AcquirerInterchangeReimbursementFeeSSBC>
<AcquirerInterchangeReimbursementFeeSSBCDebitCreditInd>DR
</AcquirerInterchangeReimbursementFeeSSBCDebitCreditInd>
<AcquirerCashbackInterchangeReimbursementFeeSSBC currency=
"000">0</AcquirerCashbackInterchangeReimbursementFeeSSBC>
<AcquirerCashbackInterchangeReimbursementFeeSSBCDebitCreditInd>
</AcquirerCashbackInterchangeReimbursementFeeSSBCDebitCreditInd>
<InterchangeFeeAmount>0</InterchangeFeeAmount>
<AgentUniqueId>String</AgentUniqueId>
</DisputeProcessingSettingsData>
<DisputeTransactionDetailsData>
    <TransactionCodeQualifier>String</TransactionCodeQualifier>
    <OrigTranCPD>2017-08-13</OrigTranCPD>
    <MemberPurchaseAgreementDate>2017-08-13
    </MemberPurchaseAgreementDate>
    <InstallmentPaymentCount> </InstallmentPaymentCount>
    <NumberOfPayments>0</NumberOfPayments>
    <MerchantDiscountRate>0</MerchantDiscountRate>
    <PaymentMethod/>
```

```
<LocalAuthNum>0</LocalAuthNum>
<CashbackAmount currency="000">0</CashbackAmount>
<BusinessAppId>aa</BusinessAppId>
<TerminalEntryCapability>0</TerminalEntryCapability>
<SurchargeAmount>0</SurchargeAmount>
<NetworkID>0002</NetworkID>
<SwitchRsnCode>0000</SwitchRsnCode>
<MessageType>aaaa</MessageType>
<MemberSuppliedFee currency="000"
<MemberSuppliedFeeDebitCreditInd="CR">0
</MemberSuppliedFee>
<DCCInd>a</DCCInd>
<AccountType1Identification>a</AccountType1Identification>
<TransactionType>String</TransactionType>
<AccountType>a</AccountType>
</DisputeTransactionDetailsData>
<DisputeNNSDetailsData>
    <JapanTCR2NumberOfBonusPayments>
        0</JapanTCR2NumberOfBonusPayments>
    <JapanTCR2BonusAmount>0</JapanTCR2BonusAmount>
    <JapanTCR2BonusCode>a</JapanTCR2BonusCode>
    <JapanTCR2BonusMonthSign>a</JapanTCR2BonusMonthSign>
    <PaymentCd>01</PaymentCd>
    <JapanTCR2PaymentMode>0</JapanTCR2PaymentMode>
    <JapanTCR2TaxAndMailingCharge>0
    </JapanTCR2TaxAndMailingCharge>
    <JapanTCR2SalesDraftReferenceNumber>0
    </JapanTCR2SalesDraftReferenceNumber>
    <JapanTCR2TextMessage>a</JapanTCR2TextMessage>
    <JapanTCR2AcquirerInvoiceAmount>0
    </JapanTCR2AcquirerInvoiceAmount>
    <JapanTCR2NationalChargebackReasonCode>a
    </JapanTCR2NationalChargebackReasonCode>
    <IRFCalculationBaseAmt currency="000">0</
    IRFCalculationBaseAmt>
    <AmountBaseAddValueTaxReturn currency="000">0
    </AmountBaseAddValueTaxReturn>
    <ConsumptionTaxAmount currency="000">0</ConsumptionTaxAmount>
    <AddValueTaxAmount currency="000">0</AddValueTaxAmount>
    <CostTransactionFlag>String</CostTransactionFlag>
    <EMVTransactionFlag>String</EMVTransactionFlag>
    <DomesticCBReasonCode>String</DomesticCBReasonCode>
    <ArgDeferredSettlementDateOfOriginal>1957-08-13
    </ArgDeferredSettlementDateOfOriginal>
    <ChileCountryCd>a</ChileCountryCd>
    <ChileSaleTypeIndicator>0</ChileSaleTypeIndicator>
    <ChileNumberOfInstallmentPayments>0
    </ChileNumberOfInstallmentPayments>
    <ChileInstallmentPaymentNumber>0
    </ChileInstallmentPaymentNumber>
    <ChileIssuerPromoFlag>0</ChileIssuerPromoFlag>
    <ChileDeferredFlag>0</ChileDeferredFlag>
    <ChileDeferredPeriod>0</ChileDeferredPeriod>
    <ChileGracePeriodFlag>0</ChileGracePeriodFlag>
    <ChileGracePeriod>0</ChileGracePeriod>
    <ChileInstallmentPaymentTotalAmt>0
    </ChileInstallmentPaymentTotalAmt>
```

```
<ChileInstallmentPaymentAmt>0</ChileInstallmentPaymentAmt>
<ChileInstallmentInterestRate>0
</ChileInstallmentInterestRate>
<ChileVATRate>0</ChileVATRate>
<ChileInterchangeReimbursementFee>0
</ChileInterchangeReimbursementFee>
<ChileVATNationalInterchangeReimbursementFee>0
</ChileVATNationalInterchangeReimbursementFee>
<ChileDeferredSettlementDate>1957-08-13
</ChileDeferredSettlementDate>
<ChileDeferredSettlementDateOfOriginal>1957-08-13
</ChileDeferredSettlementDateOfOriginal>
</DisputeNNSDetailsData>
</ResponseData>
</SIGetDisputeFinancialDetailsResponse>
```

### Request Message—REST-based Services

```
{
  "SIGetDisputeFinancialDetailsRequest": {
    "RequestHeader": {
      "MemberRole": "A"
    },
    "requestData": {
      "VisaCaseNumber": 1234567890,
      "DisputeFinId": 12345
    }
  }
}
```

### Response Message—REST-based Services

```
{
  "SIGetDisputeFinancialDetailsResponse": {
    "Status": [
      {
        "Code": "a",
        "Message": "a",
        "ErrorIn": "String"
      }
    ],
    "responseData": {
      "IssuerCaseNumber": "a",
      "AcquirerCaseNumber": "a",
      "TransactionInfo": {
        "AccountNumber": "000000000000",
        "Token": "0000000000000000",
        "TokenAssuranceLevel": "a",
        "Transaction": {
          "@id": "a",
          "TranDate": "2017-08-13",
          "Amount": {
            "value": 3000,
            "currency": "840"
          }
        }
      },
      "MerchantName": "a",
      ...
    }
  }
}
```

```
"MerchantCity": "a",
"MerchantCountrySubEntityCode": "a",
"MerchantCountryCode": "AA",
"MerchantPostalCode": "a",
" MCC": "4111",
"ARN": "00000000000000000000000000000000",
"ECMOTO": "a",
"ReimbursementAttribute": "a",
"SpecialConditionInd": "a",
"CPD": "2017-08-13",
"TransactionCategory": "F",
"AcquirerBID": "00000000",
"NetworkID": "0002",
"AcquirerBIN": "000000",
"RetrievalReferenceNumber": "000000000001",
"SystemTraceAuditNumber": "000001",
"CardholderName": "a",
"TranDestAmt":{
  "value": 3000,
  "currency": "840"
},
"ProcessingCode": "aa",
"IssuerName": "String",
"AcquirerName": "String",
"AccountType": "a",
"AccountType1Identification": "a",
"DCCInd": "a",
"DCCIndDescription": "a",
"DomesticProcessingInd": "true",
"DisputeJurisdiction":{
  "@jurisdictionCd": "REG",
  "Jurisdiction": "String",
  "IssuerRegion": "LAC",
  "AcquirerRegion": "LAC",
  "CountryCd": "AA"
},
"BusinessAppId": "aa"
},
"VisaCaseNumber": 1234567890,
"DisputeFinId": 123456,
"VROLFinancialID": 123456789,
"DisputeFinancialData":{
  "DisputeCategory": 11,
  "DisputeCategoryDesc": "Authorization",
  "DisputeCondition": 9,
  "DisputeConditionDesc": "String",
  "SrcAmt":{
    "value": 3000,
    "currency": "840"
  },
  "DestAmt":{
    "value": 3000,
    "currency": "840"
  },
  "SettlementAmount":{
    "value": 3000,
    "currency": "840"
  }
}
```

```
        },
        "PartialInd": "a",
        "CPD": "2016-01-01",
        "SettlementDate": "2016-01-01",
        "ChargebackRefNum": "000000",
        "DocumentationInd": "String",
        "MemberMsgEditText": "a",
        "ExcludedTranIdReason": "a",
        "CRSProcessingCode": "a",
        "RetrievalReferenceNumber": "000000000001",
        "SystemTraceAuditNumber": "000001",
        "DisputeStatus": "P1",
        "VROLBundleID": 0,
        "UsageCode": 0,
        "Code": "a",
        "Message": "a"
    },
    "DisputeMerchantData":{
        "MerchantName": "a",
        "MerchantNameKorea": "a",
        "MerchantNameSweden": "a",
        "JapanTCR2MerchantNameKatakana": "a",
        "JapanTCR2MerchantNameKanji": "a",
        "JapanTCR2AcquiringBankMerchantNumber": "a",
        "MerchantId": 0,
        "MerchantType": 0,
        "DepartmentId": "a",
        "SpecialMerchantIdentifier": "a",
        "MerchantLocation": "a",
        "MerchantLocationKorea": "a",
        "MerchantCitySweden": "a",
        "MerchContactPhone": "a",
        "IrfCategory": 0,
        "IRFTax": 0,
        "MexicoAccountType": "a",
        "IRFType": "01",
        "MCC": "0000",
        "POSCondCode": "a",
        "MerchantVerificationValue": "a",
        "CardAcceptorId": "a",
        "AcquirerBID": "00000000",
        "TerminalId": "a",
        "ARN": "00000000000000000000000000000000"
    },
    "DisputePOSIInfoData":{
        "DraftTransactionCode": "String",
        "CatInd": "a",
        "PosEntryModeCd": "a",
        "PosEnvironment": "a",
        "PosTermCapability": "a",
        "ECMOTO": "a",
        "RequestedPaymentServicesInd": "a",
        "PrepaidCardInd": "a",
        "PurchaseIdentifierFmt": "String",
        "PurchaseIdentifier": "String",
        "PaymentInd": "a",
        "CardTypeLocal": "a",
```

```
"ElectronicCommerceGoodsInd": "a",
"ServiceDevelopmentField": "String",
"ProductId": "a",
"SpndQlfdInd": "a",
"ProgramId": "a",
"TerminalCapabilityProfile": "a",
"RemoteTerminalInd": "String",
"JapanTCR2NetSalesAmount": 0,
"JapanTCR2NetGrossFlag": "a"
},
"DisputeAuthInfoData": {
  "AuthRespCode": "a",
  "AuthCode": "a",
  "AuthAmt": {
    "value": 3000,
    "currency": "840"
  },
  "TotalAuthAmount": {
    "value": 3000,
    "currency": "840"
  },
  "AuthSrcCode": "String",
  "FloorLimitInd": "a",
  "CardholderIdMethod": "a",
  "AvsResponseCode": "a",
  "PcasInd": "a",
  "CrbExceptionFileInd": "a",
  "SpecialConditionInd": "a",
  "RISIInd": "a",
  "MerchantTranInd": "a",
},
"DisputeProcessingSettingsData": {
  "ReimbursementAttribute": 0,
  "NetworkIdentificationCode": "String",
  "ContactInfo": "a",
  "AdjustmentProcessingInd": "a",
  "NationalReimbursementFee": {
    "value": 3000,
    "currency": "840"
  },
  "NationalTax": 0,
  "InterchangeReimbursementFeeMultiplier": "00",
  "SettlementType": 0,
  "AuthCharCd": "a",
  "MerchantVolumeInd": "a",
  "FeeProgramInd": "a",
  "ChargebackRightsInd": "a",
  "SettlementFlag": "a",
  "InternationalFeeInd": "a",
  "IssuerInterchangeReimbursementFee": {
    "@currency": "000",
    "@IssuerInterchangeReimbursementFeeDebitCreditInd": "CR",
    "$": 0
  },
  "IssuerCashbackInterchangeReimbursementFee": {
    "@currency": "000",
    "@IssuerCashbackInterchangeReimbursementFeeDebitCreditInd": "DR",
    "$": 0
  }
}
```

```
        "$": 0
    },
    "IssuerInterchangeReimbursementFeeSSBC": {
        "@currency": "000",
        "$": 0
    },
    "IssuerInterchangeReimbursementFeeSSBCDebitCreditInd": "CR",
    "IssuerCashbackInterchangeReimbursementFeeSSBC": {
        "@currency": "000",
        "$": 0
    },
    "IssuerCashbackInterchangeReimbursementFeeSSBCDebitCreditInd": "CR",
    "AcquirerInterchangeReimbursementFee": {
        "@currency": "000",
        "@AcquirerInterchangeReimbursementFeeDebitCreditInd": "CR",
        "$": 0
    },
    "AcquirerCashbackInterchangeReimbursementFee": {
        "@currency": "000",
        "@AcquirerCashbackInterchangeReimbursementFeeDebitCreditInd": "DR",
        "$": 0
    },
    "AcquirerInterchangeReimbursementFeeSSBC": {
        "@currency": "000",
        "$": 0
    },
    "AcquirerInterchangeReimbursementFeeSSBCDebitCreditInd": "DR",
    "AcquirerCashbackInterchangeReimbursementFeeSSBC": {
        "@currency": "000",
        "$": 0
    },
    "AcquirerCashbackInterchangeReimbursementFeeSSBCDebitCreditInd": "DR",
    "InterchangeFeeAmount": 0,
    "AgentUniqueId": "String"
},
"DisputeTransactionDetailsData": {
    "TransactionCodeQualifier": "String",
    "OrigTranCPD": "2017-08-13",
    "MemberPurchaseAgreementDate": "2017-08-13",
    "InstallmentPaymentCount": "",
    "NumberOfPayments": 0,
    "MerchantDiscountRate": 0,
    "PaymentMethod": "",
    "LocalAuthNum": 0,
    "CashbackAmount": {
        "@currency": "000",
        "$": 0
    },
    "BusinessAppId": "aa",
    "TerminalEntryCapability": 0,
    "SurchargeAmount": 0,
    "NetworkID": "0002",
    "SwitchRsnCode": "a",
    "MessageType": "aaaa",
    "MemberSuppliedFee": {
        "@currency": "000",
        "@MemberSuppliedFeeDebitCreditInd": "CR",
    }
}
```

```
"$": 0
},
"DCCIInd": "a",
"AccountType1Identification": "a",
"TransactionType": "String",
"AccountType": "a"
},
"DisputeNNSDetailsData": {
    "JapanTCR2NumberOfBonusPayments": 0,
    "JapanTCR2BonusAmount": 0,
    "JapanTCR2BonusCode": "a",
    "JapanTCR2BonusMonthSign": "a",
    "PaymentCd": "01",
    "JapanTCR2PaymentMode": 0,
    "JapanTCR2TaxAndMailingCharge": 0,
    "JapanTCR2SalesDraftReferenceNumber": 0,
    "JapanTCR2TextMessage": "a",
    "JapanTCR2AcquirerInvoiceAmount": 0,
    "JapanTCR2NationalChargebackReasonCode": "a",
    "IRFCalculationBaseAmt": {
        "@currency": "000",
        "$": 0
    },
    "AmountBaseAddValueTaxReturn": {
        "@currency": "000",
        "$": 0
    },
    "ConsumptionTaxAmount": {
        "@currency": "000",
        "$": 0
    },
    "AddValueTaxAmount": {
        "@currency": "000",
        "$": 0
    },
    "CostTransactionFlag": "String",
    "EMVTransactionFlag": "String",
    "DomesticCBReasonCode": "String"
    "ArgDeferredSettlementDateOfOriginal": "1957-08-13",
    "ChileCountryCd": "a",
    "ChileSaleTypeIndicator": "0",
    "ChileNumberOfInstallmentPayments": 0
    "ChileInstallmentPaymentNumber": 0
    "ChileIssuerPromoFlag": 0
    "ChileDeferredFlag": 0
    "ChileDeferredPeriod": 0
    "ChileGracePeriodFlag": 0
    "ChileGracePeriod": 0
    "ChileInstallmentPaymentTotalAmt": {
        "@currency": "000",
        "$": 0
    },
    "ChileInstallmentPaymentAmt": {
        "value": 3000,
        "currency": "840"
    },
    "ChileInstallmentInterestRate": {
```

```
        "value": 3000,
        "currency": "840"
    },
    "ChileVATRate":{
        "value": 3000,
        "currency": "840"
    },
    "ChileInterchangeReimbursementFee":{
        "value": 3000,
        "currency": "840"
    },
    "ChileVATNationalInterchangeReimbursementFee":{
        "value": 3000,
        "currency": "840"
    },
    "ChileDeferredSettlementDate": "1957-08-13",
    "ChileDeferredSettlementDateOfOriginal": "1957-08-13"
}
}
```

## SIGetDisputeFinancialAdviceDetailsRequest

SIGetDisputeFinancialAdviceDetailsRequest allows RTSI users to request the dispute financial advice data in XML.

The following elements should be considered when retrieving the financial advice details.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.
DisputeAdviceId	A unique VROL-generated ID for each financial advice.

### Request Message—SOAP-based Services

```
<SIGetDisputeFinancialAdviceDetailsRequest>
    <RequestHeader>
        <MemberRole>A</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1234567890</VisaCaseNumber>
        <DisputeAdviceId>123456</DisputeAdviceId>
    </requestData>
</SIGetDisputeFinancialAdviceDetailsRequest>
```

### Response Message—SOAP-based Services

SIGetDisputeFinancialAdviceDetailsResponse is the response service that contains the dispute financial advice details in a case and is displayed as a response to the corresponding retrieval request.

On the issuer side, the response shows the issuer version of dispute amount and currencies. On the acquirer side, the response shows the acquirer version of dispute amount and currencies.

```
<SIGetDisputeFinancialAdviceDetailsResponse>
  <Status>
    <Code>I-300000000</Code>
    <Message>Successfully completed Operation.</Message>
  </Status>
  <responseData>
    <TransactionInfo>
      <AccountNumber>410000000000</AccountNumber>
      <Transaction id="4000000000000001">
        <TranDate>2017-05-24</TranDate>
        <Amount currency="344">8.24</Amount>
      </Transaction>
      <MerchantName>MERCHANT NAME</MerchantName>
      <MerchantCity>MERCHANT CITY</MerchantCity>
      <MerchantCountryCode>NL</MerchantCountryCode>
      <MerchantPostalCode>10000</MerchantPostalCode>
      <MCC>4121</MCC>
      <ARN>700000000000000000000001</ARN>
      <ECMOTO>7</ECMOTO>
      <ReimbursementAttribute>0</ReimbursementAttribute>
      <CPD>2018-08-16</CPD>
      <TransactionCategory>D</TransactionCategory>
      <AcquirerBID>00000001</AcquirerBID>
      <NetworkID>0002</NetworkID>
      <AcquirerBIN>000002</AcquirerBIN>
      <RetrievalReferenceNumber>000000000001
      </RetrievalReferenceNumber>
      <SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
      <CardholderName>cardholder name</CardholderName>
      <TranDestAmt currency="344">33.08</TranDestAmt>
      <IssuerName>issuer name</IssuerName>
      <AcquirerName>acquirer name</AcquirerName>
      <AccountType>0</AccountType>
      <DisputeJurisdiction jurisdictionCd="INT">
        <Jurisdiction>INTERREGIONAL-AP, VE</Jurisdiction>
        <IssuerRegion>AP</IssuerRegion>
        <AcquirerRegion>Europe</AcquirerRegion>
        <CountryCd>HK</CountryCd>
      </DisputeJurisdiction>
    </TransactionInfo>
    <VisaCaseNumber>1234567890</VisaCaseNumber>
    <DisputeAdviceId>123456</DisputeAdviceId>
    <ProcessingCode>string</ProcessingCode>
    <CPD>2019-07-30</CPD>
    <DisputeCategory>10</DisputeCategory>
    <DisputeCondition>4</DisputeCondition>
    <DisputeStatus>F1</DisputeStatus>
    <DisputeTransactionCode>15</DisputeTransactionCode>
    <DisputeAmountInUSD currency="344">33.08</DisputeAmountInUSD>
    <VROLFinancialID>10.4-1000</VROLFinancialID>
    <MerchantName>Any merchant</MerchantName>
    <MerchantCity>Any city</MerchantCity>
    <MerchantCountryCode>NL</MerchantCountryCode>
    <MerchantPostalCode>12345</MerchantPostalCode>
    <MCC>4121</MCC>
    <CardAcceptorId>4000000000000001</CardAcceptorId>
    <AcquirerBID>00000001</AcquirerBID>
```

```
<ARN>00000000000000000000000000000001</ARN>
<AcquirerBIN>000002</AcquirerBIN>
<NetworkID>0002</NetworkID>
<PosEntryModeCd>01</PosEntryModeCd>
<AuthCode>500008</AuthCode>
<ReimbursementAttribute>0</ReimbursementAttribute>
<IssuerFinancialBIN>400004</IssuerFinancialBIN>
<AcquirerFinancialBIN>400000</AcquirerFinancialBIN>
<SettlementFlag>0</SettlementFlag>
<CrissCrossIndicator>N</CrissCrossIndicator>
<AccountNumber>4000000000000004</AccountNumber>
<TransactionID>0</TransactionID>
<MultipleClearingSequenceCount>1</MultipleClearingSequenceCount>
<TranDate>2018-08-15</TranDate>
<SrcAmt currency="344">33.08</SrcAmt>
<DestAmt currency="978">3.78</DestAmt>
<ProductId>P</ProductId>
<UsageCode>9</UsageCode>
</ResponseData>
</SIGetDisputeFinancialAdviceDetailsResponse>
```

### Request Message—REST-based Services

```
{
  "SIGetDisputeFinancialAdviceDetailsRequest": {
    "RequestHeader": {
      "MemberRole": "A"
    },
    "requestData": {
      "VisaCaseNumber": 1234567890,
      "DisputeAdviceId": 123456
    }
  }
}
```

### Response Message—REST-based Services

```
{
  "SIGetDisputeFinancialAdviceDetailsResponse": {
    "Status": [
      {
        "Code": "I-300000000",
        "Message": "Successfully completed Operation."
      }
    ],
    "responseData": {
      "TransactionInfo": {
        "AccountNumber": "410000000000",
        "Transaction": {
          "TranDate": "2018-08-15",
          "Amount": {
            "value": 33.08,
            "currency": "344"
          }
        },
        "id": "4000000000000001"
      },
      "MerchantName": "MERCHANT NAME",
      "MerchantCity": "MERCHANT CITY",
    }
  }
}
```

```
"MerchantCountryCode": "NL",
"MerchantPostalCode": "10000",
"MCC": "4121",
"ARN": "70000000000000000000000000000000",
"ECMOTO": "7",
"ReimbursementAttribute": "0",
"CPD": "2018-08-16",

"TransactionCategory": "D",
"AcquirerBID": "00000001",
"NetworkID": "0002",
"AcquirerBIN": "000002",
"RetrievalReferenceNumber": "000000000001",
"SystemTraceAuditNumber": "000001",
"CardholderName": "cardholder name",
"TranDestAmt":{
    "value": 33.08,
    "currency": "344"
},
"IssuerName": "issuer name",
"AcquirerName": "acquirer name",
"AccountType": "0",
"DisputeJurisdiction":{
    "Jurisdiction": "INTERREGIONAL-AP,VE",
    "IssuerRegion": "AP",
    "AcquirerRegion": "Europe",
    "CountryCd": "HK",
    "jurisdictionCd": "INT"
},
"VisaCaseNumber": 1234567890,
"DisputeAdviceId": 123456,
"CPD": "2019-07-30",
"DisputeCategory": {"value": "10"},
"DisputeCondition": "4",
"DisputeStatus": "F1",
"DisputeTransactionCode": "15",
"DisputeAmountInUSD":{
    "value": 33.08,
    "currency": "344"
},
"VROLFinancialID": "10.4-1000",
"IssuerCaseNumber": "2654456",
"MerchantName": "Any merchant",
"MerchantCity": "Any city",
"MerchantCountryCode": "NL",
"MerchantPostalCode": "12345",
"MCC": "4121",
"CardAcceptorId": "4000000000000001",
"AcquirerBID": "00000001",
"ARN": "00000000000000000000000000000001",
"AcquirerBIN": "000002",
"NetworkID": "0002",
"PosEntryModeCd": "01",
"AuthCode": "500008",
"ReimbursementAttribute": "0",
"IssuerFinancialBIN": "400004",
```

```
"AcquirerFinancialBIN": "400000",
"SettlementFlag": "0",
"CrissCrossIndicator": "N",
"AccountNumber": "4000000000000004",
"TransactionID": "0",
"MultipleClearingSequenceCount": 1,
"TranDate": "2018-08-15",
"SrcAmt": {
    "value": 33.08,
    "currency": "344"
},
"DestAmt": {
    "value": 3.78,
    "currency": "978"
},
"ProductId": "P",
"UsageCode": 9
}
}
```

## Financial Fields Returned in Immediate Response

This section lists the various financial fields that are displayed in the immediate response of the following RTSI Web services along with VisaCaseNumber and other IDs, as applicable:

- SISubmitDisputeQuestionnaireResponse
- SICreateDisputeResponseResponse
- SIAcceptDisputeResponse (see [Chapter 17, Queues](#))
- SICreateDisputePreArbResponseResponse (see [Chapter 10, Dispute Pre-Filings](#))

**Table 8-1: Financial Fields**

XML Element Names	Notes
VisaCaseNumber	
VROLFinancialSentInd	Domain values: <ul style="list-style-type: none"><li>• True=Indicates that a dispute-related financial was generated by VROL.</li><li>• False=Indicates that a dispute-related financial is required but the member needs to generate outside of VROL using the supplied fields or values.</li></ul> Field not present indicates the scenarios where the response could, but does not trigger the need for a financial. For example, SIAcceptDispute response for an incoming allocation dispute.
VROLFinancialID	
FinancialTransactionType	
FinancialAmt	
DisputeCategory	

**Table 8–1: Financial Fields (cont'd.)**

XML Element Names	Notes
DisputeCondition	See the IES - Dispute Financial tab for the domain values
VROLBundleCaseNumber	
TransactionID	
IssuerCaseNumber	
AcquirerCaseNumber	
OpposerExpectedRespDate	
DisputeFinId	

# Adjustments

9

VROL enables SMS acquirers to send and receive adjustment transactions through V.I.P.

## Adjustment Process

This section describes the process and XML operations used by acquirers to perform the task.

### Send Adjustment

(SMS)

1. Prior to sending an adjustment, acquirer may perform a TI search and create a case from the TI results. See [Chapter 5, Case Creation](#), for the information to perform this TI search, create a case, and retrieve the associated VROL Case Number.

**Important:**

Once transaction information is established in a case, there is no need to provide it in subsequent actions. In addition, <TransactionInfo> is no longer required if a case is created from a TI result.

2. To create an adjustment, acquirer sends request SISubmitAdjustmentRequest with adjustment financial information. To update an existing case, the VROL Case Number is required.

VROL sends response with the VROL Case Number, Adjustment ID, and success status.

3. Once the adjustment has been processed by BASE II/SMS and EM (CPS transaction), the acquirer can determine its success as described in the following sections:
  - [Retrieve Adjustment Network Approval Status](#)
  - [Check for Adjustment Rejects](#)

## Retrieve Adjustment Network Approval Status

### (SMS Only)

Network approval is not returned in the response when sending an adjustment from VROL. The sender can check the reject queue, but may prefer confirmation that a financial was processed successfully by V.I.P. (and EM when applicable for CPS).

Acquirers that send adjustments from VROL can determine if the transactions sent have been approved or rejected by V.I.P. by sending the detail request each day until they receive a status in the response that the approval or rejection has occurred. This process typically takes 1–2 days. If the transaction is still waiting for a response from V.I.P., it will be indicated in the response.

1. To retrieve adjustment network approval status, acquirer sends request SIGetAdjustmentDetailsRequest using VROL Case Number, Adjustment ID, and QN type. The <ReturnDetails> field is used to indicate that the user wants to receive the network approval status message.
2. VROL sends response SIGetFinancialDetailsResponse including the network status.

## Check for Adjustment Rejects

1. To check for adjustment FINs rejected by V.I.P. (if applicable), acquirer sends request SIGetQueueRequest for MY\_REJECTS or ALL\_REJECTS queue.  
VROL sends response with VROL Case Number, Reject ID, and success status.
2. Acquirer sends request SIGetAdjustmentDetailsRequest to confirm that the FIN was successful.  
VROL sends response with details.

## Repair Adjustment Rejects

1. To check for adjustment rejects, acquirer sends a queue request SIGetQueueRequest for MY\_REJECTS or ALL\_REJECTS queue.  
VROL sends response with VROL Case Number, Reject ID, and success status.
2. To repair the adjustment reject, acquirer sends request SISubmitAdjustmentRequest for the reject to be repaired along with adjustment financial information.
  - If the case contains a single unrepaired reject of the same type as the request, VROL marks the reject as repaired.
  - If the case contains multiple unrepaired rejects of the same type as the request, VROL marks the most recent reject as repaired.
  - If the case does not contain a reject of the same type as the request, VROL does not mark any transaction as repaired.

VROL adds the incoming transaction to the case and sends response with success status.

## Retrieve Adjustments

1. The issuer retrieves incoming adjustments in one of the following ways:
  - a. User sends queue request SIGetQueueRequest to retrieve incoming adjustments from either ALL\_INCOMING\_ADJUSTMENTS or queue. See [Chapter 17, Queues](#), for more information.
  - b. Some issuers opt to receive adjustments using Bulk SI Download files.

2. To retrieve AdjustmentID, the issuer sends a Hypersearch request. See [Chapter 19, Hypersearch Requests](#), for more information.
3. To retrieve adjustment details, the issuer sends request SIGetAdjustmentDetailsRequest using the VROL Case Number and AdjustmentID. VROL sends response with details of the adjustment.
4. To close the adjustment and remove it from the queue, issuer sends close adjustment request SICloseTransactionRequest with CloseItem Type as ADJ. This step prevents the transaction from being returned again in the next queue request. See [Chapter 17, Queues](#), for more information.

**Note:** *SICloseTransactionRequest closes the transaction and removes it from its respective queue but does not close the case. For more information, see [SICloseTransactionRequest](#).*

VROL sends response with success status.

## Adjustment Operations

RTSI exposes the following adjustment operations.

**SISubmitAdjustmentRequest**—Used by acquirer to submit adjustment request.

**SIGetAdjustmentDetailsRequest**—Used by acquirer to retrieve adjustment details.

**SIIgnoreRejectRequest**—Used to ignore a rejected adjustment transaction. See [Chapter 17, Queues](#), for details.

### SISubmitAdjustmentRequest

This operation allows SMS members to submit an adjustment transaction. It is used in request and response messages as shown in the following sample messages.

Refer to the table in [Chapter 3, RTSI Request Operations](#) for XML elements to consider when submitting an adjustment request.

**Note:** *Only XML elements associated with SMS transactions apply to adjustments.*

#### Request Message—SOAP-based Services

```
<SISubmitAdjustmentRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <AdjustmentData>
      <AdjustmentInfo>
        <AdjustmentReasonCode>2004</AdjustmentReasonCode>
        <ProcessingCode>22</ProcessingCode>
        <AdjustmentAmount>400.0</AdjustmentAmount>
        <TranCurrency>0840</TranCurrency>
        <ContactName>Mary</ContactName>
        <ContactPhone>1234567890</ContactPhone>
        <GFCollectionInd>true</GFCollectionInd>
        <Comments>Test-Adjustment Transaction</Comments>
      </AdjustmentInfo>
    </AdjustmentData>
  </requestData>
</SISubmitAdjustmentRequest>
```

**Response Message—SOAP-based Services**

```
<SISubmitAdjustmentResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <AdjustmentID>0000000</AdjustmentID>
    </responseData>
</SISubmitAdjustmentResponse>
```

**Request Message—REST-based Services**

```
{
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": "0000000000",
        "AdjustmentData": {
            "TransactionInfo": {
                "AccountNumber": "111111111111",
                "Token": "11111111111111",
                "TokenAssuranceLevel": "a",
                "Transaction": {
                    "id": "a",
                    "TranDate": "2016-01-01",
                    "Amount": {
                        "currency": "111",
                        "Text": 1
                    }
                },
                "MerchantName": "a",
                "MerchantCity": "a",
                "MerchantCountrySubEntityCode": "a",
                "MerchantCountryCode": "AA",
                "MerchantPostalCode": "a",
                "MCC": "1111",
                "ARN": "11111111111111111111111111111111",
                ...
            },
            "AdjustmentInfo": {
                "AdjustmentReasonCode": "a",
                "ProcessingCode": "aa",
                "AdjustmentAmount": 1,
                "AdjustmentAmountType": "CR",
                ...
            },
            "AgentUniqueId": "A"
        }
    }
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "VisaCaseNumber": "0000000000",  
        "AdjustmentID": "0000000"  
    }  
}
```

## SIGetAdjustmentDetailsRequest

This operation allows members to request details for an adjustment transaction. It is used in request and response messages as shown in the following sample messages.

The following table provides information about some XML elements that should be considered when submitting an adjustment detail request or retrieving a response.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.  <b>Note:</b> In the schema, the VROL Case Number is <VisaCaseNumber>.
AdjustmentID	Uniquely identifies an adjustment transaction and is used to access the adjustment information in a subsequent request.
ReturnDetails  <b>Note:</b> SMS only.	The domain value of <ReturnDetails> indicates if the user wants to receive the SMS approval status message as follows: <ul style="list-style-type: none"><li>• (not provided)—Existing details only (no SMS status information).</li><li>• FULL—Existing details and SMS status information.</li><li>• STATUS_ONLY—SMS status only (no financial details)</li></ul> See " <a href="#">Network Approval Status Messages</a> " later in this chapter.

## Request Message—SOAP-based Services

```
<SIGetAdjustmentDetailsRequest>  
    <RequestHeader>...</RequestHeader>  
    <requestData>  
        <VisaCaseNumber>0000000000</VisaCaseNumber>  
        <AdjustmentID>0000000</AdjustmentID>  
    </requestData>  
</SIGetAdjustmentDetailsRequest>
```

## Response Message—SOAP-based Services

```

<SIGetAdjustmentDetailsResponse>
    <Status>...</Status>
    <responseData>
        <AdjustmentDetailsResponse>
            <VisaCaseNumber>0000000000</VisaCaseNumber>
            <TransactionInfo>...</TransactionInfo>
            <AdjustmentData>
                <AdditionalTransactionInfo>
                    ...
                </AdditionalTransactionInfo>
                <Adjustment>
                    <AdjustmentReasonCode>2004</AdjustmentReasonCode>
                    <ProcessingCode>22</ProcessingCode>
                    <AdjustmentAmount>500.00</AdjustmentAmount>
                    <AdjustmentAmountType>CR</AdjustmentAmountType>
                    <SurchargeAmount>1000.00</SurchargeAmount>
                    <SurchargeAmountType>CR</SurchargeAmountType>
                    <TranCurrency>0840</TranCurrency>
                    <Comments>Test-Adjustment Transaction</Comments>
                    <AdjustmentDate>2014-08-07</AdjustmentDate>
                    <AdjustmentReasonDesc>Acquirer correction
                    </AdjustmentReasonDesc>
                    <AdjustmentProcessingDesc>Adjustment Credit
                    </AdjustmentProcessingDesc>
                    <SettlementAmount>589.77</SettlementAmount>
                    <MerchantLocation>MerchantCity</MerchantLocation>
                    <IssuerName>Benet</IssuerName>
                    <AcquirerName>Michel</AcquirerName>
                    <ReceivingInstCountryCode>US
                    </ReceivingInstCountryCode>
                    <TerminalEntryCapability>1
                    </TerminalEntryCapability>
                </Adjustment>
            </AdjustmentData>
        </AdjustmentDetailsResponse>
    </responseData>
</SIGetAdjustmentDetailsResponse>

```

## Request Message—REST-based Services

```
{
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": "0000000000",
        "AdjustmentID": "0000000"
    }
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "adjustmentDetailsResponse": {  
            "VisaCaseNumber": "0000000000",  
            "transactionInfo": {  
                "AccountNumber": "111111111111",  
                "Token": "111111111111",  
                "TokenAssuranceLevel": "a",  
                ...  
            },  
            "adjustmentData": {  
                "additionalTransactionInfo": {  
                    "TerminalId": "a",  
                    "CardAcceptorId": "a",  
                    "MerchantPostalCode": "a",  
                    ...  
                },  
                "adjustment": {  
                    "adjustmentReasonCode": "a",  
                    "processingCode": "aa",  
                    "adjustmentAmount": 1,  
                    "adjustmentAmountType": "CR",  
                    "surchargeAmount": 1,  
                    "surchargeAmountType": "DR",  
                    "tranCurrency": "aaaa",  
                    "contactName": "a",  
                    "contactPhone": "a",  
                    "GFCollectionInd": true,  
                    "comments": "a",  
                    ...  
                },  
            },  
        },  
    },  
}
```

## Network Approval Status Messages

### SMS Only

For SMS members only, the value for ReturnDetails in the SIGetAdjustmentDetailsRequest (FULL or STATUS\_ONLY) determines the request and response messages. One of two messages appear in the StatusMessage of a successful response: Acknowledged or Pending Network Response.

If an SMS response has not yet been received by VROL from SMS at the time the RTSI user sends the request, an error is returned (E-140000010). If the RTSI user receives Pending Network Response or this error, they should repeat the request later.

**Request Message—SOAP-based Services**

```
<SIGetAdjustmentDetailsRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <AdjustmentID>0000000</AdjustmentID>
        <ReturnDetails>...</ReturnDetails>
    </requestData>
</SIGetAdjustmentDetailsRequest>
```

**Response Message—ReturnDetails=FULL—SOAP-based Services**

Following is the response if successful or no advice returned as yet from VisaNet.

```
<SIGetAdjustmentDetailsResponse>
    <Status>...</Status>
    <responseData>
        <AdjustmentDetailsResponse>
            <VisaCaseNumber>0000000000</VisaCaseNumber>
            <TransactionInfo>...</TransactionInfo>
            <AdjustmentData>
                <AdditionalTransactionInfo>
                    ...
                </AdditionalTransactionInfo>
                <Adjustment>...</Adjustment>
                <SwitchingSystemStatus>
                    <StatusMessage>Acknowledged</StatusMessage>
                </SwitchingSystemStatus>
            </AdjustmentData>
        </AdjustmentDetailsResponse>
    </responseData>
</SIGetAdjustmentDetailsResponse>
```

Following is the response if request is rejected. Note that <StatusCode> and <StatusMessage> repeat up to six times if multiple rejects/returns.

```
<SIGetAdjustmentDetailsResponse>
    <Status>...</Status>
    <responseData>
        <AdjustmentDetailsResponse>
            <VisaCaseNumber>0000000000</VisaCaseNumber>
            <TransactionInfo>...</TransactionInfo>
            <AdjustmentData>
                <AdditionalTransactionInfo>
                    ...
                </AdditionalTransactionInfo>
                <Adjustment>...</Adjustment>
                <SwitchingSystemStatus>
                    <StatusCode>RC14</StatusCode>
                    <StatusMessage>Invalid account number...</StatusMessage>
                </SwitchingSystemStatus>
            </AdjustmentData>
        </AdjustmentDetailsResponse>
    </responseData>
</SIGetAdjustmentDetailsResponse>
```

## Response Message—ReturnDetails=STATUS\_ONLY—SOAP-based Services

### Important:

The successful and reject response differ only in the value of the StatusCode and description. Care must be taken to ensure the reject response is corrected and resent.

Following is the response if successful or no advice returned as yet from VisaNet.

```
<SIGetAdjustmentDetailsResponse>
    <Status>...</Status>
    <responseData>
        <AdjustmentSwitchStatusResponse>
            <VisaCaseNumber>0000000000</VisaCaseNumber>
            <AdjustmentID>0000000</AdjustmentID>
            <SwitchingSystemStatus>
                <StatusMessage>Acknowledged</StatusMessage>
            </SwitchingSystemStatus>
        </AdjustmentSwitchStatusResponse>
    </responseData>
</SIGetAdjustmentDetailsResponse>
```

Following is the response if request is rejected. Note that <StatusCode> and <StatusMessage> repeat up to six times if multiple rejects/returns.

```
<SIGetAdjustmentDetailsResponse>
    <Status>...</Status>
    <responseData>
        <AdjustmentSwitchStatusResponse>
            <VisaCaseNumber>0000000000</VisaCaseNumber>
            <AdjustmentID>xxxxxx</AdjustmentID>
            <SwitchingSystemStatus>
                <StatusCode>RC14</StatusCode>
                <StatusMessage>Invalid account number...</StatusMessage>
            </SwitchingSystemStatus>
        </AdjustmentSwitchStatusResponse>
    </responseData>
</SIGetAdjustmentDetailsResponse>
```

## Request Message—REST-based Services

```
{
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": "0000000000",
        "AdjustmentID": 1,
        "ReturnDetails": "..."
    }
}
```

**Response Message—ReturnDetails=FULL—REST-based Services**

```
{  
  "Status": "...",  
  "responseData":{  
    "adjustmentDetailsResponse":{  
      "VisaCaseNumber": "0000000000",  
      "transactionInfo": "...",  
      "adjustmentData":{  
        "additionalTransactionInfo": "...",  
        "adjustment": "...",  
        "switchingSystemStatus":{  
          "statusMessage": "Acknowledged"  
        }  
      }  
    }  
  }  
}
```

**Response Message—ReturnDetails=STATUS\_ONLY—REST-based Services**

```
{  
  "Status": "...",  
  "responseData":{  
    "adjustmentDetailsResponse":{  
      "VisaCaseNumber": "0000000000",  
      "transactionInfo": "...",  
      "adjustmentData":{  
        "additionalTransactionInfo": "...",  
        "adjustment": "...",  
        "switchingSystemStatus":{  
          "statusMessage": "Acknowledged"  
        }  
      }  
    }  
  }  
}
```

# Dispute Pre-Filings

10

*Dispute Pre-filing* is used generically to refer to dispute Pre-Arbitration or Pre-Compliance, while *dispute case filing* is used generically to refer to dispute arbitration or compliance.

- **Pre-Arbitration**—Before filing for arbitration, the requesting member must try to resolve the dispute with the opposing member by attempting Pre-Arbitration when new information or documentation is being provided about the dispute, or when the initiator changes the dispute category. When changing the dispute category, the members can change from Consumer to Processing Error, or Processing Error to Consumer. They cannot change to either Fraud or Authorization. If Pre-Arbitration fails, the member can pursue arbitration. Pre-Arbitration can be triggered for all dispute categories.
- **Pre-Compliance**—User can initiate Pre-Compliance to dispute a violation of the *Visa Core Rules* and *Visa Product and Service Rules*. If Pre-Compliance fails, the member can pursue compliance.

See [Chapter 11, Dispute Case Filings](#) for details on arbitration and compliance.

When sending a Pre-filing, the user must give the reason for initiating the Pre-filing. The opposing party, when receiving the Pre-filing, sends a Pre-filing response, and can accept full or partial liability or continue the process by declining the Pre-filing.

## Pre-Arbitration Process

Users can initiate a Pre-Arbitration depending on the workflow and the dispute phase that they are in.

- For disputes following the allocation workflow and where VROL assigns the liability to the opposer, the opposer can respond to the incoming dispute by initiating a Pre-Arbitration.
- For disputes following the collaboration workflow, the initiator can respond to an incoming Dispute Response by initiating a Pre-Arbitration.

The acquirer can initiate Pre-Arbitration on a received dispute (allocation workflow) or the issuer can initiate Pre-Arbitration on a received dispute response (collaboration workflow) if the received dispute request has not been recalled, the response time frame has not expired, and the case has not yet advanced to a filing stage.

If a Pre-Arbitration questionnaire is not submitted within time frame, VROL generates and submits an "Acceptance Notification" for the side whose time frame has expired.

This notification is generated on the calendar day after the time frame to submit a Pre-Arbitration questionnaire has expired.

If the time frame for responding to the Pre-Arbitration has expired, VROL generates and submits a Pre-Arbitration Response (Accept Full or Accept Partial) on behalf of the non-responding side. This response is generated on the calendar day after the response time frame has expired.

## Associated Transaction Research (ATR) during Pre-Arbitration

When an issuer initiates a Pre-Arbitration questionnaire, VROL searches for additional associated transactions related to the dispute. This subsequent check is performed between those dates when the initial associated transaction research (ATR) was performed and when the Pre-Arbitration was initiated. If VROL finds additional transactions during the subsequent ATR that require the user to review or match, VROL returns the list of these transactions and allows the user to complete the required actions. See [Chapter 4, Transaction Inquiries](#) for details on associated transactions.

When an issuer performs Pre-Arbitration during the collaboration workflow, VROL allows the issuer to review Credit, Reversal, Adjustment (CRA) ATR transactions that were associated since the dispute request submission. The issuer must associate the new CRA ATR transactions before they can proceed on submitting the Pre-Arbitration request.

VROL allows the user to update the data of the associated transactions selection if there is a related saved (not submitted) Pre-Arbitration questionnaire. Only the Associated values for new associated transactions that occurred since the Dispute questionnaire submission are editable. See [Update Associated Transactions Selection Data](#).

**Note:**

- *The issuer cannot edit the four CRA ATR transactions that were previously or initially associated during the dispute request submission.*
- *If the user provides an associated value for both additional CRA ATR and previously associated CRA ATR transactions, VROL returns an informational message indicating that the system considers and applies the association value only for the new additional CRA ATR since dispute was submitted. All other association values are ignored.*

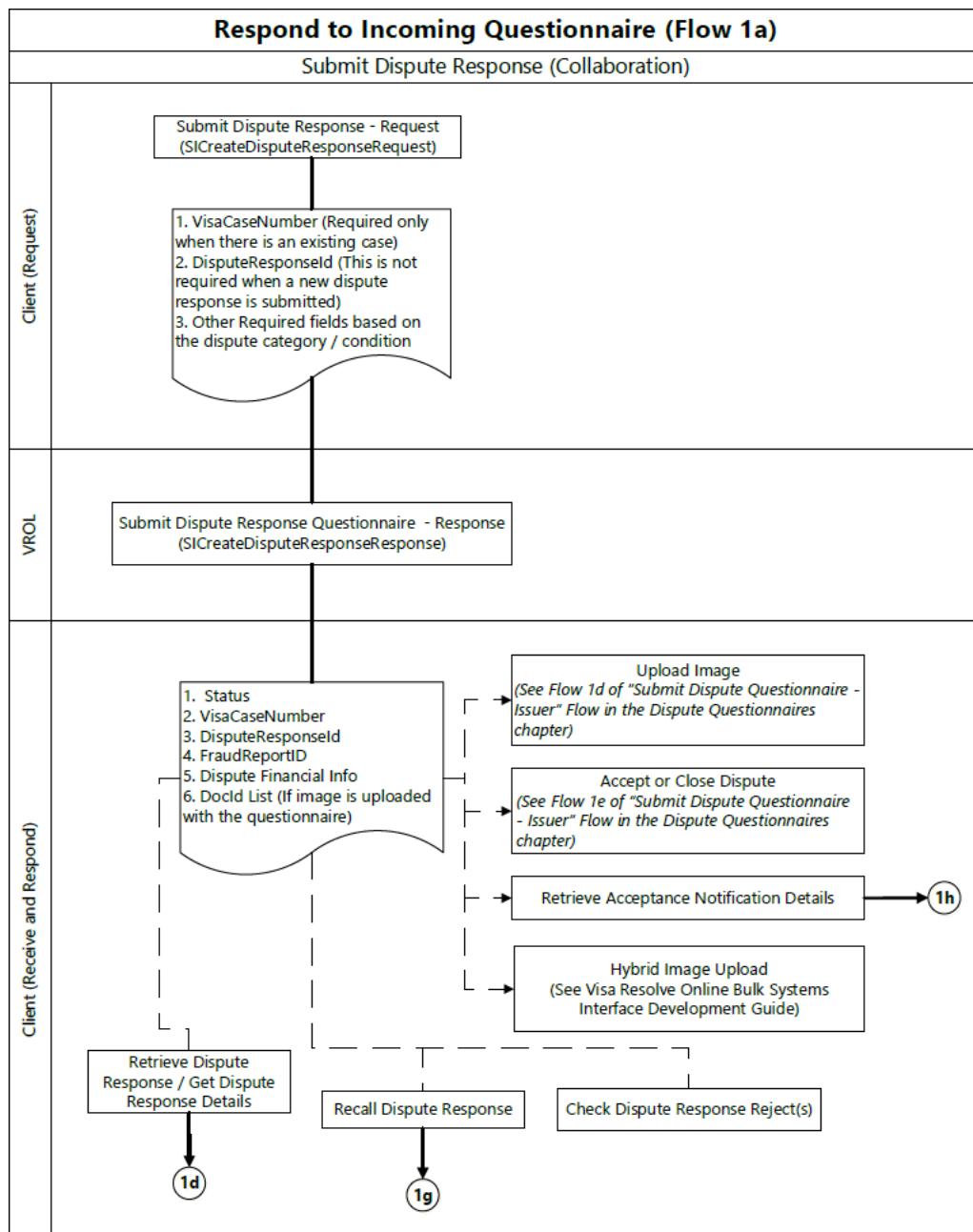
## Pre-Arbitration Flows

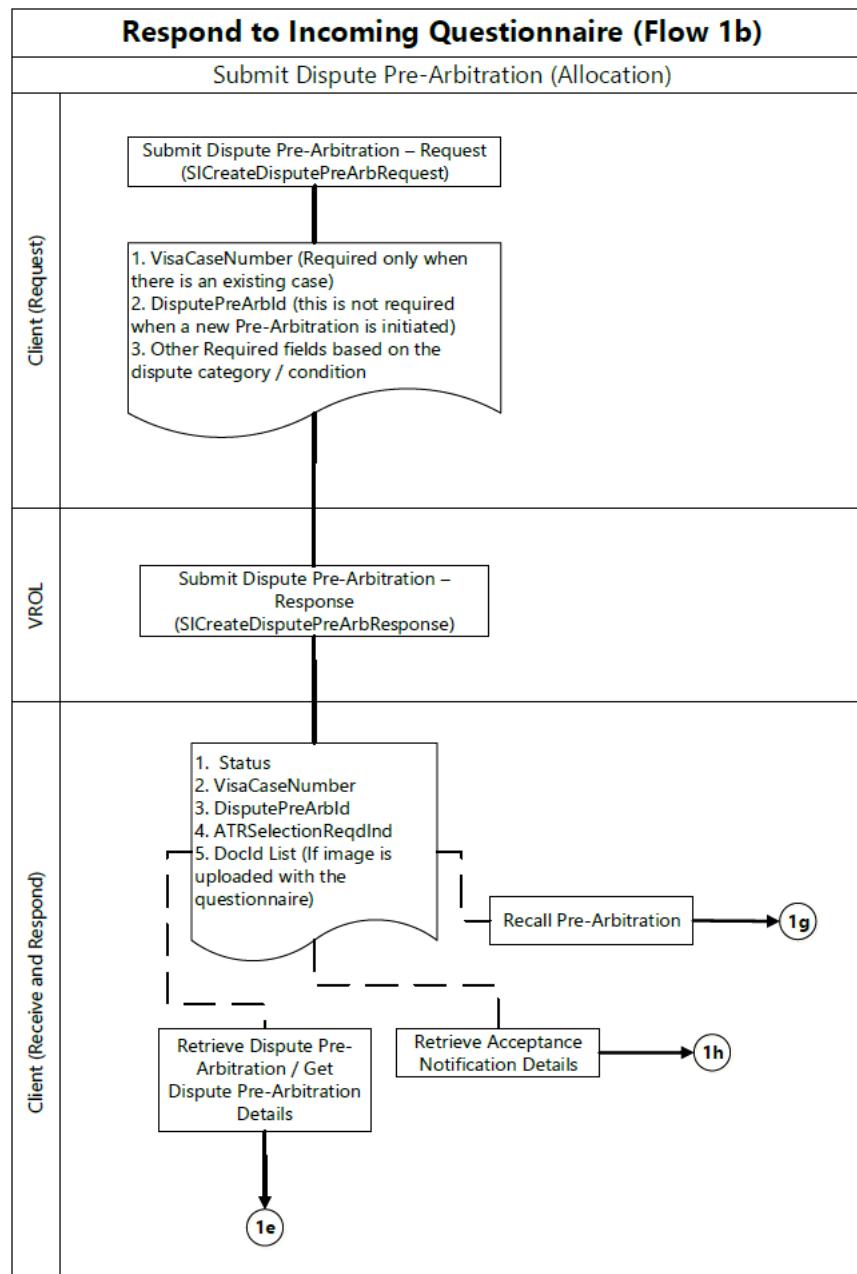
The following diagrams illustrate the Pre-Arbitration process.

**Note:**

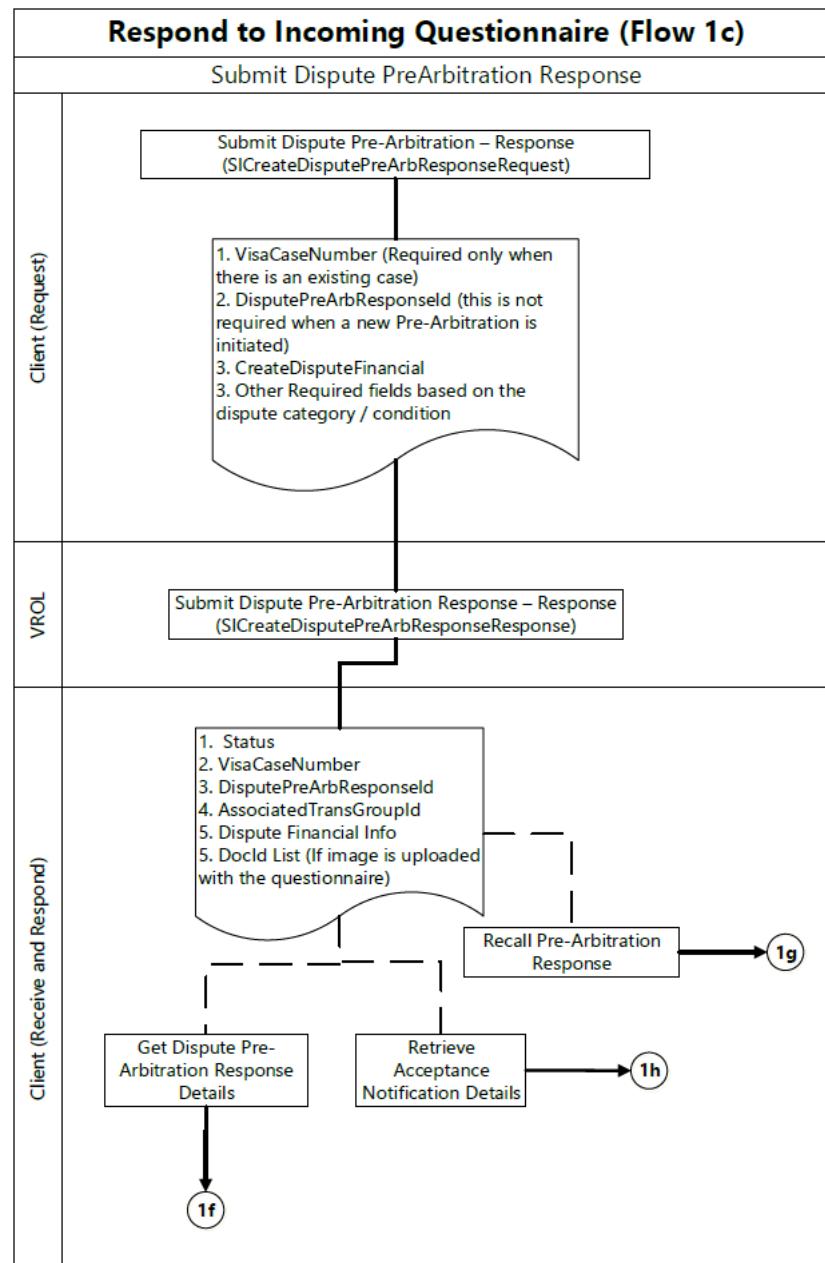
- *The diagrams list only the important elements. For the complete elements, refer to the Dispute tab of the Interface Element Specification (IES).*
- *The dashed lines in the flow refer to optional functions that the user can use, depending on the business process that the user wants to implement.*

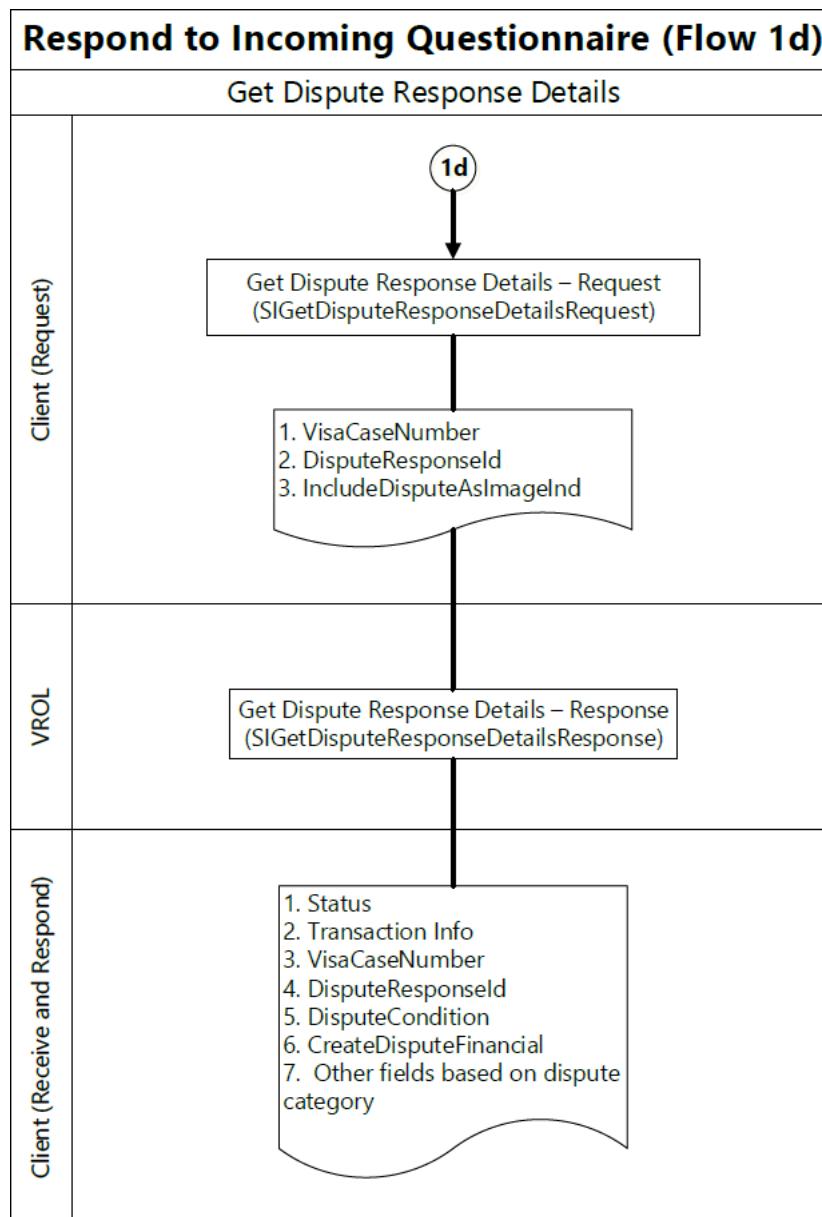
**Figure 10-1: Respond to Incoming Questionnaires—Flow 1a**



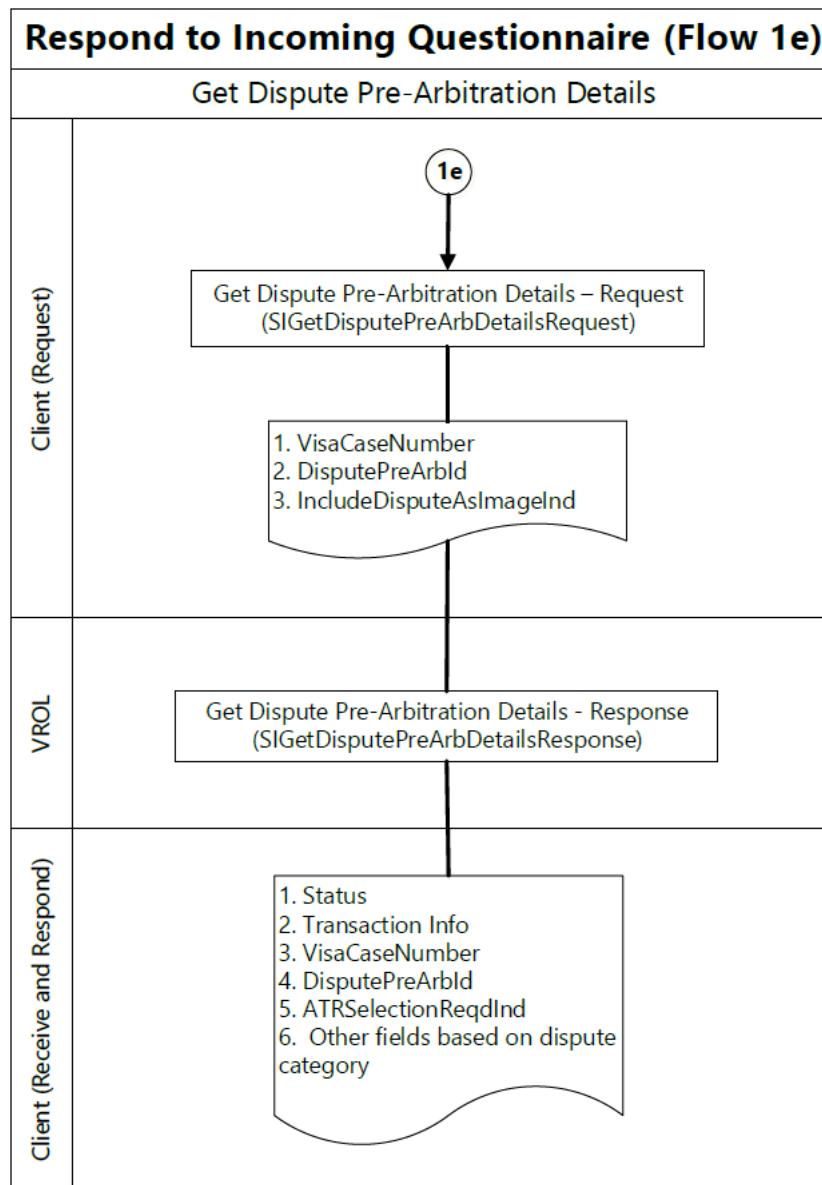
**Figure 10–2: Respond to Incoming Questionnaires—Flow 1b**

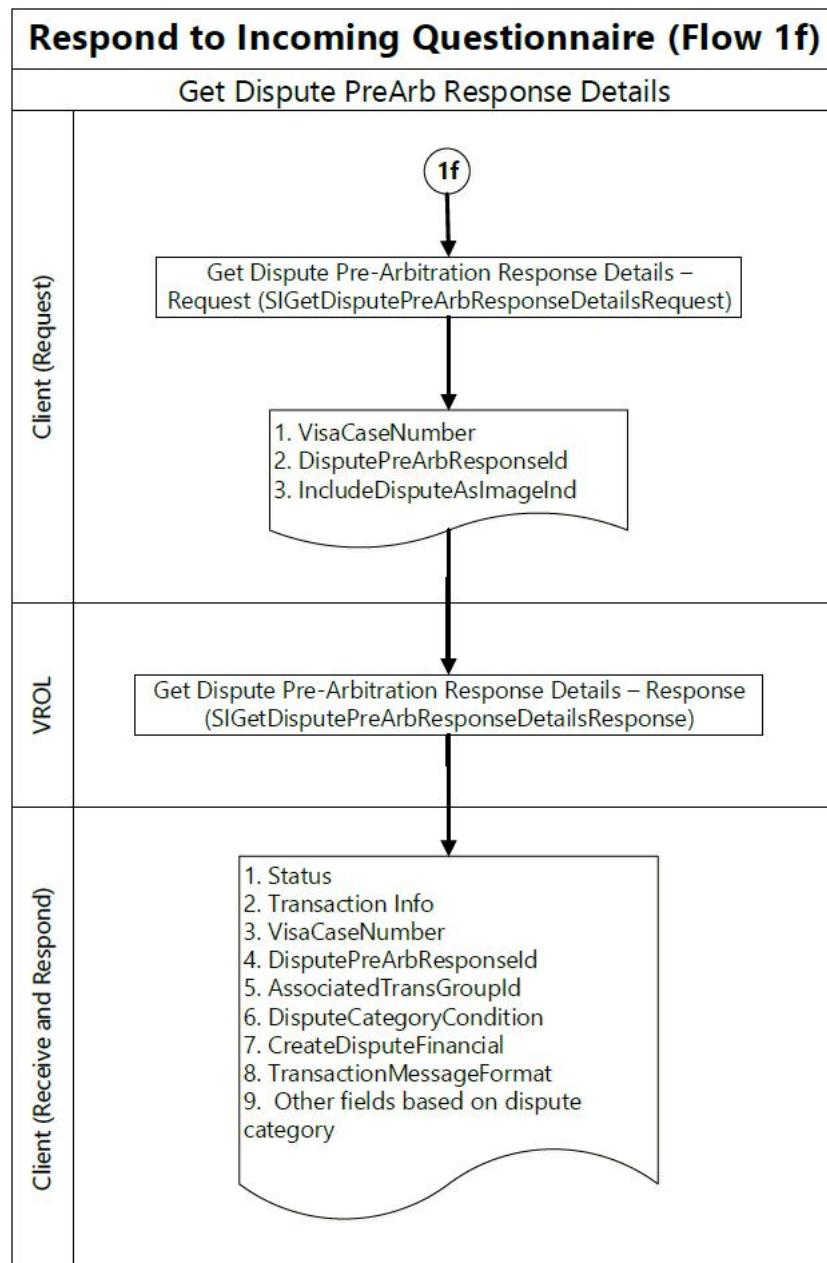
**Figure 10–3: Respond to Incoming Questionnaires—Flow 1c**



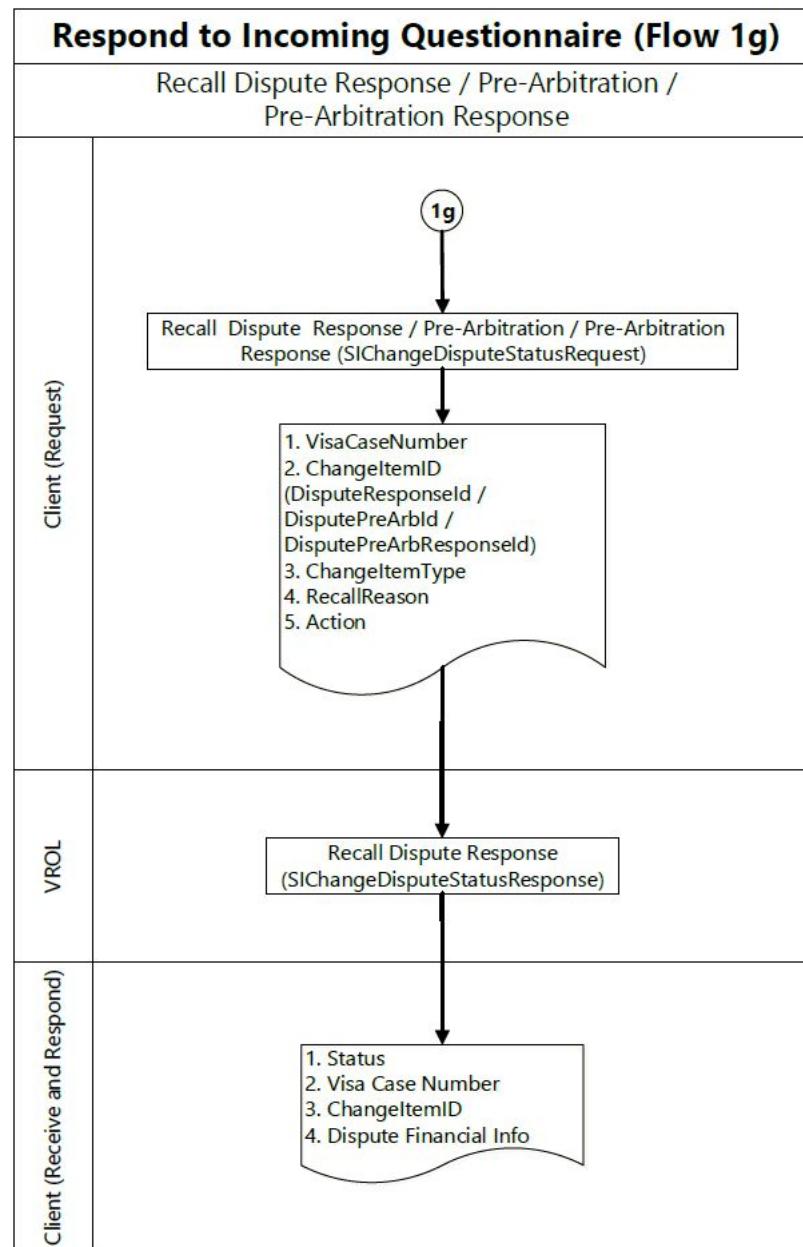
**Figure 10–4: Respond to Incoming Questionnaires—Flow 1d**

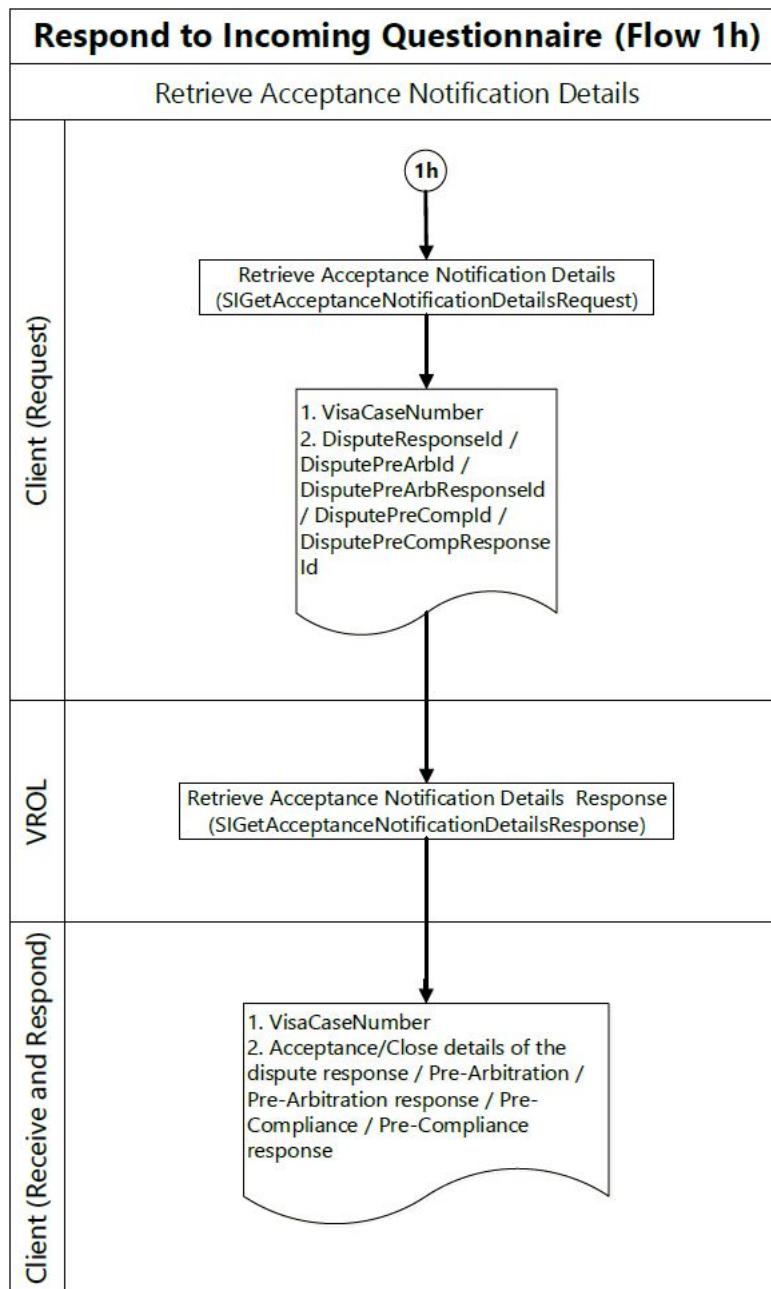
**Figure 10–5: Respond to Incoming Questionnaires—Flow 1e**



**Figure 10–6: Respond to Incoming Questionnaires—Flow 1f**

**Figure 10-7: Respond to Incoming Questionnaires—Flow 1g**



**Figure 10–8: Respond to Incoming Questionnaires—Flow 1h**

## Create Pre-Arbitration

- To create a Pre-Arbitration, the member host system triggers a SICreateDisputePreArbRequest supplying the VisaCaseNumber and Pre-Arbitration information. SICreateDisputePreArbRequest can be saved or submitted. VROL returns a SICreateDisputePreArbResponse message with the DisputePreArbId, DocIds for any attached images, AssociatedTransGroupId, and success status.

- Save

A SICreateDisputePreArbRequest message can be saved. When it is saved, it generates a DisputePreArbId.

- Submit

If a SICreateDisputePreArbRequest message is submitted for a new Pre-Arbitration, a DisputePreArbId is generated in the response.

If a SICreateDisputePreArbRequest message is submitted for a previously-saved Pre-Arbitration, there is no new DisputePreArbId generated in the response.

If an issuer initiates the Pre-Arbitration in a collaboration workflow, VROL also performs a subsequent associated transaction research (ATR) for additional transactions between the dates when ATR was performed during dispute submission and when the Pre-Arbitration was initiated. If VROL finds additional transactions during the subsequent ATR, a status message appears and the ATRSelectionReqdInd is also returned in the SICreateDisputePreArbResponse message, which can have any of the following values:

- Required

VROL saves the submitted Pre-Arbitration regardless of whether the specified action for the SICreateDisputePreArbRequest message is Save or Submit. The user must associate these additional ATR transactions using the SIAssociatedTranSelectionRequest message. See [Chapter 4, Transaction Inquiries](#) for details.

**Note:** VROL returns ATRSelectionReqdInd in the SICreateDisputePreArbResponse message only if the action in the SICreateDisputePreArbRequest message is Submit.

- NotRequired

VROL automatically associates additional ATRs found and the Pre-Arbitration is submitted successfully.

See the table in [SICreateDisputePreArbRequest](#) for additional descriptions and conditions of the ATRSelectionReqdInd domain values.

**Note:** For a saved Pre-Arbitration with ATR(s), the user can only submit the Pre-Arbitration once the ATR association is complete.

2. To attach one or more images to the questionnaire, include the DisputeAttachmentDescriptor group element in the request message.

Members can also use the Bulk SI Hybrid Image upload functionality to generate DocIds in bulk which can be supplied in the SICreateDisputePreArbRequest against the DocId element. For details, see [Hybrid Image Uploads](#) and [Visa Resolve Online Bulk Systems Interface Development Guide](#).

3. If the user wants to use the Bulk SI Hybrid Image upload functionality to supply the documents, including image details, to a saved Pre-Arbitration questionnaire, set the <SupplyDocInd>=true.

- If the <SupplyDocInd>=true, then the Pre-Arbitration questionnaire will always get saved even if the user sets the Action value to Submit.
- If the <SupplyDocInd>=false, then the Pre-Arbitration questionnaire will get submitted when the user submits the Pre-Arbitration questionnaire.

If the user fails to attach the image to a saved questionnaire through the Bulk SI Hybrid Image upload functionality, the user must supply the missing elements through RTSI, save the questionnaire, and try to attach the image again using the same Bulk SI service.

## Retrieve Pre-Arbitration from Queue

1. User retrieves Pre-Arbitration questionnaires in one of the following ways:
  - The member host system triggers a SIGetQueueRequest message or a SIGetBatchQueueRequest message to retrieve Pre-Arbitration questionnaires from the corresponding Dispute Pre-Filing queues. See [Chapter 17, Queues](#) and [Chapter 18, RTSI Batch Queues](#).
  - VROL returns a list of the Pre-Arbitration cases in the queue, including their VisaCaseNumber.
  - Some members choose to receive Pre-Arbitration questionnaires using Bulk SI Download files.
2. To close the Pre-Arbitration or Pre-Arbitration response and remove it from the queue, user sends close Pre-Arbitration request SICloseTransactionRequest. See [Chapter 17, Queues](#) for more information.

**Note:** *SICloseTransactionRequest closes the transaction and removes it from its respective queue but does not close the case. For more information, see [SICloseTransactionRequest](#).*

VROL sends response with success status.

## Get Pre-Arbitration Details

1. To retrieve Pre-Arbitration details, the member host system triggers a SIGetDisputePreArbDetailsRequest supplying the VisaCaseNumber and DisputePreArbId.
- VROL returns a SIGetDisputePreArbDetailsResponse message with Pre-Arbitration and image details, along with the AssociatedTransGroupId.
- **Note:** *The Pre-Arbitration questionnaire may be returned as an image (TIFF/PDF), in addition to XML, as part of this response message (if optional <IncludeDisputeAsImageInd>=true is included in the request). However, any images attached to the questionnaire must be retrieved separately (step 2 below).*
- **Note:** *Any image attached to the Pre-Arbitration questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.*
2. To retrieve the attached images, the member host system triggers a SIGetImageRequest message supplying the VisaCaseNumber and DocId.
- **Note:** *DocIds are generated in bulk when the member uses the Bulk SI Hybrid Image upload functionality to attach image(s) in the Pre-Arbitration questionnaire. For details, see [Hybrid Image Uploads](#) and Visa Resolve Online Bulk Systems Interface Development Guide.*

VROL returns a SIGetImageResponse message with image data.

## Respond to a Received Pre-Arbitration

**Note:** *Prior to submitting a response, an RTSI user should retrieve the Pre-Arbitration details in the case.*

1. To create a Pre-Arbitration response, the member host system triggers a SICreateDisputePreArbResponseRequest message supplying the VisaCaseNumber and Pre-Arbitration Response information.

SICreateDisputePreArbResponseRequest can be saved or submitted. VROL returns a SICreateDisputePreArbResponseResponse with DisputePreArbResponseId, DocIds for any attached images, financial fields, AssociatedTransGroupId, and success status.

- Save

A SICreateDisputePreArbResponseRequest message can be saved. When it is saved, it generates a DisputePreArbResponseId.

- Submit

If a SICreateDisputePreArbResponseRequest message is submitted for a new Pre-Arbitration response, a DisputePreArbResponseId is generated in the response.

If a SICreateDisputePreArbResponseRequest message is submitted for a previously-saved Pre-Arbitration response, there is no new DisputePreArbResponseId generated in the response.

See [Financial Fields Returned in Immediate Response](#) for a list of financial fields returned in the response. See [Chapter 8, Dispute Financials](#) for an explanation of what these fields represent and how to use them.

2. To attach one or more images to the questionnaire, include the DisputeAttachmentDescriptor group element in the request message.

Members can also use the Bulk SI Hybrid Image upload functionality to generate DocIds in bulk which can be supplied in the SICreateDisputePreArbResponseRequest against the DocId element. For details, see [Hybrid Image Uploads](#) and [Visa Resolve Online Bulk Systems Interface Development Guide](#).

3. If the user wants to use the Bulk SI Hybrid Image upload functionality to supply the documents, including image details, to a saved Pre-Arbitration response questionnaire, set the <SupplyDocInd>=true.

- If the <SupplyDocInd>=true, then the Pre-Arbitration response questionnaire will always get saved even if the user sets the Action value to Submit.
- If the <SupplyDocInd>=false, then the Pre-Arbitration response questionnaire will get submitted when the user submits the Pre-Arbitration response questionnaire.

If the user fails to attach the image to a saved questionnaire through the Bulk SI Hybrid Image upload functionality, the user must supply the missing elements through RTSI, save the questionnaire, and try to attach the image again using the same Bulk SI service.

## Get Pre-Arbitration Response Details

1. To retrieve the details of a submitted or received Pre-Arbitration response, the member host system triggers a SIGetDisputePreArbResponseDetailsRequest message supplying the VisaCaseNumber and DisputePreArbResponseId.

VROL returns a SIGetDisputePreArbResponseDetailsResponse message with the details of the Pre-Arbitration response.

**Note:** *The Pre-Arbitration Response questionnaire may be returned as an image (TIFF/PDF), in addition to XML, as part of this response message (if optional <IncludeDisputeAsImageInd>=true is included in the request). However, any images attached to the questionnaire must be retrieved separately (step 2 below).*

**Note:** *Any image attached to the Pre-Arbitration Response questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.*

2. To retrieve the attached images, the member host system triggers a SIGetImageRequest message supplying the VisaCaseNumber and DocId.

**Note:** *DocIds are generated in bulk when the member uses the Bulk SI Hybrid Image upload functionality to attach image(s) in the Pre-Arbitration Response questionnaire. For details, see [Hybrid Image Uploads](#) and Visa Resolve Online Bulk Systems Interface Development Guide.*

VROL returns a SIGetImageResponse message with image data.

## Retrieve Acceptance Notification Details

The RTSI user can retrieve the acceptance notification details for Pre-Arbitrations or Pre-Arbitration responses that were accepted and closed.

The user can accept and close a Pre-Arbitration or Pre-Arbitration response by triggering the SIAcceptDisputeRequest message. For more information, see [SIAcceptDisputeRequest](#). Optionally, the user can close the Pre-Arbitration or Pre-Arbitration response by triggering the SICloseTransactionRequest message. For more information, see [SICloseTransactionRequest](#).

1. To retrieve accepted Pre-Arbitration responses from the dispute acceptance queue, the member host system triggers a SiGetQueueRequest message. See [Chapter 17, Queues](#).

VROL returns a list of the accepted Pre-Arbitration response cases in the queue including the VisaCaseNumber for each case.

2. To retrieve details of the acceptance notification for a Pre-Arbitration or Pre-Arbitration response, the member host system triggers a SIGetAcceptanceNotificationDetailsRequest message supplying the VisaCaseNumber, DisputePreArbId, or DisputePreArbResponseId.

VROL returns a SIGetAcceptanceNotificationDetailsResponse message with the VisaCaseNumber along with the acceptance and close details of the Pre-Arbitration or Pre-Arbitration response item.

## Delete Pre-Arbitration

The initiator can delete saved Pre-Arbitrations.

1. To delete a Pre-Arbitration, the member host system triggers a SIChangeDisputeStatusRequest message supplying the VisaCaseNumber, ChangeItemID, ChangeItemType, and Action value=Delete.

**Note:** *ChangeItemID includes the unique ID value for the corresponding Pre-Arbitration (for example, DisputePreArbId for Pre-Arbitration and DisputePreArbResponseId for Pre-Arbitration response).*

VROL returns a SIChangeDisputeStatusResponse message with the success status.

## Recall Pre-Arbitration

The initiator can recall a submitted Pre-Arbitration or Pre-Arbitration response if the following conditions are met:

**Note:** *The initiator can only recall a Pre-Arbitration response with "Decline" value.*

- The Pre-Arbitration or Pre-Arbitration response has been submitted.
  - The Pre-Arbitration or Pre-Arbitration response has not been previously recalled.
  - The opposing member has not yet responded to or fully/partially accepted the received Pre-Arbitration or Pre-Arbitration response.
  - The current date is within 3 calendar days of the Pre-Arbitration or Pre-Arbitration response's submission date (1 day if the pinned original transaction in the dispute case is an Original Credit Transaction).
  - The case is still in the Pre-Arbitration status.
  - There is no case filing event in the case yet.
  - The response time frame has not yet expired.
1. To retrieve recalled Pre-Arbitrations from the recall queue, the member host system triggers a `SiGetQueueRequest` message. See [Chapter 17, Queues](#).  
VROL returns a list of the recalled Pre-Arbitration cases in the queue including the `VisaCaseNumber` for each case.
  2. To recall an initiated Pre-Arbitration, the member host system triggers a `SIChangeDisputeStatusRequest` message supplying the `VisaCaseNumber`, `ChangeltemID`, `ChangeltemType`, and `Action` value=Recall.

**Note:** *ChangeltemID refers to the unique ID value of the submitted Pre-Arbitration.*

VROL returns a `SIChangeDisputeStatusResponse` message with the success status.

VROL returns a hard-stop error if a recall request is submitted into a case that does not qualify for recall.

## Check for Pre-Arbitration Rejects

If the dispute-related Pre-Arbitration questionnaire or Pre-Arbitration response questionnaire fails one or more validation edits, VROL blocks the questionnaire and places it in the `MY_REJECTS` or `ALL_REJECTS` queue.

1. To check for rejected Pre-Arbitration questionnaires, the member host system triggers a `SiGetQueueRequest` message for `MY_REJECTS` or `ALL_REJECTS` queue. See [Chapter 17, Queues](#), for more information.

VROL returns a `SiGetQueueResponse` message with the `VisaCaseNumber` and `RejectID`.

2. To ignore a rejected Pre-Arbitration questionnaire when using the reject queue, the member host system triggers a `SiIgnoreRejectRequest` message supplying the `VisaCaseNumber`, `RejectID`, and `RejectedTransactionType`=`"Pre-Arbitration"` or `"Pre-Arbitration Response"`. This step prevents the transaction from being returned again in the next queue request. See [Chapter 17, Queues](#), for more information.

VROL returns a `SiIgnoreRejectResponse` message with the success status.

3. To repair a rejected Pre-Arbitration or Pre-Arbitration response questionnaire, the member host system triggers a `SiCreateDisputePreArbRequest` (for Pre-Arbitration QN) or `SiCreateDisputePreArbResponseRequest` (for Pre-Arbitration Response QN)

message along with the updated Pre-Arbitration or Pre-Arbitration response questionnaire information.

- If the case contains a single unrepaired reject of the same type as the request, VROL marks the reject as repaired.
- If the case contains multiple unrepaired rejects of the same type as the request, VROL marks the most recent reject as repaired.
- If the case does not contain a reject of the same type as the request, VROL does not mark any transaction as repaired. VROL adds the incoming transaction to the case and sends response with success status.

VROL returns a SICreateDisputePreArbResponse (for Pre-Arbitration QN) or SICreateDisputePreArbResponseResponse (for Pre-Arbitration Response QN) message with the success status.

## Pre-Compliance Process

Typically, issuers initiate Pre-Compliance to dispute and correct the issues arising out of dispute compliance violations as per *Visa Core Rules and Visa Product and Service Rules*. The Pre-Compliance process is invoked when a member decides to initiate a Pre-Compliance for a compliance issue. Pre-Compliance can be initiated by either issuer or acquirer, and from a single Transaction Inquiry (TI) transaction.

Both issuers and acquirers can initiate a Pre-Compliance for transactions for all jurisdictions.

A Pre-Compliance can be initiated from an existing case regardless of the case status, up to the point of Pre-Compliance or case filing. Initiators can no longer initiate a Pre-Compliance once the case contains a Pre-Compliance or an active (no final decision) case filing. Initiators can also initiate a Pre-Compliance from a closed case. This action reopens the case for the side that is initiating the Pre-Compliance. Delivering a Pre-Compliance to the opposer reopens the case for the opposer if the case was previously closed.

When responding to a Pre-Compliance, the user can choose to accept liability (full or partial) or to decline the incoming request. If the user chooses to accept the liability, then the Transfer Funds Now option can also be selected. This capability allows the automated transfer of funds (miscellaneous fees) between the two members if a member accepts liability in the Pre-Compliance Response. VROL automatically generates Fee Collection/Funds Disbursement messages from the appropriate VROL BIN to the member BINs. See [Chapter 12, Miscellaneous Fees](#) for details.

RTSI supports submission of single Pre-Compliance or single Pre-Compliance response only. Multiple submissions are not supported.

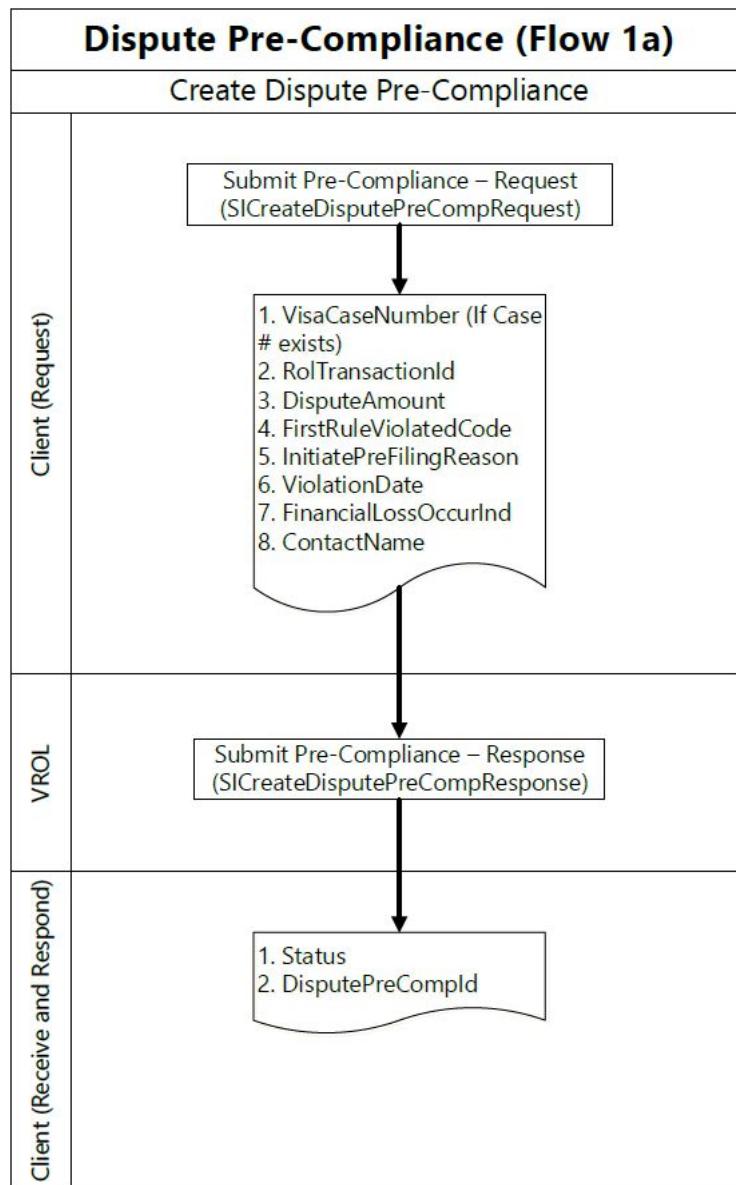
## Pre-Compliance Flows

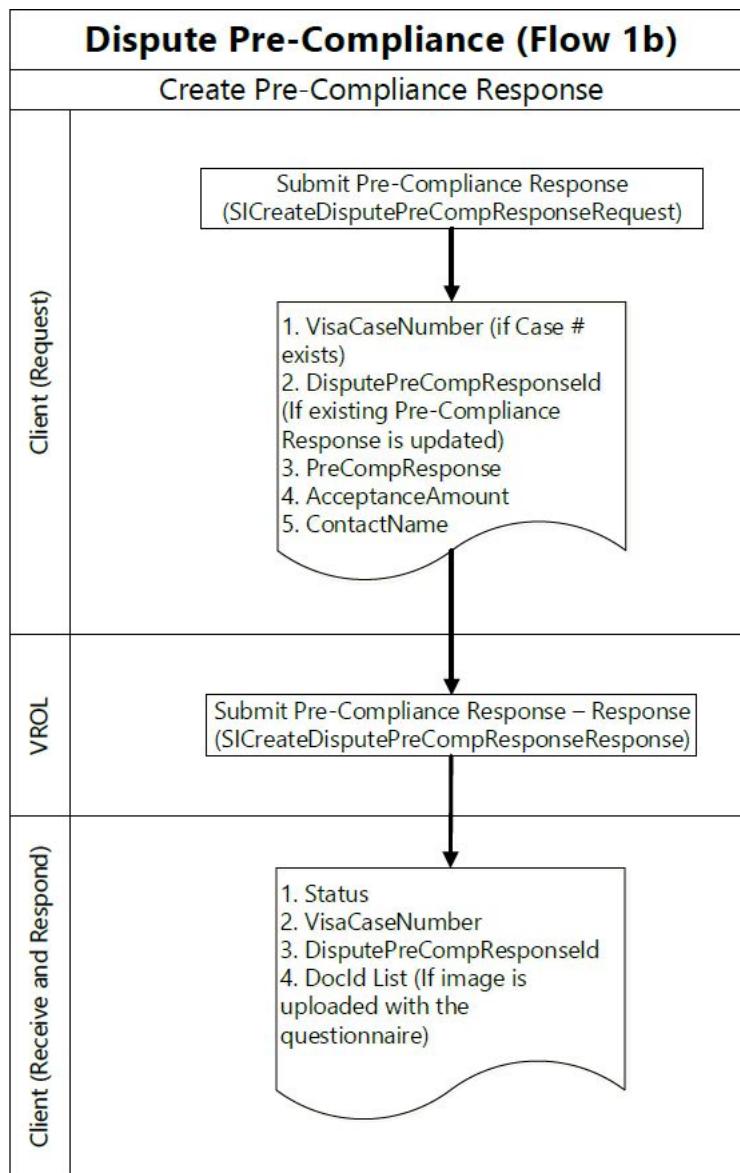
The following diagrams illustrate the Pre-Compliance process.

**Note:**

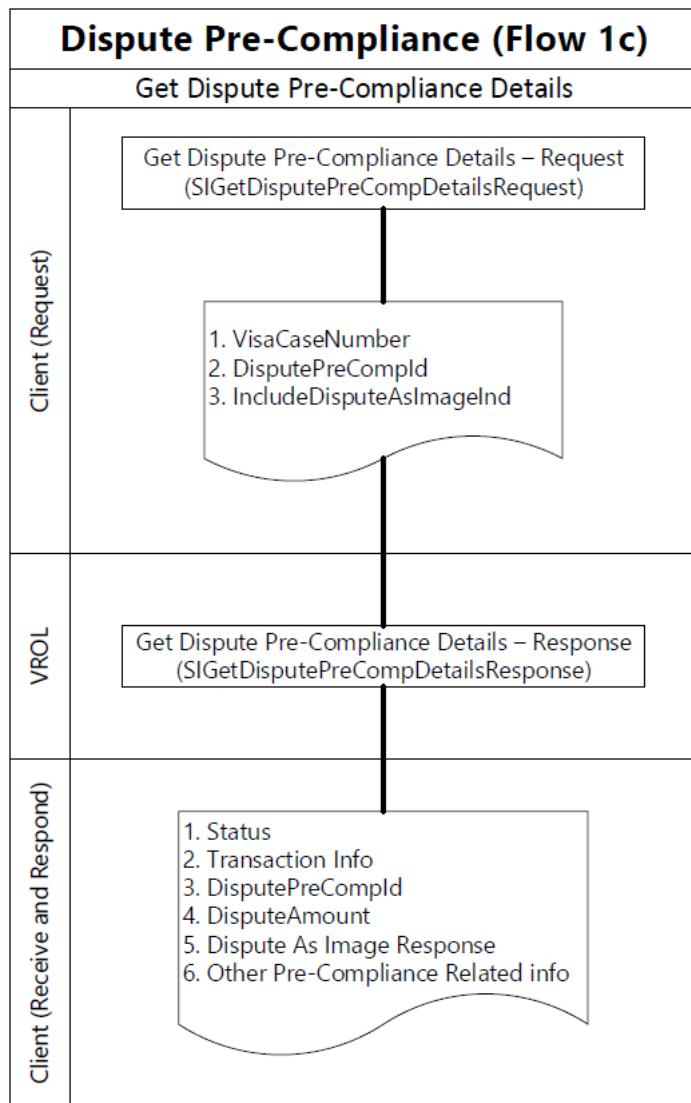
- The diagrams list only the important elements. For the complete elements, refer to the Dispute tab of the Interface Element Specification (IES).
- The dashed lines in the flow refer to optional functions that the user can use, depending on the business process that the user wants to implement.

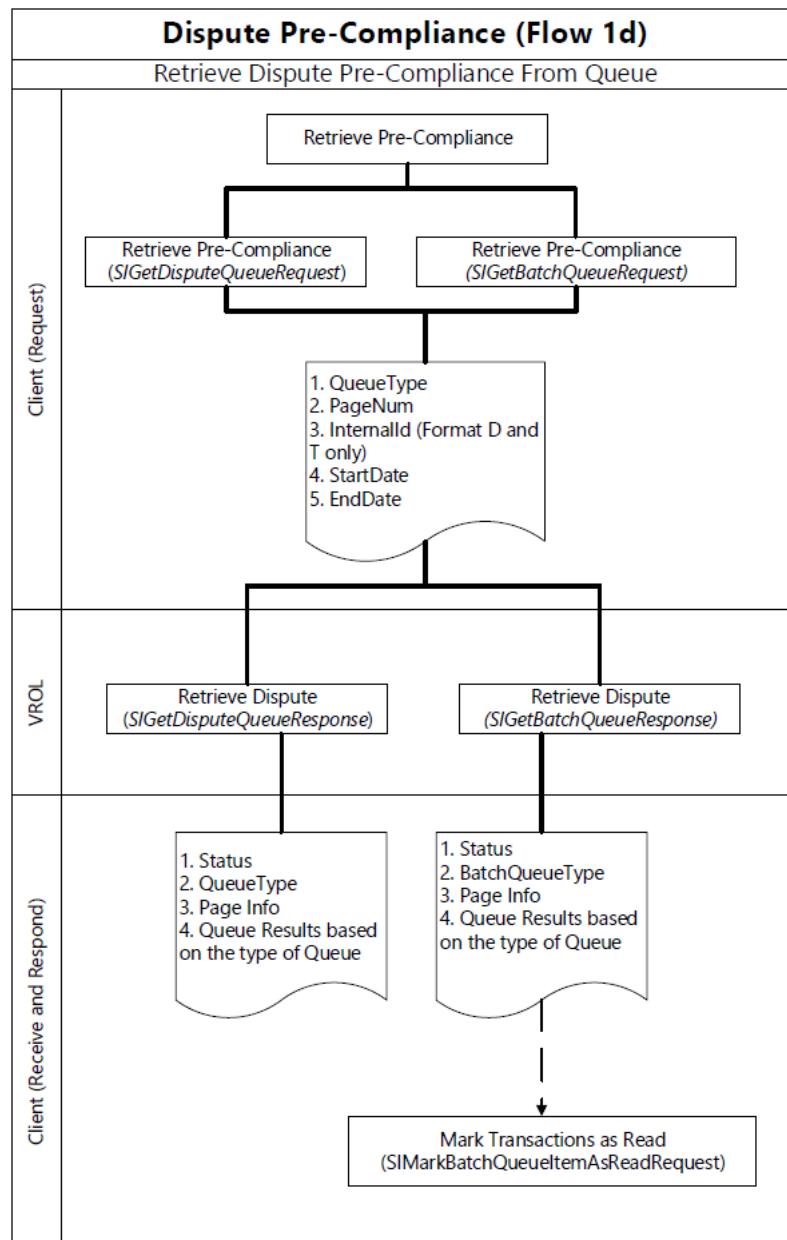
Figure 10–9: Dispute Pre-Compliance—Flow 1a



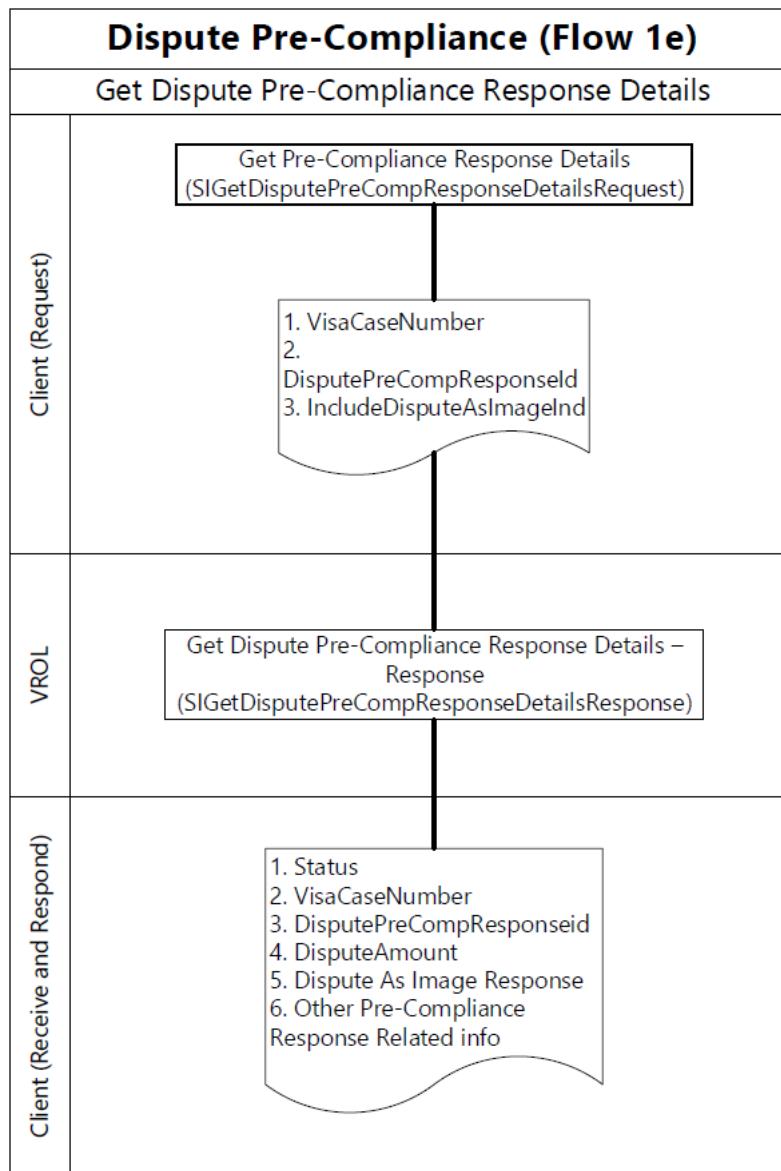
**Figure 10–10: Dispute Pre-Compliance—Flow 1b**

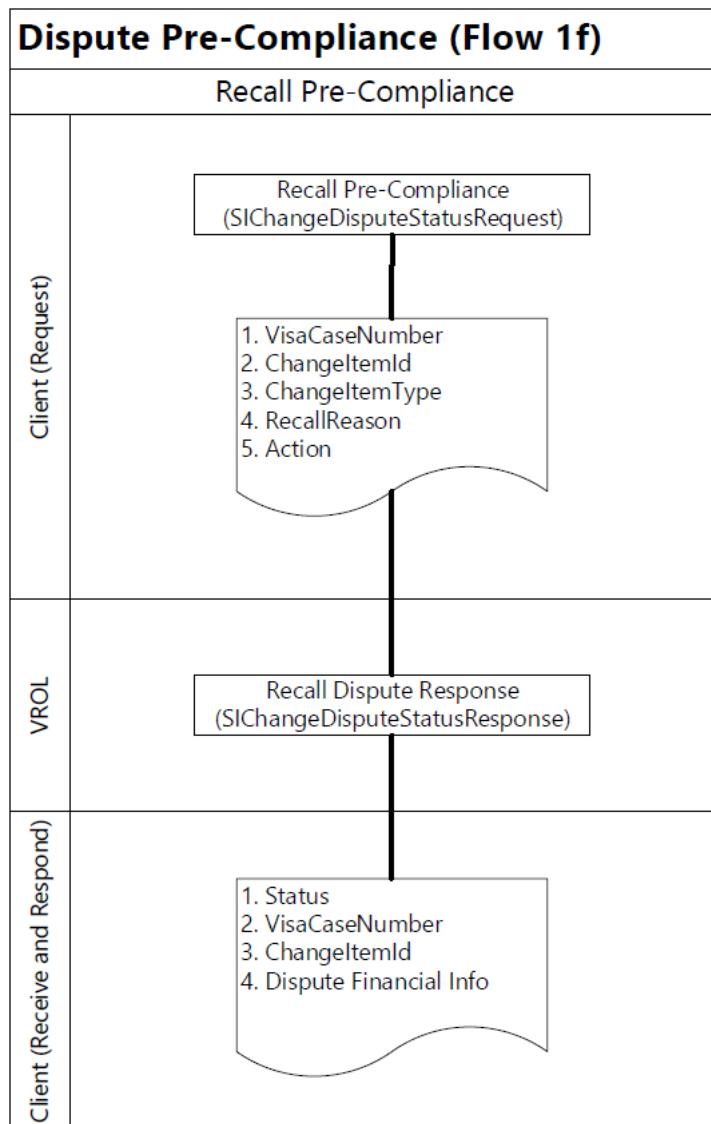
**Figure 10-11: Dispute Pre-Compliance—Flow 1c**



**Figure 10–12: Dispute Pre-Compliance—Flow 1d**

**Figure 10–13: Dispute Pre-Compliance—Flow 1e**



**Figure 10-14: Dispute Pre-Compliance—Flow 1f**

## Create Pre-Compliance

Pre-Compliance can be created from an original transaction that is previously retrieved through Transaction Inquiry (TI) or from an existing case that contains a selected original transaction. See [Chapter 4, Transaction Inquiries](#).

This involves the following actions:

- Member submits or saves a Pre-Compliance.
- VROL accepts the Pre-Compliance questionnaire field details.
- VROL applies the Pre-Compliance validation edits.
- VROL saves or submits the dispute.

1. To create a Pre-Compliance, the member host system triggers a SICreateDisputePreCompRequest message supplying the RolTransactionId and details of the Pre-Compliance questionnaire.

When creating a Pre-Compliance from an existing case, provide only the VisaCaseNumber. A new case will be created if VisaCaseNumber is not provided.

SICreateDisputePreCompRequest can be saved or submitted. VROL returns a SICreateDisputePreCompResponse message with the DisputePreCompld and success status.

- Save

A SICreateDisputePreCompRequest can be saved. When it is saved, it generates a DisputePreCompld.

- Submit

If a SICreateDisputePreCompRequest message is submitted for a new Pre-Compliance, DisputePreCompld and AccountingEntryID are generated in the response.

If a SICreateDisputePreCompRequest message is submitted for a previously-saved Pre-Compliance, there is no new DisputePreCompld generated in the response.

**Note:**

- VROL returns the appropriate error message and code for every failed scenarios.
- Required/optional submission of Provisional Credit can be done by the issuer by using the corresponding legacy RTSI Web services. See [Chapter 16, Accounting Entry Memos](#) for more information about Provisional Credit.

2. To attach one or more images to the questionnaire, include the DisputeAttachmentDescriptor group element in the request message.

Members can also use the Bulk SI Hybrid Image upload functionality to generate DocIds in bulk which can be supplied in the SICreateDisputePreCompRequest against the DocId element. For details, see [Hybrid Image Uploads](#) and [Visa Resolve Online Bulk Systems Interface Development Guide](#).

3. If the user wants to use the Bulk SI Hybrid Image upload functionality to supply the documents, including image details, to a saved Pre-Compliance questionnaire, set the <SupplyDocInd>=true.

- If the <SupplyDocInd>=true, then the Pre-Compliance questionnaire will always get saved even if the user sets the Action value to Submit.
- If the <SupplyDocInd>=false, then the Pre-Compliance questionnaire will get submitted when the user submits the Pre-Compliance questionnaire.

If the user fails to attach the image to a saved questionnaire through the Bulk SI Hybrid Image upload functionality, the user must supply the missing elements through RTSI, save the questionnaire, and try to attach the image again using the same Bulk SI service.

## Retrieve Pre-Compliance from Queue

1. User retrieves Pre-Compliance questionnaires in one of the following ways:

- The member host system triggers a SIGetQueueRequest message or a SIGetBatchQueueRequest message to retrieve Pre-Compliance questionnaires from the corresponding Dispute Pre-Filing queues. See [Chapter 17, Queues](#) and [Chapter 18, RTSI Batch Queues](#).

VROL returns a list of the Pre-Compliance cases in the queue, including their VisaCaseNumber.

- Some members choose to receive Pre-Compliance questionnaires using Bulk SI Download files.
- 2. To close the Pre-Compliance or Pre-Compliance response and remove it from the queue, user sends close Pre-Compliance request SICloseTransactionRequest. See [Chapter 17, Queues](#) for more information.

**Note:** *SICloseTransactionRequest closes the transaction and removes it from its respective queue but does not close the case. For more information, see [SICloseTransactionRequest](#).*

VROL sends response with success status.

## Get Pre-Compliance Details

1. To retrieve Pre-Compliance details, the member host system triggers a SIGetDisputePreCompDetailsRequest supplying the VisaCaseNumber and DisputePreCompld.
- VROL returns a SIGetDisputePreCompDetailsResponse message with the Pre-Compliance and image details.
- Note:** *The Pre-Compliance questionnaire may be returned as an image (TIFF/PDF), in addition to XML, as part of this response message (if optional <IncludeDisputeAsImageInd>=true is included in the request). However, any images attached to the questionnaire must be retrieved separately (step 2 below).*
- Note:** *Any image attached to the Pre-Compliance questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.*
2. To retrieve the attached images, the member host system triggers a SIGetImageRequest message supplying the VisaCaseNumber and DocId.
- Note:** *DocIds are generated in bulk when the member uses the Bulk SI Hybrid Image upload functionality to attach image(s) in the Pre-Compliance questionnaire. For details, see [Hybrid Image Uploads](#) and Visa Resolve Online Bulk Systems Interface Development Guide.*

VROL returns a SIGetImageResponse message with image data.

## Respond to a Received Pre-Compliance

The opposer can respond to a received Pre-Compliance if the Pre-Compliance Response time frame has not expired, and the case has not yet advanced to a filing stage.

**Note:** *Prior to submitting a response, an RTSI user should retrieve the Pre-Compliance details in the case.*

1. To create a Pre-Compliance Response, the member host system triggers a SICreateDisputePreCompResponseRequest message supplying the VisaCaseNumber and Pre-Compliance Response information.
- SICreateDisputePreCompResponseRequest can be saved or submitted. VROL returns a SICreateDisputePreCompResponseResponse with DisputePreCompResponseId and success status.

- Save  
A SICreateDisputePreCompResponseRequest can be saved. When it is saved, it generates a DisputePreCompResponseId.
  - Submit  
If a SICreateDisputePreCompResponseRequest message is submitted for a new Pre-Compliance response, a DisputePreCompResponseId is generated in the response.  
If a SICreateDisputePreCompResponseRequest message is submitted for a previously-saved Pre-Compliance response, there is no new DisputePreCompResponseId generated in the response.
2. To attach one or more images to the questionnaire, include the DisputeAttachmentDescriptor group element in the request message.
- Members can also use the Bulk SI Hybrid Image upload functionality to generate DocIds in bulk which can be supplied in the SICreateDisputePreCompResponseRequest against the DocId element. For details, see [Hybrid Image Uploads](#) and [Visa Resolve Online Bulk Systems Interface Development Guide](#).
3. If the user wants to use the Bulk SI Hybrid Image upload functionality to supply the documents, including image details, to a saved Pre-Compliance response questionnaire, set the <SupplyDocInd>=true.
- If the <SupplyDocInd>=true, then the Pre-Compliance response questionnaire will always get saved even if the user sets the Action value to Submit.
  - If the <SupplyDocInd>=false, then the Pre-Compliance response questionnaire will get submitted when the user submits the Pre-Compliance response questionnaire.
- If the user fails to attach the image to a saved questionnaire through the Bulk SI Hybrid Image upload functionality, the user must supply the missing elements through RTSI, save the questionnaire, and try to attach the image again using the same Bulk SI service.

## Get Pre-Compliance Response Details

1. To retrieve the details of a submitted or received Pre-Compliance Response, the member host system triggers a SIGetDisputePreCompResponseDetailsRequest message supplying the VisaCaseNumber and DisputePreCompResponseId.  
VROL returns a SIGetDisputePreCompResponseDetailsResponse message with the details of the Pre-Compliance Response.  
**Note:** *The Pre-Compliance Response questionnaire may be returned as an image (TIFF/PDF), in addition to XML, as part of this response message (if optional <IncludeDisputeAsImageInd>=true is included in the request). However, any images attached to the questionnaire must be retrieved separately (step 2 below).*  
**Note:** *Any image attached to the Pre-Compliance Response questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.*
2. To retrieve the attached images, the member host system triggers a SIGetImageRequest message supplying the VisaCaseNumber and DocId.

**Note:** DocIds are generated in bulk when the member uses the Bulk SI Hybrid Image upload functionality to attach image(s) in the Pre-Compliance Response questionnaire. For details, see [Hybrid Image Uploads](#) and Visa Resolve Online Bulk Systems Interface Development Guide.

VROL returns a SIGetImageResponse message with image data.

## Retrieve Acceptance Notification Details

The RTSI user can retrieve the acceptance notification details for Pre-Compliances or Pre-Compliance responses that were accepted and closed.

The user can accept and close a Pre-Compliance or Pre-Compliance response by triggering the SIAcceptDisputeRequest message. For more information, see [SIAcceptDisputeRequest](#). Optionally, the user can close the Pre-Compliance or Pre-Compliance response by triggering the SICloseTransactionRequest message. For more information, see [SICloseTransactionRequest](#).

1. To retrieve accepted Pre-Compliance responses from the dispute acceptance queue, the member host system triggers a SiGetQueueRequest message. See [Chapter 17, Queues](#).

VROL returns a list of the accepted Pre-Compliance response cases in the queue including the VisaCaseNumber for each case.

2. To retrieve details of the acceptance notification for a Pre-Compliance or Pre-Compliance response, the member host system triggers a SIGetAcceptanceNotificationDetailsRequest message supplying the VisaCaseNumber, DisputePreCompld, or DisputePreCompResponsd.

VROL returns a SIGetAcceptanceNotificationDetailsResponse message with the VisaCaseNumber along with the acceptance and close details of the Pre-Compliance or Pre-Compliance response item.

## Delete Pre-Compliance

The initiator can delete a saved Pre-Compliance.

1. To delete a Pre-Compliance, the member host system triggers a SIChangeDisputeStatusRequest message supplying the VisaCaseNumber, ChangeltemID, ChangeltemType, and Action value=Delete.

**Note:** ChangeltemID includes the unique ID value for the corresponding Pre-Compliance (for example, DisputePreCompld for Pre-Compliance and DisputePreCompResponsd for Pre-Compliance response).

VROL returns a SIChangeDisputeStatusResponse message with the success status.

## Recall Pre-Compliance

The initiator can recall a submitted Pre-Compliance if the following conditions are met:

**Note:** Only the Pre-Compliance can be recalled but not the Pre-Compliance response.

- The opposing member has not yet responded.
- The current date is within 3 calendar days of the Pre-Compliance's submission date (1 day if the pinned original transaction in the dispute case is an Original Credit Transaction).
- The case is still in the Pre-Compliance status.

- There is no case filing event in the case yet.
  - The response time frame has not yet expired.
1. To retrieve recalled Pre-Compliances from the recall queue, the member host system triggers a SiGetQueueRequest message. See [Chapter 17, Queues](#).  
VROL returns a list of the recalled Pre-Compliance cases in the queue including the VisaCaseNumber for each case.
  2. To recall an initiated Pre-Compliance, the member host system triggers a SIChangeDisputeStatusRequest message supplying the VisaCaseNumber, ChangeltemID, ChangeltemType, and Action value=Recall.  
**Note:** *ChangeltemID refers to the unique ID value of the submitted Pre-Compliance.*  
VROL returns a SIChangeDisputeStatusResponse message with the success status.  
VROL returns a hard-stop error if a recall request is submitted into a case that does not qualify for recall.

## Check for Pre-Compliance Rejects

If the dispute-related Pre-Compliance questionnaire or Pre-Compliance response questionnaire fails one or more validation edits, VROL blocks the questionnaire and places it in the MY\_REJECTS or ALL\_REJECTS queue.

1. To check for rejected Pre-Compliance questionnaires, the member host system triggers a SiGetQueueRequest message for MY\_REJECTS or ALL\_REJECTS queue. See [Chapter 17, Queues](#), for more information.  
VROL returns a SiGetQueueResponse message with the VisaCaseNumber and RejectID.
2. To ignore a rejected Pre-Compliance questionnaire when using the reject queue, the member host system triggers a SIIgnoreRejectRequest message supplying the VisaCaseNumber, RejectID, and RejectedTransactionType="Pre-Compliance" or "Pre-Compliance Response". This step prevents the transaction from being returned again in the next queue request. See [Chapter 17, Queues](#), for more information.  
VROL returns a SIIgnoreRejectResponse message with the success status.
3. To repair a rejected Pre-Compliance questionnaire, the member host system triggers a SICreateDisputePreCompRequest message along with the updated Pre-Compliance questionnaire information.
  - If the case contains a single unrepaired reject of the same type as the request, VROL marks the reject as repaired.
  - If the case contains multiple unrepaired rejects of the same type as the request, VROL marks the most recent reject as repaired.
  - If the case does not contain a reject of the same type as the request, VROL does not mark any transaction as repaired. VROL adds the incoming transaction to the case and sends response with success status.VROL returns a SICreateDisputePreCompResponse message with the success status.

## Pre-Filing Operations

RTSI exposes the following pre-filing operations:

**SICreateDisputePreArbRequest**—Used to create a Pre-Arbitration.

- SIGetDisputePreArbDetailsRequest**—Used to retrieve Pre-Arbitration details.
- SICreateDisputePreArbResponseRequest**—Used to respond to Pre-Arbitration.
- SIGetDisputePreArbResponseDetailsRequest**—Used to retrieve details of a submitted or received Pre-Arbitration response.
- SICreateDisputePreCompRequest**—Used to create a Pre-Compliance.
- SIGetDisputePreCompDetailsRequest**—Used to retrieve Pre-Compliance details.
- SICreateDisputePreCompResponseRequest**—Used to respond to Pre-Compliance.
- SIGetDisputePreCompResponseDetailsRequest**—Used to retrieve details of a submitted or received Pre-Compliance Response.
- SIChangeDisputeStatusRequest**—Used to recall a submitted Pre-Arbitration, Pre-Arbitration response, or Pre-Compliance; or delete a saved Pre-filing (Pre-Arbitration, Pre-Arbitration response, Pre-Compliance, or Pre-Compliance response).
- SIGetAcceptanceNotificationDetailsRequest**—Used to retrieve the acceptance notification details for Pre-Arbitrations, Pre-Arbitration responses, Pre-Compliances, or Pre-Compliance responses that were accepted and closed.

## SICreateDisputePreArbRequest

SICreateDisputePreArbRequest allows RTSI users to create a Pre-Arbitration.

The following elements should be considered when creating a Pre-Arbitration or receiving a response from this service.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.
DisputePreArbId	A unique VROL-generated ID for each saved or submitted Pre-Arbitration.
ATRSelectionReqdInd	<p>Indicates in the response if the user is required to associate ATR transactions.</p> <p>Domain values:</p> <ul style="list-style-type: none"> <li>• Required=Indicates that ATR selection is required if the subsequent ATR contains additional transactions (Credit, Reversal, or Authorization) other than the dispute transactions (For example, Dispute, Dispute Response, Dispute Reversals, or Dispute Response Reversal).</li> <li>• NotRequired=Indicates that ATR selection is not required if the subsequent ATR contains only the dispute transactions (For example, Dispute, Dispute Response, Dispute Reversals, or Dispute Response Reversal).</li> <li>• Unknown=Indicates that VROL was not able to determine the correct status yet. (SIGetDisputePreArbDetailsRequest can be submitted to check the status.)</li> </ul> <p><b>Note:</b> <i>ATRSelectionReqdInd is displayed in the SICreateDisputePreArbResponse message in a collaboration workflow.</i></p>
AssociatedTransGroupId	Identifies a group of transactions associated to the selected transaction, and is used to access the associated transaction detail information.

SICreateDisputePreArb contains request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SICreateDisputePreArbRequest>
    <RequestHeader>
        <User id="abcde1234f" type="internalId"/>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <MemberCaseNumber>555555555555</MemberCaseNumber>
        <Note>CP with fake CreditTransactionID</Note>
        <InitiatePreArbReason>CP</InitiatePreArbReason>
        <MultipleCredits>
            <Credit id="1">
                <CreditDate>2018-01-18</CreditDate>
                <CreditAmount currency="344">213.64</CreditAmount>
                <MatchingCreditTranId>8000000</MatchingCreditTranId>
            </Credit>
        </MultipleCredits>
        <DisputeCategory>11</DisputeCategory>
        <ARNandDateCreditReversal>a</ARNandDateCreditReversal>
        <Action>Submit</Action>
    </requestData>
</SICreateDisputePreArbRequest>
```

### Response Message—SOAP-based Services

```
<SICreateDisputePreArbResponse>
    <Status>
        <Code>I-301100084</Code>
        <Message>No matching credit transaction data found.</Message>
    </Status>
    <responseData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <DisputePreArbId>123456</DisputePreArbId>
        <ATRSelectionReqdInd>NotRequired</ATRSelectionReqdInd>
    </responseData>
</SICreateDisputePreArbResponse>
```

## Request Message—REST-based Services

```
{
  "RequestHeader": {
    "MemberRole": "A"
  },
  "requestData": {
    "VisaCaseNumber": "1000000000",
    "MemberCaseNumber": "555555555555",
    "Note": "A credit or Reversal issued by the Merchant was not addressed by the Issuer in the Dispute.",
    "InitiatePreArbReason": "CP",
    "MultipleCredits": {
      "Credit": [
        {
          "id": "1",
          "CreditDate": "2018-01-18",
          "CreditAmount": {
            "currency": "344",
            "value": 213.64
          },
          "MatchingCreditTranId": 8234629
        },
        {
          "id": "2",
          "CreditDate": "2018-01-18",
          "CreditAmount": {
            "currency": "840",
            "value": 1.60
          },
          "MatchingCreditTranId": 8234637
        }
      ]
    },
    "WhyCreditReversalNotApply": "any reason",
    "Action": "Submit"
  }
}
```

## Response Message—REST-based Services

```
{
  "Status": {
    "Code": "I-300000000",
    "Message": "Successfully completed Operation."
  },
  "responseData": {
    "VisaCaseNumber": 1000000000,
    "DisputePreArbId": 123456
    "ATRSelectionReqdInd": "NotRequired"
  }
}
```

## SIGetDisputePreArbDetailsRequest

SIGetDisputePreArbDetailsRequest allows RTSI users to request for the Pre-Arbitration questionnaire data to be returned in image format, as well as in XML. The returned

image will be in TIFF or PDF format. To retrieve the dispute as an image, the user must set <IncludeDisputeAsImageInd>=true. The image data will be returned in the <Image> group in the SIGetDisputePreArbDetailsResponse.

**Note:** Only the Pre-Arbitration questionnaire will be returned as an image. Any attached documents must be retrieved using the AttachmentId value(s) in this response as input to the DocId field(s) of SISGetImageRequest.

**Note:** Any image attached to the Pre-Arbitration questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.

The following elements should be considered when retrieving the Pre-Arbitration details.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.
DisputePreArbId	A unique VROL-generated ID for each saved or submitted Pre-Arbitration.
AssociatedTransGroupId	Identifies a group of transactions associated to the selected transaction, and is used to access the associated transaction detail information.

`SIGetDisputePreArbDetails` contains request and response messages as shown in the following sample messages.

## Request Message—SOAP-based Services

```
<SIGetDisputePreArbDetailsRequest>
    <RequestHeader>
        <User id="abcde1234f" type="internalId"/>
        <MemberRole>A</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <DisputePreArbId>123456</DisputePreArbId>
    </requestData>
</SIGetDisputePreArbDetailsRequest>
```

### **Response Message—SOAP-based Services**

**Note:** The `SIGetDisputePreArbDetailsResponseRequest` message includes the dispute history, which consists of information from all previously-submitted disputes and dispute responses that are part of the same dispute thread.

```
<SIGetDisputePreArbDetailsResponse>
    <Status>
        <Code>I-300000000</Code>
        <Message>Successfully completed Operation.</Message>
    </Status>
    <responseData>
        <TransactionInfo>
            <AccountNumber>410000000000</AccountNumber>
            <Transaction id="8000000000000000">
                <TranDate>2019-06-23</TranDate>
                <Amount currency="840">3000.00</Amount>
            </Transaction>
        </TransactionInfo>
    </responseData>
</SIGetDisputePreArbDetailsResponse>
```

```
<MerchantName>MERCHANT NAME</MerchantName>
<MerchantCity>MERCHANT CITY</MerchantCity>
<MerchantCountrySubEntityCode>CA</MerchantCountrySubEntityCode>
<MerchantCountryCode>MU</MerchantCountryCode>
<MerchantPostalCode>90005</MerchantPostalCode>
<MCC>4111</MCC>
<ARN>70000000000000000000000000000001</ARN>
<ECMOTO>1</ECMOTO>
<ReimbursementAttribute>A</ReimbursementAttribute>
<CPD>2019-06-24</CPD>
<TransactionCategory>D</TransactionCategory>
<AcquirerBID>00000001</AcquirerBID>
<NetworkID>0002</NetworkID>
<AcquirerBIN>400003</AcquirerBIN>
<RetrievalReferenceNumber>700000000000
/RetrievalReferenceNumber>
<SystemTraceAuditNumber>080000</SystemTraceAuditNumber>
<TranDestAmt currency="840">3000.00</TranDestAmt>
<ProcessingCode>00</ProcessingCode>
<IssuerName>issuer name</IssuerName>
<AcquirerName>acquirer name</AcquirerName>
<AccountType>00</AccountType>
<DisputeJurisdiction jurisdictionCd="DOM">
    <Jurisdiction>DOMESTIC-MU</Jurisdiction>
    <IssuerRegion>CEMEA</IssuerRegion>
    <AcquirerRegion>CEMEA</AcquirerRegion>
    <CountryCd>MU</CountryCd>
</DisputeJurisdiction>
</TransactionInfo>
<VisaCaseNumber>1000000000</VisaCaseNumber>
<DisputePreArbId>123456</DisputePreArbId>
<MemberCaseNumber>555555555555</MemberCaseNumber>
<DisputeCategory>10</DisputeCategory>
<DisputeCondition>4</DisputeCondition>
<MerchantContactInfo>
    <ContactCity>MERCHANT CITY</ContactCity>
    <ContactStateCode>CA</ContactStateCode>
    <ContactPostalCode>90005</ContactPostalCode>
    <ContactCountryCode>MU</ContactCountryCode>
    <ContactOptInInd>false</ContactOptInInd>
</MerchantContactInfo>
<DisputeHistorySummary>
    <DisputeHistory>
        <SubmittedBy>Acquirer</SubmittedBy>
        <DisputeCategory>10</DisputeCategory>
        <DisputeItemType>DISPREQ</DisputeItemType>
        <RecallInd>false</RecallInd>
        <EventDate>2019-06-28T05:36:40.244</EventDate>
        <UserOrOrgName>user or organization name
        </UserOrOrgName>
        <DisputeItemID>111111</DisputeItemID>
    </DisputeHistory>
</DisputeHistorySummary>
<CaseHistorySummary>
    <CaseHistory>
        <Event>Dispute</Event>
        <CPD>2019-06-28</CPD>
```

```
<DisputeAmount currency="840">3000.00</DisputeAmount>
<DisputeCategoryCondition>10.4 Fraud - Card Absent
Environment</></DisputeCategoryCondition>
</CaseHistory>
<CaseHistory>
<Event>Pre-Arbitration</Event>
<CPD>2019-06-28</CPD>
<DisputeAmount currency="840">3000.00</DisputeAmount>
<DisputeCategoryCondition>10.4 Fraud - Card Absent
Environment</></DisputeCategoryCondition>
</CaseHistory>
</CaseHistorySummary>
<DisputeAmountChangeInd>N</DisputeAmountChangeInd>
<DisputeAmount currency="840">3000.00</DisputeAmount>
<InitiatePreArbReason>ID</InitiatePreArbReason>
<IssuerAcquirerContactInfo>
<ContactName>name</ContactName>
</IssuerAcquirerContactInfo>
<DisputeInvalidReason>IDRC7</DisputeInvalidReason>
<RecallInd>false</RecallInd>
<DisputeImageAttachment>
<Attachment id="7000003">
<ContentType>image/tiff</ContentType>
<Comment>any comment</Comment>
<ImageData>image data</ImageData>
<DocType>OTHER</DocType>
</Attachment>
</DisputeImageAttachment>
<NotListedDisputeInvalidReason>reason
</NotListedDisputeInvalidReason>
</ResponseData>
</SIGetDisputePreArbDetailsResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": {
    "User": {
      "id": "abcde1234f",
      "type": "internalId"
    },
    "MemberRole": "A"
  },
  "requestData": {
    "VisaCaseNumber": "1000000000",
    "DisputePreArbId": "123456"
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": {
    "Code": "I-3000000000",
    "Message": "Successfully completed Operation."
  },
  "responseData": {
```

```
"TransactionInfo":{  
    "AccountNumber": "0000000000000006",  
    "Transaction":{  
        "TranDate": "2019-06-23",  
        "Amount":{  
            "value": 3000.00,  
            "currency": "840"  
        },  
        "id": "8000000000000001"  
    },  
    "MerchantName": "MERCHANT NAME",  
    "MerchantCity": "MERCHANT CITY",  
    "MerchantCountrySubEntityCode": "CA",  
    "MerchantCountryCode": "MU",  
    "MerchantPostalCode": "90005",  
    "MCC": "4111",  
    "ARN": "00000000000000000000000000000001",  
    "ECMOTO": "1",  
    "ReimbursementAttribute": "A",  
    "CPD": "2019-06-24",  
    "TransactionCategory": "D",  
    "AcquirerBID": "00000000",  
    "NetworkID": "0002",  
    "AcquirerBIN": "000002",  
    "RetrievalReferenceNumber": "70000000000",  
    "SystemTraceAuditNumber": "300001",  
    "TranDestAmt":{  
        "value": 3000.00,  
        "currency": "840"  
    },  
    "ProcessingCode": "00",  
    "IssuerName": "Issuer Name",  
    "AcquirerName": "Acquirer Name",  
    "AccountType": "0",  
    "DisputeJurisdiction":{  
        "Jurisdiction": "DOMESTIC-MU",  
        "IssuerRegion": "CEMEA",  
        "AcquirerRegion": "CEMEA",  
        "CountryCd": "MU",  
        "jurisdictionCd": "DOM"  
    },  
    "VisaCaseNumber": 1000000000,  
    "DisputePreArbId": 123456,  
    "MemberCaseNumber": "555555555555",  
    "DisputeCategory":{  
        "value": "10"  
    },  
    "DisputeCondition": "4",  
    "MerchantContactInfo":{  
        "ContactCity": "MERCHANT CITY",  
        "ContactStateCode": "CA",  
        "ContactPostalCode": "90005",  
        "ContactCountryCode": "MU",  
        "ContactOptInInd": "false"  
    },  
    "DisputeHistorySummary":{
```

```
"DisputeHistory": [{}  
    "SubmittedBy": "Issuer",  
    "DisputeCategory":{  
        "value": "10"  
    },  
    "DisputeItemType": "DISPREQ",  
    "RecallInd": false,  
    "EventDate": "2019-06-28",  
    "UserOrOrgName": "ABC Bank - SMS",  
    "DisputeItemID": 111111  
}],  
},  
"CaseHistorySummary":{  
    "CaseHistory": [{}  
        "Event": "Dispute",  
        "CPD": "2019-06-28",  
        "DisputeAmount":{  
            "value": 3000.00,  
            "currency": "840"  
        },  
        "DisputeCategoryCondition": "10.4 Fraud - Card  
Absent Environment"  
    ],  
    {  
        "Event": "Pre-Arbitration",  
        "CPD": "2019-06-28",  
        "DisputeAmount":{  
            "value": 3000.00,  
            "currency": "840"  
        },  
        "DisputeCategoryCondition": "10.4 Fraud - Card  
Absent Environment"  
    ]  
},  
"DisputeAmountChangeInd": "N",  
"DisputeAmount":{  
    "value": 3000.00,  
    "currency": "840"  
},  
"InitiatePreArbReason": "ID",  
"IssuerAcquirerContactInfo":{  
    "ContactName": "name"  
},  
"DisputeInvalidReason": "IDRC7",  
"RecallInd": "false",  
"DisputeImageAttachment":{  
    "Attachment": [ {}  
        "ContentType": "image/tiff",  
        "Comment": "testtttt",  
        "ImageData": "image data",  
        "DocType": "OTHER",  
        "id": "7000003"  
    ]  
},  
"NotListedDisputeInvalidReason": "reason"  
}  
}
```

## SICreateDisputePreArbResponseRequest

SICreateDisputePreArbResponseRequest allows RTSI users to respond to a Pre-Arbitration.

The following XML elements should be considered when responding to a Pre-Arbitration or receiving a response from this service.

Element	Description
Response	Indicates the user's response to the Pre-Arbitration.  Domain values: <ul style="list-style-type: none"> <li>• FULL=Accept Full</li> <li>• PART=Accept Partial</li> <li>• DECL=Decline</li> </ul>
DisputePreArbResponseId	A unique VROL-generated ID for each response to a saved or submitted Pre-Arbitration.
AssociatedTransGroupId	Identifies a group of transactions associated to the selected transaction, and is used to access the associated transaction detail information.

**Note:** *SICreateDisputePreArbResponseResponse also contains the financial fields. See [Financial Fields Returned in Immediate Response](#) for a list of financial fields returned in the response.*

SICreateDisputePreArbResponse contains request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SICreateDisputePreArbResponseRequest>
    <RequestHeader>
        <User id="abcde1234f" type="internalId"/>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <MemberCaseNumber>555555555555</MemberCaseNumber>
        <Note>note</Note>
        <PreArbResponse>DECL</PreArbResponse>
        <LiabilityNotAcceptedFully>explanation</LiabilityNotAcceptedFully>
        <Action>Submit</Action>
    </requestData>
</SICreateDisputePreArbResponseRequest>
```

## Response Message—SOAP-based Services

```
<SICreateDisputePreArbResponseResponse>
    <Status>
        <Code>I-300000000</Code>
        <Message>Successfully completed Operation.</Message>
    </Status>
    <responseData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <DisputePreArbResponseId>123456</DisputePreArbResponseId>
        <DisputeFinancialInfo/>
    </responseData>
</SICreateDisputePreArbResponseResponse>
```

## Request Message—REST-based Services

```
{
    "RequestHeader": {
        "User": {
            "id": "abcde1234f",
            "type": "internalId"
        },
        "MemberRole": "I"
    },
    "requestData": {
        "VisaCaseNumber": 1000000000,
        "MemberCaseNumber": "555555555555",
        "Note": "note",
        "PreArbResponse": "DECL",
        "LiabilityNotAcceptedFully": "explanation",
        "Action": "Submit"
    }
}
```

## Response Message—REST-based Services

```
{
    "Status": {
        "Code": "I-300000000",
        "Message": "Successfully completed Operation."
    },
    "responseData": {
        "VisaCaseNumber": 1000000000,
        "DisputePreArbResponseId": 123456,
        "DisputeFinancialInfo": {}
    }
}
```

## SIGetDisputePreArbResponseDetailsRequest

SIGetDisputePreArbResponseDetailsRequest allows RTSI users to request for the Pre-Arbitration Response questionnaire data to be returned in image format, as well as in XML. The returned image will be in TIFF or PDF format. To retrieve the dispute as an image, the user must set `<IncludeDisputeAsImageInd>=true`. The image data will be returned in the `<Image>` group in the SIGetDisputePreArbResponseDetailsResponse.

**Note:** Only the Pre-Arbitration Response questionnaire will be returned as an image. Any attached documents must be retrieved using the AttachmentId value(s) in this response as input to the DocId(s) field of `SIGetImageRequest`.

**Note:** Any image attached to the Pre-Arbitration Response questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the `<DownloadImageInd>=false` in the request.

The following elements should be considered when retrieving the Pre-Arbitration Response details.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.
DisputePreArbResponseId	A unique VROL-generated ID for each response to a saved or submitted Pre-Arbitration Response.
AssociatedTransGroupId	Identifies a group of transactions associated to the selected transaction, and is used to access the associated transaction detail information.

`SIGetDisputePreArbResponseDetails` contains request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SIGetDisputePreArbResponseDetailsRequest>
    <RequestHeader>
        <User id="tesve8763u" type="internalId"/>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <DisputePreArbResponseId>123456</DisputePreArbResponseId>
    </requestData>
</SIGetDisputePreArbResponseDetailsRequest>
```

### Response Message—SOAP-based Services

**Note:** The `SIGetDisputePreArbResponseDetailsResponse` message includes the dispute history, which consists of information from all previously-submitted disputes and dispute responses that are part of the same dispute thread.

```
<SIGetDisputePreArbResponseDetailsResponse>
    <Status>
        <Code>I-920000001</Code>
        <Message>No associated transactions found.</Message>
    </Status>
    <responseData>
        <TransactionInfo>
            <AccountNumber>410000000000</AccountNumber>
            <Transaction id="0000000000000000">
                <TranDate>2019-06-23</TranDate>
                <Amount currency="344">8.24</Amount>
            </Transaction>
            <MerchantName>MERCHANT NAME</MerchantName>
            <MerchantCity>MERCHANT CITY</MerchantCity>
            <MerchantCountrySubEntityCode>CA
```

```
</MerchantCountrySubEntityCode>
<MerchantCountryCode>MU</MerchantCountryCode>
<MerchantPostalCode>94425</MerchantPostalCode>
<MCC>4111</MCC>
<ARN>00000000000000000000000000000001</ARN>
<ECMOTO>1</ECMOTO>
<ReimbursementAttribute>A</ReimbursementAttribute>
<CPD>2019-06-24</CPD>
<TransactionCategory>D</TransactionCategory>
<AcquirerBID>10000000</AcquirerBID>
<NetworkID>0002</NetworkID>
<AcquirerBIN>400001</AcquirerBIN>
<RetrievalReferenceNumber>7100000000
</RetrievalReferenceNumber>
<SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
<TranDestAmt currency="344">8.24</TranDestAmt>
<ProcessingCode>00</ProcessingCode>
<IssuerName>issuer name</IssuerName>
<AcquirerName>acquirer name</AcquirerName>
<AccountType>0</AccountType>
<DisputeJurisdiction jurisdictionCd="DOM">
    <Jurisdiction>DOMESTIC-MU</Jurisdiction>
    <IssuerRegion>CEMEA</IssuerRegion>
    <AcquirerRegion>CEMEA</AcquirerRegion>
    <CountryCd>MU</CountryCd>
</DisputeJurisdiction>
</TransactionInfo>
<VisaCaseNumber>1000000000</VisaCaseNumber>
<DisputePreArbResponseId>123456</DisputePreArbResponseId>
<MemberCaseNumber>case</MemberCaseNumber>
<DisputeCategoryCondition>10.4 - Fraud: Card Absent Environment
</DisputeCategoryCondition>
<DisputeCategory>10</DisputeCategory>
<DisputeCondition>4</DisputeCondition>
<DisputeCardholderInfo>
    <NameType>PN</NameType>
    <CardholderCountryCode>US</CardholderCountryCode>
    <CardholderOptInInd>false</CardholderOptInInd>
</DisputeCardholderInfo>
<DisputeHistorySummary>
    <DisputeHistory>
        <SubmittedBy>Issuer</SubmittedBy>
        <DisputeCategory>10</DisputeCategory>
        <DisputeItemType>DISPREQ</DisputeItemType>
        <RecallInd>false</RecallInd>
        <EventDate>2019-06-28T05:36:40.244</EventDate>
        <UserOrOrgName>String</UserOrOrgName>
        <DisputeItemID>0</DisputeItemID>
    </DisputeHistory>
</DisputeHistorySummary>
<Note>note</Note>
<CaseHistorySummary>
    <CaseHistory>
        <Event>name</Event>
        <CPD>2019-06-28</CPD>
        <DisputeAmount currency="840">3000.00</DisputeAmount>
        <DisputeCategoryCondition>
```

```

          10.4 Fraud - Card Absent Environment
          </DisputeCategoryCondition>
        </CaseHistory>
      </CaseHistorySummary>
      <PreArbResponse>DECL</PreArbResponse>
      <LiabilityNotAcceptedFully>String
      </LiabilityNotAcceptedFully>
      <IssuerAcquirerContactInfo>
        <ContactName>name</ContactName>
      </IssuerAcquirerContactInfo>
      <RecallInd>false</RecallInd>
    </ResponseData>
  </SIGetDisputePreArbResponseDetailsResponse>

```

### Request Message—REST-based Services

```

{
  "RequestHeader": {
    "User": {
      "id": "tesve8763u",
      "type": "internalId"
    },
    "MemberRole": "I"
  },
  "requestData": {
    "VisaCaseNumber": 1000000000,
    "DisputePreArbResponseId": 123456,
  }
}

```

### Response Message—REST-based Services

```

{
  "Status": {
    "Code": "I-920000001",
    "Message": "No associated transactions found."
  },
  "responseData": {
    "TransactionInfo": {
      "AccountNumber": "00000000001"
    },
    "Transaction": {
      "TranDate": "2019-06-23",
      "Amount": {
        "value": 3000.00,
        "currency": "840"
      },
      "id": "8111111111111111"
    },
    "MerchantName": "MERCHANT NAME",
    "MerchantCity": "MERCHANT CITY",
    "MerchantCountrySubEntityCode": "CA",
    "MerchantCountryCode": "MU",
    "MerchantPostalCode": "94425",
    "MCC": "4111",
    "ARN": "74000000000000000000000000000000",
    "ECMOTO": "1",
    "ReimbursementAttribute": "A",
  }
}

```

```
"CPD": "2019-06-24",
"TransactionCategory": "D",
"AcquirerBID": "10000011",
"NetworkID": "0002",
"AcquirerBIN": "400000",
"RetrievalReferenceNumber": "900000000000",
"SystemTraceAuditNumber": "200000",
"TranDestAmt": {
    "value": 3000.00,
    "currency": "840"
},
"ProcessingCode": "00",
"IssuerName": "Bank",
"AcquirerName": "Bank",
"AccountType": "0",
"DisputeJurisdiction": {
    "Jurisdiction": "DOMESTIC-MU",
    "IssuerRegion": "CEMEA",
    "AcquirerRegion": "CEMEA",
    "CountryCd": "MU",
    "jurisdictionCd": "DOM"
},
"VisaCaseNumber": 1000000000,
"DisputePreArbResponseId": 123456,
"MemberCaseNumber": "memberCaseNumber",
"DisputeCategoryCondition": "10.4 - Fraud: Card
Absent Environment",
"DisputeCategory": {
    "value": "10"
},
"DisputeCondition": "4",
"DisputeCardholderInfo": {
    "NameType": "PN",
    "CardholderCountryCode": "US",
    "CardholderOptInInd": false
},
"DisputeHistorySummary": {
    "DisputeHistory": {
        "SubmittedBy": "Issuer",
        "DisputeCategory": {
            "value": "10"
        }
    },
    "DisputeItemType": "DISPREQ",
    "RecallInd": false,
    "EventDate": "2019-12-17T09:30:47Z",
    "UserOrOrgName": "String",
    "DisputeItemID": 0
},
"Note": "note",
"CaseHistorySummary": {
    "CaseHistory": {
        "Event": "name",
        "CPD": "2016-01-01",
        "DisputeAmount": {
            "value": 3000.00,
            "currency": "840"
        }
    }
}
```

```

        "currency": "840"
    },
    "DisputeCategoryCondition": "10.4 Fraud - Card
Absent Environment"
}, {
    "Event": "Pre-Arbitration",
    "CPD": "2019-06-28",
    "DisputeAmount": {
        "value": 3000.00,
        "currency": "840"
    },
    "DisputeCategoryCondition": "10.4 Fraud - Card
Absent Environment"
}, {
    "Event": "Pre-Arbitration Response",
    "CPD": "2019-06-28",
    "DisputeAmount": {
        "value": 3000.00,
        "currency": "840"
    },
    "DisputeCategoryCondition": "10.4 Fraud - Card
Absent Environment"
} ]
},
"PreArbResponse": "DECL",
"LiabilityNotAcceptedFully": "Explanation...",
"IssuerAcquirerContactInfo": {
    "ContactName": "Test"
},
"RecallInd": false
}
}
}

```

## SICreateDisputePreCompRequest

SICreateDisputePreCompRequest allows RTSI users to create the Pre-Compliance by providing details of the Pre-Compliance.

The following elements should be considered when creating Pre-Compliance or receiving a response from this service.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.  When creating a Pre-Compliance from an existing case, provide only the VisaCaseNumber. A new case is created if VisaCaseNumber is not supplied in the request.
RolTransactionId	Uniquely identifies an original transaction retrieved from VROL's Transaction Inquiry data repository.  This element is required when creating a Pre-Compliance from a selected original transaction.
DisputePreCompid	A unique VROL-generated ID for each saved or submitted Pre-Compliance.

SICreateDisputePreComp contains request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SICreateDisputePreCompRequest>
    <RequestHeader>
        <User_id="abcde1234f" type="internalId"/>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <DisputeAmount currency="000">0</DisputeAmount>
        <DisputeAmountChangeReason>reason</DisputeAmountChangeReason>
        <InitiatePreFilingReason>reason</InitiatePreFilingReason>
        <FirstRuleViolatedCode>C050</FirstRuleViolatedCode>
        <RulesViolatedAdditionalInfo>text</RulesViolatedAdditionalInfo>
        <ViolationDate>2019-07-01</ViolationDate>
        <FinancialLossOccurInd>Y</FinancialLossOccurInd>
        <FinancialLossExplanation>explanation
        </FinancialLossExplanation>
        <Action>Submit</Action>
    </requestData>
</SICreateDisputePreCompRequest>
```

### Response Message—SOAP-based Services

```
<SICreateDisputePreCompResponse>
    <Status>
        <Code>I-920000001</Code>
        <Message>No associated transactions found.</Message>
    </Status>
    <responseData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <DisputePreCompId>123456</DisputePreCompId>
    </responseData>
</SICreateDisputePreCompResponse>
```

## Request Message—REST-based Services

```
{
    "RequestHeader": {
        "User": {
            "id": "abcde1234f",
            "type": "internalId"
        },
        "MemberRole": "I"
    },
    "requestData": {
        "VisaCaseNumber": 1000000000,
        "MemberCaseNumber": "test",
        "DisputeAmount": {
            "currency": "840",
            "value": "11906.57"
        },
        "Note": "note",
        "DisputeAmountChangeReason": "reason",
        "InitiatePreFilingReason": "reason",
        "FirstRuleViolatedCode": "C050",
        "ViolationDate": "2019-07-01",
        "RulesViolatedAdditionalInfo": "text",
        "FinancialLossOccurInd": "N",
        "Action": "Submit",
    }
}
```

## Response Message—REST-based Services

```
{
    "Status": {
        "Code": "I-300000000",
        "Message": "Successfully completed Operation."
    },
    "responseData": {
        "VisaCaseNumber": 1000000000,
        "DisputePreCompId": 123456,
    }
}
```

## **SIGetDisputePreCompDetailsRequest**

SIGetDisputePreCompDetailsRequest allows RTSI users to request for the Pre-Compliance questionnaire data to be returned in image format, as well as in XML. The returned image will be in TIFF or PDF format. To retrieve the dispute as an image, the user must set <IncludeDisputeAsImageInd>=true. The image data will be returned in the <Image> group in the SIGetDisputePreCompDetailsResponse.

**Note:** Only the Pre-Compliance questionnaire will be returned as an image. Any attached documents must be retrieved using the AttachmentId value(s) in this response as input to the DocId field(s) of SIGetImageRequest.

**Note:** Any image attached to the Pre-Compliance questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=<false> in the request.

The following elements should be considered when retrieving the Pre-Compliance details.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.
DisputePreCompld	A unique VROL-generated ID for each saved or submitted Pre-Compliance.

`SIGetDisputePreCompDetails` contains request and response messages as shown in the following sample messages.

## Request Message—SOAP-based Services

```
<SIGetDisputePreCompDetailsRequest>
    <RequestHeader>
        <MemberRole>I</MemberRole>
        <User id="tesve8763u" type="internalId"/>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <DisputePreCompId>123456789</DisputePreCompId>
        <IncludeDisputeAsImageInd>false</IncludeDisputeAsImageInd>
        <DownloadImageInd>false</DownloadImageInd>
    </requestData>
</SIGetDisputePreCompDetailsRequest>
```

## Response Message—SOAP-based Services

```
<SIGetDisputePreCompDetailsResponse>
    <Status>
        <Code>I-920000001</Code>
        <Message>No associated transactions found.</Message>
    </Status>
    <responseData>
        <TransactionInfo>
            <AccountNumber>410000000000</AccountNumber>
            <Transaction id="0000000000000000">
                <TranDate>2019-06-23</TranDate>
                <Amount currency="344">8.24</Amount>
            </Transaction>
            <MerchantName>MERCHANT NAME</MerchantName>
            <MerchantCity>MERCHANT CITY</MerchantCity>
            <MerchantCountrySubEntityCode>CA
            </MerchantCountrySubEntityCode>
            <MerchantCountryCode>US</MerchantCountryCode>
            <MerchantPostalCode>94416</MerchantPostalCode>
            <MCC>5411</MCC>
            <ARN>000000000000000000000001</ARN>
            <ECMOTO>1</ECMOTO>
            <ReimbursementAttribute>A</ReimbursementAttribute>
            <CPD>2019-06-24</CPD>
            <TransactionCategory>D</TransactionCategory>
            <AcquirerBID>10000000</AcquirerBID>
        </TransactionInfo>
    </responseData>
</SIGetDisputePreCompDetailsResponse>
```

```

<NetworkID>0002</NetworkID>
<AcquirerBIN>400001</AcquirerBIN>
<RetrievalReferenceNumber>7100000000
</RetrievalReferenceNumber>
<SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
<TranDestAmt currency="840">11906.57</TranDestAmt>
<ProcessingCode>00</ProcessingCode>
<IssuerName>issuer name</IssuerName>
<AcquirerName>acquirer name</AcquirerName>
<AccountType>0</AccountType>
<DisputeJurisdiction jurisdictionCd="DOM">
    <Jurisdiction>DOMESTIC-US</Jurisdiction>
    <IssuerRegion>US</IssuerRegion>
    <AcquirerRegion>US</AcquirerRegion>
    <CountryCd>US</CountryCd>
</DisputeJurisdiction>
</TransactionInfo>
<VisaCaseNumber>1000000000</VisaCaseNumber>
<DisputePreCompId>123456789</DisputePreCompId>
<DisputeAmount currency="840">11906.57</DisputeAmount>
<CardholderInfo>
    <NameType>PN</NameType>
    <CardholderCountryCode>US</CardholderCountryCode>
    <CardholderOptInInd>false</CardholderOptInInd>
</CardholderInfo>
<IssuerAcquirerContactInfo>
    <ContactName>a</ContactName>
</IssuerAcquirerContactInfo>
<InitiatePreFilingReason>a</InitiatePreFilingReason>
<FirstRuleViolatedCode>C050</FirstRuleViolatedCode>
<RulesViolatedAdditionalInfo>a</RulesViolatedAdditionalInfo>
<ViolationDate>2019-01-01</ViolationDate>
<FinancialLossOccurInd>Y</FinancialLossOccurInd>
<FinancialLossExplanation>a</FinancialLossExplanation>
<RecallInd>false</RecallInd>
</ResponseData>
</SIGetDisputePreCompDetailsResponse>

```

### Request Message—REST-based Services

```

{
    "RequestHeader": {
        "MemberRole": "I"
    },
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": 1000000000,
        "DisputePreCompId": 123456789,
        "IncludeDisputeAsImageInd": "true",
        "DownloadImageInd": "true"
    }
}

```

### Response Message—REST-based Services

```
{
    "Status": {

```

```
        "Code": "I-920000001",
        "Message": "No associated transactions found."
    },
    "ResponseData": {
        "TransactionInfo": {
            "AccountNumber": "00000000001",
            "$": "..."
        },
        "VisaCaseNumber": 1000000000,
        "DisputePreCompId": 123456789,
        "MemberCaseNumber": "a",
        "QuestionnaireRecalledDate": "2016-08-13",
        "DisputeAmount": {
            "@currency": "000",
            "$": 0
        },
        "CardholderInfo": "...",
        "MerchantContactInfo": "...",
        "Note": "a",
        "IssuerAcquirerContactInfo": {
            "ContactName": "a",
            "ContactPhone": 6501111111,
            "$": "..."
        },
        "DisputeAmountChangeReason": "test",
        "ExplanationOfCreditPresented": "test",
        "InitiatePreFilingReason": "a",
        "FirstRuleViolatedCode": "a",
        "SecondRuleViolatedCode": "a",
        "ThirdRuleViolatedCode": "a",
        "RulesViolatedAdditionalInfo": "a",
        "ViolationDate": "2016-01-01",
        "ViolationDiscoveryDate": "2016-01-01",
        "ViolationDiscoveryDateInfo": "a",
        "FinancialLossOccurInd": "N",
        "FinancialLossExplanation": "a",
        "RecallInd": "true",
        "RecallReason": "String",
        "Image": {
            "RemoteFileName": "a",
            "DocComment": "a",
            "MimeType": "image/jpeg",
            "DocType": "VSEXT",
            "ImageType": "DOCUMENT",
            "ImageSize": {
                "@unit": "BY",
                "$": 0
            },
            "TransitName": "JDM00000",
            "AttachmentId": "String"
        },
        "DisputeAsImageResponseDescriptor": {
            "ContentType": "application/pdf",
            "ImageData": "xxxxxx"
        }
    }
}
```

## SICreateDisputePreCompResponseRequest

SICreateDisputePreCompResponseRequest allows RTSI users to respond to a Pre-Compliance.

The following XML elements should be considered when responding to a Pre-Compliance or receiving a response from this service.

Element	Description
Response	Indicates the user's response to the Pre-Compliance.  Domain values: <ul style="list-style-type: none"> <li>• FULL=Accept Full</li> <li>• PART=Accept Partial</li> <li>• DECL=Decline</li> </ul>
DisputePreCompResponseId	A unique VROL-generated ID for each response to a saved or submitted Pre-Compliance.

SICreateDisputePreCompResponse contains request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SICreateDisputePreCompResponseRequest>
    <RequestHeader>
        <User id="abcde1234f" type="internalId"/>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <MemberCaseNumber>555555555555</MemberCaseNumber>
        <Note>a</Note>
        <PreCompResponse>DECL</PreCompResponse>
        <ContinuePreFilingReason>reason</ContinuePreFilingReason>
        <IssuerAcquirerContactInfo>
            <ContactName>name</ContactName>
            <ContactPhone>6501111111</ContactPhone>
            <ContactFax>6504000000</ContactFax>
            <ContactOther>6500000001</ContactOther>
            <ContactEmail>abc@defgmail.com</ContactEmail>
            <Action>Submit</Action>
        </IssuerAcquirerContactInfo>
    </requestData>
</SICreateDisputePreCompResponseRequest>
```

## Response Message—SOAP-based Services

```
<SICreateDisputePreCompResponseResponse>
  <Status>
    <Code>I-300000000</Code>
    <Message>Successfully completed Operation.</Message>
  </Status>
  <responseData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <DisputePreCompResponseId>123456</DisputePreCompResponseId>
    <DocIdList>
      <DocId>1</DocId>
    </DocIdList>
  </responseData>
</SICreateDisputePreCompResponseResponse>
```

## Request Message—REST-based Services

```
{
  "RequestHeader": {
    "User": {
      "id": "abcde1234f",
      "type": "internalId"
    },
    "MemberRole": "I"
  },
  "requestData": {
    "VisaCaseNumber": 1000000000,
    "MemberCaseNumber": "555555555555",
    "MerchantContactInfo": {
      "ContactName": "name",
      "ContactAddress": "Salt Lake",
      "ContactCity": "FC",
      "ContactStateCode": "CA",
      "ContactPostalCode": 11111,
      "ContactCountryCode": "FC",
      "ContactPhone": 11111111,
      "ContactFax": 22222222,
      "ContactEmail": "abc@xyz.com",
      "ContactURL": "www.abc.com",
      "ContactBestTimeToCall": "6-9PM",
      "ContactOptInInd": "true"
    },
    "PreCompResponse": "DECL",
    "ContinuePreFilingReason": "reason",
    "IssuerAcquirerContactInfo": {
      "ContactName": "a",
      "ContactPhone": 6501111111,
      "ContactFax": 22222222,
      "ContactEmail": "abc@xyz.com",
      "Action": "Submit",
    }
}
```

## Response Message—REST-based Services

```
{
  "Status": {
    "Code": "I-300000000",
    "Message": "Successfully completed Operation."
  },
  "ResponseData": {
    "VisaCaseNumber": 1000000000,
    "DisputePreCompResponseId": 123456,
    "DocIdList": [
      "DocId": 1
    ]
  }
}
```

## SIGetDisputePreCompResponseDetailsRequest

SIGetDisputePreCompResponseDetailsRequest allows RTSI users to request for the Pre-Compliance Response questionnaire data to be returned in image format, as well as in XML. The returned image will be in TIFF or PDF format. To retrieve the dispute as an image, the user must set <IncludeDisputeAsImageInd>=true. The image data will be returned in the <Image> group in the SIGetDisputePreCompResponseDetailsResponse.

**Note:** Only the Pre-Compliance Response questionnaire will be returned as an image. Any attached documents must be retrieved using the AttachmentId value(s) in this response as input to the DocId(s) field of SIGetImageRequest.

**Note:** Any image attached to the Pre-Compliance Response questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.

The following elements should be considered when retrieving the Pre-Compliance Response details.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.
DisputePreCompResponseId	A unique VROL-generated ID for each response to a saved or submitted Pre-Compliance Response.

SIGetDisputePreCompResponseDetails contains request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SIGetDisputePreCompResponseDetailsRequest>
  <RequestHeader>
    <User id="tesve8763u" type="internalId"/>
    <MemberRole>I</MemberRole>
  </RequestHeader>
  <requestData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <DisputePreCompResponseId>123456789</DisputePreCompResponseId>
    <IncludeDisputeAsImageInd>false</IncludeDisputeAsImageInd>
    <DownloadImageInd>false</DownloadImageInd>
  </requestData>
</SIGetDisputePreCompResponseDetailsRequest>
```

### Response Message—SOAP-based Services

```
<SIGetDisputePreCompResponseDetailsResponse>
  <Status>
    <Code>I-920000001</Code>
    <Message>No associated transactions found.</Message>
  </Status>
  <responseData>
    <TransactionInfo>
      <AccountNumber>410000000000</AccountNumber>
      <Transaction id="0000000000000000">
        <TranDate>2019-06-23</TranDate>
        <Amount currency="840">11906.57</Amount>
      </Transaction>
      <MerchantName>MERCHANT NAME</MerchantName>
      <MerchantCity>MERCHANT CITY</MerchantCity>
      <MerchantCountrySubEntityCode>CA
      </MerchantCountrySubEntityCode>
      <MerchantCountryCode>US</MerchantCountryCode>
      <MerchantPostalCode>94416</MerchantPostalCode>
      <MCC>5411</MCC>
      <ARN>00000000000000000001</ARN>
      <ECMOTO>1</ECMOTO>
      <ReimbursementAttribute>A</ReimbursementAttribute>
      <CPD>2019-06-24</CPD>
      <TransactionCategory>D</TransactionCategory>
      <AcquirerBID>10000000</AcquirerBID>
      <NetworkID>0002</NetworkID>
      <AcquirerBIN>400001</AcquirerBIN>
      <RetrievalReferenceNumber>7100000000
      </RetrievalReferenceNumber>
      <SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
      <TranDestAmt currency="840">11906.57</TranDestAmt>
      <ProcessingCode>00</ProcessingCode>
      <IssuerName>issuer name</IssuerName>
      <AcquirerName>acquirer name</AcquirerName>
      <AccountType>0</AccountType>
      <DisputeJurisdiction jurisdictionCd="DOM">
        <Jurisdiction>DOMESTIC-US</Jurisdiction>
        <IssuerRegion>US</IssuerRegion>
        <AcquirerRegion>US</AcquirerRegion>
        <CountryCd>US</CountryCd>
      </DisputeJurisdiction>
    </TransactionInfo>
  </responseData>
</SIGetDisputePreCompResponseDetailsResponse>
```

```

        </DisputeJurisdiction>
    </TransactionInfo>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <DisputePreCompResponseId>123456789</DisputePreCompResponseId>
    <MemberCaseNumber>a</MemberCaseNumber>
    <DisputeAmount currency="840">11906.57</DisputeAmount>
    <CardholderInfo>
        <NameType>PN</NameType>
        <CardholderCountryCode>US</CardholderCountryCode>
        <CardholderOptInInd>false</CardholderOptInInd>
    </CardholderInfo>
    <MerchantContactInfo>
        <ContactName>Test</ContactName>
        <ContactAddress>salt lake</ContactAddress>
        <ContactCity>FC</ContactCity>
        <ContactStateCode>CA</ContactStateCode>
        <ContactPostalCode>11111</ContactPostalCode>
        <ContactCountryCode>FC</ContactCountryCode>
        <ContactPhone>11111111</ContactPhone>
        <ContactFax>22222222</ContactFax>
        <ContactEmail>abc@xyz.com</ContactEmail>
        <ContactURL>www.abc.com</ContactURL>
        <ContactBestTimeToCall>6-9PM</ContactBestTimeToCall>
        <ContactOptInInd>true</ContactOptInInd>
    </MerchantContactInfo>
    <PreCompResponse>DECL</PreCompResponse>
    <ContinuePreFilingReason>reason</ContinuePreFilingReason>
    <IssuerAcquirerContactInfo>
        <ContactName>a</ContactName>
        <ContactPhone>6501111111</ContactPhone>
        <ContactFax>22222222</ContactFax>
        <ContactEmail>abc@xyz.com</ContactEmail>
    </IssuerAcquirerContactInfo>
</ResponseData>
</SIGetDisputePreCompResponseDetailsResponse>

```

### Request Message—REST-based Services

```
{
    "RequestHeader": {
        "User": {
            "id": "tesve8763u",
            "type": "loginId"
        },
        "MemberRole": "I"
    },
    "requestData": {
        "VisaCaseNumber": 1000000000,
        "DisputePreCompResponseId": 123456789,
        "IncludeDisputeAsImageInd": "false",
        "DownloadImageInd": "false"
    }
}
```

### Response Message—REST-based Services

```
{
```

```
"Status": {
    "Code": "I-920000001",
    "Message": "No associated transactions found."
},
"ResponseData": {
    "TransactionInfo": {
        "AccountNumber": "00000000001",
        "Transaction": {
            "TranDate": "2019-07-12",
            "Amount": {
                "value": 11906.57,
                "currency": "840"
            },
            "id": "81111111111110"
        },
        "MerchantName": "MERCHANT NAME",
        "MerchantCity": "MERCHANT CITY",
        "MerchantCountrySubEntityCode": "CA",
        "MerchantCountryCode": "US",
        "MerchantPostalCode": "94416",
        "MCC": "5411",
        "ARN": "20000000000000000000000000",
        "ECMOTO": "1",
        "ReimbursementAttribute": "A",
        "CPD": "2019-07-13",
        "TransactionCategory": "D",
        "AcquirerBID": "10000001",
        "NetworkID": "0002",
        "AcquirerBIN": "200000",
        "RetrievalReferenceNumber": "91111111111",
        "SystemTraceAuditNumber": "100000",
        "TranDestAmt": {
            "value": 11906.57,
            "currency": "840"
        },
        "ProcessingCode": "00",
        "IssuerName": "Bank",
        "AcquirerName": "Bank",
        "AccountType": "0",
        "DisputeJurisdiction": {
            "Jurisdiction": "DOMESTIC-US",
            "IssuerRegion": "US",
            "AcquirerRegion": "US",
            "CountryCd": "US",
            "jurisdictionCd": "DOM"
        }
    },
    "VisaCaseNumber": 1000000000,
    "DisputePreCompResponseId": 123456789,
    "MemberCaseNumber": "a",
    "DisputeAmount": {
        "value": 11906.57,
        "currency": "840"
    },
    "CardholderInfo": {
        "NameType": "PN",
        "CardholderCountryCode": "US",
        "CardholderName": "John Doe"
    }
}
```

```
        "CardholderOptInInd": false
    },
    "MerchantContactInfo": {
        "ContactName": "Test",
        "ContactAddress": "salt lake",
        "ContactCity": "FC",
        "ContactStateCode": "CA",
        "ContactPostalCode": "11111",
        "ContactCountryCode": "FC",
        "ContactPhone": "11111111",
        "ContactFax": "22222222",
        "ContactEmail": "abc@xyz.com",
        "ContactURL": "www.abc.com",
        "ContactBestTimeToCall": "6-9PM",
        "ContactOptInInd": true
    },
    "PreCompResponse": "DECL",
    "ContinuePreFilingReason": "reason",
    "IssuerAcquirerContactInfo": {
        "ContactName": "a",
        "ContactPhone": 6501111111,
        "ContactFax": "22222222",
        "ContactEmail": "abc@xyz.com"
    },
}
}
```

## SIGetAcceptanceNotificationDetailsRequest

**SIGetAcceptanceNotificationDetailsRequest** allows RTSI users to retrieve the acceptance notification details for Pre-Arbitrations, Pre-Arbitration responses, Pre-Compliances, or Pre-Compliance responses that were accepted and closed.

The following elements should be considered when retrieving the acceptance notification details for Pre-Arbitrations, Pre-Arbitration responses, Pre-Compliances, or Pre-Compliance responses that were accepted and closed.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.
DisputePreArbId	A unique VROL-generated ID for each saved or submitted Pre-Arbitration.
DisputePreArbResponselD	A unique VROL-generated ID for each response to a saved or submitted Pre-Arbitration.
DisputePreComplId	A unique VROL-generated ID for each saved or submitted Pre-Compliance.
DisputePreCompResponselD	A unique VROL-generated ID for each response to a saved or submitted Pre-Compliance.

`SIGetAcceptanceNotificationDetails` request and response messages are shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SIGetAcceptanceNotificationDetailsRequest>
  <RequestHeader>
    <User id="abcde1234f" type="internalId"/>
    <MemberRole>A</MemberRole>
  </RequestHeader>
  <requestData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <DisputePreArbResponseId>400001</DisputePreArbResponseId>
  </requestData>
</SIGetAcceptanceNotificationDetailsRequest>
```

## Response Message—SOAP-based Services

```
<SIGetAcceptanceNotificationDetailsResponse>
    <Status>
        <Code>I-300000000</Code>
        <Message>Successfully completed Operation.</Message>
    </Status>
    <responseData>
        <TransactionInfo>
            <AccountNumber>4000000000000005</AccountNumber>
            <Transaction id="8000000000000002">
                <TranDate>2019-07-12</TranDate>
                <Amount currency="840">13702.87</Amount>
            </Transaction>
            <MerchantName>MERCHANT NAME</MerchantName>
            <MerchantCity>MERCHANT CITY</MerchantCity>
            <MerchantCountrySubEntityCode>CA
            </MerchantCountrySubEntityCode>
            <MerchantCountryCode>US</MerchantCountryCode>
            <MerchantPostalCode>90006</MerchantPostalCode>
            <MCC>5411</MCC>
            <ARN>200000000000000000000001</ARN>
            <ECMOTO>1</ECMOTO>
            <ReimbursementAttribute>A</ReimbursementAttribute>
            <CPD>2019-07-12</CPD>
            <TransactionCategory>D</TransactionCategory>
            <AcquirerBID>10000009</AcquirerBID>
            <NetworkID>0002</NetworkID>
            <AcquirerBIN>200008</AcquirerBIN>
            <RetrievalReferenceNumber>900000000003
            </RetrievalReferenceNumber>
            <SystemTraceAuditNumber>100003</SystemTraceAuditNumber>
            <CardholderName>Name</CardholderName>
            <TranDestAmt currency="840">13702.87</TranDestAmt>
            <ProcessingCode>00</ProcessingCode>
            <IssuerName>ABC Bank</IssuerName>
            <AcquirerName>DEF Bank</AcquirerName>
            <AccountType>00</AccountType>
            <DisputeJurisdiction jurisdictionCd="DOM">
                <Jurisdiction>DOMESTIC-US</Jurisdiction>
                <IssuerRegion>US</IssuerRegion>
                <AcquirerRegion>US</AcquirerRegion>
                <CountryCd>US</CountryCd>
            </DisputeJurisdiction>
        </TransactionInfo>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <MemberCaseNumber>555555555555</MemberCaseNumber>
        <AcceptItemType>PREARBRESP</AcceptItemType>
        <DisputeCategory>10/DisputeCategory>
        <AcceptanceAmount currency="840">13702.87</AcceptanceAmount>
        <AcceptanceDateTime>2019-07-17T10:10:24.000Z
        </AcceptanceDateTime>
    </responseData>
</SIGetAcceptanceNotificationDetailsResponse>
```

### Request Message—REST-based Services

```
{  
    "RequestHeader": {  
        "User": {  
            "id": "abcde1234f",  
            "type": "internalId"  
        }  
        "MemberRole": "A"  
    },  
    "requestData": {  
        "DisputePreArbResponseId": 000001  
        "VisaCaseNumber": 1000000000,  
    }  
}
```

### Response Message—REST-based Services

```
{  
    "Status": [ {  
        "Code": "I-3000000000",  
        "Message": "Successfully completed Operation."  
    } ],  
    "responseData": {  
        "TransactionInfo": {  
            "AccountNumber": "4000000000000005",  
            "Transaction": {  
                "TranDate": "2019-07-12",  
                "Amount": {  
                    "value": 13702.87,  
                    "currency": "840"  
                },  
                "id": "8000000000000002"  
            },  
            "MerchantName": "MERCHANT NAME",  
            "MerchantCity": "MERCHANT CITY",  
            "MerchantCountrySubEntityCode": "CA",  
            "MerchantCountryCode": "US",  
            "MerchantPostalCode": "90006",  
            "MCC": "5411",  
            "ARN": "20000000000000000000000000000001",  
            "ECMOTO": "1",  
            "ReimbursementAttribute": "A",  
            "CPD": "2019-07-12",  
            "TransactionCategory": "D",  
            "AcquirerBID": "10000009",  
            "NetworkID": "0002",  
            "AcquirerBIN": "200008",  
            "RetrievalReferenceNumber": "900000000003",  
            "SystemTraceAuditNumber": "100003",  
            "CardholderName": "Name",  
            "TranDestAmt": {  
                "value": 13702.87,  
                "currency": "840"  
            },  
            "ProcessingCode": "00",  
            "IssuerName": "ABC Bank",  
        }  
    }  
}
```

```
        "AcquirerName": "DEF Bank",
        "AccountType": "00",
        "DisputeJurisdiction": {
            "Jurisdiction": "DOMESTIC-US",
            "IssuerRegion": "US",
            "AcquirerRegion": "US",
            "CountryCd": "US",
            "jurisdictionCd": "DOM"
        },
        "VisaCaseNumber": 1000000000,
        "MemberCaseNumber": "555555555555",
        "AcceptItemType": "PREARBRESP",
        "DisputeCategory": {
            "value": "10"
        },
        "AcceptanceAmount": {
            "value": 13702.87,
            "currency": "840"
        },
        "AcceptanceDateTime": "2019-07-17T10:10:24.000Z"
    }
}
```

## SIChangeDisputeStatusRequest

SIChangeDisputeStatusRequest allows RTSI users to do the following actions:

- Delete a saved Pre-filing (Pre-Arbitration, Pre-Arbitration response, Pre-Compliance, or Pre-Compliance response)
- Recall a submitted Pre-Arbitration, Pre-Arbitration response, or Pre-Compliance

**Note:** Only the Pre-Compliance can be recalled but not the Pre-Compliance response.

The following elements should be considered when deleting a saved Pre-filing, or recalling a submitted Pre-Arbitration, Pre-Arbitration response, or Pre-Compliance.

Element	Description
ChangeItemID	<p>Uniquely identifies the Pre-filing item being recalled or deleted.</p> <p>ChangeItemID includes the following unique ID values for the corresponding Pre-filing item:</p> <ul style="list-style-type: none"> <li>• DisputePreArbId (for Pre-Arbitration)</li> <li>• DisputePreArbRespsId (for Pre-Arbitration response)</li> <li>• DisputePreComplId (for Pre-Compliance)</li> <li>• DisputePreCompRespsId (for Pre-Compliance response)</li> </ul> <p><b>Note:</b> <i>DisputePreCompRespsId is applicable only for the Delete action.</i></p>
ChangeItemType	<p>Indicates the type of Pre-filing being recalled or deleted.</p> <p>Domain values:</p> <ul style="list-style-type: none"> <li>• PREARB=PreArbitration</li> <li>• PREARBRESP=PreArbitration Response</li> <li>• PRECOMP=PreCompliance</li> <li>• PRECOMPRESP=PreCompliance Response</li> </ul> <p><b>Note:</b> <i>PRECOMPRESP is applicable only for the Delete action.</i></p>
Action	<p>Indicates the corresponding action to be applied to the Pre-filing item.</p> <p>Domain values: Recall, Delete</p>

SICheckDisputeStatus request and response messages are shown in the following sample messages.

### Request Message—Recall action—SOAP-based Services

```
<SICheckDisputeStatusRequest>
  <RequestHeader>
    <User id="abcde1234f" type="internalId"/>
    <MemberRole>I</MemberRole>
  </RequestHeader>
  <requestData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <ChangeItemID>123456</ChangeItemID>
    <ChangeItemType>PREARB</ChangeItemType>
    <RecallReason>reason</RecallReason>
    <Action>Recall</Action>
  </requestData>
</SICheckDisputeStatusRequest>
```

**Note:** For Cardholder No Longer Disputes, add the keyword '[CLD]' at the beginning of the reason text in the <RecallReason> element to allow VROL to recall the dispute transaction even after 3 calendar days (or 1 calendar day if the pinned original transaction in the dispute case is an Original Credit Transaction) provided that the liability is not shifted to the issuer.

### Response Message—Recall action—SOAP-based Services

```
<SIChangeDisputeStatusResponse>
  <Status>
    <Code>I-300000000</Code>
    <Message>Successfully completed Operation.</Message>
  </Status>
  <responseData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <ChangeItemID>123456</ChangeItemID>
    <DisputeFinancialInfo/>
  </responseData>
</SIChangeDisputeStatusResponse>
```

### Request Message—Recall action—REST-based Services

```
{
  "RequestHeader": {
    "User": {
      "id": "abcde1234f",
      "type": "internalId"
    },
    "MemberRole": "I"
  },
  "requestData": {
    "VisaCaseNumber": 1234567890,
    "ChangeItemID": 123456,
    "ChangeItemType": "PREARB",
    "RecallReason": "reason",
    "Action": "Recall"
  }
}
```

**Note:** For Cardholder No Longer Disputes, add the keyword '[CLD]' at the beginning of the reason text in the <RecallReason> element to allow VROL to recall the dispute transaction even even after 3 calendar days (or 1 calendar day if the pinned original transaction in the dispute case is an Original Credit Transaction) provided that the liability is not shifted to the issuer.

### Response Message—Recall action—REST-based Services

```
{
  "Status": {
    "Code": "I-300000000",
    "Message": "Successfully completed Operation."
  },
  "responseData": {
    "VisaCaseNumber": 1000000000,
    "ChangeItemID": 123456,
    "DisputeFinancialInfo": {}
  }
}
```

### Request Message—Delete action—SOAP-based Services

```
<SIChangeDisputeStatusRequest>
  <RequestHeader>
    <User id="abcde1234f" type="internalId"/>
    <MemberRole>I</MemberRole>
  </RequestHeader>
  <requestData>
    <VisaCaseNumber>1234567890</VisaCaseNumber>
    <ChangeItemID>1234</ChangeItemID>
    <ChangeItemType>PREARB</ChangeItemType>
    <Action>Delete</Action>
  </requestData>
</SIChangeDisputeStatusRequest>
```

### Response Message—Delete action—SOAP-based Services

```
<SIChangeDisputeStatusResponse>
  <Status>
    <Code>I-300000000</Code>
    <Message>Successfully completed Operation.</Message>
  </Status>
  <responseData>
    <VisaCaseNumber>1234567890</VisaCaseNumber>
    <ChangeItemID>123456</ChangeItemID>
    <DisputeFinancialInfo/>
  </responseData>
</SIChangeDisputeStatusResponse>
```

### Request Message—Delete action—REST-based Services

```
{
  "RequestHeader": {
    "User": {
      "id": "abcde1234f",
      "type": "internalId"
    },
    "MemberRole": "I"
  },
  "requestData": {
    "VisaCaseNumber": 1234567890,
    "ChangeItemID": 123456,
    "ChangeItemType": "PREARB",
    "Action": "Delete"
  }
}
```

**Response Message—Delete action—REST-based Services**

```
{  
    "Status": {  
        "Code": "I-300000000",  
        "Message": "Successfully completed Operation."  
    },  
    "responseData": {  
        "VisaCaseNumber": 1234567890,  
        "ChangeItemID": 123456,  
        "DisputeFinancialInfo": {}  
    }  
}
```

# Dispute Case Filings

11

*Dispute Case filing* is used generically to refer to arbitration and compliance cases that a member files with Visa.

- **Arbitration**—Allows Visa to assign liability for a disputed transaction. If a member cannot resolve a dispute by Pre-Arbitration, the member can request arbitration from Visa within the allowable time limits.
- **Compliance**—Allows a member that has no dispute right to file a complaint against another member for a violation of the *Visa Core Rules* and *Visa Product and Service Rules*.

These filings are used when the members have not been able to resolve a dispute without Visa's involvement.

Dispute case filings are valid on Visa, Interlink, and Plus networks.

Issuers and acquirers are required to use VROL to submit dispute case filings to Visa for domestic, regional, and inter-regional disputes. Both issuers and acquirers may file compliance and arbitration cases. The organization's setup, user's role, and user-level permissions and the status of the case affect the allowed actions that can be performed in a case. When a case is filed, VROL assigns a jurisdiction to the case based on the issuer and merchant region codes.

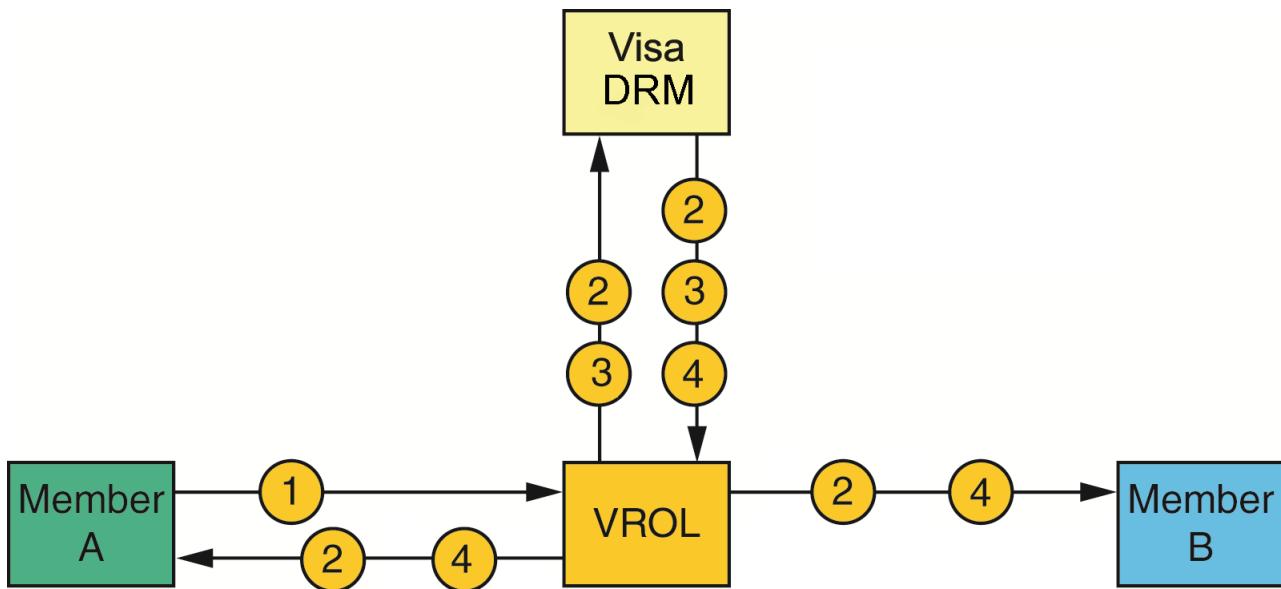
VROL automatically transfers funds when the final ruling is made. VROL initiates the resulting fee collection and funds disbursement transactions from the appropriate VROL BIN to the member BINs. See [Chapter 12, Miscellaneous Fees](#), for more information.

This chapter describes the case filing functions supported by RTSI.

The process is illustrated in [Figure 11–1](#).

## Dispute Case Filing Process—Summary

**Figure 11-1: Dispute Case Filing Process Flow**



1. The initiating member (issuer or acquirer) submits a dispute case filing request.
- Note:** *For submission of bundled case filing, the user can only select a maximum of 10 cases in a bundle. VROL returns an error if the user selects more than the allowed cases for the bundle.*
2. Visa acknowledges (accepts) or forwards the dispute case filing for rejection.
  - If the dispute case filing is accepted, an acknowledgment letter is sent to both the initiating and opposing members. The opposing member receives the incoming dispute case filing request details.
  - If the dispute case filing is forwarded for rejection, a rejection letter is sent to the initiating member and the process ends. Both members receive the rejection letter only if the case was closed with a rejection.
3. Visa may send a contact message to either member.
4. Visa rules on the case. The final ruling is sent to both members as a final decision letter.

VROL automatically transfers funds for disputes, Pre-Arbitrations, and arbitrations by processing the financial messages to transfer the liability when the final ruling, except for Rejects, is made on a case and is applicable for all jurisdiction. The VROL financial message(s) are for the opposer's dispute amount. For a final decision ruling of a Reject, there is no transfer of liability.

**Note:** *Automatic transfer of funds does not apply to Pre-Compliances and Compliances.*

Upon submission of the final decision for a case with a dispute category, the type of financial message processed by VROL to transfer the liability are as follows, depending on the type of case filing and workflow that the dispute follows:

### For Arbitration case filing:

- In an allocation workflow, if the acquirer wins in the final decision, VROL processes a Dispute Reversal (0422 for SMS/TC 35, TC 36, or TC 37 for BASE II).

**Note:** *The Dispute Reversal transaction amount has the same currency as the Dispute transaction amount.*

- In a collaboration workflow, if the issuer wins in the final decision, VROL processes a Dispute Response Reversal (0220 for SMS/TC 25, TC 26, or TC 27 for BASE II).

**Note:** *The Dispute Response Reversal transaction amount has the same currency as the Dispute Response transaction amount.*

#### For Compliance case filing:

Miscellaneous Fees (0220 for SMS/TC 10 for BASE II or 0422 for SMS/TC 20 for BASE II) are the financial messages used for the following Compliance scenarios:

- Issuer (Initiator) to Issuer (Opposer)
- Acquirer (Initiator) to Acquirer (Opposer)
- Acquirer (Initiator) to Issuer (Opposer)
- Issuer (Initiator) to Acquirer (Opposer)

For more information about dispute financials, see [Chapter 8, Dispute Financials](#).

**Note:** *For the final decision ruling of a bundled case filing, the liability shift financial occurs solely on the primary case in the bundle.*

**Note:** *The term "Initiating Member"/"Initiator" refers to the member who moved the dispute into the filing stage. The term "Opposing Member"/"Opposer" refers to the member who did not initiate the dispute case filing.*

## Dispute Case Filing Currencies

For Arbitration cases, the arbitration case filing amount currency is the same as the initiator's dispute amount currency.

**Note:** *The case's dispute amount tracks the amount from the last dispute financial present in the case.*

For compliance cases, case filings or appeals can be created with the filing amount in the following currencies per jurisdiction:

Europe	<ul style="list-style-type: none"><li>• USD</li><li>• British Pounds</li><li>• Euros</li></ul>
Visa Canada	<ul style="list-style-type: none"><li>• Canadian Dollars</li><li>• USD</li></ul>
All other jurisdictions	USD

The fees and penalties currency can be different from the case filing currency.

## Dispute Case Filing Process—Details

This section describes the process and XML operations used by issuers and acquirers to perform the tasks.

Depending on the workflow that the dispute follows, the user can file Arbitration or Compliance if the following conditions are met:

#### **Arbitration case filing:**

The issuer or acquirer can file Arbitration for dispute cases if the following conditions are met:

- For Fraud or Authorization disputes that follow the allocation workflow:
  - The issuer declined or partially declined a Pre-Arbitration request that is previously allocated against the acquirer
  - The acquirer is filing Arbitration
- For Processing Error or Consumer disputes that follow the collaboration workflow:
  - The acquirer declined or partially declined a Pre-Arbitration request
  - The issuer is filing Arbitration

#### **Compliance case filing:**

Users can file Compliance for dispute cases if the following conditions are met:

- The initiator filed the Pre-Compliance request
- The opposer declined or partially declined the Pre-Compliance request, or the current date is 30 days or more from the first Pre-Compliance date
- The initiator is filing Compliance (the initiator may be the issuer or acquirer)

SISubmitDisputeFilingRequest Web service can be used to perform the following actions:

1. Submitting dispute arbitration or compliance filing
2. Submitting dispute arbitration or compliance withdrawal request
3. Submitting dispute arbitration or compliance appeal request (post final ruling)

Similarly, SIGetDisputeFilingDetailsRequest Web service can be used to perform the following actions:

1. Retrieving dispute arbitration or compliance filing details
2. Retrieving dispute arbitration or compliance withdrawal request details
3. Retrieving dispute arbitration or compliance appeal request (post final ruling) details
4. Retrieving dispute case filing letters (Acknowledgement, Rejection, and Decision Letters) details

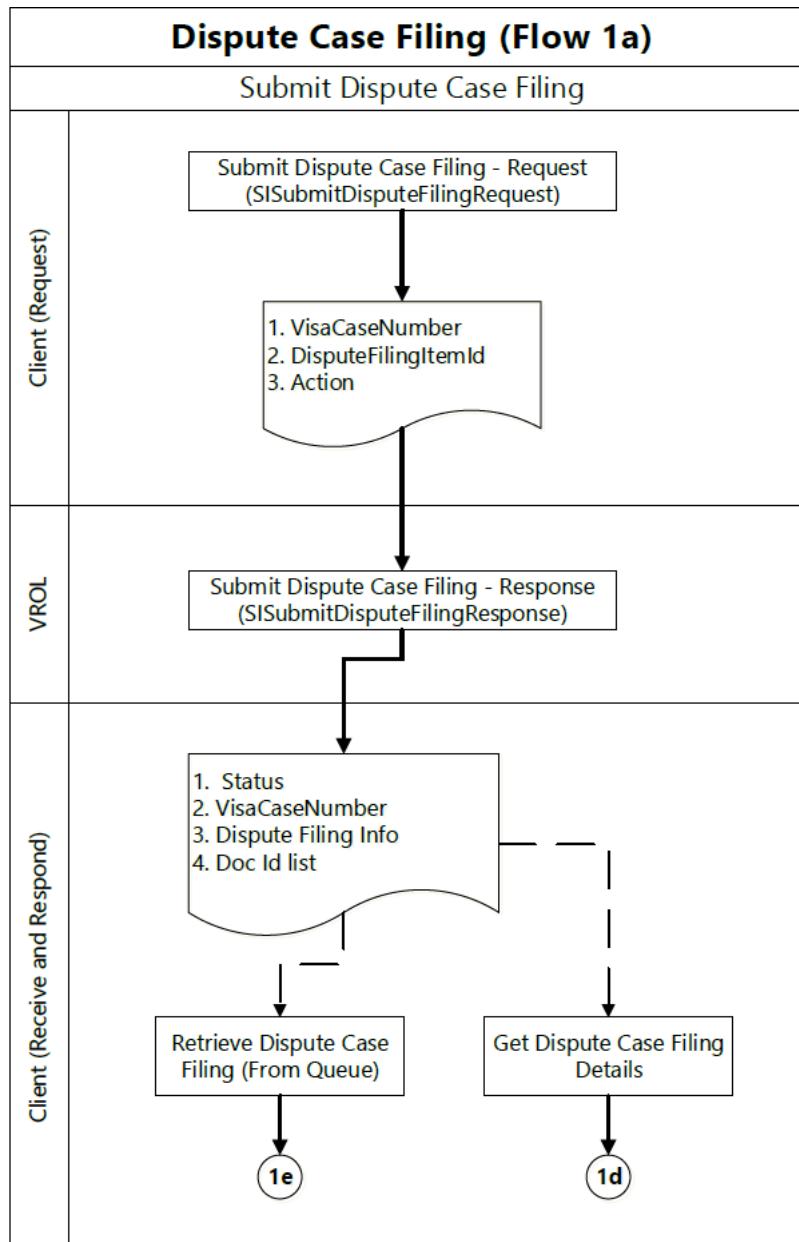
## Dispute Case Filing Flows

The following diagrams illustrate the dispute case filing process.

#### **Note:**

- *The diagrams list only the important elements. For the complete elements, refer to the Dispute tab of the Interface Element Specification (IES).*
- *The dashed lines in the flow refer to optional functions that the user can use, depending on the business process that the user wants to implement.*

**Figure 11–2: Dispute Case Filing—Flow 1a**



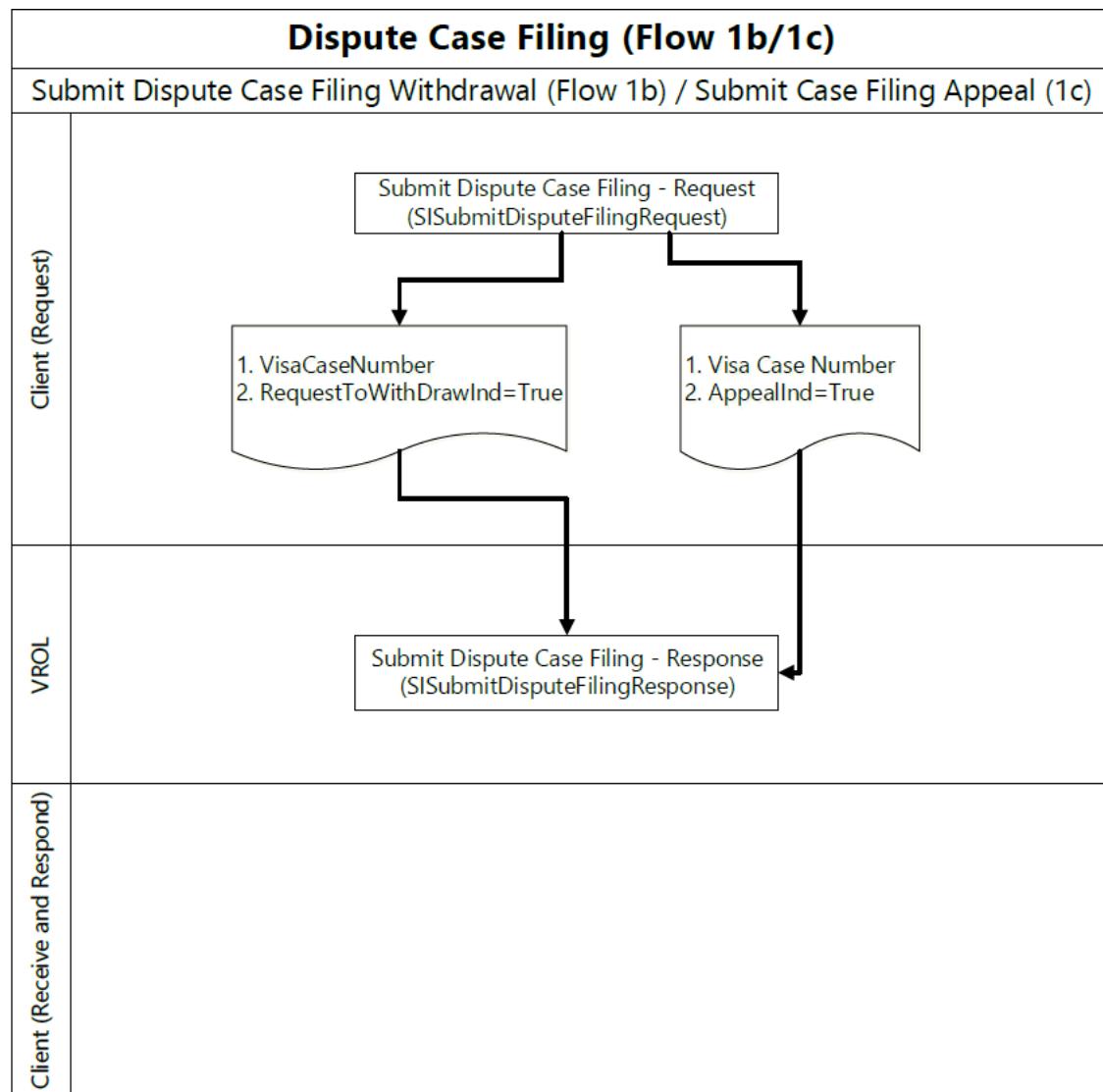
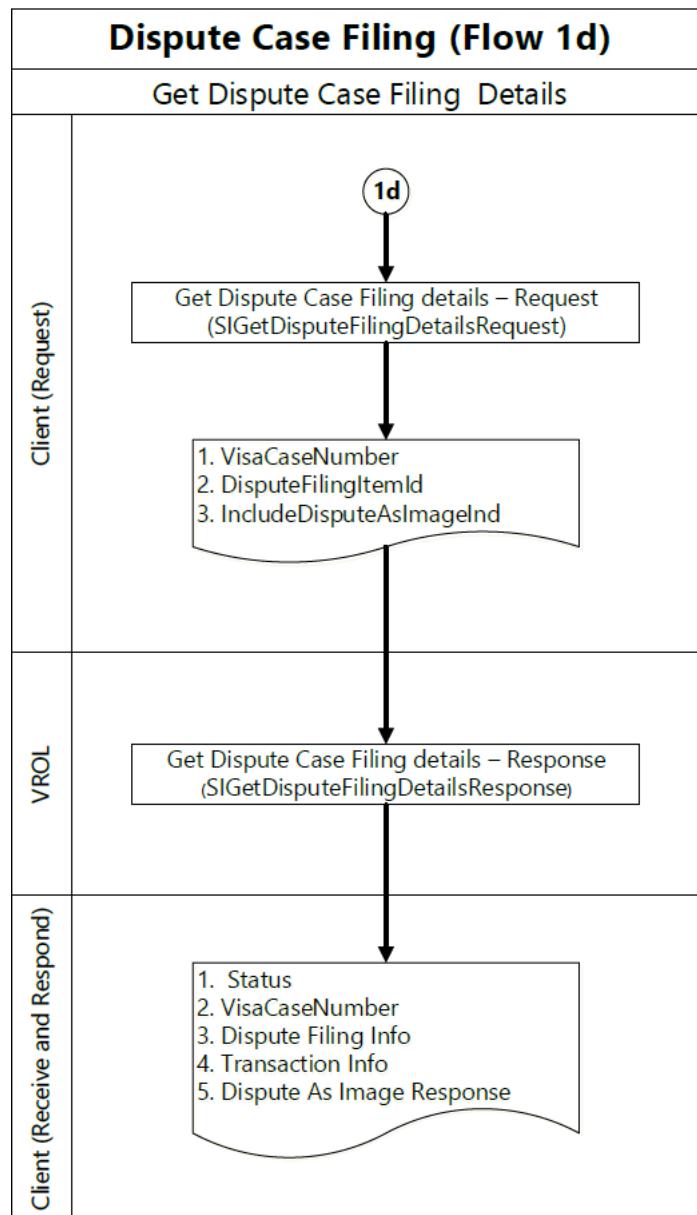
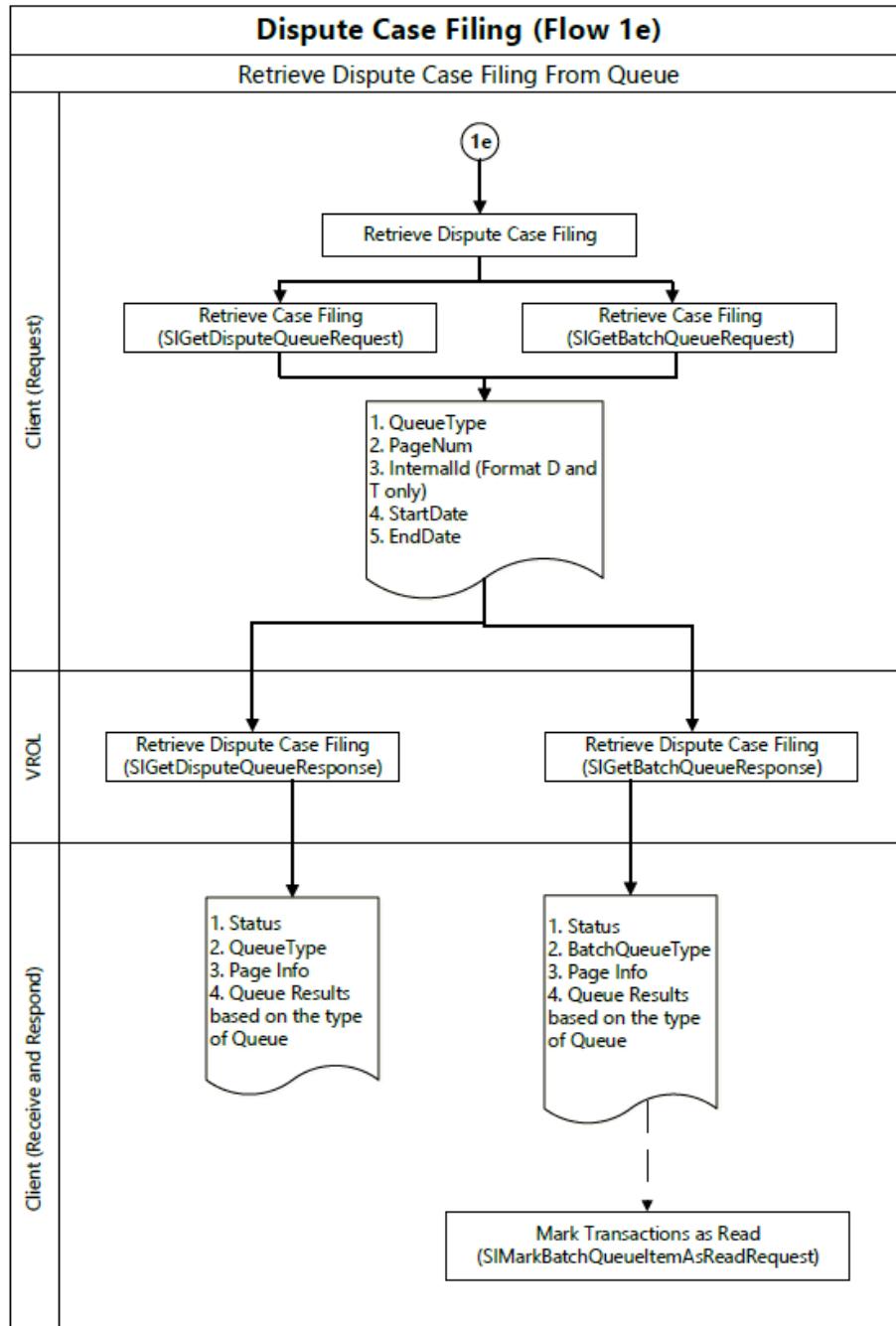
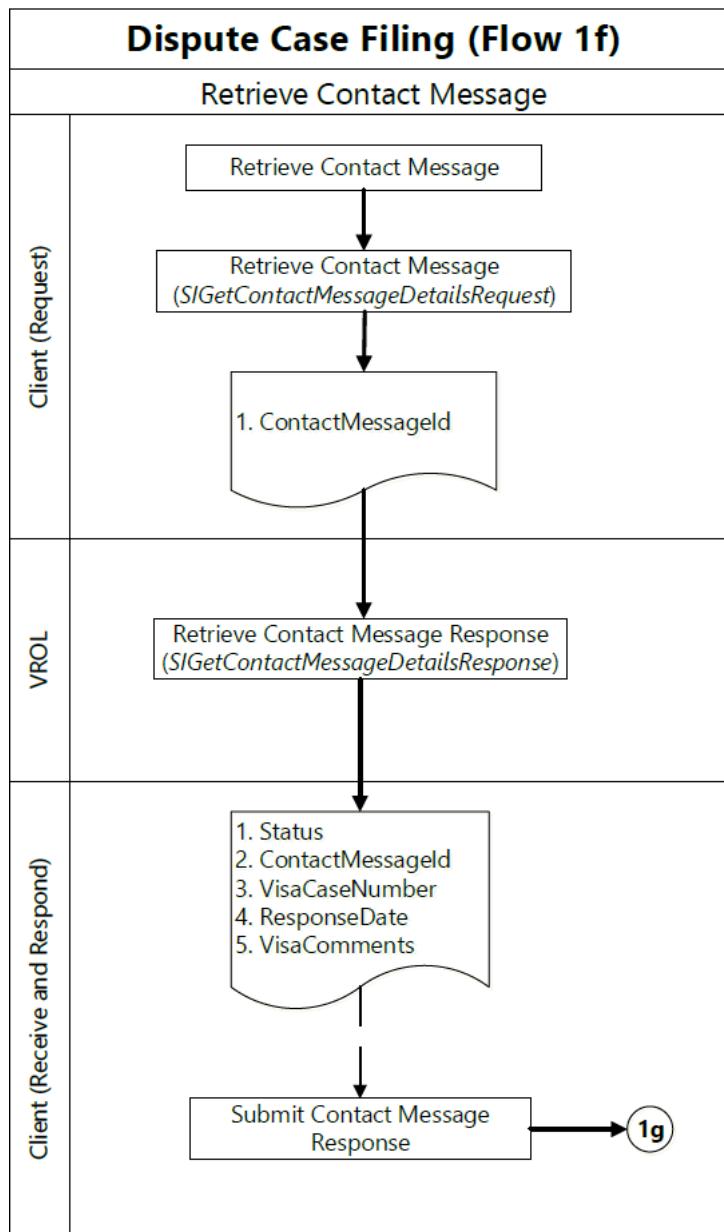
**Figure 11–3: Dispute Case Filing—Flow 1b/1c**

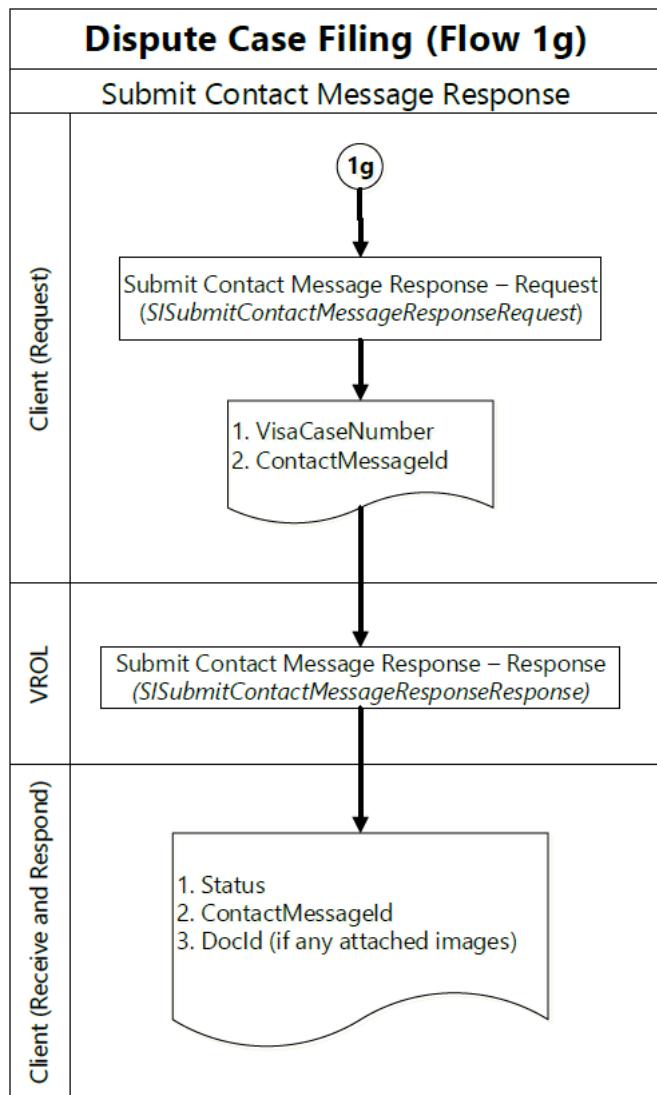
Figure 11–4: Dispute Case Filing—Flow 1d



**Figure 11–5: Dispute Case Filing—Flow 1e**

**Figure 11–6: Dispute Case Filing—Flow 1f**



**Figure 11–7: Dispute Case Filing—Flow 1g**

## Create Dispute Case Filing

1. To create a dispute case filing, the member host system triggers a `SISubmitDisputeFilingRequest` message supplying the `VisaCaseNumber` and dispute case filing information (either arbitration or compliance). This request can include image(s). User can Save or Submit the dispute case filing request by setting the `Action` value.

VROL returns a `SISubmitDisputeFilingResponse` message with the `VisaCaseNumber`, `DisputeFilingItemId`, `DisputeFilingItemType`, and the `DocIds` for any attached images.

**Note:** *DisputeFilingItemId* uniquely identifies a dispute filing transaction while a *DisputeFilingItemType* identifies the type of the dispute case filing. See [Table 19–1](#) in [Chapter 19, Hypersearch Requests](#) for more information.

2. To attach one or more images to the questionnaire, include the DisputeAttachmentDescriptor group element in the request message.  
Members can also use the Bulk SI Hybrid Image upload functionality to generate DocIds in bulk which can be supplied in the SISubmitDisputeFilingRequest against the DocId element. For details, see [Hybrid Image Uploads](#) and *Visa Resolve Online Bulk Systems Interface Development Guide*.
3. If the user wants to use the Bulk SI Hybrid Image upload functionality to supply the documents, including image details, to a saved dispute case filing questionnaire, set the <SupplyDocInd>=true.
  - If the <SupplyDocInd>=true, then the dispute case filing questionnaire will always get saved even if the user sets the Action value to Submit.
  - If the <SupplyDocInd>=false, then the dispute case filing questionnaire will get submitted when the user submits the dispute case filing questionnaire.  
If the user fails to attach the image to a saved questionnaire through the Bulk SI Hybrid Image upload functionality, the user must supply the missing elements through RTSI, save the questionnaire, and try to attach the image again using the same Bulk SI service.

## Retrieve Dispute Case Filing

1. To retrieve case filing from the corresponding case filing queues, the member host system triggers a SIGetQueueRequest message or a SIGetBatchQueueRequest message. See [Chapter 17, Queues](#) and [Chapter 18, RTSI Batch Queues](#).

VROL returns a list of the dispute case filings with "Acknowledgement", "Visa Contact Messages", "Response made available to other side", or "Final Decision" as Action value in the queue, including their VisaCaseNumber.

**Note:** *Action field categorizes the various states when the case is in a case filing stage such as "Acknowledgement", "Visa Contact Messages", "Response made available to other side", or "Final Decision".*

2. To retrieve the DisputeFilingItemId, the member host system triggers a Hypersearch request. See [Chapter 19, Hypersearch Requests](#), for more information.

VROL returns the DisputeFilingItemId along with its DisputeFilingItemType.

3. To retrieve dispute case filing details, the member host system triggers a SIGetDisputeFilingDetailsRequest message supplying the VisaCaseNumber and DisputeFilingItemId.

VROL returns a SIGetDisputeFilingDetailsResponse message with dispute case filing and image details along with DisputeFilingItemType.

**Note:** *The dispute case filing questionnaire may be returned as an image (TIFF/PDF), in addition to XML, as part of this response message (if optional <IncludeDisputeAsImageInd>=true is included in the request). However, any images attached to the questionnaire must be retrieved separately (step 4 below).*

**Note:** *Any image attached to the dispute case filing questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.*

4. To retrieve the attached images, the member host system triggers a SIGetImageRequest message supplying the VisaCaseNumber and DocId.

**Note:** DocIds are generated in bulk when the member uses the Bulk SI Hybrid Image upload functionality to attach image(s) in the case filing questionnaire. For details, see [Hybrid Image Uploads](#) and [Visa Resolve Online Bulk Systems Interface Development Guide](#).

VROL returns a SIGetImageResponse message with image data.

## Submit Dispute Case Filing Withdrawal Request

User can **Request to Withdraw** if he wants to withdraw the case filing and accept liability.

1. To retrieve case filing from the corresponding case filing queues, the member host system triggers a SIGetQueueRequest message or a SIGetBatchQueueRequest message. See [Chapter 17, Queues](#) and [Chapter 18, RTSI Batch Queues](#).

VROL returns a list of the dispute case filings in the queue, including their VisaCaseNumber.

2. To create a dispute case filing withdrawal request, the member host system triggers a SISubmitDisputeFilingRequest message supplying the VisaCaseNumber and RequestToWithdraw=TRUE. This request can include comment(s) or image(s).

VROL returns a SISubmitDisputeFilingResponse message with the success status and the DocIds for any attached images.

3. To attach one or more images to the questionnaire, include the DisputeAttachmentDescriptor group element in the request message.

Members can also use the Bulk SI Hybrid Image upload functionality to generate DocIds in bulk which can be supplied in the SISubmitDisputeFilingRequest against the DocId element. For details, see [Hybrid Image Uploads](#) and [Visa Resolve Online Bulk Systems Interface Development Guide](#).

## Retrieve Dispute Case Filing Withdrawal Request

1. To retrieve case filing from the corresponding case filing queues, the member host system triggers a SIGetQueueRequest message or a SIGetBatchQueueRequest message. See [Chapter 17, Queues](#) and [Chapter 18, RTSI Batch Queues](#).

VROL returns a list of the dispute case filings in the queue, including their VisaCaseNumber.

2. To retrieve dispute case filing withdrawal request details, the member host system triggers a SIGetDisputeFilingDetailsRequest message supplying the VisaCaseNumber and DisputeFilingItemId.

VROL returns a SIGetDisputeFilingDetailsResponse message with details of withdrawal request.

**Note:** The contact message questionnaire may be returned as an image (TIFF/PDF), in addition to XML, as part of this response (if optional <IncludeDisputeAsImageInd>=true is included in the request).

**Note:** Any image attached to the dispute case filing questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.

## Retrieve Contact Message, Create Contact Message Response

1. To retrieve case filing from the corresponding case filing queues, the member host system triggers a SIGetQueueRequest message or a SIGetBatchQueueRequest message. See [Chapter 17, Queues](#) and [Chapter 18, RTSI Batch Queues](#).

VROL returns a list of the dispute case filings with "Visa Contact Messages" as Action value in the queue, including their VisaCaseNumber.

**Note:** Action field categorizes the various states when the case is in a case filing stage such as "Visa Contact Messages".

2. To retrieve the ContactMessageId, the member host system triggers a Hypersearch request. See [Chapter 19, Hypersearch Requests](#), for more information.

VROL returns a list of ContactMessageIds, including their timestamp.

3. To retrieve contact message request details, the member host system triggers a SIGetContactMessageDetailsRequest message supplying the VisaCaseNumber and ContactMessageId.

VROL returns a SIGetContactMessageDetailsResponse message with contact message request and image details.

**Note:** Any image attached to the contact message request is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.

4. To retrieve the attached images with contact message request, the member host system triggers a SIGetImageRequest message supplying the VisaCaseNumber and DocId.

**Note:** DocIds are generated in bulk when the member uses the Bulk SI Hybrid Image upload functionality to attach image(s) in the case filing questionnaire. For details, see [Hybrid Image Uploads](#) and [Visa Resolve Online Bulk Systems Interface Development Guide](#).

VROL returns a SIGetImageResponse message with image data.

5. To create contact message response, the member host system triggers a SISubmitContactMessageResponseRequest message supplying the VisaCaseNumber, ContactMessageId, and contact message response information. User can submit the contact message response request.

**Note:** The ContactMessageId refers to the ID of the contact message from DRM to the member.

VROL returns a SISubmitContactMessageResponseResponse message with the success status, including the DocIds for any attached images.

6. To attach one or more images to the contact message data, include the DisputeAttachmentDescriptor group element in the request message.

Members can also use the Bulk SI Hybrid Image upload functionality to generate DocIds in bulk which can be supplied in the SISubmitContactMessageResponseRequest against the DocId element. For details, see [Hybrid Image Uploads](#) and [Visa Resolve Online Bulk Systems Interface Development Guide](#).

7. If the user wants to use the Bulk SI Hybrid Image upload functionality to supply the documents, including image details, to a saved contact message response, set the <SupplyDocInd>=true.

- If the <SupplyDocInd>=true, then the contact message response will always get saved even if the user sets the Action value to Submit.
  - If the <SupplyDocInd>=false, then the contact message response will get submitted when the user submits the contact message response.
- If the user fails to attach the image to a saved questionnaire through the Bulk SI Hybrid Image upload functionality, the user must supply the missing elements through RTSI, save the questionnaire, and try to attach the image again using the same Bulk SI service.
8. To retrieve the ContactMessageResponseId, the member host system triggers a Hypersearch request. See [Chapter 19, Hypersearch Requests](#), for more information.
  9. To retrieve contact message response details, the member host system triggers a SIGetContactMessageResponseDetailsRequest message supplying the ContactMessageResponseId.
- VROL returns a SIGetContactMessageResponseDetailsResponse message with details of contact message response.

**Note:** *Any image attached to the contact message response is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.*

## Create Dispute Case Filing Appeal, Retrieve Dispute Case Filing Appeal Request

Prior to sending a dispute case filing appeal, the user must meet the following conditions to perform an appeal on a filed case:

- Members cannot appeal the final decision made on an appeal.
- The member completely or partially lost the case.
  - The total dispute amount for the transaction is greater than US\$5,000.
  - The member can provide new evidence that is not previously available.
  - An appeal has not been performed on the case.
1. To create a dispute case filing appeal, the member host system triggers a SISubmitDisputeFilingRequest message supplying the VisaCaseNumber and AppealInd=True.
- VROL returns a SISubmitDisputeFilingResponse message with the success status.
2. To retrieve case filing from the corresponding case filing queues, the member host system triggers a SIGetQueueRequest message or a SIGetBatchQueueRequest message. See [Chapter 17, Queues](#) and [Chapter 18, RTSI Batch Queues](#).
- VROL returns a list of the dispute case filings in the queue, including their VisaCaseNumber.
3. To retrieve the DisputeFilingItemId, the member host system triggers a Hypersearch request. See [Chapter 19, Hypersearch Requests](#), for more information.
  4. To retrieve dispute case filing appeal details, the member host system triggers a SIGetDisputeFilingDetailsRequest supplying the VisaCaseNumber and DisputeFilingItemId.
- VROL sends response with details of dispute case filing appeal, including AppealInd=True.

**Note:** *The dispute case filing questionnaire may be returned as an image (TIFF/PDF), in addition to XML, as part of this response (if optional <IncludeDisputeAsImageInd>=true is included in the request). However, any images attached to the questionnaire must be retrieved separately (step 5 below).*

**Note:** *Any image attached to the dispute case filing questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.*

5. To retrieve the attached images with dispute case filing appeal, the member host system triggers a SIGetImageRequest message supplying the VisaCaseNumber and DocID.

VROL returns a SIGetImageResponse message with image data.

## Retrieve Acknowledgment/Rejection/Final Decision Letters

To retrieve the letters, the member host system triggers a hypersearch request supplying the VisaCaseNumber and retrieves the DisputeFilingItemId, which is used in SIGetDisputeFilingDetailsRequest. See [Chapter 19, Hypersearch Requests](#), for more information about hypersearch.

**Note:** *Any image attached to the dispute case filing questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.*

VROL sends response with letter details.

## Delete Case Filing

The initiator can delete a saved arbitration or compliance.

1. To delete an arbitration or compliance, the member host system triggers a SIChangeDisputeStatusRequest message supplying the VisaCaseNumber, ChangeItemID, ChangeItemType, and Action value=Delete.

**Note:** *ChangeItemID includes the unique ID value for the Case Filing item (that is, DisputeFilingItemId for arbitration or compliance).*

VROL returns a SIChangeDisputeStatusResponse message with the success status.

## Check for Case Filing Rejects

If the dispute-related arbitration or compliance case filing questionnaire fails one or more validation edits, VROL blocks the questionnaire and places it in the MY\_REJECTS or ALL\_REJECTS queue.

1. To check for rejected case filing questionnaires, the member host system triggers a SIGetQueueRequest message for MY\_REJECTS or ALL\_REJECTS queue. See [Chapter 17, Queues](#), for more information.

VROL returns a SIGetQueueResponse message with the VisaCaseNumber and RejectID.

2. To ignore a rejected case filing questionnaire when using the reject queue, the member host system triggers a SIIgnoreRejectRequest message supplying the VisaCaseNumber, RejectID, and RejectedTransactionType="Arbitration" or "Compliance". This step prevents the transaction from being returned again in the next queue request. See [Chapter 17, Queues](#), for more information.

VROL returns a SISignoreRejectResponse message with the success status.

3. To repair a rejected case filing questionnaire, the member host system triggers a SISubmitDisputeFilingRequest message along with the updated case filing questionnaire information.
  - If the case contains a single unrepaired reject of the same type as the request, VROL marks the reject as repaired.
  - If the case contains multiple unrepaired rejects of the same type as the request, VROL marks the most recent reject as repaired.
  - If the case does not contain a reject of the same type as the request, VROL does not mark any transaction as repaired. VROL adds the incoming transaction to the case and sends response with success status.

VROL returns a SISubmitDisputeFilingResponse message with the success status.

## Dispute Case Filing Operations

RTSI exposes the following dispute case filing operations:

**SISubmitDisputeFilingRequest**—Used to submit a dispute case filing request, withdrawal, or appeal.

**SIGetDisputeFilingDetailsRequest**—Used to retrieve the details submitted in a case filing request.

**SIGetContactMessageDetailsRequest**—Used to retrieve the details of a contact message request.

**SISubmitContactMessageResponseRequest**—Used to submit a response to a contact message request.

**SIGetContactMessageResponseDetailsRequest**—Used to retrieve the details of a contact message response.

**SIChangeDisputeStatusRequest**—Used to delete a saved case filing (arbitration or compliance).

### SISubmitDisputeFilingRequest

A dispute case filing (arbitration or compliance) must be added to an existing case. To add to an existing case, SISubmitDisputeFilingRequest requires the VisaCaseNumber and DisputeFilingItemId.

**Note:** *DisputeFilingItemId is required when updating a previously-saved dispute case filing. The field should not be supplied when creating a new dispute case filing.*

This request requires the transaction details for the dispute questionnaire request (see the Dispute tab of the IES) and is used in request and response messages as shown in the following sample message.

The following elements should be considered when submitting a dispute case filing, withdrawal, or appeal request, or when receiving a response from this service.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.
DisputeFilingItemId	A unique VROL-generated ID for each saved or submitted dispute case filing request.
DisputeFilingItemType	Indicates in the response whether the dispute case filing item type is any of the following domain values: <ul style="list-style-type: none"> <li>• Arbitration</li> <li>• Compliance</li> <li>• FilingResponse</li> <li>• AcknowledgementLetter</li> <li>• RejectionLetter</li> <li>• DecisionLetter</li> </ul>
RequestToWithdrawInd	Indicates in the response whether or not the request is a withdrawal of a case filing.  This element is only applicable for dispute case filing withdrawal.
AppealInd	Indicates in the response whether or not the request is an appeal of a prior case filing.  This element is only applicable for dispute case filing appeal.

SISubmitDisputeFiling contains request and response messages as shown in the following sample messages.

**Important:**

Once transaction information is established in a case, there is no need to provide it in subsequent actions. In addition, <TransactionInfo> is no longer required if a case is created from a TI result.

### Request Message—Arbitration—SOAP-based Services

```
<SISubmitDisputeFilingRequest>
  <RequestHeader>
    <MemberRole>I</MemberRole>
  </RequestHeader>
  <requestData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <MemberCaseNumber>555555555555</MemberCaseNumber>
    <Arbitration>
      <CaseFilingAmount currency="334">50.00</CaseFilingAmount>
      <Comments>any comment</Comments>
      <FilingReason>filing reason</FilingReason>
    </Arbitration>
    <Action>Submit</Action>
  </requestData>
</SISubmitDisputeFilingRequest>
```

### Request Message—Compliance—SOAP-based Services

```
<SISubmitDisputeFilingRequest>
  <RequestHeader>
    <MemberRole>I</MemberRole>
```

```
</RequestHeader>
<requestData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <DisputeFilingItemId>0000001</DisputeFilingItemId>
    <MemberCaseNumber>555555555555</MemberCaseNumber>
    <Compliance>
        <FirstRuleViolatedCode>a</FirstRuleViolatedCode>
        <SecondRuleViolatedCode>a</SecondRuleViolatedCode>
        <ThirdRuleViolatedCode>a</ThirdRuleViolatedCode>
        <RulesViolatedAdditionalInfo>a
        </RulesViolatedAdditionalInfo>
        <ViolationDate>2016-08-13</ViolationDate>
        <ViolationDiscoveryDate>2016-08-13
        </ViolationDiscoveryDate>
        <FinancialLossExplanation>any explanation
        </FinancialLossExplanation>
        <FilingReason>any reason</FilingReason>
        <CaseFilingAmount currency="000">0</CaseFilingAmount>
        <Comments>any comment</Comments>
        <IssuerAcquirerContactInfo>
            <ContactName>a</ContactName>
            <ContactPhone>123456789</ContactPhone>
            <ContactFax>1234567891</ContactFax>
            <ContactOther>a</ContactOther>
            <ContactEmail>aa@hotmail.com</ContactEmail>
        </IssuerAcquirerContactInfo>
        <AppealInd>true</AppealInd> <- For dispute case filing appeal only
    </Compliance>
    <FilingResp>
        <IssuerAcquirerContactInfo>
            <ContactName>a</ContactName>
            <ContactPhone>123456789</ContactPhone>
            <ContactFax>1234567891</ContactFax>
            <ContactOther>a</ContactOther>
            <ContactEmail>aa@hotmail.com</ContactEmail>
        </IssuerAcquirerContactInfo>
        <RequestToWithdrawInd>true</RequestToWithdrawInd> <- For dispute case filing withdrawal only
    </FilingResp>
    <DisputeAttachmentDescriptor>
        <Attachment id="String">
            <ContentType>application/pdf</ContentType>
            <Comment>comment</Comment>
            <ImageData>xxxxxx</ImageData>
            <DocType>VSEXT</DocType>
        </Attachment>
    </DisputeAttachmentDescriptor>
    <DocIdList>
        <DocId>0</DocId>
    </DocIdList>
    <Action>Save</Action>
    <SupplyDocInd>true</SupplyDocInd>
</requestData>
</SISubmitDisputeFilingRequest>
```

## Response Message—SOAP-based Services

```
<SISubmitDisputeFilingResponse>
  <Status>
    <Code>W-121005162</Code>
    <Message>Transaction amount is equal or less than low
    dollar write off limit. The Case Filing
    was processed successfully.</Message>
  </Status>
  <responseData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <DisputeFilingInfo>
      <DisputeFilingItemId>000001</DisputeFilingItemId>
      <DisputeFilingItemType>Arbitration
      </DisputeFilingItemType>
    </DisputeFilingInfo>
  </responseData>
</SISubmitDisputeFilingResponse>
```

## Request Message—Arbitration—REST-based Services

```
{
  "RequestHeader": {
    "MemberRole": "I"
  },
  "requestData": {
    "VisaCaseNumber": 1000000000,
    "MemberCaseNumber": "test",
    "Arbitration": {
      "RequestingMemberBIN": 442828,
      "RequestorName": "test56",
      "OpposerMemberBIN": 471705,
      "OpposerName": "test567",
      "CaseFilingAmount": {
        "@currency": 334,
        "$": 50.00
      },
      "Comments": "test_comment",
      "FilingReason": "test56"
    },
    "Action": "Submit"
  }
}
```

## Request Message—Compliance—REST-based Services

```
{
  "RequestHeader": {
    "MemberRole": "I"
  },
  "requestData": {
    "VisaCaseNumber": "1000000000",
    "DisputeFilingItemId": "1234567",
    "MemberCaseNumber": "test",
    "Compliance": {
      "FinancialLossExplanation": "Test",
      "FilingReason": "Test",
      "CaseFilingAmount": {
        "Currency": "840",
        "Value": "10"
      },
      "Comments": "Test",
      "IssuerAcquirerContactInfo": {
        "ContactName": "a",
        "ContactPhone": "123456789",
        "ContactFax": "123456789",
        "ContactEmail": "abc@xyz.com"
      }
    },
    "Action": "Submit"
  }
}
```

## Response Message—REST-based Services

```
{
  "Status": [
    {
      "Code": "W-126000043",
      "Message": "Please ensure that you are within case filing time frame."
    }
  ],
  "responseData": {
    "VisaCaseNumber": 1000000000,
    "DisputeFilingInfo": {
      "DisputeFilingItemId": 000001,
      "DisputeFilingItemType": "Arbitration"
    }
  }
}
```

## SIGetDisputeFilingDetailsRequest

SIGetDisputeFilingDetailsRequest requires the VisaCaseNumber and DisputeFilingItemId to retrieve the details submitted in a dispute case filing request, appeal, or withdrawal request.

SIGetDisputeFilingDetailsRequest allows RTSI users to request for the dispute case filing questionnaire data to be returned in image format, as well as in XML. The returned image will be in TIFF or PDF format. To retrieve the dispute as an image, the user

must set <IncludeDisputeAsImageInd>=true. The image data will be returned in the <DisputeAttachmentDescriptor> in the SIGetDisputeFilingDetailsResponse.

**Note:** Only the dispute case filing questionnaire will be returned as an image. Any attached documents must be retrieved using the AttachmentId value(s) in this response as input to the DocId field(s) of SIGetImageRequest.

**Note:** Any image attached to the dispute case filing questionnaire is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.

**Note:** In the response message, <CaseHistorySummary> is not applicable for Letter.

The SIGetDisputeFilingDetails contains request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SIGetDisputeFilingDetailsRequest>
  <RequestHeader>
    <MemberRole>I</MemberRole>
  </RequestHeader>
  <requestData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <DisputeFilingItemId>000001</DisputeFilingItemId>
    <IncludeDisputeAsImageInd>true</IncludeDisputeAsImageInd>
    <DownloadImageInd>true</DownloadImageInd>
  </requestData>
</SIGetDisputeFilingDetailsRequest>
```

### Response Message—Arbitration—SOAP-based Services

```
<SIGetDisputeFilingDetailsResponse>
  <Status>
    <Code>I-920000001</Code>
    <Message>No associated transactions found.</Message>
  </Status>
  <responseData>
    <TransactionInfo>
      <AccountNumber>410000000000</AccountNumber>
      <Transaction id="0000000000000000">
        <TranDate>2017-05-24</TranDate>
        <Amount currency="344">8.24</Amount>
      </Transaction>
      <MerchantName>MERCHANT NAME</MerchantName>
      <MerchantCity>MERCHANT CITY</MerchantCity>
      <MerchantCountrySubEntityCode>CA
      </MerchantCountrySubEntityCode>
      <MerchantCountryCode>HK</MerchantCountryCode>
      <MerchantPostalCode>94408</MerchantPostalCode>
      <MCC>5000</MCC>
      <ARN>00000000000000000001</ARN>
      <ReimbursementAttribute>A</ReimbursementAttribute>
      <CPD>2017-05-24</CPD>
      <TransactionCategory>D</TransactionCategory>
      <AcquirerBID>10000006</AcquirerBID>
      <NetworkID>0002</NetworkID>
      <AcquirerBIN>400000</AcquirerBIN>
      <RetrievalReferenceNumber>700000000000
    </TransactionInfo>
  </responseData>
</SIGetDisputeFilingDetailsResponse>
```

```
</RetrievalReferenceNumber>
<SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
<TranDestAmt currency="344">8.24</TranDestAmt>
<ProcessingCode>00</ProcessingCode>
<IssuerName>issuer name</IssuerName>
<AcquirerName>acquirer name</AcquirerName>
<AccountType>00</AccountType>
<DisputeJurisdiction jurisdictionCd="DOM">
    <Jurisdiction>DOMESTIC-HK</Jurisdiction>
    <IssuerRegion>AP</IssuerRegion>
    <AcquirerRegion>AP</AcquirerRegion>
    <CountryCd>HK</CountryCd>
</DisputeJurisdiction>
</TransactionInfo>
<VisaCaseNumber>1000000000</VisaCaseNumber>
<DisputeFilingInfo>
    <DisputeFilingItemId>0000000001</DisputeFilingItemId>
    <DisputeFilingItemType>Compliance</DisputeFilingItemType>
</DisputeFilingInfo>
<MemberCaseNumber>555555555555</MemberCaseNumber>
<Arbitration>
    <CaseHistorySummary>
        <CaseHistory>
            <Event>name</Event>
            <CPD>2016-01-01</CPD>
            <DisputeAmount currency="840">100
            </DisputeAmount>
            <DisputeCategoryCondition>13
            </DisputeCategoryCondition>
        </CaseHistory>
    </CaseHistorySummary>
<DaysBetweenEvents>
    <Event>name</Event>
    <Days>01</Days>
</DaysBetweenEvents>
<RequestingMemberBIN>requestor BIN</RequestingMemberBIN>
<RequestorName>requestor name</RequestorName>
<OpposerMemberBIN>opposer BIN</OpposerMemberBIN>
<OpposerName>opposer name</OpposerName>
<CaseFilingAmount currency="000">0
</CaseFilingAmount>
<Comments>any comment</Comments>
<IssuerAcquirerContactInfo>
    <ContactName>a</ContactName>
    <ContactPhone>123456789</ContactPhone>
    <ContactFax>1234567891</ContactFax>
    <ContactOther>a</ContactOther>
    <ContactEmail>aa@hotmail.com</ContactEmail>
</IssuerAcquirerContactInfo>
<AppealInd>true</AppealInd> <- For dispute case filing appeal only
<FilingReason>reason</FilingReason>
<DisputeAttachmentDescriptor>
    <Attachment id="String">
        <ContentType>application/pdf</ContentType>
        <Comment>any comment</Comment>
        <ImageData>xxxxxx</ImageData>
```

```
<DocType>TEFOL</DocType>
</Attachment>
</DisputeAttachmentDescriptor>
</Arbitration>
<FilingResp>
    <CaseHistorySummary>
        <CaseHistory>
            <Event>name</Event>
            <CPD>2016-01-01</CPD>
            <DisputeAmount currency="840">100
            </DisputeAmount>
            <DisputeCategoryCondition>12
            </DisputeCategoryCondition>
        </CaseHistory>
    </CaseHistorySummary>
    <DaysBetweenEvents>
        <Event>name</Event>
        <Days>01</Days>
    </DaysBetweenEvents>
    <RequestingMemberBIN>requestor BIN</RequestingMemberBIN>
    <RequestorName>requestor name</RequestorName>
    <OpposerMemberBIN>opposer BIN</OpposerMemberBIN>
    <OpposerName>opposer name</OpposerName>
    <IssuerAcquirerContactInfo>
        <ContactName>a</ContactName>
        <ContactPhone>123456789</ContactPhone>
        <ContactFax>1234567891</ContactFax>
        <ContactOther>a</ContactOther>
        <ContactEmail>aa@hotmail.com</ContactEmail>
    </IssuerAcquirerContactInfo>
    <RequestToWithdrawInd>true</RequestToWithdrawInd> <-
For dispute case filing withdrawal only
    <DisputeAttachmentDescriptor>
        <Attachment id="String">
            <ContentType>image/tiff</ContentType>
            <Comment>any comment</Comment>
            <ImageData>xxxxxx</ImageData>
            <DocType>VSEXT</DocType>
        </Attachment>
    </DisputeAttachmentDescriptor>
</FilingResp>
<Letter type="Final Decision">
    <LetterId>1</LetterId>
    <Text>text</Text>
    <Jurisdiction>LAC</Jurisdiction>
    <AppealInd>true</AppealInd> <-
For dispute case filing appeal only
</Letter>
<DisputeAsImageResponseDescriptor>
    <ContentType>application/pdf</ContentType>
    <ImageData>xxxxxx</ImageData>
</DisputeAsImageResponseDescriptor>
</ResponseData>
</SIGetDisputeFilingDetailsResponse>
```

**Response Message—Compliance—SOAP-based Services**

```
<SIGetDisputeFilingDetailsResponse>
    <Status>
        <Code>I-920000001</Code>
        <Message>No associated transactions found.</Message>
    </Status>
    <responseData>
        <TransactionInfo>
            <AccountNumber>410000000000</AccountNumber>
            <Transaction id="0000000000000000">
                <TranDate>2017-05-24</TranDate>
                <Amount currency="344">8.24</Amount>
            </Transaction>
            <MerchantName>MERCHANT NAME</MerchantName>
            <MerchantCity>MERCHANT CITY</MerchantCity>
            <MerchantCountrySubEntityCode>CA
            </MerchantCountrySubEntityCode>
            <MerchantCountryCode>HK</MerchantCountryCode>
            <MerchantPostalCode>94408</MerchantPostalCode>
            <MCC>5000</MCC>
            <ARN>000000000000000000000001</ARN>
            <ReimbursementAttribute>A</ReimbursementAttribute>
            <CPD>2017-05-24</CPD>
            <TransactionCategory>D</TransactionCategory>
            <AcquirerBID>10000006</AcquirerBID>
            <NetworkID>0002</NetworkID>
            <AcquirerBIN>400000</AcquirerBIN>
            <RetrievalReferenceNumber>700000000000
            </RetrievalReferenceNumber>
            <SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
            <TranDestAmt currency="344">8.24</TranDestAmt>
            <ProcessingCode>00</ProcessingCode>
            <IssuerName>issuer name</IssuerName>
            <AcquirerName>acquirer name</AcquirerName>
            <AccountType>00</AccountType>
            <DisputeJurisdiction jurisdictionCd="DOM">
                <Jurisdiction>DOMESTIC-HK</Jurisdiction>
                <IssuerRegion>AP</IssuerRegion>
                <AcquirerRegion>AP</AcquirerRegion>
                <CountryCd>HK</CountryCd>
            </DisputeJurisdiction>
        </TransactionInfo>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <DisputeFilingInfo>
            <DisputeFilingItemId>0000000001</DisputeFilingItemId>
            <DisputeFilingItemType>Compliance</DisputeFilingItemType>
        </DisputeFilingInfo>
        <MemberCaseNumber>555555555555</MemberCaseNumber>
        <Compliance>
            <CaseHistorySummary>
                <CaseHistory>
                    <Event>name</Event>
                    <CPD>2016-01-01</CPD>
                    <DisputeAmount currency="840">100
                    </DisputeAmount>
                    <DisputeCategoryCondition>11
```

```
        </DisputeCategoryCondition>
    </CaseHistory>
</CaseHistorySummary>
<DaysBetweenEvents>
    <Event>name</Event>
    <Days>01</Days>
</DaysBetweenEvents>
<RequestingMemberBIN>requestor BIN</RequestingMemberBIN>
<RequestorName>requestor name</RequestorName>
<OpposerMemberBIN>opposer BIN</OpposerMemberBIN>
<OpposerName>opposer name</OpposerName>
<FirstRuleViolatedCode>a</FirstRuleViolatedCode>
<SecondRuleViolatedCode>a</SecondRuleViolatedCode>
<ThirdRuleViolatedCode>a</ThirdRuleViolatedCode>
<RulesViolatedAdditionalInfo>a
</RulesViolatedAdditionalInfo>
<ViolationDate>2016-01-01</ViolationDate>
<ViolationDiscoveryDate>2016-01-01
</ViolationDiscoveryDate>
<FinancialLossExplanation>explanation
</FinancialLossExplanation>
<FilingReason>reason</FilingReason>
<CaseFilingAmount currency="000">0
</CaseFilingAmount>
<Comments>any comment</Comments>
<IssuerAcquirerContactInfo>
    <ContactName>a</ContactName>
    <ContactPhone>123456789</ContactPhone>
    <ContactFax>1234567891</ContactFax>
    <ContactOther>a</ContactOther>
    <ContactEmail>aa@hotmail.com</ContactEmail>
</IssuerAcquirerContactInfo>
<AppealInd>true</AppealInd> <- For dispute case filing appeal only
<DisputeAttachmentDescriptor>
    <Attachment id="String">
        <ContentType>image/tiff</ContentType>
        <Comment>any comment</Comment>
        <ImageData>xxxxxx</ImageData>
        <DocType>VSEXT</DocType>
    </Attachment>
</DisputeAttachmentDescriptor>
</Compliance>
<FilingResp>
    <CaseHistorySummary>
        <CaseHistory>
            <Event>name</Event>
            <CPD>2016-01-01</CPD>
            <DisputeAmount currency="840">100
            </DisputeAmount>
            <DisputeCategoryCondition>12
            </DisputeCategoryCondition>
        </CaseHistory>
    </CaseHistorySummary>
    <DaysBetweenEvents>
        <Event>name</Event>
        <Days>01</Days>
```

```

        </DaysBetweenEvents>
        <RequestingMemberBIN>requestor BIN</RequestingMemberBIN>
        <RequestorName>requestor name</RequestorName>
        <OpposerMemberBIN>opposer BIN</OpposerMemberBIN>
        <OpposerName>opposer name</OpposerName>
        <IssuerAcquirerContactInfo>
            <ContactName>a</ContactName>
            <ContactPhone>123456789</ContactPhone>
            <ContactFax>1234567891</ContactFax>
            <ContactOther>a</ContactOther>
            <ContactEmail>aa@hotmail.com</ContactEmail>
        </IssuerAcquirerContactInfo>
        <RequestToWithdrawInd>true</RequestToWithdrawInd> <-
For dispute case filing withdrawal only
        <DisputeAttachmentDescriptor>
            <Attachment id="String">
                <ContentType>image/tiff</ContentType>
                <Comment>any comment</Comment>
                <ImageData>xxxxxx</ImageData>
                <DocType>VSEXT</DocType>
            </Attachment>
        </DisputeAttachmentDescriptor>
    </FilingResp>
    <Letter type="Final Decision">
        <LetterId>1</LetterId>
        <Text>text</Text>
        <Jurisdiction>LAC</Jurisdiction>
        <AppealInd>true</AppealInd> <-
For dispute case filing appeal only
    </Letter>
    <DisputeAsImageResponseDescriptor>
        <ContentType>application/pdf</ContentType>
        <ImageData>xxxxxx</ImageData>
    </DisputeAsImageResponseDescriptor>
</ResponseData>
</SIGetDisputeFilingDetailsResponse>

```

### Request Message—REST-based Services

```
{
    "RequestHeader": {
        "MemberRole": "A"
    },
    "requestData": {
        "DisputeFilingItemId": 1652562,
        "IncludeDisputeAsImageInd": true,
        "VisaCaseNumber": 1002318018
    }
}
```

### Response Message—Arbitration—REST-based Services

```
{
    "Status": [
        {
            "Code": "I-300000000",
            "Message": "Successfully completed Operation."
        }
    ]
}
```

```
        },
    "ResponseData": {
        "TransactionInfo": {
            "AccountNumber": "410000000000",
            "Transaction": {
                "TranDate": "2017-05-06",
                "Amount": {
                    "value": 56,
                    "currency": "840"
                },
                "MerchantName": "MERCHANT NAME",
                "MerchantCity": "MERCHANT CITY",
                "MerchantCountrySubEntityCode": "NJ",
                "MerchantCountryCode": "US",
                "MerchantPostalCode": "07652",
                "MCC": "5000",
                "ARN": "00000000000000000000000000000001",
                "ECMOTO": "7",
                "ReimbursementAttribute": "A",
                "CPD": "2017-05-07",
                "TransactionCategory": "D",
                "AcquirerBID": "10000006",
                "NetworkID": "0002",
                "AcquirerBIN": "400000",
                "RetrievalReferenceNumber": "700000000000",
                "SystemTraceAuditNumber": "080000",
                "CardholderName": "Name",
                "TranDestAmt": {
                    "value": 56,
                    "currency": "840"
                },
                "ProcessingCode": "00",
                "IssuerName": "AAA BANK",
                "AcquirerName": "BBB BANK",
                "AccountType": "0",
                "DisputeJurisdiction": {
                    "Jurisdiction": "DOMESTIC-US",
                    "IssuerRegion": "US",
                    "AcquirerRegion": "US",
                    "CountryCd": "US",
                    "jurisdictionCd": "DOM"
                }
            },
            "VisaCaseNumber": 1000000000,
            "DisputeFilingInfo": {
                "DisputeFilingItemId": 0000000001,
                "DisputeFilingItemType": "Arbitration"
            },
            "MemberCaseNumber": "555555555555",
            "Arbitration": {
                "CaseHistorySummary": {
                    "CaseHistory": [
                        {
                            "Event": "Dispute",
                            "CPD": "2017-06-29",
                            "DisputeAmount": {
                                "value": 56,
                                "currency": "840"
                            }
                        }
                    ]
                }
            }
        }
    }
}
```

```
        },
        "DisputeCategoryCondition":
        "10.3 Fraud - Card Present Environment"
    },
    ],
    "DaysBetweenEvents": [
        {
            "Event": "Dispute from the Transaction",
            "Days": 53
        }
    ],
    "RequestingMemberBIN": "401308",
    "RequestorName": "ABC BANK",
    "OpposerMemberBIN": "400002",
    "OpposerName": "CDE BANK",
    "CaseFilingAmount": {
        "value": 56,
        "currency": "840"
    },
    "Comments": "Dspt TEST",
    "IssuerAcquirerContactInfo": {
        "ContactName": "a",
        "ContactPhone": "12345678",
        "ContactOther": "cntct",
        "ContactEmail": "abc@visa.com"
    },
    "AppealInd": false,
    "FilingReason": "tsttTEST"
}
}
```

## Response Message—Compliance—REST-based Services

```
"ReimbursementAttribute": "A",
"CPD": "2017-05-07",
"TransactionCategory": "D",
"AcquirerBID": "00000000",
"NetworkID": "0002",
"AcquirerBIN": "400000",
"RetrievalReferenceNumber": "700000000000",
"SystemTraceAuditNumber": "80000",
"CardholderName": "Name",
"TranDestAmt": {
    "value": 123.89,
    "currency": "554"
},
"ProcessingCode": "00",
"IssuerName": "AAA BANK",
"AcquirerName": "BBB BANK",
"AccountType": "0",
"DisputeJurisdiction": {
    "Jurisdiction": "DOMESTIC-NZ",
    "IssuerRegion": "AP",
    "AcquirerRegion": "AP",
    "CountryCd": "NZ",
    "jurisdictionCd": "DOM"
}
},
"VisaCaseNumber": 1000000000,
"DisputeFilingInfo": {
    "DisputeFilingItemId": 0000000001,
    "DisputeFilingItemType": "Compliance"
},
"MemberCaseNumber": "555555555555",
"Compliance": {
    "CaseHistorySummary": {"CaseHistory": [
        {
            "Event": "Pre-Compliance",
            "CPD": "2017-10-30",
            "DisputeAmount": {
                "value": 123.89,
                "currency": "554"
            },
            "DisputeCategoryCondition": "Delayed or Amended Charges"
        }
    ]},
    "DaysBetweenEvents": [
        {
            "Event": "Pre-compliance from the Transaction",
            "Days": 164
        }
    ],
    "RequestingMemberBIN": "499916",
    "RequestorName": "ABC BANK",
    "OpposerMemberBIN": "400000",
    "OpposerName": "CDE BANK",
    "FirstRuleViolatedCode": "C030",
    "ViolationDate": "2017-10-30",
    "FinancialLossExplanation": "Test",
    "FilingReason": "Test",
```

```
        "CaseFilingAmount": {
            "value": 10,
            "currency": "840"
        },
        "Comments": "Test",
        "IssuerAcquirerContactInfo": {
            "ContactName": "a",
            "ContactPhone": "9593545773",
            "ContactFax": "123456789",
            "ContactEmail": "abc@xyz.com"
        },
        "AppealInd": false
    },
    "DisputeAsImageResponseDescriptor": {
        "ContentType": "image/tiff",
        "ImageData": "xxxxxx"
    }
}
```

## SIGetContactMessageDetailsRequest

**SIGetContactMessageDetailsRequest** is used to retrieve the details of a contact message.

`SIGetContactMessageDetailsRequest` allows RTSI users to request for the contact message questionnaire data to be returned in image format, as well as in XML. The returned image will be in TIFF or PDF format. To retrieve the dispute as an image, the user must set `<IncludeDisputeAsImageInd>=true`. The image data will be returned in the `<DisputeAttachmentDescriptor>` in the `SIGetContactMessageDetailsResponse`.

**Note:** Only the contact message questionnaire will be returned as an image. Any attached documents must be retrieved using the AttachmentId value(s) in this response as input to the DocId field(s) of SISGetImageRequest.

**Note:** Any image attached to the contact message request is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.

**Note:** In the response message, <CaseHistorySummary> is not applicable for Letter.

In order to retrieve an incoming contact message request details, `SIGetContactMessageDetailsRequest` requires the `VisaCaseNumber` and `ContactMessageId`.

`SIGetContactMessageDetails` contains request and response messages as shown in the following sample messages.

## Request Message—SOAP-based Services

```
<SIGetContactMessageDetailsRequest>
    <RequestHeader>
        <User id="abcde1234f" type="internalId"/>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1234567890</VisaCaseNumber>
        <ContactMessageId>000001</ContactMessageId>
        <IncludeDisputeAsImageInd>false</IncludeDisputeAsImageInd>
        <DownloadImageInd>false</DownloadImageInd>
    </requestData>
</SIGetContactMessageDetailsRequest>
```

## **Response Message—SOAP-based Services**

```
<SIGetContactMessageDetailsResponse>
    <Status>
        <Code>I-300000000</Code>
        <Message>Successfully completed Operation.</Message>
    </Status>
    <responseData>
        <TransactionInfo>
            <AccountNumber>410000000000</AccountNumber>
            <Transaction id="0000000000000000">
                <TranDate>2017-05-24</TranDate>
                <Amount currency="344">8.24</Amount>
            </Transaction>
            <MerchantName>MERCHANT NAME</MerchantName>
            <MerchantCity>MERCHANT CITY</MerchantCity>
            <MerchantCountrySubEntityCode>CA
            </MerchantCountrySubEntityCode>
            <MerchantCountryCode>US</MerchantCountryCode>
            <MerchantPostalCode>94403</MerchantPostalCode>
            <MCC>5411</MCC>
            <ARN>00000000000000000001</ARN>
            <ECMOTO>1</ECMOTO>
            <ReimbursementAttribute>A</ReimbursementAttribute>
            <CPD>2019-03-02</CPD>
            <TransactionCategory>D</TransactionCategory>
            <AcquirerBID>10000006</AcquirerBID>
            <NetworkID>0002</NetworkID>
            <AcquirerBIN>400000</AcquirerBIN>
            <RetrievalReferenceNumber>700000000000
            </RetrievalReferenceNumber>
            <CardholderName>Name</CardholderName>
            <SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
            <TranDestAmt currency="840">14208.76</TranDestAmt>
            <ProcessingCode>00</ProcessingCode>
            <IssuerName>issuer name</IssuerName>
            <AcquirerName>acquirer name</AcquirerName>
            <AccountType>00</AccountType>
            <DisputeJurisdiction jurisdictionCd="DOM">
                <Jurisdiction>DOMESTIC-US</Jurisdiction>
                <IssuerRegion>US</IssuerRegion>
                <AcquirerRegion>US</AcquirerRegion>
```

```
<CountryCd>US</CountryCd>
</DisputeJurisdiction>
</TransactionInfo>
<VisaCaseNumber>1000000000</VisaCaseNumber>
<ContactMessageId>0000001</ContactMessageId>
<MemberCaseNumber>555555555555</MemberCaseNumber>
<DisputeCategory>13</DisputeCategory>
<CaseHistorySummary>
    <CaseHistory>
        <Event>Dispute</Event>
        <CPD>2019-03-06</CPD>
        <DisputeAmount currency="840">14208.76</DisputeAmount>
        <DisputeCategoryCondition>13.3 Consumer -
            Not as Described</DisputeCategoryCondition>
    </CaseHistory>
    <CaseHistory>
        <Event>Dispute Response</Event>
        <CPD>2019-03-06</CPD>
        <DisputeAmount currency="840">14208.76</DisputeAmount>
        <DisputeCategoryCondition>13.3 Consumer -
            Not as Described</DisputeCategoryCondition>
    </CaseHistory>
    <CaseHistory>
        <Event>Pre-Arbitration</Event>
        <CPD>2019-03-06</CPD>
        <DisputeAmount currency="840">14208.76</DisputeAmount>
        <DisputeCategoryCondition>13.3 Consumer -
            Not as Described</DisputeCategoryCondition>
    </CaseHistory>
    <CaseHistory>
        <Event>Pre-Arbitration Response</Event>
        <CPD>2019-03-06</CPD>
        <DisputeAmount currency="840">14208.76</DisputeAmount>
        <DisputeCategoryCondition>13.3 Consumer -
            Not as Described</DisputeCategoryCondition>
    </CaseHistory>
    <CaseHistory>
        <Event>Arbitration</Event>
        <CPD>2019-03-06</CPD>
        <DisputeAmount currency="840">14208.76</DisputeAmount>
        <DisputeCategoryCondition>13.3 Consumer -
            Not as Described</DisputeCategoryCondition>
    </CaseHistory>
</CaseHistorySummary>
<DaysBetweenEvents>
    <Event>Dispute from the Transaction</Event>
    <Days>4</Days>
</DaysBetweenEvents>
<DaysBetweenEvents>
    <Event>Dispute Response from the Dispute</Event>
    <Days>0</Days>
</DaysBetweenEvents>
<DaysBetweenEvents>
    <Event>Pre-arbitration from the Dispute Response</Event>
    <Days>0</Days>
</DaysBetweenEvents>
<DaysBetweenEvents>
```

```
<Event>Pre-arbitration Response from the
Pre-arbitration</Event>
<Days>0</Days>
</DaysBetweenEvents>
<DaysBetweenEvents>
<Event>Arbitration from the Pre-arbitration
Response</Event>
<Days>0</Days>
</DaysBetweenEvents>
<ResponseDate>2019-05-01</ResponseDate>
<VisaComments>comment</VisaComments>
</responseData>
</SIGetContactMessageDetailsResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": {
    "MemberRole": "I",
    "User": {
      "id": "abcde1234f",
      "type": "internalId"
    }
  },
  "requestData": {
    "VisaCaseNumber": 123456789,
    "ContactMessageId": 000001,
    "IncludeDisputeAsImageInd": "false",
    "DownloadImageInd": "false"
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": {
    "Code": "I-300000000",
    "Message": "Successfully completed Operation."
  },
  "responseData": {
    "TransactionInfo": {
      "AccountNumber": 410000000000,
      "Transaction": {
        "TranDate": "2017-05-26",
        "Amount": {
          "value": 14208.76,
          "currency": "840"
        }
      },
      "id": 10000000000001,
    },
    "MerchantName": "MERCHANT NAME",
    "MerchantCity": "MERCHANT CITY",
    "MerchantCountrySubEntityCode": "CA",
    "MerchantCountryCode": "US",
    "MerchantPostalCode": 94403,
    "MCC": 5411,
    "ARN": 00000000000000000000000000000000
  }
}
```

```
"ECMOTO": "1",
"ReimbursementAttribute": "A",
"CPD": "2019-03-02",
"TransactionCategory": "D",
"AcquirerBID": 10000000,
"NetworkID": "0002",
"AcquirerBIN": 000001,
"RetrievalReferenceNumber": 700000000000,
"SystemTraceAuditNumber": 200007,
"CardholderName": "Name",
"TranDestAmt":{
  "value": 14208.76,
  "currency": "840",
},
"ProcessingCode": "00",
"IssuerName": "ABC BANK",
"AcquirerName": "BBB BANK",
"AccountType": "00",
"DisputeJurisdiction":{
  "Jurisdiction": "DOMESTIC-US",
  "IssuerRegion": "US",
  "AcquirerRegion": "US",
  "CountryCd": "US"
  "@jurisdictionCd": "DOM",
}
},
"VisaCaseNumber": 1000000000,
>ContactMessageId": 000001,
"MemberCaseNumber": "ABCDE",
"DisputeCategory": {"value": "13"}, 
"CaseHistorySummary": {
  "CaseHistory": [
    {
      "Event": "Dispute",
      "CPD": "2019-03-06",
      "DisputeAmount": {
        "value": 14208.76,
        "currency": "840"
      },
      "DisputeCategoryCondition": "13.3 Consumer - Not as Described"
    },
    {
      "Event": "Dispute Response",
      "CPD": "2019-03-06",
      "DisputeAmount": {
        "value": 14208.76,
        "currency": "840"
      },
      "DisputeCategoryCondition": "13.3 Consumer - Not as Described"
    },
    {
      "Event": "Pre-Arbitration",
      "CPD": "2019-03-06",
      "DisputeAmount": {
```

```
        "value": 14208.76,
        "currency": "840"
    },
    "DisputeCategoryCondition": "13.3 Consumer - Not as Described"
},
{
    "Event": "Pre-Arbitration Response",
    "CPD": "2019-03-06",
    "DisputeAmount":{
        "value": 14208.76,
        "currency": "840"
    },
    "DisputeCategoryCondition": "13.3 Consumer - Not as Described"
},
{
    "Event": "Arbitration",
    "CPD": "2019-03-06",
    "DisputeAmount":{
        "value": 14208.76,
        "currency": "840"
    },
    "DisputeCategoryCondition": "13.3 Consumer - Not as Described"
},
],
},
"DaysBetweenEvents": [
{
    "Event": "Dispute from the Transaction",
    "Days": 4
},
{
    "Event": "Dispute Response from the Dispute",
    "Days": 0
},
{
    "Event": "Pre-Arbitration from the Dispute Response",
    "Days": 0
},
{
    "Event": "Pre-Arbitration Response from the Pre-Arbitration",
    "Days": 0
},
{
    "Event": "Arbitration from the Pre-arbitration Response",
    "Days": 0
},
],
"ResponseDate": "2019-05-01",
"VisaComments": "TEST123CONTACT"
}
```

## SISubmitContactMessageResponseRequest

SISubmitContactMessageResponseRequest is used to submit a response to a contact message.

The following elements should be considered when submitting a contact message response or when receiving a response from this service.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.
ContactMessageId	A unique VROL-generated ID for each contact message.

SISubmitContactMessageResponse contains request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SISubmitContactMessageResponseRequest>
    <RequestHeader>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <ContactMessageId>000001</ContactMessageId>
        <MemberCaseNumber>555555555555</MemberCaseNumber>
        <Comments>any comment</Comments>
        <DisputeAttachmentDescriptor>
            <Attachment id="1">
                <ContentType>application/pdf</ContentType>
                <Comment>comment</Comment>
                <ImageData>xxxxxx</ImageData>
                <DocType>OTHER</DocType>
            </Attachment>
        </DisputeAttachmentDescriptor>
        <Action>Submit</Action>
    </requestData>
</SISubmitContactMessageResponseRequest>
```

### Response Message—SOAP-based Services

```
<SISubmitContactMessageResponseResponse>
    <Status>
        <Code>I-300000000</Code>
        <Message>Successfully completed Operation.</Message>
    </Status>
    <responseData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <ContactMessage>
            <ContactMessageId>000001</ContactMessageId>
            <ContactMessageResponseId>000002</ContactMessageResponseId>
        </ContactMessage>
    </responseData>
</SISubmitContactMessageResponseResponse>
```

### Request Message—REST-based Services

```
{  
    "RequestHeader": {  
        "MemberRole": "I"  
    },  
    "requestData": {  
        "VisaCaseNumber": 1000000000,  
        "ContactMessageId": 000001,  
        "MemberCaseNumber": "test ",  
        "Comments": "test_response",  
        "DisputeAttachmentDescriptor": {  
            "Attachment": {  
                "@id": 1,  
                "ContentType": "application/pdf",  
                "Comment": "test",  
                "ImageData": "data",  
                "DocType": "OTHER"  
            }  
        },  
        "Action": "Submit"  
    }  
}
```

### Response Message—REST-based Services

```
{  
    "Status": {  
        "Code": "I-300000000",  
        "Message": "Successfully completed Operation."  
    },  
    "responseData": {  
        "VisaCaseNumber": 1000000000,  
        "ContactMessage": {  
            "ContactMessageId": 000001,  
            "ContactMessageResponseId": 000002  
        }  
    }  
}
```

## SIGetContactMessageResponseDetailsRequest

SIGetContactMessageResponseDetailsRequest is used to retrieve the details of a contact message response.

**Note:** Any image attached to the contact message response is downloaded by default. If the user does not want to download the attached image, then they need to set the <DownloadImageInd>=false in the request.

**Note:** In the response message, <CaseHistorySummary> is not applicable for Letter.

SIGetContactMessageResponseDetails contains request and response messages as shown in the following sample messages.

## Request Message—SOAP-based Services

```
<SIGetContactMessageResponseDetailsRequest>
    <RequestHeader>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <ContactMessageResponseId>000001</ContactMessageResponseId>
        <IncludeDisputeAsImageInd>true</IncludeDisputeAsImageInd>
    </requestData>
</SIGetContactMessageResponseDetailsRequest>
```

## Response Message—SOAP-based Services

```
<SIGetContactMessageResponseDetailsResponse>
    <Status>
        <Code>I-300000000</Code>
        <Message>Successfully completed Operation.</Message>
    </Status>
    <responseData>
        <TransactionInfo>
            <AccountNumber>410000000000</AccountNumber>
            <Transaction id="0000000000000000">
                <TranDate>2017-05-24</TranDate>
                <Amount currency="344">8.24</Amount>
            </Transaction>
            <MerchantName>MERCHANT NAME</MerchantName>
            <MerchantCity>MERCHANT CITY</MerchantCity>
            <MerchantCountrySubEntityCode>CA
            </MerchantCountrySubEntityCode>
            <MerchantCountryCode>HK</MerchantCountryCode>
            <MerchantPostalCode>94408</MerchantPostalCode>
            <MCC>5000</MCC>
            <ARN>000000000000000000000001</ARN>
            <ReimbursementAttribute>A</ReimbursementAttribute>
            <CPD>2017-05-24</CPD>
            <TransactionCategory>D</TransactionCategory>
            <AcquirerBID>10000006</AcquirerBID>
            <NetworkID>0002</NetworkID>
            <AcquirerBIN>400000</AcquirerBIN>
            <RetrievalReferenceNumber>700000000000
            </RetrievalReferenceNumber>
            <SystemTraceAuditNumber>000001</SystemTraceAuditNumber>
            <TranDestAmt currency="344">8.24</TranDestAmt>
            <ProcessingCode>00</ProcessingCode>
            <IssuerName>issuer name</IssuerName>
            <AcquirerName>acquirer name</AcquirerName>
            <AccountType>00</AccountType>
            <DisputeJurisdiction jurisdictionCd="DOM">
                <Jurisdiction>DOMESTIC-HK</Jurisdiction>
                <IssuerRegion>AP</IssuerRegion>
                <AcquirerRegion>AP</AcquirerRegion>
                <CountryCd>HK</CountryCd>
            </DisputeJurisdiction>
        </TransactionInfo>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
    </responseData>
</SIGetContactMessageResponseDetailsResponse>
```

```

<ContactMessage>
    <ContactMessageId>000001</ContactMessageId>
    <ContactMessageResponseId>000002
    </ContactMessageResponseId>
</ContactMessage>
<MemberCaseNumber>555555555555</MemberCaseNumber>
<CaseHistorySummary>
    <CaseHistory>
        <Event>Dispute</Event>
        <CPD>2016-01-01</CPD>
        <DisputeAmount currency="840">100</DisputeAmount>
        <DisputeCategoryCondition>12.2 Processing Error - Incorrect Transaction Code</DisputeCategoryCondition>
    </CaseHistory>
</CaseHistorySummary>
<DisputeCategory>12</DisputeCategory>
<DaysBetweenEvents>
    <Event>Dispute from the Transaction</Event>
    <Days>5</Days>
</DaysBetweenEvents>
<ResponseDate>2016-01-01</ResponseDate>
<VisaComments>comment</VisaComments>
<Comments>any comment</Comments>
<DisputeImageAttachment>
    <AttachType>Mime</AttachType>
    <Attachment>
        <AttachType>Mime</AttachType>
        <Attachment id="String">
            <ContentType>application/pdf</ContentType>
            <Comment>any comment</Comment>
            <ImageData>xxxxxx</ImageData>
        </Attachment>
        <DocId>0</DocId>
    </DisputeImageAttachment>
    <DisputeAsImageResponseDescriptor>
        <ContentType>application/pdf</ContentType>
        <ImageData>xxxxxx</ImageData>
    </DisputeAsImageResponseDescriptor>
</ResponseData>
</SIGetContactMessageResponseDetailsResponse>

```

### Request Message—REST-based Services

```
{
    "RequestHeader": {
        "MemberRole": "I"
    },
    "requestData": {
        "VisaCaseNumber": 1000000000,
        "ContactMessageResponseId": 000001,
        "IncludeDisputeAsImageInd": "true"
    }
}
```

### Response Message—REST-based Services

```
{
```

```
"Status": {
    "Code": "I-300000000",
    "Message": "Successfully completed Operation."
},
"ResponseData": {
    "TransactionInfo": {
        "AccountNumber": 410000000000,
        "Transaction": {
            "@id": 0000000000000000,
            "TranDate": "2017-05-26",
            "Amount": {
                "@currency": 554,
                "$": 85.11
            }
        },
        "MerchantName": "MERCHANT NAME",
        "MerchantCity": "MERCHANT CITY",
        "MerchantCountrySubEntityCode": "CA",
        "MerchantCountryCode": "NZ",
        "MerchantPostalCode": 94421,
        "MCC": 5000,
        "ARN": 00000000000000000000000000000001,
        "ReimbursementAttribute": "A",
        "CPD": "2017-05-26",
        "TransactionCategory": "D",
        "AcquirerBID": 10000000,
        "NetworkID": "0002",
        "AcquirerBIN": 400000,
        "RetrievalReferenceNumber": 700000000000,
        "SystemTraceAuditNumber": 080000,
        "CardholderName": "name",
        "TranDestAmt": {
            "@currency": 554,
            "$": 79.01
        },
        "ProcessingCode": "00",
        "IssuerName": "AAA BANK",
        "AcquirerName": "BBB BANK",
        "AccountType": "00",
        "DisputeJurisdiction": {
            "@jurisdictionCd": "DOM",
            "Jurisdiction": "DOMESTIC-NZ",
            "IssuerRegion": "AP",
            "AcquirerRegion": "AP",
            "CountryCd": "NZ"
        }
    },
    "VisaCaseNumber": 1000000000,
    "ContactMessage": {
        "ContactMessageId": 000001,
        "ContactMessageResponseId": 000002
    },
    "MemberCaseNumber": "test",
    "CaseHistorySummary": {
        "CaseHistory": [
            {
                "Event": "Dispute",
                "CaseHistory": [
                    {
                        "Event": "Dispute"
                    }
                ]
            }
        ]
    }
}
```

```
"CPD": "2017-05-31",
"DisputeAmount": {
    "@currency": 554,
    "$": 68.01
},
"DisputeCategoryCondition": "12.2 Processing Error - Incorrect
Transaction Code"
],
},
"DisputeCategory": 12,
"DaysBetweenEvents": [
{
    "Event": "Dispute from the Transaction",
    "Days": 5
},
],
"ResponseDate": "2017-06-15",
"VisaComments": "TEST123CONTACT",
"Comments": "test_contactmessage response",
"DisputeImageAttachment": [
{
    "AttachType": "Mime",
    "Attachment": {
        "AttachType": "Mime",
        "Attachment": [
            {
                "@id": "String",
                "ContentType": "image/jpeg",
                "Comment": "a",
                "ImageData": "data"
            },
            ...
        ]
    }
},
"DocId": 0
},
"DisputeAsImageResponseDescriptor": {
    "ContentType": "image/tiff",
    "ImageData": "data"
}
}
}
```

## SIChangeDisputeStatusRequest

SIChangeDisputeStatusRequest can be used to delete a saved case filing (arbitration or compliance).

The following elements should be considered when deleting a saved case filing.

Element	Description
ChangeItemID	Uniquely identifies the case filing item being deleted.  ChangeItemID includes the unique ID value for the corresponding case filing item, which is the DisputeFilingItemId (for arbitration and compliance).
ChangeItemType	Indicates the type of case filing being deleted.  Domain values: <ul style="list-style-type: none"><li>• ARB=Arbitration</li><li>• COMP=Compliance</li></ul>
Action	Indicates the corresponding action to be applied to the case filing item.  Domain value: Delete

SICChangeDisputeStatus request and response messages are shown in the following sample messages.

### Request Message—Delete action—SOAP-based Services

```
<SICChangeDisputeStatusRequest>
  <RequestHeader>
    <MemberRole>I</MemberRole>
  </RequestHeader>
  <requestData>
    <VisaCaseNumber>1234567890</VisaCaseNumber>
    <ChangeItemID>1234</ChangeItemID>
    <ChangeItemType>ARB</ChangeItemType>
    <Action>Delete</Action>
  </requestData>
</SICChangeDisputeStatusRequest>
```

### Response Message—Delete action—SOAP-based Services

```
<SICChangeDisputeStatusResponse>
  <status>
    <Code>I-300000000</Code>
    <Message>Successfully completed Operation.</Message>
  </status>
  <responseData>
    <VisaCaseNumber>1234567890</VisaCaseNumber>
    <ChangeItemID>1234</ChangeItemID>
  </responseData>
</SICChangeDisputeStatusResponse>
```

### Request Message—Delete action—REST-based Services

```
{  
  "RequestHeader": {  
    "MemberRole": "I"  
  },  
  "requestData": {  
    "VisaCaseNumber": 1234567890,  
    "ChangeItemID": 1234,  
    "ChangeItemType": "ARB",  
    "Action": "Delete"  
  }  
}
```

### Response Message—Delete action—REST-based Services

```
{  
  "Status": {  
    "Code": "I-3000000000",  
    "Message": "Successfully completed Operation."  
  },  
  "responseData": {  
    "VisaCaseNumber": 1234567890,  
    "ChangeItemID": 1234  
  }  
}
```

THIS PAGE INTENTIONALLY LEFT BLANK.

# Miscellaneous Fees

12

Miscellaneous fee (fee collection and funds disbursement) transactions are financial exception transactions that are sent through VisaNet to debit or credit a member's account. Users can initiate a stand-alone miscellaneous fee or create one within an existing case.

Miscellaneous fees are valid on Visa, Interlink, and Plus networks.

Members can elect to automatically send miscellaneous fees on the basis of a pre-filing (pre-arbitration, pre-compliance) response. They can also send miscellaneous fee transactions as part of the case filing (arbitration, compliance) process.

Issuers and acquirers should check with their Visa representative to determine if this feature is available in their region and review their internal processes to determine how this might affect their proprietary systems.

VROL supports the *submission* of Fee Collection and Funds Disbursements (FC/FD) for most valid reason codes and the *receipt* of FC/FD with any valid reason code. These valid reason codes are identified in the IES.

**Note:** Reason codes 0190, 0300, 5xxx, and some 6xxx (xxx represents any three digit number) are documented in BASE II and/or V.I.P. manuals, however, VROL does not allow a user to select these codes when submitting an FC/FD.

A user may dispute/return an FC/FD that has been received for reason codes 0190 and 6xxx as long as the reason code is not from a VROL Visa Source BIN.

To send or receive member-to-member miscellaneous fees in VROL, the members financial BINs must be set up in CORE and Config, and the VROL organization setting must be enabled, including identifying the organization as BASE II or SMS.

Because VROL does not use the member's BIN to send Visa-initiated miscellaneous fees, no CORE or Config setting or VROL organization setting is required for Visa-initiated fees.

**Note:** Visa BIN 412063 is used for Visa-initiated miscellaneous fees for all jurisdictions. The client must not send miscellaneous fees to this BIN.

A user must be entitled in their user profile in order to send miscellaneous fees.

## Miscellaneous Fees Process

This section describes the process and XML operations used by issuers and acquirers to send and receive miscellaneous fee transactions.

Miscellaneous fees can be created based on an existing case found during a user's workflow (see [Chapter 17, Queues](#)). Otherwise, a new case can be created (see [Chapter 5, Case Creation](#)). In both scenarios, VROL returns the VROL Case Number.

### Send Misc Fees

1. To send a miscellaneous fee request, user sends request SISubmitMiscFeeRequest with miscellaneous fee data.  
VROL sends response.
2. Once the miscellaneous fee has been processed by BASE II/SMS and EM (CPS transaction), the user can determine its success as described in the following sections:
  - [Retrieve Misc Fees Network Approval Status](#)
  - [Check for Misc Fee Rejects](#)

### Retrieve Misc Fees Network Approval Status

#### (SMS Only)

Network approval is not returned in the response when sending a misc. fee from VROL. The sender can check the reject queue, but may prefer confirmation that a misc. fee was processed successfully by V.I.P. (and EM when applicable for CPS).

Users that send misc. fees from VROL can determine if the transactions sent have been approved or rejected by V.I.P. by sending the detail request each day until they receive a status in the response that the approval or rejection has occurred. This process typically takes 1-2 days. If the transaction is still waiting for a response from V.I.P., it will be indicated in the response.

1. To retrieve Misc Fees network approval status, user sends request SIGetMiscFeeDetailsRequest using VROL Case Number, Misc Fee ID, and QN type. The <ReturnDetails> field is used to indicate that the user wants to receive the network approval status message.
2. VROL sends response SIGetMiscFeeDetailsResponse including the network status.

### Check for Misc Fee Rejects

**Note:** It takes 1–2 days for V.I.P. to send a reject/return message to VROL.

1. To check for miscellaneous fees rejected BASE II or SMS, user sends request SIGetQueueRequest for MY\_REJECTS or ALL\_REJECTS queue.  
VROL returns the VROL Case Number, Reject ID, and success status.
2. To ignore a miscellaneous fee reject when using the reject queue, send request SIIgnoreRejectRequest using VROL Case Number, Reject ID, and RejectedTransactionType= Funds Disbursement or Fee Collection. This step prevents the transaction from being returned again in the next queue request.
3. To repair, resend the original including the Visa Case Number. This is equivalent to a second fee collection/funds disbursement.

## Retrieve Misc Fees

1. To get miscellaneous fee details, user sends request **SIGetMiscFeeDetailsRequest** using VROL Case Number and Misc Fee ID.
2. VROL sends response. The Miscellaneous Fees Type can be used to distinguish between a fee collection and a funds disbursement.
3. To close the miscellaneous fee and remove it from the queue, issuer sends close misc fee request **SICloseMiscFeeRequest** or **SICloseTransactionRequest** with CloseItem Type as MISCFEE. This step prevents the transaction from being returned again in the next queue request. See [Chapter 17, Queues](#), for more information.

**Note:** *SICloseTransactionRequest closes the transaction and removes it from its respective queue but does not close the case. For more information, see [SICloseTransactionRequest](#).*

VROL sends response with success status.

## Miscellaneous Fee Operations

RTSI exposes the following miscellaneous fee operations.

**SISubmitMiscFeeRequest**—Used to request a miscellaneous fee with data.

**SIGetMiscFeeDetailsRequest**—Used to request details of a miscellaneous fee.

**SIgnoreRejectRequest**—Used to ignore a rejected miscellaneous fee transaction. See [Chapter 17, Queues](#), for details.

### SISubmitMiscFeeRequest

This operation allows members to submit miscellaneous fees and is used in request and response messages as shown in the following sample messages.

The following table provides information about the MessageText element that should be considered when processing requests or retrieving a response.

Element	Description
MessageText	<ul style="list-style-type: none"><li>• For BASE II records, the maximum length of this field is 70 characters. BASE II is defined as either the member is configured in VROL for BASE II financials or Network ID value = 0002.</li><li>• For SMS records, the maximum length is 255.</li><li>• For US Domestic jurisdiction, if RC=150 and the card type is Visa Card/Visa Electron Card, then MMT should include "Code 10" after the cardholder name.</li><li>• For non-US jurisdiction, if RC=402 then depending on the date type, MMT should include the text "Auth" for Authorization Date or "Tran" for the Visa Prepaid Load Service Transaction Date before the date.</li></ul>

**Request Message—SOAP-based Services**

```
<SISubmitMiscFeeRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <MiscellaneousFee type="FeeCollection">
            <AccountNumber>xxxxxxxxxxxxxx</AccountNumber>
            <TransactionID>xxxxxxxxxxxxxx</TransactionID>
            <SourceBin>xxxxxx</SourceBin>
            <DestinationBin>xxxxxx</DestinationBin>
            <MiscFeeAmount currency="840">1000<MiscFeeAmount>
            <NetworkID>0004</NetworkID>
            <MiscFeeReasonCode>0170</MiscFeeReasonCode>
            <MessageText>MessageText</MessageText>
            <EventDate>2014-08-08</EventDate>
        </MiscellaneousFee>
    </requestData>
</SISubmitMiscFeeRequest>
```

**Response Message—SOAP-based Services**

```
<SISubmitMiscFeeResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <MiscFeeID>0000000</MiscFeeID>
    </responseData>
</SISubmitMiscFeeResponse>
```

### Request Message—REST-based Services

```
{  
    "RequestHeader": "...",  
    "requestData": {  
        "VisaCaseNumber": "0000000000",  
        "MiscellaneousFee": {  
            "type": "FeeCollection",  
            "ProcessingPlatform": "BASE2",  
            "AccountNumber": "111111111111",  
            "TransactionID": "a",  
            "SourceBin": "111111",  
            "DestinationBin": "111111",  
            "MiscFeeAmount": {  
                "currency": "111",  
                "Text": 1  
            },  
            "SettlementAmount": {  
                "currency": "111",  
                "Text": 1  
            },  
            "NetworkID": "aaa",  
            "ATMTransactionInd": true,  
            "MerchantCountryCode": "AA",  
            "MiscFeeReasonCode": "A",  
            "MessageText": "a",  
            ...  
        }  
    }  
}
```

### Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "VisaCaseNumber": "0000000000",  
        "MiscFeeID": "0000000"  
    }  
}
```

## SIGetMiscFeeDetailsRequest

This operation allows members to request details for a miscellaneous fee and is used in request and response messages as shown in the following sample messages.

The following table provides information about some XML elements that should be considered when submitting a miscellaneous fees detail request or retrieving a response.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.  <b>Note:</b> In the schema, the VROL Case Number is <VisaCaseNumber>.
MiscFeeID	Uniquely identifies a miscellaneous fees transaction and is used to access the miscellaneous fees information in a subsequent request.
ReturnDetails  <b>Note:</b> SMS Only	The domain value of <ReturnDetails> indicates if the user wants to receive the SMS approval status message as follows: <ul style="list-style-type: none"><li>• (not provided)—Existing details only (no SMS status information).</li><li>• FULL—Existing details and SMS status information.</li><li>• STATUS_ONLY—SMS status only (no financial details)</li></ul> See <a href="#">Network Approval Status Messages</a> later in this chapter.

## Request Message—SOAP-based Services

```
<SIGetMiscFeeDetailsRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <MiscFeeID>0000000</MiscFeeID>
    </requestData>
</SIGetMiscFeeDetailsRequest>
```

## Response Message—SOAP-based Services

```
<SIGetMiscFeeDetailsResponse>
    <Status>...</Status>
    <responseData>
        <MiscFeeDetailsResponse>
            <VisaCaseNumber>0000000000</VisaCaseNumber>
            <MiscFeeID>0000000</MiscFeeID>
            <MiscellaneousFee type="FeeCollection">
                <AccountNumber>xxxxxxxxxxxxxxxx</AccountNumber>
                <TransactionID>xxxxxxxxxxxxxxxx</TransactionID>
                <SourceBin>xxxxxx</SourceBin>
                <DestinationBin>xxxxxx</DestinationBin>
                <MiscFeeAmount currency="840">1000</MiscFeeAmount>
                <SettlementAmount currency="840">1000</SettlementAmount>
                <NetworkID>0004</NetworkID>
                <MiscFeeReasonCode>0170</MiscFeeReasonCode>
                <MessageText>MessageText</MessageText>
                <EventDate>2014-08-08</EventDate>
                <CPD>2014-08-13</CPD>
            </MiscellaneousFee>
        </MiscFeeDetailsResponse>
    </responseData>
</SIGetMiscFeeDetailsResponse>
```

## Request Message—REST-based Services

```
{  
    "RequestHeader": "...",  
    "requestData":{  
        "VisaCaseNumber": "0000000000",  
        "MiscFeeID": "0000000"  
    }  
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData":{  
        "MiscFeeDetailsResponse":{  
            "VisaCaseNumber": "0000000000",  
            "MiscFeeID": "0000000",  
            "MiscellaneousFee":{  
                "type": "FeeCollection",  
                "ProcessingPlatform": "SMS",  
                "AccountNumber": "1111111111",  
                "TransactionID": "a",  
                "SourceBin": "111111",  
                "DestinationBin": "111111",  
                "MiscFeeAmount":{  
                    "currency": "111",  
                    "Text": 1  
                },  
                "SettlementAmount":{  
                    "currency": "111",  
                    "Text": 1  
                },  
                "NetworkID": "aaa",  
                ...  
            },  
        }  
    }  
}
```

## Network Approval Status Messages

### SMS Only

For SMS members only the value for ReturnDetails in the SIGetMiscFeeDetailsRequest (FULL or STATUS\_ONLY) determines the request and response messages. One of two messages appear in the StatusMessage of a successful response: Acknowledged or Pending Network Response.

If a response has not yet been received by VROL from the network at the time the RTSI user sends the request, an error is returned (E-140000010). If the RTSI user receives Pending Network Response or this error, they should repeat the request later.

### Request Message—SOAP-based Services

```
<SIGetMiscFeeDetailsRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <MiscFeeID>0000000</MiscFeeID>
        <ReturnDetails>...</ReturnDetails>
    </requestData>
</SIGetMiscFeeDetailsRequest>
```

### Response Message—ReturnDetails=FULL—SOAP-based Services

Following is the response if successful or no advice returned as yet from VisaNet.

```
<SIGetMiscFeeDetailsResponse>
    <Status>...</Status>
    <responseData>
        <MiscFeeDetailsResponse>
            <VisaCaseNumber>0000000000</VisaCaseNumber>
            <MiscFeeID>0000000</MiscFeeID>
            <MiscellaneousFee type="FeeCollection">...</MiscellaneousFee>
            <SwitchingSystemStatus>
                <StatusMessage>Acknowledged</StatusMessage>
            </SwitchingSystemStatus>
        </MiscFeeDetailsResponse>
    </responseData>
</SIGetMiscFeeDetailsResponse>
```

Following is the response if request is rejected. Note that <StatusCode> and <StatusMessage> repeat up to six times if multiple rejects/returns.

```
<SIGetMiscFeeDetailsResponse>
    <Status>...</Status>
    <responseData>
        <MiscFeeDetailsResponse>
            <VisaCaseNumber>0000000000</VisaCaseNumber>
            <MiscFeeID>0000000</MiscFeeID>
            <MiscellaneousFee type="FeeCollection">
                <AccountNumber>xxxxxxxxxxxxxxxx</AccountNumber>
                <TransactionID>xxxxxxxxxxxxxxxx</TransactionID>
                <SourceBin>xxxxxx</SourceBin>
                <DestinationBin>xxxxxx</DestinationBin>
                <MiscFeeAmount currency="840">800.00</MiscFeeAmount>
                <NetworkID>0004</NetworkID>
                <MiscFeeReasonCode>0170</MiscFeeReasonCode>
                <MessageText>MessageText</MessageText>
            </MiscellaneousFee>
            <SwitchingSystemStatus>
                <StatusCode>RC14</StatusCode>
                <StatusMessage>Invalid account number...</StatusMessage>
            </SwitchingSystemStatus>
        </MiscFeeDetailsResponse>
    </responseData>
</SIGetMiscFeeDetailsResponse>
```

### Response Message—ReturnDetails=STATUS\_ONLY—SOAP-based Services

#### Important:

The successful and reject response differ only in the value of the StatusCode and description. Care must be taken to ensure the reject response is corrected and resent.

Following is the response if successful or no advice returned as yet from VisaNet.

```
<SIGetMiscFeeDetailsResponse>
    <Status>...</Status>
    <responseData>
        <MiscFeeSwitchStatusResponse>
            <VisaCaseNumber>0000000000</VisaCaseNumber>
            <MiscFeeID>0000000</MiscFeeID>
            <SwitchingSystemStatus>
                <StatusMessage>Acknowledged</StatusMessage>
            </SwitchingSystemStatus>
        </MiscFeeSwitchStatusResponse>
    </responseData>
</SIGetMiscFeeDetailsResponse>
```

Following is the response if request is rejected. Note that <StatusCode> and <StatusMessage> repeat up to six times if multiple rejects/returns.

```
<SIGetMiscFeeDetailsResponse>
    <Status>...</Status>
    <responseData>
        <MiscFeeSwitchStatusResponse>
            <VisaCaseNumber>0000000000</VisaCaseNumber>
            <MiscFeeID>xxxxxx</MiscFeeID>
            <SwitchingSystemStatus>
                <StatusCode>RC14</StatusCode>
                <StatusMessage>Invalid account number...</StatusMessage>
            </SwitchingSystemStatus>
        </MiscFeeSwitchStatusResponse>
    </responseData>
</SIGetMiscFeeDetailsResponse>
```

### Request Message—REST-based Services

```
{
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": "0000000000",
        "MiscFeeID": "0000000",
        "ReturnDetails": "..."
    }
}
```

**Response Message—ReturnDetails=FULL—REST-based Services**

```
{
  "Status": "...",
  "responseData": {
    "MiscFeeDetailsResponse": {
      "VisaCaseNumber": "0000000000",
      "MiscFeeID": "0000000",
      "MiscellaneousFee": {
        "type": "FeeCollection",
        "ProcessingPlatform": "SMS",
        "AccountNumber": "11111111111",
        "TransactionID": "a",
        "SourceBin": "111111",
        "DestinationBin": "111111",
        "MiscFeeAmount": {
          "currency": "111",
          "Text": 1
        },
        "SettlementAmount": {
          "currency": "111",
          "Text": 1
        },
        "NetworkID": "aaa",
        ...
      },
      "SwitchingSystemStatus": {
        "StatusCode": "A",
        "StatusMessage": "Invalid account number..."
      }
    }
  }
}
```

**Response Message—ReturnDetails=STATUS\_ONLY—REST-based Services**

```
{
  "Status": "...",
  "responseData": {
    "MiscFeeSwitchStatusResponse": {
      "VisaCaseNumber": "0000000000",
      "MiscFeeID": "0000000",
      "SwitchingSystemStatus": {
        "StatusMessage": "Acknowledged"
      }
    }
  }
}
```

# Fraud Reporting

13

RTSI provides Web services that allow the member host system (MHS) to:

- Add, change, delete, or reactivate a fraud advice from a new or existing case.
  - View details of a previously submitted fraud advice.
  - Repair or ignore a rejected fraud advice. (FRS fraud only: Visa, Interlink, Plus networks only.)
  - Retrieve a list of cases and determine which ones include a fraud advice (using the hypersearch function).
  - Retrieve a list of rejects, including fraud advice rejects (using the existing queue function for retrieving rejects). (FRS fraud only: Visa, Interlink, Plus networks only.)
  - Retrieve a list of cases and determine which ones include a delivered fraud report.
- Issuers and acquirers must be entitled in their user profile for fraud functionality.

The categories of Fraud reports that are supported in VROL are: a) NRI fraud, b) Convenience Check fraud, and c) other fraud. NRI (not received as issued) fraud occurs for example, when a card is mailed to a cardholder but is not actually received by the cardholder, that is not related to any transaction. Therefore, NRI fraud fields are related to the cardholder and are not merchant or transaction related. Convenience Check fraud is valid only for issuers in US or Canada and is not related to a transaction. Non-NRI, non-Convenience Check fraud is transaction related, and the user should do the fraud in a case that contains a TI transaction or provide the transaction fields with the fraud transaction.

The behavior of fraud transactions also depends on whether they are "FRS fraud" or not. FRS fraud transactions are Fraud transactions on the Visa, Interlink, and Plus networks (Interlink and Plus networks were added in release 13.2). FRS fraud requires: a) (non-NRI fraud only) use of the Account Sequence Number (ASN) and b) support of possible reject/repair.

Because VROL does not use the member's BIN to send fraud reports, there is no CORE or Config setup required.

## Fraud Disputes

Disputes with the fraud dispute category require that a Fraud report is submitted prior to, or in conjunction with the dispute in order for the dispute to be submitted in VROL.

When the issuer or the issuing side of a dual organization creates a Fraud dispute, VROL searches if an active and non-rejected Fraud report exists within the case, other cases within VROL, and the fraud reports in FRS for the disputed original transaction. If one or more Fraud reports are found for the disputed original transaction, the dispute can proceed. If none is found, the user must create a Fraud report and submit it, either independent of the dispute (but prior to dispute submission) or in conjunction with the dispute submission. User cannot update or edit any existing fraud report from the dispute.

If the Fraud Report is submitted in conjunction with the dispute, VROL proceeds with the dispute processing and will not wait for a fraud report confirmation to be received before delivering the dispute to the opposer.

## Fraud Report Process

Each RTSI fraud report process consists of a series of request and response transmissions managed through VROL. The following steps describe the process and XML operations that can be used by issuers and acquirers to perform the task.

The fraud report can be created based on an existing case found during a user's workflow (see [Chapter 17, Queues](#)). Otherwise, a new case can be created (see [Chapter 5, Case Creation](#)). In both these scenarios, VROL returns the VROL Case Number of the case.

### Send Fraud Reports

1. To send a fraud report, user sends a request SISubmitFraudReportRequest with fraud report information. User must provide the cardholder's amount and currency in the request. This information is also found in TI results (see [Chapter 4, Transaction Inquiries](#)).
2. VROL sends response with the Fraud Report ID, which can be used to get the success or reject status of the submitted fraud report. Check for the rejected fraud report as described in [Check for Fraud Report Rejects](#).

### Retrieve Delivered Fraud Reports from Queue

1. To retrieve delivered fraud reports from the queue, the member host system triggers a SIGetQueueRequest message. See [Chapter 17, Queues](#).  
VROL returns a list of the delivered fraud report cases in the queue including their VisaCaseNumber.
2. To accept and change the status of a delivered fraud report from the queue, the member host system triggers a SIChangeQueueStatusRequest message supplying the VisaCaseNumber and Status as required fields. See [Chapter 17, Queues](#) for more information.  
VROL sends response with success status.

### Retrieve Fraud Report Information

1. To request for information about a previously submitted fraud report, user sends a fraud report details request SIGetFraudReportDetailsRequest. This request uses the Fraud Report ID returned previously.
2. VROL sends response with fraud details along with VROL Case Number.

## Check for Fraud Report Rejects

**Note:** It takes 1-2 days for VROL to receive a reject/return message.

1. To check for rejected fraud reports, user sends queue request **SIGetQueueRequest** for MY\_REJECTS or ALL\_REJECTS queue.  
VROL sends response with VROL Case Number, Reject ID, and success status.
2. To ignore rejected fraud report, user sends request **SIgnoreRejectRequest** using VROL Case Number, Reject ID, and RejectedTransactionType=Fraud Report. This step prevents the transaction being returned again in the next queue request.  
VROL sends response with success status.

Repair the rejected fraud report as described in [Repair a Rejected Fraud Advice](#).

## Repair a Rejected Fraud Advice

1. User sends a queue request **SIGetQueueRequest** for MY\_REJECTS or ALL\_REJECTS queue.  
VROL sends response with VROL Case Number, Reject ID, and success status.
2. To repair the rejected fraud advice, user sends request **SISubmitFraudReportRequest** for the reject to be repaired along with fraud information.
  - If the case contains a single unrepairsed reject of the same type as the request, VROL marks the reject as repaired.
  - If the case contains multiple unrepairsed rejects of the same type as the request, VROL marks the most recent reject as repaired.
  - If the case does not contain a reject of the same type as the request, VROL does not mark any transaction as repaired.

VROL adds the incoming transaction to the case and sends response with success status.

**Note:** A user can always repair a Fraud Advice reject, even if it is not the latest Fraud Advice in the case.

**Note:** A repaired and re-submitted Fraud reject will be treated in the system like the submission of a new Fraud Advice. If it is successfully submitted, it will supersede any existing Fraud Advices in the case.

## Fraud Report Operations

RTSI exposes the following fraud report operations:

**SISubmitFraudReportRequest**—Used to submit a fraud report for a new or existing case to VROL.

**SIGetFraudReportDetailsRequest**—Used to request additional information regarding a previously submitted fraud report.

**SIGetQueueRequest**—Used to get the reject codes of a rejected fraud report.

**SIgnoreRejectRequest**—Used to ignore a rejected fraud transaction. See [Chapter 17, Queues](#), for details.

## SISubmitFraudReportRequest

Use SISubmitFraudReportRequest to submit a fraud report to VROL, including all the required fraud data as listed in the IES.

The following table provides information about some XML elements that should be considered when submitting fraud report request or retrieving a response.

Element	Description
FraudType	<ul style="list-style-type: none"> <li>Valid values (issuers only):           <ul style="list-style-type: none"> <li><b>0</b> = Lost</li> <li><b>1</b> = Stolen</li> <li><b>2</b> = Card not received as Issued (NRI)</li> <li><b>3</b> = Fraudulent application (cardholder never applied for the card)</li> <li><b>4</b> = Issuer-reported counterfeit</li> <li><b>5</b> = Miscellaneous</li> <li><b>6</b> = Fraudulent use of account number</li> <li><b>A</b> = Incorrect Processing</li> <li><b>B</b> = Account or Credentials Takeover</li> </ul> </li> <li>Valid value (acquirer only):           <ul style="list-style-type: none"> <li><b>9</b> = Acquirer-reported counterfeit</li> </ul> </li> </ul>
FraudTypeCategory	<p>Domain values:</p> <ul style="list-style-type: none"> <li><b>NRI</b> - Card was not received as issued and there is no associated transaction (this choice is not available to acquirer).</li> <li><b>CONV</b> - Convenience Check Fraud Reporting (US and Canada only).</li> <li><b>CARDTXN</b> - Card Transaction Fraud.</li> <li>Default values: <b>CARDTXN</b> for card based transaction.</li> </ul>
NotificationCd (NC)	<p>Used to specify the action being performed on the fraud report. Domain values:</p> <ul style="list-style-type: none"> <li><b>1</b>=Addition</li> <li><b>2</b>=Addition of dup</li> <li><b>3</b>=Change</li> <li><b>4</b>=Delete</li> <li><b>5</b>=Reactivate. No edits restricting values based on the value of existing fraud report in case.</li> </ul>
VisaCaseNumber	<p>Case tracking number assigned by VROL to uniquely identify a case in VROL.</p> <p><b>Note:</b> In the schema, the VROL Case Number is &lt;VisaCaseNumber&gt;.</p>
FraudReportID	A unique VROL-generated ID for each successfully submitted fraud report.

Element	Description
AcctSeqNum (ASN), AutoAssignAccountSequenceNumInd	<p>The ASN identifies a transaction within a cardholder account number in FRS. So, for each fraud transaction for the same account number, the ASN should be unique, or the fraud will be considered a duplicate.</p> <p>Valid only for FRS non-NRI Fraud, that is, non-NRI Fraud on networks 2, 3, and 4. (For NRI Fraud and fraud on other networks, these fields are not valid and will be ignored.)</p> <p>The auto-sequence number (ASN) is used to distinguish between different fraud transactions in a particular account.</p> <p>When updating or deleting an existing fraud transaction (Notification Code=3, 4, or 5), the SI user should send the correct value of &lt;AcctSeqNum&gt; to identify the affected fraud transaction. &lt;AutoAssignAccountSequenceNumInd&gt; is invalid when updating or deleting an existing fraud transaction.</p> <p>Domain Values for ASN: If NotificationCd=3, 4, or 5, the user must use the value returned when creating the Fraud transaction. For NotificationCd = 1 or 2, ASN is auto-assigned.</p>
CloseFraudCaseInd	If <CloseFraudCaseInd>=true, VROL will update the case resolution status of a case to Closed-Resolved on successful submission of the fraud report.

## Non-NRI FRS Fraud With Notification Code=1 and ASN=4001

### Request Message—SOAP-based Services

```

<SISubmitFraudReportRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <FraudType>1</FraudType>
        <FraudTypeCategory>CARDEXN</FraudTypeCategory>
        <NotificationCd>1</NotificationCd>
        <IssuerGenAuth>Y</IssuerGenAuth>
        <AcctSeqNum>4001</AcctSeqNum>
        <CloseFraudCaseInd>false</CloseFraudCaseInd>
        <FraudInvestigativeStatus>11</FraudInvestigativeStatus>
    </requestData>
</SISubmitFraudReportRequest>

```

### Response Message—SOAP-based Services

```

<SISubmitFraudReportResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <FraudReportID>000000</FraudReportID>
    </responseData>
</SISubmitFraudReportResponse>

```

**Request Message—REST-based Services**

```
{
  "RequestHeader": "...",
  "requestData": {
    "VisaCaseNumber": "0000000000",
    "FraudType": 1,
    "FraudTypeCategory": "CARDTXN",
    "AccountNumber": "1111111111",
    "NotificationCd": 1,
    "IssuerGenAuth": "Y",
    "AcctSeqNum": 4001,
    "CloseFraudCaseInd": false,
    "FraudInvestigativeStatus": "11",
    ...
  }
}
```

**Response Message—REST-based Services**

```
{
  "Status": "...",
  "responseData": {
    "VisaCaseNumber": "0000000000",
    "FraudReportID": "0000000"
  }
}
```

**Non-NRI FRS Fraud With Notification Code=1 and Auto-Assign ASN=true****Request Message—SOAP-based Services**

```
<SISubmitFraudReportRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <FraudType>1</FraudType>
    <FraudTypeCategory>CARDTXN</FraudTypeCategory>
    <AccountNumber>xxxxxxxxxxxxxx</AccountNumber>
    ...
    <NotificationCd>1</NotificationCd>
    <IssuerGenAuth>X</IssuerGenAuth>
    <AutoAssignAccountSequenceNumInd>true
    </AutoAssignAccountSequenceNumInd>
    <CloseFraudCaseInd>false</CloseFraudCaseInd>
    <CardCapability>1</CardCapability>
    <FraudInvestigativeStatus>11</FraudInvestigativeStatus>
  </requestData>
</SISubmitFraudReportRequest>
```

### Response Message—SOAP-based Services

```
<SISubmitFraudReportResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <FraudReportID>0000000</FraudReportID>
  </responseData>
</SISubmitFraudReportResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData": {
    "VisaCaseNumber": "0000000000",
    "FraudType": 1,
    "FraudTypeCategory": "CARDEXN",
    "AccountNumber": "111111111111",
    ...
    "NotificationCd": 1,
    "IssuerGenAuth": "X",
    "AutoAssignAccountSequenceNumInd": true,
    "CloseFraudCaseInd": false,
    "CardCapability": "a",
    "FraudInvestigativeStatus": "11",
    ...
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": "...",
  "responseData": {
    "VisaCaseNumber": "0000000000",
    "FraudReportID": "0000000"
  }
}
```

### NRI FRS Fraud

**Note:** ASN is not valid for NRI Fraud.

**Request Message—SOAP-based Services**

```
<SISubmitFraudReportRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <FraudType>2</FraudType>
    <FraudTypeCategory>NRI</FraudTypeCategory>
    <AccountNumber>xxxxxxxxxxxxxx</AccountNumber>
    <NetworkID>0002</NetworkID>
    <NotificationCd>2</NotificationCd>
    <IssuerGenAuth>Y</IssuerGenAuth>
    <CloseFraudCaseInd>false</CloseFraudCaseInd>
    ...
  </requestData>
</SISubmitFraudReportRequest>
```

**Response Message—SOAP-based Services**

```
<SISubmitFraudReportResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <FraudReportID>0000000</FraudReportID>
  </responseData>
</SISubmitFraudReportResponse>
```

**Request Message—REST-based Services**

```
{
  "RequestHeader": "...",
  "requestData": {
    "VisaCaseNumber": "0000000000",
    "FraudType": 2,
    "FraudTypeCategory": "NRI",
    "AccountNumber": "11111111111",
    "NetworkID": "aaa",
    "NotificationCd": 2,
    "IssuerGenAuth": "Y",
    "CloseFraudCaseInd": false,
    ...
  }
}
```

**Response Message—REST-based Services**

```
{
  "Status": "...",
  "responseData": {
    "VisaCaseNumber": "0000000000",
    "FraudReportID": "0000000"
  }
}
```

## SIGetFraudReportDetailsRequest

Use SIGetFraudReportDetailsRequest to request additional information regarding a previously submitted fraud report. This request requires the Fraud Report ID and returns a SIGetFraudReportDetailsResponse from VROL.

The following table provides information about some XML elements that should be considered when submitting fraud report details request or retrieving a response:

Element	Description
CardholderNotificationDate	The date on which the cardholder was notified about Fraud.

SIGetFraudReportDetailsRequest is used in a request and response format as shown in the following sample messages.

### Non-NRI FRS Fraud

#### Request Message—SOAP-based Services

```
<SIGetFraudReportDetailsRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <FraudReportID>0000000</FraudReportID>
    </requestData>
</SIGetFraudReportDetailsRequest>
```

#### Response Message—SOAP-based Services

```
<SIGetFraudReportDetailsResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <FraudReportID>0000000</FraudReportID>
        <FraudType>1</FraudType>
        <FraudTypeCategory>CARDTXN</FraudTypeCategory>
        <AccountNumber>xxxxxxxxxxxxxx</AccountNumber>
        <Transaction id="000000000000000">...</Transaction>
        <MerchantName>Name</MerchantName>
        <MerchantCity>yellowknife</MerchantCity>
        <MerchantCountrySubEntityCode>NT</MerchantCountrySubEntityCode>
        <MerchantCountryCode>CA</MerchantCountryCode>
        <MerchantPostalCode>12345</MerchantPostalCode>
        <MCC>5999</MCC>
        <ARN>xxxxxxxxxxxxxxxxxxxxxx</ARN>
        <NetworkID>0002</NetworkID>
        ...
        <NotificationCd>1</NotificationCd>
        <IssuerGenAuth>X</IssuerGenAuth>
        <AcctSeqNum>4102</AcctSeqNum>
    </responseData>
</SIGetFraudReportDetailsResponse>
```

**Request Message—REST-based Services**

```
{
  "RequestHeader": "...",
  "requestData": {
    "FraudReportID": "0000000"
  }
}
```

**Response Message—REST-based Services**

```
{
  "Status": "...",
  "responseData": {
    "VisaCaseNumber": "0000000000",
    "FraudReportID": "0000000",
    "FraudType": 1,
    "AccountNumber": "111111111111",
    "Transaction": {
      "id": "a",
      "TranDate": "2016-01-01",
      "Amount": {
        "currency": "111",
        "Text": 1
      },
      ...
    },
    "MerchantName": "Name",
    "MerchantCity": "City",
    "MerchantCountrySubEntityCode": "CA",
    "MerchantCountryCode": "US",
    "MerchantPostalCode": "1011",
    "MCC": "5999",
    "ARN": "11111111111111111111111111",
    "NetworkID": "VSN",
    ...
    "NotificationCd": 1,
    "IssuerGenAuth": "Y",
    "AcctSeqNum": 4102,
  }
}
```

**NRI Fraud****Request Message—SOAP-based Services**

```
<SIGetFraudReportDetailsRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <FraudReportID>0000000</FraudReportID>
  </requestData>
</SIGetFraudReportDetailsRequest>
```

### Response Message—SOAP-based Services

```
<SIGetFraudReportDetailsResponse>
  <Status>...</Status>
  <ResponseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <FraudReportID>0000000</FraudReportID>
    <FraudType>2</FraudType>
    <FraudTypeCategory>NRI</FraudTypeCategory>
    <AccountNumber>xxxxxxxxxxxxxx</AccountNumber>
    <NetworkID>0002</NetworkID>
    <NotificationCd>2</NotificationCd>
    ...
    <FraudInvestigativeStatus>11</FraudInvestigativeStatus>
    <CardholderNotificationDate>2014-08-07</CardholderNotificationDate>
  </ResponseData>
</SIGetFraudReportDetailsResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData":{
    "FraudReportID": "0000000"
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": "...",
  "responseData":{
    "VisaCaseNumber": "0000000000",
    "FraudReportID": "0000000",
    "FraudType": 2,
    "AccountNumber": "111111111111",
    "NetworkID": "aaa",
    "NotificationCd": 2,
    ...
    "FraudInvestigativeStatus": 11,
    "CardholderNotificationDate": "2014-08-07"
  }
}
```

## SIGetQueueRequest

Use SIGetQueueRequest to get the reject codes of rejected fraud reports.

**Request Message—SOAP-based Services**

```
<SIGetQueueRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <QueueType>ALL_REJECTS</QueueType>
  </requestData>
</SIGetQueueRequest>
```

**Response Message—SOAP-based Services**

```
<FormatRQueueItem>
  <VisaCaseNumber>1000000000</VisaCaseNumber>
  <RejectedTransactionType type="FRD">Fraud Report</RejectedTransactionType>
  <DaysAged>277</DaysAged>
  <Amount currency="840">20.00</Amount>
  <AccountNumber>xxxxxxxxxxxxxxxxxxxx</AccountNumber>
  <User>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</User>
  <RejectID>392931</RejectID>
  <RejectReason code="R41">Invalid account number. Must be ...</RejectReason>
  <RejectReason code="R82">Transactions exceeding 180 days...</RejectReason>
  <RejectReason code="R41">Invalid account number. Must be ...</RejectReason>
  <RejectReason code="R82">Transactions exceeding 180 days...</RejectReason>
  <ARN>2000000000000000000000000008</ARN>
  <RetrievalReferenceNumber>900000000000</RetrievalReferenceNumber>
</FormatRQueueItem>
```

**Request Message—REST-based Services**

```
{
  "RequestHeader": "...",
  "requestData": {
    "QueueType": "ALL_REJECTS",
  }
}
```

### Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "QueueType": "ALL_REJECTS",  
        "PageInfo": {  
            "PageNum": 1,  
            "TotalPages": 1  
        },  
        "Queue": {  
            "FormatRQueueItem": {  
                "VisaCaseNumber": "0000000000",  
                "RejectedTransactionType": {  
                    "type": "FRD",  
                    "Text": "Fraud Report</"  
                },  
                "Reason": {  
                    "code": "A",  
                    "Text": "A"  
                },  
                "DaysAged": 1,  
                "Amount": {  
                    "currency": "840",  
                    "Text": 1  
                },  
                "AccountNumber": "1111111111",  
                "User": "A",  
                "RejectID": 392931,  
                "RejectReason": [  
                    {  
                        "code": "R41",  
                        "Text": "Invalid account number. Must be ..."  
                    }, {  
                        "code": "R82",  
                        "Text": "Transactions exceeding 180 days..."  
                    }, {  
                        "code": "R41",  
                        "Text": "Invalid account number. Must be ..."  
                    }, {  
                        "code": "R82",  
                        "Text": "Transactions exceeding 180 days..."  
                    }  
                ],  
                "ARN": 200000000000000000000000000008,  
                "RetrievalReferenceNumber": 900000000000  
            }  
        }  
    }  
}
```

THIS PAGE INTENTIONALLY LEFT BLANK.

# Exception File Listings

14

*Exception File Listings* (EFLs) are entered by issuers and used to help members recover lost, stolen, and counterfeit cards. For example, once a "bad" card number is listed in V.I.P. by means of an EFL, VisaNet can decline or return a message to the acquirer. Another use for EFL is to limit the number of transactions per day for a card number.

A Cardholder Transaction Amount Limit transaction is a type of V.I.P. Listing EFL that allows a custom Transaction Amount Limit. Users can only add, update, or delete Cardholder Transaction Amount Limit EFL transactions if the users and their VROL organization are configured for this transaction.

The submitted EFL is sent to V.I.P., and a response (success/reject) is returned immediately from V.I.P. Only the last successfully submitted EFL for a particular card number is stored by V.I.P., and a VROL real-time EFL inquiry will only retrieve this last transaction. As a result, if an EFL is submitted in one VROL case and a second EFL *for the same card number* is submitted in the same case or in a second VROL case, a subsequent VROL EFL inquiry will retrieve the second EFL. The history of EFL transactions for a card number can be retrieved from the VROL EF History Inquiry that is based on the V.I.P. transaction logs. The EF History Inquiry (EFHI) is only available in the UI.

VROL does not use the member's BIN to send Exception File Listings, so there is no CORE change needed.

VROL EFL functionality enables issuers to:

- Add/update or delete EFL from a new or existing case.
- Inquire about existing EFLs from within a case or outside a case.

When the issuer or the issuing side of a dual organization creates a Fraud or Authorization dispute, VROL searches for the existence of an active negative Exception File Listing for the same card/account number as the dispute. If one is found, the dispute can proceed. If none is found, the user can optionally create/submit an EFL request with the Fraud dispute. A new negative EFL listing cannot be created for Authorization disputes.

The following permissions are required for an issuer to use the EFL functions:

- The member's VROL organization setting must be enabled for add/update/delete functionality. This setting does not have to be enabled for real-time exception file inquiry and history inquiry.
- A user can be entitled in their user profile for EFL inquiry only, EFL inquiry and update, or disable EFL.

If the user has "Inquiry-only" permissions, they will only be able to do real-time Inquiry, history inquiry, and view from a case—they will not be able to add/update or delete an EFL.

**Note:** The term "update" in this context actually results in the addition of a second event.

## Exception File Listing Process

Each RTSI exception file listing process consists of a series of request and response transmissions managed through VROL. The following steps describe the process and XML operations used by an issuer to add, update, and delete exception file listings, and to make inquiries.

The exception file listing (EFL) can be created based on an existing case found during a user's workflow (see [Chapter 17, Queues](#)). Otherwise, a new case can be created (see [Chapter 5, Case Creation](#)). In both scenarios, VROL returns the VROL Case Number.

1. To add, update, or delete an EFL, issuer sends request SISubmitExceptionRequest with data.

**Note:** If a member submits an Exception File Listing (EFL) with purge days value that is greater than the Maximum Purge Days value as defined in the Member Preferences Screen, and the value of the Exception File Listing - Maximum Purge Days field in the Member Preferences screen is not blank or zero, then VROL returns an error message, indicating the maximum number of days allowed to purge.

VROL sends response to indicate the success or reject status of the submitted request and returns the Exception File ID.

2. To get EFL details, issuer sends request SIGetExceptionDetailsRequest using Exception File ID.

VROL sends response.

3. To get information about an existing exception file listing, issuer sends request for history SIIinquiryExceptionRequest.

VROL sends response.

## Exception File Listing Operations

RTSI exposes the following exception file operations:

**SISubmitExceptionRequest**—Used by the issuer to submit an exception file listing request for a new or existing case to VROL. This includes negative listing, V.I.P. limits, and delete.

**SIGetExceptionDetailsRequest**—Used by the issuer to request information regarding a previously submitted successful exception file listing.

**SIIinquiryExceptionRequest**—Used by the issuer to obtain updated information on an existing exception file listing from V.I.P.

### SISubmitExceptionRequest

Issuers use SISubmitExceptionRequest to add, update, or delete an exception file listing from a new or existing case. It is also used to repair a rejected exception file listing; when resending the request all the fields are resent, not just the corrected ones.

If no case matching information is supplied with the request when adding an exception file listing, VROL creates a new case. If no match is found, or multiple matches are found, VROL returns an error.

There are three values used with the SISubmitExceptionRequest:

- ListNegativeExceptionFile
- ListVIPExceptionFile
- DeleteExceptionFile

SISubmitExceptionRequest requires the following fields.

Field	Data Requirement
AccountNumber	If not supplied, value in case matching information is used.
ActivityLimit	Required when adding or updating "Set VIP Limit" request, except for Cardholder Transaction Amount Limit EFL transactions.
CardholderTransactionLimit	Required when adding or updating Cardholder Transaction Amount Limit EFL transactions.
ActionCode	Required when adding or updating "Negative File Listing" request.
CardRecoveryBulletinRegions	Valid and required if BASE I ListingType=Negative File Listing and ActionCode=04, 07, 41, or 43.
PurgeDate	Valid when submitting a Negative or V.I.P. Exception File Listing. When valid, it is required if DoNotPurgeInd=false or if DoNotPurgeInd is not supplied.  <b>Note:</b> VROL returns an error message that indicates the maximum number of days allowed to purge if the following conditions occur: <ul style="list-style-type: none"><li>• The PurgeDate value of the submitted Exception File Listing (EFL) is greater than the Maximum Purge Days that the EFL can be purged as defined in the Member Preferences screen.</li><li>• The value of the Exception File Listing - Maximum Purge Days field in the Member Preferences screen is not blank or zero. The Visa Administrator can set the value of this setting when configuring member settings for the organization.</li></ul>
DoNotPurgeInd	Valid when submitting a Negative or V.I.P. Exception File Listing. When valid, it is required if PurgeDate is not supplied.

SISubmitExceptionRequest is used in a request and response format as shown in the following sample messages.

### ListNegativeExceptionFile

#### Request Message—SOAP-based Services

```
<SISubmitExceptionRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <ListNegativeExceptionFile>
      <AccountNumber>xxxxxxxxxxxxxxxx</AccountNumber>
      <PurgeDate>2014-09-09</PurgeDate>
      <ActionCode>04</ActionCode>
    </ListNegativeExceptionFile>
  </requestData>
</SISubmitExceptionRequest>
```

## Response Message—SOAP-based Services

The following example shows a sample of RTSI response containing the confirmation of a successful transmission. The actual elements returned in a message depend on the exception file listing data available in VROL. For a full set of possible fields, refer to the XML schema or the IES.

```
<SISubmitExceptionResponse>
    <Status>...</Status>
    <responseData>
        <exceptionFileID>000000</exceptionFileID>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <ActionCode>04</ActionCode>
    </responseData>
</SISubmitExceptionResponse>
```

## Request Message—REST-based Services

```
{
    "RequestHeader": "...",
    "requestData": {
        "listNegativeExceptionFile": {
            "VisaCaseNumber": "0000000000",
            "AccountNumber": "111111111111",
            "PurgeDate": "2016-01-01",
            "ActionCode": "a",
            "cardRecoveryBulletinRegions": {
                "Region": "B"
            }
        },
    }
}
```

## Response Message—REST-based Services

```
{
    "status": "...",
    "responseData": {
        "exceptionFileID": "000000",
        "VisaCaseNumber": "0000000000",
        "ActionCode": "a",
        ...
    }
}
```

## ListVIPExceptionFile (non-Cardholder Transaction Amount Limit transaction)

### Request Message—SOAP-based Services

```
<SISubmitExceptionRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <ListVIPExceptionFile>
      <AccountNumber>xxxxxxxxxxxxxx</AccountNumber>
      <DoNotPurgeInd>true</DoNotPurgeInd>
      <ActivityLimit>A1</ActivityLimit>
    </ListVIPExceptionFile>
  </requestData>
</SISubmitExceptionRequest>
```

### Response Message—SOAP-based Services

```
<SISubmitExceptionResponse>
  <Status>...</Status>
  <responseData>
    <ExceptionFileID>000000</ExceptionFileID>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <ActivityLimit>A1</ActivityLimit>
  </responseData>
</SISubmitExceptionResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData": {
    "ListVIPExceptionFile": {
      "VisaCaseNumber": "0000000000",
      "AccountNumber": "1111111111",
      "DoNotPurgeInd": true,
      "ActivityLimit": "A5",
    }
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": "...",
  "responseData": [
    {
      "ExceptionFileID": "000000",
      "VisaCaseNumber": "0000000000",
      "ActivityLimit": "A5",
      ...
    },
  ]
}
```

**ListVIPExceptionFile (Cardholder Transaction Amount Limit transaction)**

**Note:** A Cardholder Transaction Amount Limit transaction is a type of V.I.P. Listing EFL that allows a custom Transaction Amount Limit. Users can only add, update, or delete Cardholder Transaction Amount Limit EFL transactions if the users and their VROL organization are configured for this transaction.

**Request Message—SOAP-based Services**

```
<SISubmitExceptionRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <ListVIPExceptionFile>
            <AccountNumber>xxxxxxxxxxxxxx</AccountNumber>
            <DoNotPurgeInd>true</DoNotPurgeInd>
            <CardholderTransactionLimit>55</CardholderTransactionLimit>
        </ListVIPExceptionFile>
    </requestData>
</SISubmitExceptionRequest>
```

**Response Message—SOAP-based Services**

```
<SISubmitExceptionResponse>
    <Status>...</Status>
    <responseData>
        <ExceptionFileID>000000</ExceptionFileID>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <CardholderTransactionLimit>55</CardholderTransactionLimit>
    </responseData>
</SISubmitExceptionResponse>
```

**Request Message—REST-based Services**

```
{
    "RequestHeader": "...",
    "requestData": {
        "ListVIPExceptionFile": {
            "VisaCaseNumber": "0000000000",
            "AccountNumber": "1111111111",
            "DoNotPurgeInd": true,
            "CardholderTransactionLimit": 55
        }
    }
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        ""ExceptionFileID": "000000",  
        "VisaCaseNumber": "0000000000",  
        "CardholderTransactionLimit": 55  
        ...  
    },  
}
```

## DeleteExceptionFile

**Note:** If the transaction is a Cardholder Transaction Amount Limit transaction, which is a type of V.I.P. Listing EFL, the element <CardholderTransactionLimitInd>=true is required to delete the transaction.

## Request Message—SOAP-based Services

```
<SISubmitExceptionRequest>  
    <RequestHeader>...</RequestHeader>  
    <requestData>  
        <DeleteExceptionFile>  
            <AccountNumber>xxxxxxxxxxxxxxxx</AccountNumber>  
            <CardholderTransactionLimitInd>true  
            </CardholderTransactionLimitInd> <- Only if deleting  
                a Cardholder Transaction Amount Limit transaction  
        </DeleteExceptionFile>  
    </requestData>  
</SISubmitExceptionRequest>
```

## Response Message—SOAP-based Services

```
<SISubmitExceptionResponse>  
    <Status>...</Status>  
    <responseData>  
        <ExceptionFileID>000000</ExceptionFileID>  
        <VisaCaseNumber>0000000000</VisaCaseNumber>  
        <AccountNumber>xxxxxxxxxxxxxxxx</AccountNumber>  
        <ListingType>EFDEL</ListingType>  
        <IsPurgeDateInfinite>false</IsPurgeDateInfinite>  
    </responseData>  
</SISubmitExceptionResponse>
```

**Request Message—REST-based Services**

```
{
  "RequestHeader": "...",
  "requestData": {
    "DeleteExceptionFile": {
      "VisaCaseNumber": "0000000000",
      "AccountNumber": "11111111111",
      "CardholderTransactionLimitInd": true <- Only if deleting
        a Cardholder Transaction Amount Limit transaction
    }
  }
}
```

**Response Message—REST-based Services**

```
{
  "Status": "...",
  "responseData": {
    "ExceptionFileID": "000000",
    "VisaCaseNumber": "0000000000",
    "AccountNumber": "11111111111",
    "ListingType": "EFDEL",
    "IsPurgeDateInfinite": false
  }
}
```

**SIGetExceptionDetailsRequest**

Issuers may use SIGetExceptionDetailsRequest to view details of a previously submitted successful exception file listing. This returns information already in VROL and does not communicate with V.I.P.

SIGetExceptionDetailsRequest requires the Exception File ID and account number (if not supplied in case matching information) for the listing. SIGetExceptionDetailsResponse provides exception file listing details information from VROL.

SIGetExceptionDetailsRequest requires the following field.

Field	Data Requirement
ExceptionFileID	Unique identifier used to identify EFL, which can be retrieved from either a hypersearch response or the response when submitting a new EFL.

**Request Message—SOAP-based Services**

```
<SIGetExceptionDetailsRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <ExceptionFileID>000000</ExceptionFileID>
  </requestData>
</SIGetExceptionDetailsRequest>
```

**Response Message—SOAP-based Services**

The following example shows a sample of RTSI response containing the confirmation of a successful transmission and exception file listing detail information.

**Note:** The actual elements returned in a message depend on the exception file listing data available in VROL. For all possible elements, see the IES.

```
<SIGetExceptionDetailsResponse>
  <Status>...</Status>
  <ResponseData>
    <ExceptionFileID>000000</ExceptionFileID>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <AccountNumber>xxxxxxxxxxxxxx</AccountNumber>
    <RequestDate>2009-08-13</RequestDate>
    <ListingType>String</ListingType>
    <IsPurgeDateInfinite>false<IsPurgeDateInfinite>false
    ...
  <ResponseData>
</SIGetExceptionDetailsResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData":{
    ""ExceptionFileID": "000000"
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": "...",
  "responseData":{
    ""ExceptionFileID": "000000",
    "VisaCaseNumber": "0000000000",
    "AccountNumber": "1111111111",
    "RequestDate": "2016-01-01T19:31:47Z",
    "ListingType": "EFDEL",
    "IsPurgeDateInfinite": "false<IsPurgeDateInfinite>false
    ...
  }
}
```

## SIIinquiryExceptionRequest

Issuers use SIIinquiryExceptionRequest to obtain updated information on an existing exception file listing. If the inquiry results in an error, the request can be resent with all fields, not just the corrected ones. If no case matching information is supplied with the request, VROL does not create a new case before submitting the request to V.I.P. If no matches or multiple matches are found, VROL returns an error.

SIIinquiryExceptionRequest requires the VisaCaseNumber and AccountNumber fields.

SIIinquiryExceptionRequest is used in a request and response format as shown in the following sample messages.

SIIinquiryExceptionRequest requires the following field.

Field	Data Requirement
CardholderTransactionLimitInd	Required when the transaction is a Cardholder Transaction Amount Limit transaction, which is a type of V.I.P. Listing EFL. This can be retrieved from the SIGetExceptionDetailsRequest response.

### Request Message—SOAP-based Services

```
<SIInquiryExceptionRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <AccountNumber>xxxxxxxxxxxxxx</AccountNumber>
        <CardholderTransactionLimitInd>true
        </CardholderTransactionLimitInd> <- Only if inquiring for
        a Cardholder Transaction Amount Limit transaction
    </requestData>
</SIInquiryExceptionRequest>
```

### Response Message—SOAP-based Services

The following example shows a sample of RTSI response containing the exception file listing data.

**Note:** *The actual elements returned in a message depend on the exception file listing data available in VROL. For all possible elements, see the IES.*

```
<SIInquiryExceptionResponse>
    <Status>...</Status>
    <responseData>
        <ExceptionFileID>000000</ExceptionFileID>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <ActivityLimit>A1</ActivityLimit>
        <LastUpdate>2014-08-20T05:11:18+00:00</LastUpdate>
    </responseData>
</SIInquiryExceptionResponse>
```

### Request Message—REST-based Services

```
{
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": "0000000000",
        "AccountNumber": "11111111111",
        "CardholderTransactionLimitInd": true <- Only if inquiring for
        a Cardholder Transaction Amount Limit transaction
    }
}
```

### Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "ExceptionFileID": "000000",  
        "VisaCaseNumber": "0000000000",  
        "ActivityLimit": "A1",  
        "LastUpdate": "2014-08-20T05:11:18+00:00",  
    },  
}
```

THIS PAGE INTENTIONALLY LEFT BLANK.

# Stop Payments

15

Issuers can stop automatic bill payments on a cardholder's behalf. This function is part of the Preauthorized Payment Cancellation Service (PPCS).

After the cardholder contacts the merchant to cancel, they may contact their card issuer to place a stop order for one or more specific transactions. The VisaNet Cardholder Database is updated to generate a "decline" response for future authorizations related to the cancelled recurring billing.

A customer can specify to stop a specific preauthorized payment, to stop all future preauthorized payments from a particular merchant, or to stop all automatic bill payments for multiple merchants in the event an account is closed.

When the issuer or the issuing side of a dual organization creates a Consumer dispute for a recurring transaction, VROL searches for the existence of an active Stop Payment order. If one is found, the dispute can proceed. If none is found, the user can optionally create/submit a Stop Payment order request with the dispute.

VROL supports real-time inquiries against the live PPCS file.

The member's organization must be entitled for the add/update/delete functions. This setting is not required to perform real-time or history inquiries.

VROL does not use the member's BIN to send stop payments, but there is an additional flag required in CORE for V.I.P. to stop the preauthorized payment.

## Stop Payment Process

This section describes the process and XML operations used by issuers to perform the tasks.

Stop payment orders can be created based on an existing case found during a user's workflow (see [Chapter 17, Queues](#)). Otherwise, a new case can be created (see [Chapter 5, Case Creation](#)). In both these scenarios, VROL returns the VROL Case Number of the case.

1. To add or update a stop payment, user sends request SISubmitStopPaymentOrderRequest with stop payment data.

VROL sends response with the VROL Case Number, Stop Payment ID, and success status.

2. To delete a stop payment, user sends request SISubmitStopPaymentOrderRequest with VROL Case Number and Stop Payment ID.  
VROL sends response with the VROL Case Number, Stop Payment ID, and success status.
3. To get details, user sends request SIGetStopPaymentDetailsRequest using the Stop Payment ID returned previously.  
VROL sends response.
4. To get information about an existing stop payment order, user sends request for history SISubmitStopPaymentInquiryRequest.  
VROL sends response with the VROL Case Number and Stop Payment ID for all stop payment orders matching the request.

## Stop Payment Operations

RTSI exposes the following PPCS operations:

**SISubmitStopPaymentOrderRequest**—Used to add, update, or delete a stop payment.

**SIGetStopPaymentDetailsRequest**—Used to get details of stop payment request.

**SISubmitStopPaymentInquiryRequest**—Used to get information about an existing stop payment.

### SISubmitStopPaymentOrderRequest

SISubmitStopPaymentOrderRequest is used in a request and response format as shown in the following sample messages.

#### Request Message—SOAP-based Services

```
<SISubmitStopPaymentOrderRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <StopPaymentDescriptor operation="add">
      <CancellationType>R0</CancellationType>
      <AccountNumber>xxxxxxxxxxxxxx</AccountNumber>
      <CardAcceptorId>10000000</CardAcceptorId>
      <MerchantName>Any merchant</MerchantName>
      <MerchantCategoryCode>5411</MerchantCategoryCode>
      <AcquirerInstCountryCode>US</AcquirerInstCountryCode>
    </StopPaymentDescriptor>
  </requestData>
</SISubmitStopPaymentOrderRequest>
```

#### Response Message—SOAP-based Services

```
<SISubmitStopPaymentOrderResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <StopPaymentID>1000000</StopPaymentID>
  </responseData>
</SISubmitStopPaymentOrderResponse>
```

## Request Message—REST-based Services

```
{  
    "RequestHeader": "...",  
    "requestData":{  
        "VisaCaseNumber": "0000000000",  
        "StopPaymentDescriptor":{  
            "operation": "add",  
            "CancellationType": "R1 (b)",  
            "CancellationTypeDescription": "A",  
            "AccountNumber": "1111111111",  
            "CardAcceptorId": "a",  
            "MerchantName": "Any merchant",  
            "MerchantCategoryCode": "5411",  
            "AcquirerInstCountryCode": "US",  
        }  
    }  
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData":{  
        "VisaCaseNumber": "0000000000",  
        "StopPaymentID": 1000000  
    }  
}
```

## SIGetStopPaymentDetailsRequest

SIGetStopPaymentDetailsRequest is used in a request and response format as shown in the following sample messages.

## Request Message—SOAP-based Services

```
<SIGetStopPaymentDetailsRequest>  
    <RequestHeader>...</RequestHeader>  
    <requestData>  
        <VisaCaseNumber>0000000000</VisaCaseNumber>  
        <StopPaymentID>1000000</StopPaymentID>  
    </requestData>  
</SIGetStopPaymentDetailsRequest>
```

### Response Message—SOAP-based Services

```
<SIGetStopPaymentDetailsResponse>
  <Status>...</Status>
  <ResponseData>
    <StopPaymentOrder>
      <VisaCaseNumber>0000000000</VisaCaseNumber>
      <CancellationType>R0</CancellationType>
      <CancellationTypeDescription>R0-One Time Stop Payment
      </CancellationTypeDescription>
      <AccountNumber>xxxxxxxxxxxxxx</AccountNumber>
      <CardAcceptorId>1000000</CardAcceptorId>
      <MerchantName>Any merchant</MerchantName>
      <ExpirationDate>2014-09-27</ExpirationDate>
      <AcquirerInstCountryCode>US</AcquirerInstCountryCode>
    </StopPaymentOrder>
  </ResponseData>
</SIGetStopPaymentDetailsResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData": {
    "VisaCaseNumber": "0000000000",
    "StopPaymentID": 1000000
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": "...",
  "responseData": {
    "StopPaymentOrder": {
      "VisaCaseNumber": "0000000000",
      "CancellationType": R0,
      "CancellationTypeDescription": R0-One Time Stop Payment,
      "AccountNumber": "1111111111",
      "CardAcceptorId": 1000000,
      "MerchantName": "Any merchant",
      "ExpirationDate": "2016-01-01",
      "AcquirerInstCountryCode": US,
    }
  }
}
```

## SISubmitStopPaymentInquiryRequest

SISubmitStopPaymentInquiryRequest is used in a request and response format as shown in the following sample messages.

### **Request Message—SOAP-based Services**

```
<SISubmitStopPaymentInquiryRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <AccountNumber>xxxxxxxxxxxxxxxx</AccountNumber>
  </requestData>
</SISubmitStopPaymentInquiryRequest>
```

### **Response Message—SOAP-based Services**

The following response shows three of the four cancellation types (R3 - Full Account Block is not shown):

- R0 – One Time Stop Payment
- R1(a) – Merchant Stop Payment
- R1(b) – Merchant Stop Payment with MCC

```
<SISubmitStopPaymentInquiryResponse>
  <Status>...</Status>
  <ResponseData>
    <StopPaymentOrder>
      <VisaCaseNumber>1000000001</VisaCaseNumber>
      <StopPaymentID>10000-1</StopPaymentID>
      <CancellationType>R0</CancellationType>
      <CancellationTypeDescription>R0-One Time Stop Payment
      </CancellationTypeDescription>
      <AccountNumber>xxxxxxxxxxxxxx</AccountNumber>
      <CardAcceptorId>10000000</CardAcceptorId>
      <MerchantName>Any merchant</MerchantName>
      <ExpirationDate>2014-09-26</ExpirationDate>
      <AcquirerInstCountryCode>US</AcquirerInstCountryCode>
      ...
      <Code>a</Code>
      <Message>a</Message>
    </StopPaymentOrder>
    <StopPaymentOrder>
      <VisaCaseNumber>1000000002</VisaCaseNumber>
      <StopPaymentID>10000-2</StopPaymentID>
      <CancellationType>R1 (b)</CancellationType>
      <AccountNumber>xxxxxxxxxxxxxx</AccountNumber>
      <MerchantCategoryCode>6011</MerchantCategoryCode>
      <ExpirationDate>2015-01-01</ExpirationDate>
      <RequestDate>20014-07-30</RequestDate>
      ...
      <Code>a</Code>
      <Message>a</Message>
    </StopPaymentOrder>
    <StopPaymentOrder>
      <VisaCaseNumber>1000000003</VisaCaseNumber>
      <StopPaymentID>10000-3</StopPaymentID>
      <CancellationType>R1 (a)</CancellationType>
      <AccountNumber>xxxxxxxxxxxxxx</AccountNumber>
      <CardAcceptorId>10000001</CardAcceptorId>
      <MerchantVerificationValue>00000000
      </MerchantVerificationValue>
      <MerchantName>ACME ANVIL</MerchantName>
      <ExpirationDate>2015-01-01</ExpirationDate>
      <RequestDate>2014-07-30</RequestDate>
      ...
      <Code>a</Code>
      <Message>a</Message>
    </StopPaymentOrder>
  </ResponseData>
</SISubmitStopPaymentInquiryResponse>
```

## Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData": {
    "AccountNumber": "1111111111"
  }
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "stopPaymentOrder": [  
            {  
                "VisaCaseNumber": 1000000001,  
                "StopPaymentID": "10000-1",  
                "CancellationType": "R0",  
                "CancellationTypeDescription": "R0-One Time Stop Payment",  
                "AccountNumber": "xxxxxxxxxxxxxxxxxx",  
                "CardAcceptorId": 10000000,  
                "MerchantName": "Any merchant",  
                "ExpirationDate": "2014-09-26",  
                "AcquirerInstCountryCode": "US"  
            }, {  
                "VisaCaseNumber": 1000000002,  
                "StopPaymentID": "10000-2",  
                "CancellationType": "R1(b)",  
                "AccountNumber": "xxxxxxxxxxxxxxxxxx",  
                "MerchantCategoryCode": 6011,  
                "ExpirationDate": "2015-01-01",  
                "RequestDate": "20014-07-30"  
            }, {  
                "VisaCaseNumber": 1000000003,  
                "StopPaymentID": "10000-3",  
                "CancellationType": "R1(a)",  
                "AccountNumber": "xxxxxxxxxxxxxxxxxx",  
                "CardAcceptorId": 10000001,  
                "MerchantVerificationValue": "00000000",  
                "MerchantName": "ACME ANVIL",  
                "ExpirationDate": "2015-01-01",  
                "RequestDate": "2014-07-30"  
            }  
        ]  
    }  
}
```

THIS PAGE INTENTIONALLY LEFT BLANK.

# Accounting Entry Memos

16

VROL enables issuers and acquirers to create and update Accounting Entry Memos (AEMs) through RTSI. AEMs record how issuers and acquirers track their internal general ledger entries for a case. These AEM records are stored in VROL and are not submitted outside the system nor are they visible to the opposing side of a case. VROL Report ROL-150—Accounting Entry Memo Report can be used to export the data to a member's general ledger accounting system.

VROL determines if a cardholder provisional credit through Accounting Entry Memo is required/optional with a dispute, based on the VROL Member Configuration. If Provisional Credit is missing and required/optional, user can create an AEM on an existing case.

## AEM Process

An AEM can be created based on an existing case found during a user's workflow (see [Chapter 17, Queues](#)). Otherwise, a new case can be created (see [Chapter 5, Case Creation](#)). In both these scenarios, VROL returns the VROL Case Number of the case.

1. To submit a new AEM request, user sends request **SISubmitAccountingEntryRequest** with AEM data. This request requires a VROL Case Number from a previous request.  
VROL sends response with VROL Case Number, Accounting Entry ID, and success status.
2. To get AEM details, user sends request **SIGetAccountingEntryDetailsRequest** using VROL Case Number and Accounting Entry ID.  
VROL sends response for all AEM transactions matching the request.
3. To reverse an AEM, user re-sends the original AEM, but reverses the amounts from credit to debit and vice versa.

## AEM Operations

RTSI exposes the following AEM operations:

**SISubmitAccountingEntryRequest**—Used to submit a new AEM request.

**SIGetAccountingEntryDetailsRequest**—Used to request AEM details.

## SISubmitAccountingEntryRequest

SISubmitAccountingEntryRequest is used in request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SISubmitAccountingEntryRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <MemberCaseNumber>a</MemberCaseNumber>
    <AccountingEntry>
      <AccountingEntryMemoNumber>0</AccountingEntryMemoNumber>
      <AccountingEntryCurrency>840</AccountingEntryCurrency>
      <DisputeAmount postingType="CR">0</DisputeAmount>
      <ConsumerAccountNumber>0</ConsumerAccountNumber>
      <IssuerInstitutionID>000000000</IssuerInstitutionID>
      <EntryDate>2017-08-13</EntryDate>
      <EntryType>FINALE</EntryType>
      <ConsumerDescription>a</ConsumerDescription>
      <PostingAmount postingType="CR">1000</PostingAmount>
      <GLEntry>
        <GLType>01</GLType>
        <GLEntryAmount postingType="DR">1000</GLEntryAmount>
        <GLAccount>xxxxxx</GLAccount>
        <GLAccountDescription>Text desc</GLAccountDescription>
      </GLEntry>
      <GLEntry>
        ...
      </GLEntry>
      <Notes>Accounting Entry notes</Notes>
    </AccountingEntry>
  </requestData>
</SISubmitAccountingEntryRequest>
```

### Response Message—SOAP-based Services

```
<SISubmitAccountingEntryResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <AccountingEntryID>100000</AccountingEntryID>
  </responseData>
</SISubmitAccountingEntryResponse>
```

## Request Message—REST-based Services

```
{  
    "RequestHeader": "...",  
    "requestData": {  
        "VisaCaseNumber": "0000000000",  
        "MemberCaseNumber": "a",  
        "AccountingEntry": {  
            "AccountingEntryMemoNumber": 1,  
            "AccountingEntryCurrency": "111",  
            "DisputeAmount": {  
                "postingType": "DR",  
                "$": 1  
            },  
            "ConsumerAccountNumber": 1,  
            "IssuerInstitutionID": "1111111111",  
            "EntryDate": "2016-01-01",  
            "EntryType": "INILDW",  
            "ConsumerDescription": "a",  
            "PostingAmount": {  
                "postingType": "DR",  
                "$": 1  
            },  
            "GLEntry": [  
                {  
                    "GLType": "PA",  
                    "GLEntryAmount": {  
                        "postingType": "CR",  
                        "$": 1  
                    },  
                    "GLAccount": "A",  
                    "GLAccountDescription": "A"  
                },  
                {  
                    "GLType": "WO",  
                    ...  
                },  
                {  
                    "GLAccount": "A",  
                    "GLAccountDescription": "A"  
                }  
            ],  
            "Notes": "A"  
        }  
    }  
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "VisaCaseNumber": "0000000000",  
        "AccountingEntryID": 100000  
    }  
}
```

## **SIGetAccountingEntryDetailsRequest**

SIGetAccountingEntryDetailsRequest is used in request and response messages as shown in the following sample messages.

### **Request Message—SOAP-based Services**

```
<SIGetAccountingEntryDetailsRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <AccountingEntryID>100000</AccountingEntryID>
  </requestData>
</SIGetAccountingEntryDetailsRequest>
```

### **Response Message—SOAP-based Services**

```
<SIGetAccountingEntryDetailsResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <AccountingEntryID>100000</AccountingEntryID>
    <MemberCaseNumber>a</MemberCaseNumber>
    <AccountingEntry>
      <AccountingEntryMemoNumber>1</AccountingEntryMemoNumber>
      <AccountingEntryCurrency>840</AccountingEntryCurrency>
      <DisputeAmount postingType="DR">123.00</DisputeAmount>
      <ConsumerAccountNumber>0</ConsumerAccountNumber>
      <IssuerInstitutionID>0000000000</IssuerInstitutionID>
      <EntryDate>2014-08-18</EntryDate>
      <EntryType>FINALE</EntryType>
      <ConsumerDescription>Test consumer desc
      </ConsumerDescription>
      <PostingAmount postingType="CR">1000.00</PostingAmount>
      <GLEntry>
        <GLType>01</GLType>
        <GLEntryAmount postingType="DR">1000.00</GLEntryAmount>
        <GLAccount>xxxxx</GLAccount>
        <GLAccountDescription>Test desc</GLAccountDescription>
      </GLEntry>
      <GLEntry>
        ...
      </GLEntry>
      <Notes>Accounting Entry notes</Notes>
    </AccountingEntry>
  </responseData>
</SIGetAccountingEntryDetailsResponse>
```

### **Request Message—REST-based Services**

```
{
  "RequestHeader": "...",
  "requestData": {
    "AccountingEntryID": 100000
  }
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "VisaCaseNumber": "0000000000",  
        "AccountingEntryID": 0,  
        "MemberCaseNumber": "a",  
        "AccountingEntry": {  
            "AccountingEntryMemoNumber": 0,  
            "AccountingEntryCurrency": 840,  
            "DisputeAmount": {  
                "@postingType": "DR",  
                "$": 0  
            },  
            "ConsumerAccountNumber": 0,  
            "IssuerInstitutionID": "0000000000",  
            "EntryDate": "2017-08-13",  
            "EntryType": "WRTOFF",  
            "ConsumerDescription": "a",  
            "PostingAmount": {  
                "postingType": "CR",  
                "$": 1000  
            },  
            "GLEntry": [  
                {  
                    "GLType": "01",  
                    "GLEntryAmount": {  
                        "postingType": "DR",  
                        "$": 1000  
                    },  
                    "GLAccount": "xxxxx",  
                    "GLAccountDescription": "Text desc"  
                }, "..."  
            ],  
            "Notes": "Accounting Entry notes"  
        }  
    }  
}
```

THIS PAGE INTENTIONALLY LEFT BLANK.

Queues are lists of cases owned by the user's organization, either cases requiring action from the user or showing actions by the user, and are used to manage a user's workflow. RTSI supports a large subset of the queues available in the VROL UI.

**Note:** *If a user has access to Bulk SI in addition to RTSI, the Bulk SI download can be used in place of the RTSI queue functionality to retrieve transactions from the opposing member.*

Queues correspond to the different dispute stages. There are two types of queues for a dispute stage:

- All queues: All work sent to the user's organization.
- My queues: Work created by, assigned to, or worked on by the user. The user is the owner.

**Important:**

Transactions must be removed from the queue to reduce the number of transactions returned from VROL for each queue request. Once a user has retrieved all the items in a queue, they should take an action on each transaction, either responding or using a request to ignore or close the transaction.

The My Cases queue allows a user to check the cases the user created, was assigned to, or worked on. The cases in this queue do not necessarily require an action by the user.

Custom queues (Custom A...H) can be used to group cases for purposes specific to a member. For example, to associate a queue with a particular Team Lead. These custom queues can be assigned member-specific names in the member's organization settings.

The XML schema elements are based on the queue format, rather than the individual queue. VROL determines the queue format from the queue type element. This information is detailed in the following table. (I/A denotes if the queue is available only to an issuer or an acquirer.)

These queues become available for a user only when they are turned on in the VROL User Profile.

For the description of the elements in the respective queues, see the Queues tab of the IES.

In the Queues tab of the IES, you can view the elements for a particular queue by filtering the "Queue Format" column with the respective queue format and the "Function"

column with the respective queue function. For example, if you want to view the details for "MY\_AWAITING\_ACTION\_DISPUTE" queue, filter the "Queue Format" column with the queue format "D" and filter the "Function" column with "Queue Request" or "Queue Response". The details for this queue are displayed accordingly.

**Table 17–1: Queues**

<b>Category</b>	<b>UI Queue Name</b>	<b>Queue Type</b>	<b>Queue</b>	<b>I/A</b>	<b>For possible workflows, see...</b>
My Cases	My Cases	MY_CASES	S		
	My Saved/Pended	MY_SAVED_PENDED	S		
	My Sent to Cardholder	MY_SENT_TO_CARDHOLDER	S		
	My Sent to Merchant	MY_SENT_TO_MERCHANT	S		
Case Management	Pending Closure	PENDING_CLOSURE	S		
	My Correspondence	MY_CORRESPONDENCE	S		
	All Correspondence	ALL_CORRESPONDENCE	S		
	All Saved/Pended	ALL_SAVED_PENDED	S		
Review	My Reviewed	MY_REVIEWED	T		
	All Reviewed	ALL_REVIEWED	T		
	My Pending Review	MY_PENDING REVIEW	T		
	All Pending Review	ALL_PENDING REVIEW	T		
Rejects	My rejects	MY_REJECTS	R		<ul style="list-style-type: none"> <li>• <a href="#">Chapter 6, Request for Copies</a></li> <li>• <a href="#">Chapter 12, Miscellaneous Fees</a></li> </ul>
	All rejects	ALL_REJECTS	R		
Copy Requests	My Outgoing Copy Requests	MY_OUTGOING_RFC_REQUESTS	T	I	
	All Outgoing Copy Requests	ALL_OUTGOING_RFC_REQUESTS	T	I	
	My Incoming Copy Requests	MY_INCOMING_RFC_REQUESTS	T	A	<a href="#">Chapter 6, Request for Copies</a>
	All Incoming Copy Requests	ALL_INCOMING_RFC_REQUESTS	T	A	
	My RFC Responses	MY_RFC_RESPONSES	T	I	<a href="#">Chapter 6, Request for Copies</a>
	All RFC Responses	ALL_RFC_RESPONSES	T	I	

**Table 17-1: Queues (cont'd.)**

Category	UI Queue Name	Queue Type	Queue	I/A	For possible workflows, see...
Misc Fees	My Misc. Fees	MY_MISC_FEES	T		
	All Misc. Fees	ALL_MISC_FEES	T		
Adjustments	All Incoming Adjustments	ALL_INCOMING_ADJUSTMENTS	T	I	<a href="#">Chapter 9, Adjustments</a>
Case Filing	Un-Submitted Case Filings	UNSUBMITTED_CASE_FILINGS	S		
	My Arb Filings	MY_ARBITRATION	S		
	My Comp Filings	MY_COMPLIANCE	S		
	All Incoming Arb Filings	ALL_INCOMING_ARBITRATION	S		
	All Outgoing Arb Filings	ALL_OUTGOING_ARBITRATION	S		
	All Incoming Comp Filings	ALL_INCOMING_COMPLIANCE	S		
	All Outgoing Comp Filings	ALL_OUTGOING_COMPLIANCE	S		
Request Proof of Posting	My Awaiting Action Request Proof of Postings	MY_AWAITING_ACTION_RPP	T		<a href="#">Chapter 20, Messages</a>
	All Awaiting Action Request Proof of Postings	ALL_AWAITING_ACTION_RPP	T		
	My Submitted Request Proof of Postings	MY_SUBMITTED_RPP	T		
	All Submitted Request Proof of Postings	ALL_SUBMITTED_RPP	T		
Custom	Fwd to CustSrv	FORWARD_TO_CUSTOMER_SERVICE	S		
	Fwd to Fraud	FORWARD_TO_FRAUD	S		
	Custom A...H	CUSTOM_A...CUSTOM_H	S		

**Table 17-1: Queues (cont'd.)**

<b>Category</b>	<b>UI Queue Name</b>	<b>Queue Type</b>	<b>Queue</b>	<b>I/A</b>	<b>For possible workflows, see...</b>
Collaboration	My Awaiting Action Inquiry/GF/ASRR	MY_INCOMING_COLLABORATION	T		<a href="#">Chapter 22, Collaboration</a>
	All Awaiting Action Inquiry/GF/ASRR	ALL_INCOMING_COLLABORATION	T		
	My Submitted Inquiry/GF/ASRR	MY_SUBMITTED_COLLABORATION	T		
	All Submitted Inquiry/GF/ASRR	ALL_SUBMITTED_COLLABORATION	T		
Disputes	My Awaiting Action Disputes	MY_AWAITING_ACTION_DISPUTE <sup>1</sup>	D		<a href="#">Chapter 7, Dispute Questionnaires</a>
	All Awaiting Action Disputes	ALL_AWAITING_ACTION_DISPUTE <sup>1</sup>	D		
	My Submitted Disputes	MY_SUBMITTED_DISPUTE <sup>2</sup>	D		
	All Submitted Disputes	ALL_SUBMITTED_DISPUTE <sup>2</sup>	D		
	My Expired Disputes	MY_EXPIRED_DISPUTE	D		
	All Expired Disputes	ALL_EXPIRED_DISPUTE	D		
	Recalls	RECALLS	D		
	Auto Dispute Saved	AUTO_DISPUTE_SAVED	P		
	All Submitted Rapid Dispute Resolution	ALL_SUBMITTED_RAPID_DISPUTE_RESOLUTION	H	I	
Dispute Quick Forms	My Quick-form Exceptions	MY_QUICKFORM_EXCEPTION	K		<a href="#">Chapter 23, Dispute Quick Forms</a>
	All Quick-form Exceptions	ALL_QUICKFORM_EXCEPTION	K		
Dispute Financials	Dispute Financial	DISPUTE_FINANCIAL	V		<a href="#">Chapter 8, Dispute Financials</a>
Fraud	Fraud Reports Delivered	FRAUD_REPORTS_DELIVERED	Z		<a href="#">Chapter 13, Fraud Reporting</a>

**Table 17-1: Queues (cont'd.)**

Category	UI Queue Name	Queue Type	Queue	I/A	For possible workflows, see...
Visa Fraud Monitoring Program (VFMP)	Visa Fraud Monitoring Program	GMFP_ITEMS	J	I	<a href="#">Chapter 25, Visa Fraud Monitoring Program</a>
Dispute Pre-Filing <sup>3</sup>	My Awaiting Action Pre-Filing	MY_AWAITING_ACTION_PRE_FILING	D		<a href="#">Chapter 10, Dispute Pre-Filings</a>
	All Awaiting Action Pre-Filing	ALL_AWAITING_ACTION_PRE_FILING	D		
	My Submitted Pre-Filing	MY_SUBMITTED_PRE_FILING	D		
	All Submitted Pre-Filing	ALL_SUBMITTED_PRE_FILING	D		
Dispute Acceptances <sup>4</sup>	My Acceptances Sent	MY_ACCEPTANCES_SENT	D		
	All Acceptances Sent	ALL_ACCEPTANCES_SENT	D		
	My Acceptances Received	MY_ACCEPTANCES_RECEIVED	D		
	All Acceptances Received	ALL_ACCEPTANCES_RECEIVED	D		

---

<sup>1</sup> Includes saved disputes, dispute responses, and their associated recalls.

---

<sup>2</sup> Includes submitted disputes, dispute responses, and their associated recalls.

---

<sup>3</sup> Includes dispute Pre-Arbitrations and Pre-Compliances.

---

<sup>4</sup> Includes accepted dispute responses, Pre-Arbitrations, and Pre-Arbitration responses.

To remove transactions from a queue so they are not returned repeatedly, use the following functions. Samples are provided later in this chapter.

- `SICloseRFCRequest` to remove RFC transactions.
- `SICloseMiscFeeRequest` to remove miscellaneous fee transactions.
- `SICloseMessageRequest` to remove message transactions.
- `SICloseTransactionRequest` to close the following transactions and remove them from their respective queues:
  - Transactions that do not qualify for the new dispute resolution process: RFC, miscellaneous fee, message (RPP/Other), collaboration, and adjustment transactions.
  - Transactions that qualify for the new dispute resolution process: Dispute, Dispute response, Pre-Arbitration, Pre-Arbitration response, Pre-Compliance, or Pre-Compliance response transactions.

**Note:** `SICloseTransactionRequest` closes the transactions and removes them from their respective queues but does not close the case.

- `SIgnoreRejectRequest` to remove rejected transactions.
- `SIAcceptDisputeRequest` to accept the dispute liability or to accept the other side's decision, and to remove the dispute transaction from the queue.

## Queue Process

In the UI, two hundred items are displayed per page. In the `SGetQueue` request, the user notes the page number and a maximum of two hundred items is returned per page.

1. User submits a request `SGetQueueRequest` specifying the queue they want to see. By default, the user requests page 1.
2. VROL returns up to 200 items per page from the queue. VROL also returns the page number and the total number of pages.
3. User uses the total number of pages to repeatedly request all remaining pages.

## Queue Actions

The following actions can be taken on the items in a queue. The actions correspond to the buttons in the VROL UI.

Queue Format	UI Button	Actions
K	<ul style="list-style-type: none"> <li>• View Case</li> </ul>	<ul style="list-style-type: none"> <li>• View the case using existing functions or hypersearch.<sup>1</sup></li> </ul>
R	<ul style="list-style-type: none"> <li>• View Case</li> <li>• Ignore</li> <li>• Repair</li> </ul>	<ul style="list-style-type: none"> <li>• View the case using existing functions or hypersearch.<sup>1</sup></li> <li>• Ignore a case to remove it from the queue using <code>SIgnoreRejectRequest</code>.</li> <li>• Repair a rejected transaction by sending it again.</li> </ul>
S	<ul style="list-style-type: none"> <li>• View Case</li> </ul>	<ul style="list-style-type: none"> <li>• View the case using existing functions or hypersearch.<sup>1</sup></li> <li>• Close a Pre-Arbitration, Pre-Arbitration response, Pre-Compliance, or Pre-Compliance response transaction, and remove it from the queue using <a href="#">SICloseTransactionRequest</a>.</li> </ul> <p><b>Note:</b> <i>If a case is closed (automatically by VROL or using UI), VROL automatically closes the Pre-Arbitrations, Pre-Arbitration responses, Pre-Compliances, or Pre-Compliance responses.</i></p>

Queue Format	UI Button	Actions
T	<ul style="list-style-type: none"> <li>View Case</li> <li>Close RFC, Close Misc Fee, Close Message, Close Collaboration</li> </ul>	<ul style="list-style-type: none"> <li>View the case using existing functions or hypersearch.<sup>1</sup></li> <li>Close an RFC, Misc Fee case, Message, Adjustment, or Collaboration transaction and remove it from the queue using <a href="#">SICloseRFCRequest</a>, <a href="#">SICloseMiscFeeRequest</a>, or <a href="#">SICloseMessageRequest</a>.</li> <li>Close a transaction that does not qualify for the new dispute resolution process such as RFC, Misc Fee case, Message, Adjustment, or Collaboration transaction, and remove it from its respective queue using <a href="#">SICloseTransactionRequest</a>.</li> </ul> <p><b>Note:</b> <i>SICloseTransactionRequest closes the transaction and removes it from its respective queue but does not close the case. For more information, see <a href="#">SICloseTransactionRequest</a>.</i></p> <p><b>Note:</b> <i>If a case is closed (automatically by VROL or using UI), VROL automatically closes the RFCs, miscellaneous fees, messages, adjustments, collaborations, or disputes, dispute responses, Pre-Arbitrations, Pre-Arbitration responses, Pre-Compliances, or Pre-Compliance responses.</i></p>
D	<ul style="list-style-type: none"> <li>View Case</li> <li>Accept/Close Dispute</li> <li>Close Dispute</li> </ul>	<ul style="list-style-type: none"> <li>View the case using existing functions or hypersearch.<sup>1</sup></li> <li>Accept and close any of the following items using <a href="#">SIAcceptDisputeRequest</a>: <ul style="list-style-type: none"> <li>The dispute where liability has already been shifted</li> <li>The other side's full or partial acceptance, or decline response</li> <li>The other side's decision to a dispute</li> </ul> </li> <li>Close a transaction that qualifies for the new dispute resolution process such as Dispute, Dispute response, Pre-Arbitration, Pre-Arbitration response, Pre-Compliance, or Pre-Compliance response transaction, and remove it from its respective queue using <a href="#">SICloseTransactionRequest</a>.</li> </ul> <p><b>Note:</b> <i>SICloseTransactionRequest closes the transaction and removes it from its respective queue but does not close the case. For more information, see <a href="#">SICloseTransactionRequest</a>.</i></p>
V	<ul style="list-style-type: none"> <li>View Case</li> </ul>	<ul style="list-style-type: none"> <li>View the case using existing functions or hypersearch.<sup>1</sup></li> </ul>
Z	<ul style="list-style-type: none"> <li>View Case</li> <li>Accepted</li> </ul>	<ul style="list-style-type: none"> <li>View the case using existing functions or hypersearch.<sup>1</sup></li> <li>Accept and change the status of a delivered fraud report from the queue using <a href="#">SIChangeQueueStatusRequest</a>.</li> </ul>
J	<ul style="list-style-type: none"> <li>View Details</li> <li>Ignore</li> </ul>	<ul style="list-style-type: none"> <li>Retrieve details of VFMP-reported transaction using <a href="#">SIGetGMFPDetailsRequest</a> (see <a href="#">Chapter 25, Visa Fraud Monitoring Program</a>).</li> <li>Set the status of a selected transaction to 'Ignored'. The transaction remains in the VFMP queue.</li> </ul>
H	TBD	

<sup>1</sup> Queue response items will contain (Visa case number) and/or (exception ID and exception type). The exception ID and type can be used with existing functions to get details about the transaction in the queue. If Queue does not contain exception ID and type in the response, then the transactions in the case can be found using the Visa Case Number in a hypersearch request (see [Chapter 19, Hypersearch Requests](#)).

## Queue Operations

RTSI exposes the following operations:

**SIGetQueueRequest**—Used by issuers and acquirers to obtain information from their personal work queue (MyQueue).

The sort order of the results returned is the default order in the UI for each queue. Items move out of queues based on each individual queue.

The following table provides information about some XML elements that should be considered when submitting a SIGetQueue request or retrieving a response.

Element	Description
RequestName, RequestData	These elements are required in the request. The value specified in the RequestName element must be "SIGetQueueRequest".
MemberRole	<b>Note:</b> <MemberRole> is ignored in queue requests.
QueueType	This element is required in the request and must specify one of the values shown in <a href="#">Table 17-1</a> .
PageNum	This element is optional in the request. If not specified in the request and MyQueues contains multiple entries, the first entry is returned.
InternalId	Valid for queue format D, S, and T only. In queue response, this returns the ID that is needed to retrieve the transaction details.  For example, <InternalId transactionType="Message"><Id>403291</Id></InternalId> provides the Message ID that can be used in SIGetMessageRequest in <MessageID>.
GMFPSatus	Valid for queue format J only (issuers only).  The issuer can retrieve all but transactions in 'Ignored' status by including all other status values using this element.  Domain values: ADMCH, CAMCH, DSPRQ, GMEX, RPTD, IELG, and GMIG <b>Note:</b> For the definition of each domain value, refer to the IES.
ReportMonth	Valid for queue format J only (issuers only).  If this element is omitted, all VFMP transactions for the past 12 months (including current month) are returned.

### SIGetQueueRequest

SIGetQueue contains request and response messages as shown in the following sample messages.

#### Request Message—SOAP-based Services

This example is for an RFC.

```
<SIGetQueueRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <QueueType>MY_RFC_RESPONSES</QueueType>
    <PageNum>1</PageNum>
  </requestData>
</SIGetQueueRequest>
```

## Response Message—P Format—SOAP-based Services

```
<SIGetQueueResponse>
    <Status>...</Status>
    <responseData>
        <QueueType>AUTO_DISPUTE_SAVED</QueueType>
        <PageInfo>
            <PageNum>1</PageNum>
            <TotalPages>10</TotalPages>
        </PageInfo>
        <Queue>
            <FormatPQueueItem> [up to 200 items per page]
                <VisaCaseNumber>1000000000</VisaCaseNumber>
                <MasterCaseNumber>1000000000</MasterCaseNumber>
                <DisputeJurisdiction jurisdictionCd="DOM">
                    <Jurisdiction>String</Jurisdiction>
                    <IssuerRegion>Europe</IssuerRegion>
                    <AcquirerRegion>AP</AcquirerRegion>
                    <CountryCd>AA</CountryCd>
                </DisputeJurisdiction>
                <ReasonCode>String</ReasonCode>
                <DaysAged>0</DaysAged>
                <CaseAmount currency="000">0</CaseAmount>
                <AccountNumber>0000000000</AccountNumber>
                <PPCCardAccountNumber>0000000000</PPCCardAccountNumber>
                <User>String</User>
                <PendReason code="String">String</PendReason>
                <MemberCaseNumber>a</MemberCaseNumber>
                <ARN>00000000000000000000000000000000</ARN>
                <MerchantName>a</MerchantName>
                <AutoCBDate>1957-08-13</AutoCBDate>
                <QnId>0</QnId>
                <Token>0000000000000000</Token>
            </FormatPQueueItem>
        </Queue>
    </responseData>
<SIGetQueueResponse>
```

**Response Message—R Format—SOAP-based Services**

```
<SIGetQueueResponse>
    <Status>...</Status>
    <ResponseData>
        <QueueType>MY_REJECTS</QueueType>
        <PageInfo>
            <PageNum>1</PageNum>
            <TotalPages>10</TotalPages>
        </PageInfo>
        <Queue>
            <FormatRQueueItem> [up to 200 items per page]
                <VisaCaseNumber>xxxxxxxxxxxx</VisaCaseNumber>
                <RejectedTransactionType type="FRD">Fraud Report
                </RejectedTransactionType>
                <Reason code="xxx"></Reason>
                <DaysAged>number</DaysAged>
                <Amount currency="type">amount</Amount>
                <AccountNumber>xxxxxxxxxxxxxxxxxxxx</AccountNumber>
                <User>jeh0331</User/>
                <RejectID>306839</RejectID>
                <RejectReason code="xxx">Invalid account...</RejectReason>
                <MemberCaseNumber>555555555555</MemberCaseNumber>
                <FraudBundleCaseNumber>1000000000
                </FraudBundleCaseNumber>
                <ARN>xxxxxxxxxxxxxxxxxxxxxxxx</ARN>
                <RetrievalReferenceNumber>810100029004
                </RetrievalReferenceNumber>
                <Token>xxxxxxxxxxxxxxxxxxxx</Token> [for tokenized
transactions only]
                <DateAdded>2016-08-13</DateAdded>
                <ERRequestedInd>true</ERRequestedInd>
                <ParceladoCaseNumber>1000000000</ParceladoCaseNumber>
[Valid for Brazil domestic jurisdiction only]
            </FormatRQueueItem>
        </Queue>
    </ResponseData>
<SIGetQueueResponse>
```

## Response Message—S Format—SOAP-based Services

```
<SIGetQueueResponse>
    <Status>...</Status>
    <responseData>
        <QueueType>MY_CASES</QueueType>
        <PageInfo>
            <PageNum>1</PageNum>
            <TotalPages>10</TotalPages>
        </PageInfo>
        <Queue>
            <FormatSQueueItem> [up to 200 items per page]
                <VisaCaseNumber>xxxxxxxxxx</VisaCaseNumber>
                <FollowUpDate>2016-08-13</FollowUpDate>
                <CaseStatus>Iss Pre-Arb</CaseStatus>
                <ReasonCode>reason code</ReasonCode>
                <ReasonCodeDescription>String</ReasonCodeDescription>
                <DaysToAct>66</DaysToAct>
                <CaseAmount currency="840">600.00</CaseAmount>
                <OriginalTransactionAmount currency="840">34.00
                </OriginalTransactionAmount>
                <AccountNumber>xxxxxxxxxxxxxxxxxxxx</AccountNumber>
                <NetworkID>0002</NetworkID>
                <InternalId transactionType="String">
                    <Id>String</Id>
                </InternalId>
                <MemberCaseNumber>a</MemberCaseNumber>
                <FraudBundleCaseNumber>1000000000</FraudBundleCaseNumber>
                <MemberStatus>String</MemberStatus>
                <ARN>xxxxxxxxxxxxxxxxxxxxxxxxx</ARN>
                <RetrievalReferenceNumber>8000000004
                </RetrievalReferenceNumber>
                <User>jeh0331</User>
                <LastActionDate>2014-09-19T02:55:27+00:00
                </LastActionDate>
                <BIDName>name</BIDName>
                <MerchantName>ANY STORE</MerchantName>
                <MultiFilingCd>M</MultiFilingCd>
                <DisputeJurisdiction jurisdictionCd="REG">
                    <Jurisdiction>REGIONAL-AP</Jurisdiction>
                </DisputeJurisdiction>
                <Token>xxxxxxxxxxxxxxxxxx</Token> [for tokenized
                    transactions only]
                <DateToRespond>2016-08-13</DateToRespond>
                <Action>String</Action>
                <CloseItemID>0</CloseItemID>
                <CloseItemType>MISCFEE</CloseItemType>
                <ParceladoCaseNumber>1000000000</ParceladoCaseNumber>
                    [Valid for Brazil domestic jurisdiction only]
            </FormatSQueueItem>
        </Queue>
    </responseData>
<SIGetQueueResponse>
```

## Response Message—T Format—SOAP-based Services

**Note:** For RPP message queues only, the <MemberRole> field will be returned with the queue response, to distinguish between issuer and acquirer views of the RPP message queue. It is returned for all user IDs, not only for those specified as both issuer and acquirer.

```

<SIGetQueueResponse>
    <Status>...</Status>
    <ResponseData>
        <QueueType>MY_RFC_RESPONSES</QueueType>
        < PageInfo >
            <PageNum>1</PageNum>
            <TotalPages>10</TotalPages>
        </ PageInfo >
        < Queue >
            <FormatTQueueItem> [up to 200 items per page]
                <VisaCaseNumber>xxxxxxxxxxxx</VisaCaseNumber>
                <Action>copy request</Action>
                <CPD>2010-02-15</CPD>
                <CaseStatus>chargeback</CaseStatus>
                <ReasonCode>reason code</ReasonCode>
                <ReasonCodeDescription>string</ReasonCodeDescription>
                <DaysAged>number</DaysAged>
                <CaseAmount currency="840">295.50</CaseAmount>
                <AccountNumber>xxxxxxxxxxxxxxxxxxxx</AccountNumber>
                <MemberCaseNumber>555555555555</MemberCaseNumber>
                <FraudBundleCaseNumber>1000000000</FraudBundleCaseNumber>
                <MemberStatus>status</MemberStatus>
                <ARN>xxxxxxxxxxxxxxxxxxxxxxxxxx</ARN>
                <RetrievalReferenceNumber>800000000004
                </RetrievalReferenceNumber>
                <MerchantName>Any Store</MerchantName>
                <NetworkID>0002</NetworkID>
                <User>jeh0331</User>
                <InternalId transactionType="string">
                    <Id>123456789</Id>
                </InternalId>
                <BIDName>name</BIDName>
                <DisputeJurisdiction jurisdictionCd="INT">
                    <Jurisdiction>Inter-Regional-US,AP</Jurisdiction>
                </DisputeJurisdiction>
                <Token>xxxxxxxxxxxxxxxxxx</Token> [for tokenized transactions only]
                <DateToRespond>2016-08-13</DateToRespond>
                <CloseItemID>0</CloseItemID>
                <CloseItemType>MISCFEE</CloseItemType>
                <ParceladoCaseNumber>1000000000</ParceladoCaseNumber>
[Valid for Brazil domestic jurisdiction only]
            </FormatTQueueItem>
        </ Queue >
    </ ResponseData >
<SIGetQueueResponse>

```

## Response Message—K Format—SOAP-based Services

```
<SIGetQueueResponse>
  <Status>...</Status>
  <ResponseData>
    <QueueType>MY_QUICKFORM_EXCEPTION</QueueType>
    <PageInfo>
      <PageNum>1</PageNum>
      <TotalPages>10</TotalPages>
    </PageInfo>
    <Queue>
      <FormatKQueueItem> [up to 200 items per page]
        <QuickFormID>0</QuickFormID>
        <Action>Dispute</Action>
        <Source>RTSI</Source>
        <RequestDate>2016-08-13</RequestDate>
        <AccountNumber>000000000000</AccountNumber>
        <Amount currency="000">0</Amount>
        <DaysAged>0</DaysAged>
        <ARN>00000000000000000000000000000000</ARN>
        <User>String</User>
        <TransactionID>a</TransactionID>
        <ErrorCondition>String</ErrorCondition>
      </FormatKQueueItem>
    </Queue>
  </ResponseData>
<SIGetQueueResponse>
```

**Response Message—D Format—SOAP-based Services**

```
<SIGetQueueResponse>
    <Status>...</Status>
    <ResponseData>
        <QueueType>MY_AWAITING_ACTION_DISPUTE</QueueType>
        < PageInfo >
            <PageNum>1</PageNum>
            <TotalPages>10</TotalPages>
        </ PageInfo >
        <Queue>
            <FormatDQueueItem> [up to 200 items per page]
                <VisaCaseNumber>1000000000</VisaCaseNumber>
                <FollowUpDate>2016-08-13</FollowUpDate>
                <CaseStatus>String</CaseStatus>
                <DaysToAct>0</DaysToAct>
                <DateToRespond>2016-08-13</DateToRespond>
                <DisputeAmt currency="000">0</DisputeAmt>
                <AccountNumber>000000000000</AccountNumber>
                <Token>0000000000000000</Token>
                <MemberCaseNumber>a</MemberCaseNumber>
                <FraudBundleCaseNumber>1000000000</FraudBundleCaseNumber>
                <MemberStatus>a</MemberStatus>
                <ARN>00000000000000000000000000000000</ARN>
                <RetrievalReferenceNumber>a</RetrievalReferenceNumber>
                <User>String</User>
                <LastActionDate>2016-12-17T09:30:47Z</LastActionDate>
                <BIDName>String</BIDName>
                <MerchantName>String</MerchantName>
                <NetworkID>aaa</NetworkID>
                <Jurisdiction jurisdictionCd="REG">
                    <Jurisdiction>String</Jurisdiction>
                </Jurisdiction>
                <InternalId transactionType="String">
                    <Id>String</Id>
                </InternalId>
                <DCCode>String</DCCode>
                <DCCodeDescription>String</DCCodeDescription>
                <ParceladoCaseNumber>1000000000</ParceladoCaseNumber>
                [Valid for Brazil domestic jurisdiction only]
            </FormatDQueueItem>
        </Queue>
    </ResponseData>
<SIGetQueueResponse>
```

## Response Message—J Format—SOAP-based Services

```
<SIGetQueueResponse>
  <Status>...</Status>
  <responseData>
    <QueueType>GMFP_ITEMS</QueueType>
    <PageInfo>
      <PageNum>1</PageNum>
      <TotalPages>1</TotalPages>
    </PageInfo>
    <Queue>
      <FormatJQueueItem> [up to 200 items per page]
        <GMFPItemID>100000</GMFPItemID>
        <GMFPStatus>RPTD</GMFPStatus>
        <DateAdded>2015-10-01</DateAdded>
        <MerchantName>Any Merchant</MerchantName>
        <MerchantCity>BEIJING</MerchantCity>
        <MerchantCountryCode>CN</MerchantCountryCode>
        <AccountNumber>xxxxxxxxxxxxxxxxxxxx</AccountNumber>
        <OrigTranCPD>2015-04-15</OrigTranCPD>
        <DestinationAmount currency="840">500.00
        </DestinationAmount>
        <TransactionID>a</TransactionID>
        <ARN>xxxxxxxxxxxxxxxxxxxxxxxxx</ARN>
        <ECMOTO>N</ECMOTO>
        <PosEntryModeCd>02</PosEntryModeCd>
        <FraudPostDate>2015-10-01</FraudPostDate>
        <DaysToAct>1</DaysToAct>
        <DateToRespond>2016-08-13</DateToRespond>
        <Token>xxxxxxxxxxxxxxxxxxxx</Token> [for tokenized
        transactions only]
        <BIDName>name</BIDName>
      </FormatJQueueItem>
    </Queue>
  </responseData>
</SIGetQueueResponse>
```

## Response Message—V Format—SOAP-based Services

```
<SIGetQueueResponse>
  <Status>...</Status>
  <ResponseData>
    <QueueType>DISPUTE_FINANCIAL</QueueType>
    <PageInfo>
      <PageNum>1</PageNum>
      <TotalPages>0</TotalPages>
    </PageInfo>
    <Queue>
      <FormatVQueueItem> [up to 200 items per page]
        <CPD>2017-08-13</CPD>
        <DisputeStatus>F1</DisputeStatus>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <MemberCaseNumber>a</MemberCaseNumber>
        <DisputeAmt currency="000">0</DisputeAmt>
        <NetworkID>aaa</NetworkID>
        <SubmittedBy>Client</SubmittedBy>
        <IncomingOutgoing>Outgoing</IncomingOutgoing>
        <MsgType>TC25</MsgType>
      </FormatVQueueItem>
    </Queue>
  </ResponseData>
</SIGetQueueResponse>
```

## Response Message—Z Format—SOAP-based Services

```
<SIGetQueueResponse>
  <Status>...</Status>
  <responseData>
    <QueueType>FRAUD_REPORTS_DELIVERED</QueueType>
    <PageInfo>
      <PageNum>1</PageNum>
      <TotalPages>0</TotalPages>
    </PageInfo>
    <Queue>
      <FormatLQueueItem> [up to 200 items per page]
        <AccountNumber>0000000000</AccountNumber>
        <CPD>2017-08-13</CPD>
        <Amount currency="000">0</Amount>
        <MemberCaseNumber>a</MemberCaseNumber>
        <ARN>00000000000000000000000000000000</ARN>
        <RetrievalReferenceNumber>a</RetrievalReferenceNumber>
        <MerchantName>a</MerchantName>
        <NetworkID>aaa</NetworkID>
        <MCC>0000</MCC>
        <FraudReportDeliveredDate>2017-08-13
        </FraudReportDeliveredDate>
        <TransactionID>a</TransactionID>
        <AcquirerBIN>000000</AcquirerBIN>
        <SystemTraceAuditNumber>a</SystemTraceAuditNumber>
        <FraudType>0</FraudType>
        <NotificationCd>a</NotificationCd>
        <FraudChannel>String</FraudChannel>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <Status>P</Status>
      </FormatLQueueItem>
    </Queue>
  </responseData>
</SIGetQueueResponse>
```

**Response Message—H Format—SOAP-based Services**

```
<SIGetQueueResponse>
    <Status>...</Status>
    <ResponseData>
        <QueueType>ALL_SUBMITTED_RAPID_DISPUTE_RESOLUTION</QueueType>
        < PageInfo >
            <PageNum>1</PageNum>
            <TotalPages>1</TotalPages>
        </ PageInfo >
        < Queue >
            <FormatHQueueItem> [up to 200 items per page]
                <AccountNumber>000000000000</AccountNumber>
                <Token>0000000000000000</Token>
                <DisputeCategoryCondition>String</DisputeCategoryCondition>
                <DisputeAmt>0</DisputeAmt>
                <MemberCaseNumber>a</MemberCaseNumber>
                <InternalId transactionType="String">
                    <Id>String</Id>
                </InternalId>
                <MerchantName>a</MerchantName>
                <VisaCaseNumber>1000000000</VisaCaseNumber>
                <RdrSubmissionTime>2018-12-17T09:30:47Z</RdrSubmissionTime>
                <RdrExpiryTime>2018-12-17T09:30:47Z</RdrExpiryTime>
                <RdrStatus>Resolved</RdrStatus>
            </FormatHQueueItem>
        </ Queue >
    </ResponseData>
</SIGetQueueResponse>
```

## Request Message—REST-based Services

```
{  
    "RequestHeader": "...",  
    "requestData": {  
        "QueueType": "MY_RFC_RESPONSES",  
        "PageNum": 1,  
    }  
}
```

## Response Message—P Format—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "QueueType": "AUTO_DISPUTE_SAVED",  
        "PageInfo": {  
            "PageNum": 1,  
            "TotalPages": 1  
        },  
        "Queue": {  
            "FormatPQueueItem": {  
                "VisaCaseNumber": 1000000000,  
                "MasterCaseNumber": 1000000000,  
                "DisputeJurisdiction": {  
                    "@jurisdictionCd": "DOM",  
                    "Jurisdiction": "String",  
                    "IssuerRegion": "Europe",  
                    "AcquirerRegion": "AP",  
                    "CountryCd": "AA"  
                },  
                "ReasonCode": "String",  
                "DaysAged": 0,  
                "CaseAmount": {  
                    "@currency": "000",  
                    "$": 0  
                },  
                "AccountNumber": "000000000000",  
                "PPCCardAccountNumber": "000000000000",  
                "User": "String",  
                "PendReason": {  
                    "@code": "String",  
                    "$": "String"  
                },  
                "MemberCaseNumber": "a",  
                "ARN": "00000000000000000000000000000000",  
                "MerchantName": "a",  
                "AutoCBDate": "1957-08-13",  
                "QnId": 0,  
                "Token": "0000000000000000"  
            },  
        }  
    }  
}
```

**Response Message—R Format—REST-based Services**

```
{  
    "Status": "...",  
    "responseData": {  
        "QueueType": "MY_REJECTS",  
        "PageInfo": {  
            "PageNum": 1,  
            "TotalPages": 1  
        },  
        "Queue": {  
            "FormatRQueueItem": {  
                "VisaCaseNumber": "0000000000",  
                "RejectedTransactionType": {  
                    "type": "A",  
                    "Text": "A"  
                },  
                "Reason": {  
                    "code": "A",  
                    "Text": "A"  
                },  
                "DaysAged": 1,  
                "Amount": {  
                    "currency": "111",  
                    "Text": 1  
                },  
                "AccountNumber": "1111111111",  
                "User": "A",  
                "RejectID": 1000000,  
                "RejectReason": {  
                    "code": "A",  
                    "Text": "A"  
                },  
                "MemberCaseNumber": "a",  
                "ARN": "111111111111111111111111",  
                "RetrievalReferenceNumber": "a",  
                "Token": "111111111111",  
                "DateAdded": "2016-01-01",  
                "ERRequestedInd": "true",  
                "ParceladoCaseNumber": 1000000000 [Valid for Brazil domestic  
                jurisdiction only]  
            },  
        }  
    }  
}
```

## Response Message—S Format—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "queueType": "MY_CASES",  
        "pageInfo": {  
            "pageNum": 1,  
            "totalPages": 1  
        },  
        "queue": {  
            "formatSQueueItem": {  
                "visaCaseNumber": "0000000000",  
                "followUpDate": "2016-08-13",  
                "caseStatus": "A",  
                "reasonCode": "A",  
                "reasonCodeDescription": "A",  
                "daysToAct": 1,  
                "caseAmount": {  
                    "currency": "111",  
                    "text": 1  
                },  
                "originalTransactionAmount": {  
                    "currency": "111",  
                    "text": 1  
                },  
                "accountNumber": "1111111111",  
                "networkID": "aaa",  
                "internalId": {  
                    "@transactionType": "String",  
                    "id": "String"  
                },  
                "memberCaseNumber": "a",  
                "fraudBundleCaseNumber": 1000000000,  
                "memberStatus": "String",  
                "arn": "111111111111111111111111",  
                "retrievalReferenceNumber": "a",  
                "user": "A",  
                "lastactionDate": "2016-01-01T19:31:47Z",  
                "bidName": "A",  
                "merchantName": "A",  
                "disputeJurisdiction": {  
                    "jurisdictionCd": "DOM",  
                    "jurisdiction": "A",  
                },  
                "token": "11111111111111",  
                "dateToRespond": "2016-08-13",  
                "closeItemID": 0,  
                "closeItemType": "MISCfee",  
                "parceladoCaseNumber": 1000000000 [Valid for Brazil domestic jurisdiction only]  
            },  
        }  
    }  
}
```

**Response Message—T Format—REST-based Services**

```
{  
    "Status": "...",  
    "responseData": {  
        "QueueType": "MY_RFC_RESPONSES",  
        "PageInfo": {  
            "PageNum": 1,  
            "TotalPages": 1  
        },  
        "Queue": {  
            "FormatTQueueItem": {  
                "VisaCaseNumber": "0000000000",  
                "ReviewStatus": "A",  
                "Action": "A",  
                "CPD": "2016-01-01",  
                "CaseStatus": "A",  
                "ReasonCode": "A",  
                "ReasonCodeDescription": "A",  
                "DaysAged": 1,  
                "CaseAmount": {  
                    "currency": "111",  
                    "Text": 1  
                },  
                "AccountNumber": "111111111111",  
                "MemberCaseNumber": "a",  
                "FraudBundleCaseNumber": 1000000000,  
                "MemberStatus": "A",  
                "ARN": "11111111111111111111111111",  
                "RetrievalReferenceNumber": "a",  
                "MerchantName": "A",  
                "NetworkID": "aaa",  
                "User": "A",  
                "InternalId": {  
                    "transactionType": "A",  
                    "Id": "A"  
                },  
                "BIDName": "A",  
                "DisputeJurisdiction": {  
                    "jurisdictionCd": "DOM",  
                    "Jurisdiction": "A",  
                },  
                "Token": "111111111111",  
                "DateToRespond": "2016-08-13",  
                "CloseItemID": 0,  
                "CloseItemType": "MISCFEE",  
                "ParceladoCaseNumber": 1000000000 [Valid for Brazil domestic  
                jurisdiction only]  
            },  
        }  
    }  
}
```

## Response Message—D Format—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "QueueType": "MY_AWAITING_ACTION_DISPUTE",  
        "PageInfo": {  
            "PageNum": 1,  
            "TotalPages": 1  
        },  
        "Queue": {  
            "FormatDQueueItem": {  
                "VisaCaseNumber": "0000000000",  
                "FollowUpDate": "2016-01-01",  
                "CaseStatus": "A",  
                "DaysToAct": 1,  
                "DateToRespond": "2016-08-13",  
                "DisputeAmt": {  
                    "@currency": "000",  
                    "Text": 0  
                },  
                "AccountNumber": "111111111111",  
                "Token": "11111111111111",  
                "MemberCaseNumber": "a",  
                "FraudBundleCaseNumber": 1000000000,  
                "MemberStatus": "A",  
                "ARN": "11111111111111111111111111",  
                "RetrievalReferenceNumber": "a",  
                "User": "A",  
                "LastactionDate": "2016-01-01T19:31:47Z",  
                "BIDName": "A",  
                "MerchantName": "A",  
                "NetworkID": "aaa",  
                "Jurisdiction": {  
                    "jurisdictionCd": "REG",  
                    "Jurisdiction": "String",  
                },  
                "InternalId": {  
                    "transactionType": "A",  
                    "Id": "A"  
                },  
                "DCCode": "A",  
                "DCCodeDescription": "A",  
                "ParceladoCaseNumber": 1000000000 [Valid for Brazil domestic jurisdiction only]  
            }  
        }  
    }  
}
```

**Response Message—J Format—REST-based Services**

```
{  
    "Status": "...",  
    "responseData": {  
        "queueType": "GMFP_ITEMS",  
        "pageInfo": {  
            "pageNum": 1,  
            "totalPages": 1  
        },  
        "queue": {  
            "formatJQueueItem": {  
                "GMFPItemID": 1,  
                "GMFPStatus": "DSPRQ",  
                "DateAdded": "2016-01-01",  
                "MerchantName": "Any Merchant",  
                "MerchantCity": "BEIJING",  
                "MerchantCountryCode": "CN",  
                "AccountNumber": "111111111111",  
                "OrigTranCPD": "2016-01-01",  
                "destinationAmount": {  
                    "currency": "111",  
                    "text": 1  
                },  
                "transactionID": "a",  
                "ARN": "111111111111111111111111",  
                "ECMOTO": "A",  
                "PosEntryModeCd": "A",  
                "FraudPostDate": "2016-01-01",  
                "DaysToAct": 1,  
                "DateToRespond": "2016-08-13",  
                "Token": "111111111111",  
                "BIDName": "A"  
            },  
        }  
    }  
}
```

### Response Message—V Format—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "QueueType": "DISPUTE_FINANCIAL",  
        "PageInfo": {  
            "PageNum": 1,  
            "TotalPages": 0  
        },  
        "Queue": {  
            "FormatVQueueItem": {  
                "CPD": "2017-08-13",  
                "DisputeStatus": "F1",  
                "VisaCaseNumber": 1000000000,  
                "MemberCaseNumber": "a",  
                "DisputeAmt": {  
                    "@currency": "000",  
                    "$": 0  
                },  
                "NetworkID": "aaa",  
                "SubmittedBy": "Client",  
                "IncomingOutgoing": "Outgoing",  
                "MsgType": "TC25"  
            }  
        }  
    }  
}
```

## Response Message—Z Format—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "queueType": "FRAUD_REPORTS_DELIVERED",  
        "pageInfo": {  
            "pageNum": 1,  
            "totalPages": 0  
        },  
        "queue": {  
            "formatLQueueItem": {  
                "accountNumber": "000000000000",  
                "CPD": "2017-08-13",  
                "amount": {  
                    "@currency": "000",  
                    "$": 0  
                },  
                "memberCaseNumber": "a",  
                "ARN": "000000000000000000000000",  
                "retrievalReferenceNumber": "a",  
                "merchantName": "a",  
                "networkID": "aaa",  
                "MCC": "0000",  
                "fraudReportDeliveredDate": "2017-08-13",  
                "transactionID": "a",  
                "acquirerBIN": "000000",  
                "systemTraceAuditNumber": "a",  
                "FraudType": 0,  
                "notificationCd": "a",  
                "FraudChannel": "String",  
                "VisaCaseNumber": 1000000000,  
                "status": "P"  
            }  
        }  
    }  
}
```

## Response Message—K Format—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "QueueType": "MY_QUICKFORM_EXCEPTION",  
        "PageInfo": {  
            "PageNum": 1,  
            "TotalPages": 0  
        },  
        "Queue": {  
            "FormatKQueueItem": {  
                {  
                    "QuickFormID": 0,  
                    "Action": "Dispute",  
                    "Source": "RTSI",  
                    "RequestDate": "1957-08-13",  
                    "AccountNumber": "000000000000",  
                    "Amount": {  
                        "@currency": "000",  
                        "$": 0  
                    },  
                    "DaysAged": 0,  
                    "ARN": "000000000000000000000000000000",  
                    "User": "String",  
                    "TransactionID": "a",  
                    "ErrorCondition": "String"  
                }  
            }  
        }  
    }  
}
```

## Response Message—H Format—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "QueueType": "ALL_SUBMITTED_RAPID_DISPUTE_RESOLUTION",  
        "PageInfo": {  
            "PageNum": 1,  
            "TotalPages": 0  
        },  
        "Queue": {  
            "FormatHQueueItem": {  
                {  
                    "AccountNumber": "000000000000",  
                    "Token": "00000000000000",  
                    "DisputeCategoryCondition": "Test",  
                    "DisputeAmt": {  
                        "@currency": "000",  
                        "$": 0  
                    },  
                    "MemberCaseNumber": "a",  
                    "InternalId": {  
                        "@transactionType": "String",  
                        "Id": "String"  
                    },  
                    "MerchantName": "a",  
                    "VisaCaseNumber": 1000000000,  
                    "RdrSubmissionTime": 2018-12-17T09:30:47Z,  
                    "RdrExpiryTime": 2018-12-17T09:30:47Z,  
                    "RdrStatus": "Resolved"  
                }  
            }  
        }  
    }  
}
```

## Remove Transactions From Queue

A member should remove transactions from its queue to prevent the transaction from being returned again in the next queue request. This can be particularly important to reduce the bandwidth used to retrieve queue data.

### SICloseRFCRequest

SICloseRFCRequest is used in request and response messages for the following queue names (queue format T):

- My/All Incoming Copy Requests
- My/All Outgoing Copy Requests
- My/All RFC Responses

### Request Message—SOAP-based Services

```
<SICloseRFCRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <RFCID>0000000</RFCID>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
  </requestData>
</SICloseRFCRequest>
```

### Response Message—SOAP-based Services

```
<SICloseRFCResponse>
  <Status>...</Status>
  <responseData>
    <RFCID>0000000</RFCID>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
  </responseData>
</SICloseRFCResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData": {
    "RFCID": 1,
    "VisaCaseNumber": 1111111111
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": {
    "Code": "a",
    "Message": "a",
    "ErrorIn": "A"
  },
  "responseData": {
    "RFCID": 1,
    "VisaCaseNumber": 1111111111
  }
}
```

## SICloseMiscFeeRequest

SICloseMiscFeeRequest is used in request and response messages for the following queue names (queue format T):

- My Misc. Fees
- All Misc. Fees

### Request Message—SOAP-based Services

```
<SICloseMiscFeeRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <MiscFeeID>1000000</MiscFeeID>
    </requestData>
</SICloseMiscFeeRequest>
```

### Response Message—SOAP-based Services

```
<SICloseMiscFeeResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <MiscFeeID>1000000</MiscFeeID>
    </responseData>
</SICloseMiscFeeResponse>
```

### Request Message—REST-based Services

```
{
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": "0000000000",
        "MiscFeeID": 1000000
    }
}
```

### Response Message—REST-based Services

```
{
    "Status": "...",
    "responseData": {
        "VisaCaseNumber": "0000000000",
        "MiscFeeID": 1000000,
    }
}
```

## SICloseMessageRequest

SICloseMessageRequest will remove the message from the following My/All queues:

- My/All Awaiting Action Request Proof of Postings
- My/All Submitted Request Proof of Postings

**Note:** As the RPP message queues are different when viewed by issuer or acquirer in a dual organization, the role of the user ID is used to determine whether this affects the issuer or acquirer version of the queue.

**Note:** For a dual user, <MemberRole> is required in the RTSI header, <RealTimeSIRequest>. For the SICloseMessageRequest, this will be used by VROL to distinguish between issuer and acquirer views of the RPP message queue.

### Request Message—RPP—SOAP-based Services

```
<SICloseMessageRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <MessageID>0000000</MessageID>
        <MessageType>RPP</MessageType>
    </requestData>
</SICloseMessageRequest>
```

### Response Message—RPP—SOAP-based Services

```
<SICloseMessageResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <MessageID>0000000</MessageID>
        <MessageType>RPP</MessageType>
    </responseData>
</SICloseMessageResponse>
```

### Request Message—RPP—REST-based Services

```
{
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": "0000000000",
        "MessageID": "0000000",
        "MessageType": "RPP"
    }
}
```

### Response Message—RPP—REST-based Services

```
{
    "Status": "...",
    "responseData": {
        "VisaCaseNumber": "0000000000",
        "MessageID": "0000000",
        "MessageType": "DENO",
        "MemberRole": "A"
    }
}
```

## SICloseTransactionRequest

SICloseTransaction is used to close the following transactions and remove them from their respective queues (queue format S, T, and D):

- Transactions that do not qualify to the new dispute resolution process: RFC, miscellaneous fee, message (RPP/Other), collaboration, and adjustment transactions.
- Transactions that qualify to the new dispute resolution process: Dispute, dispute response, Pre-Arbitration, Pre-Arbitration response, Pre-Compliance, or Pre-Compliance response transactions.

**Note:** *SICloseTransactionRequest closes the transaction and removes it from its respective queue but does not close the case.*

The following elements should be considered when closing a transaction.

Element	Description
CloseItemID	Uniquely identifies the transaction being closed.
CloseItemType	Indicates the type of transaction being closed.  Domain values: <ul style="list-style-type: none"> <li>• For Format S: PREARB, PREARBRESP, PRECOMP, and PRECOMPRESP</li> <li>• For Format T: RFC, MISCFEE, RPP, ADJ, COLLABORATION, OTHER, PREARB, PREARBRESP, PRECOMP, and PRECOMPRESP</li> <li>• For Format D: DISPREQ, DISPRES, PREARB, PREARBRESP, PRECOMP, and PRECOMPRESP</li> </ul>

### Request Message—SOAP-based Services

```
<SICloseTransactionRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <CloseItemID>0000000</CloseItemID>
        <CloseItemType>Collaboration</CloseItemType>
    </requestData>
</SICloseTransactionRequest>
```

### Response Message—SOAP-based Services

```
<SICloseTransactionResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <CloseItemID>0000000</CloseItemID>
        <CloseItemType>Collaboration</CloseItemType>
    </responseData>
</SICloseTransactionResponse>
```

### Request Message—REST-based Services

```
{
    "Status": "...",
    "responseData": {
        "VisaCaseNumber": "0000000000",
        "CloseItemID": "0000000",
        "CloseItemType": "Collaboration"
    }
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData":{  
        "VisaCaseNumber": "0000000000",  
        "CloseItemID": "0000000",  
        "CloseItemType": "Collaboration"  
    }  
}
```

## SIAcceptDisputeRequest

SIAcceptDisputeRequest is used for the following:

- To accept dispute request where the liability has already been shifted and remove the dispute transaction from the queue.
- When a member agrees to accept the other side's response. When the other side sends back the full or partial acceptance or decline response to an incoming request and the decision is acceptable to the current side, this Web service can be used to just accept the decision and remove the dispute transaction from the queue.

**Note:**

- *Optionally, the user can use [SICloseTransactionRequest](#) to close a dispute transaction.*
- *The side accepting the decision will not be able to take the dispute to the next phase after accepting the decision.*

VROL returns a SIAcceptDisputeResponse message with VisaCaseNumber, AcceptItemID, and the financial fields or an indication that VROL generated a financial (when applicable).

See [Financial Fields Returned in Immediate Response](#) for a list of financial fields returned in the response.

For example, an issuer submits a Consumer dispute, the acquirer sends a decline response and provides evidence for declining. The issuer reviews the decline response and evidence, and submits SIAcceptDisputeRequest to agree with the acquirer.

**Note:** *For a dual user, <MemberRole> is required in the RTSI header, <RealTimeSIRequest>.*

The following elements should be considered when accepting a dispute or when receiving a response from this service.

Element	Description
AcceptItemID	Uniquely identifies the dispute item being accepted. AcceptItemID includes the ID value for the corresponding dispute item (DisputeId for dispute, DisputeRespsnId for Dispute Response, DisputePreArbId for Pre-Arbitration, and DisputePreArbRespsnId for Pre-Arbitration response).
AcceptItemType	Indicates the type of dispute being accepted. Domain values: <ul style="list-style-type: none"> <li>• DISPREQ=Dispute</li> <li>• DISPRESP=Dispute Response</li> <li>• PREARB=Pre-Arbitration</li> <li>• PREARBRESP=Pre-Arbitration Response</li> <li>• PRECOMP=Pre-Compliance</li> <li>• PRECOMPRESP=Pre-Compliance Response</li> </ul>

**Note:** Members can retrieve the Accept Item details from a Hypersearch response using the VisaCaseNumber. See [Chapter 19, Hypersearch Requests](#) for details.

### Request Message—SOAP-based Services

```
<SIAcceptDisputeRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <AcceptItemID>11111</AcceptItemID>
        <AcceptItemType>PREARBRESP</AcceptItemType>
    </requestData>
</SIAcceptDisputeRequest>
```

### Response Message—SOAP-based Services

```
<SIAcceptDisputeResponse>
    <Status>...</Status>
    <ResponseData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <AcceptItemID>11111</AcceptItemID>
        <DisputeFinancialInfo>
            <VROLFinancialSentInd>true</VROLFinancialSentInd>
            <VROLFinancialID>111111111111</VROLFinancialID>
            <FinancialTransactionType>String</FinancialTransactionType>
            <FinancialAmt currency="840">100</FinancialAmt>
            <DisputeCategory>12</DisputeCategory>
            <DisputeCondition>2</DisputeCondition>
            <DisputeId>1234567</DisputeId>
            <VROLBundleCaseNumber>1000000000</VROLBundleCaseNumber>
            <TransactionID>1234567</TransactionID>
            <IssuerCaseNumber>1111111111</IssuerCaseNumber>
            <AcquirerCaseNumber>1111111112</AcquirerCaseNumber>
            <OpposerExpectedRespDate>2016-01-01</OpposerExpectedRespDate>
        </DisputeFinancialInfo>
    </ResponseData>
</SIAcceptDisputeResponse>
```

### Request Message—REST-based Services

```
{
    "RequestHeader": "...",
    "requestData": {
        "AcceptItemType": "DISPREQ",
        "AcceptItemID": 11111,
        "VisaCaseNumber": 1000000000
    }
}
```

### Response Message—REST-based Services

```
{
    "Status": "...",
    "responseData": {
        "VisaCaseNumber": 1000000000,
        "AcceptItemID": 11111
    }
}
```

## Ignore Rejected Transactions

RTSI enables a user to ignore a rejected transaction (queue format R) for the following transactions as determined by the RejectedTransactionType.

RejectedTransactionType	Type Attribute
Request For Copy	RFC
Adjustment	ADJM

RejectedTransactionType	Type Attribute
Fee Collection	FC
Funds Disbursement	FD
Fraud	FRD
Dispute Request	DISPREQ
Dispute Response	DISPRESP
Pre-Arbitration	PREARB
Pre-Arbitration Response	PREARBRESP
Pre-Compliance	PRECOMP
Pre-Compliance Response	PRECOMPRESP
Arbitration	ARB
Compliance	COMP

## Ignore Reject Operation

RTSI exposes the following ignore reject operation.

**SIIgnoreRejectRequest**—Used to ignore a transaction reject and prevent it from being returned again in the next queue request.

## SIIgnoreRejectRequest

SIIgnoreRejectRequest is used to clear transactions from the following queues (queue format R):

- My Rejects
- All Rejects

### Request Message—SOAP-based Services

```
<SIIgnoreRejectRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <RejectID>1000000</RejectID>
    <RejectedTransactionType type="FRD">Fraud Report
    </RejectedTransactionType>
  </requestData>
</SIIgnoreRejectRequest>
```

### Response Message—SOAP-based Services

```
<SIIgnoreRejectResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <RejectID>1000000</RejectID>
  </responseData>
</SIIgnoreRejectResponse>
```

### Request Message—REST-based Services

```
{  
    "RequestHeader": "...",  
    "requestData":{  
        "VisaCaseNumber": "0000000000",  
        "RejectID": 1000000,  
        "RejectedTransactionType":{  
            "type": "FRD",  
            "Text": "Fraud Report"  
        }  
    }  
}
```

### Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData":{  
        "VisaCaseNumber": "0000000000",  
        "RejectID": 1000000,  
        "RejectedTransactionType":{  
            "type": "FRD",  
            "Text": "Fraud Report"  
        }  
    }  
}
```

## SIIgnoreGMFPRRequest

Once the issuer has reviewed each VFMP transaction in the queue, the issuer should change the status of the VFMP transaction to 'Ignored' by sending request SIIgnoreGMFPRRequest using GMFPItemID.

**Note:** *SIIgnoreGMFPRRequest is different from other queue "ignore/close" actions as the ignored VFMP transaction can still be retrieved using a queue request.*

### Request Message—SOAP-based Services

```
<SIIgnoreGMFPRRequest>  
    <RequestHeader>  
        <User id="a" type="internalId"/>  
        <MemberRole>I</MemberRole>  
    </RequestHeader>  
    <requestData>  
        <GMFPItemID>1234567</GMFPItemID>  
    </requestData>  
</SIIgnoreGMFPRRequest>
```

### Response Message—SOAP-based Services

```
<SIIgnoreGMFPResponse>
  <Status>
    <Code>a</Code>
    <Message>a</Message>
    <ErrorIn>String</ErrorIn>
  </Status>
  <responseData>
    <GMFPItemID>1234567</GMFPItemID>
  </responseData>
</SIIgnoreGMFPResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": {
    "User": {
      "@id": "a",
      "@type": "internalId"
    },
    "MemberRole": "I"
  },
  "requestData": {
    "GMFPItemID": 1234567
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": [
    {
      "Code": "a",
      "Message": "a",
      "ErrorIn": "String"
    }
  ],
  "responseData": {
    "GMFPItemID": 1234567
  }
}
```

## Change Queue Status

RTSI enables a user to accept and change the status of a delivered fraud report transaction from the queue (queue format Z).

### Change Queue Status Operation

RTSI exposes the following change queue status operation.

**SIChangeQueueStatusRequest**—Used to change the status of a delivered fraud report.

## SIChangeQueueStatusRequest

SIChangeQueueStatusRequest is used to accept and change the status of a fraud report from the Fraud Reports Delivered queue (queue format Z). This function also includes the required <Status> element, which indicates the new status you want to set for the fraud report. Refer to the IES (Queue Actions tab) for a list of all the possible domain values for <Status>.

### Request Message—SOAP-based Services

```
<SIChangeQueueStatusRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <TransactionID>a</TransactionID>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <Status>C</Status>
  </requestData>
</SIChangeQueueStatusRequest>
```

### Response Message—SOAP-based Services

```
<SIChangeQueueStatusResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <Status>A</Status>
  </responseData>
</SIChangeQueueStatusResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData":{
    "TransactionID": "a",
    "VisaCaseNumber": 1000000000,
    "Status": "C"
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": "...",
  "responseData":{
    "VisaCaseNumber": 1000000000,
    "Status": "A"
  }
}
```

THIS PAGE INTENTIONALLY LEFT BLANK.

# RTSI Batch Queues

18

*RTSI Batch Queues* are lists of cases or transactions that are owned by a user's organization and that require action from the user. These queues are used to manage a user's workflow. RTSI Batch Queues support a subset of the queues available in the VROL UI. They are only available for queues with incoming items.

**Note:** *If a user has access to Bulk SI in addition to RTSI, the Bulk SI download can be used in place of the RTSI queue functionality to retrieve transactions from the opposing member. The RTSI Batch queues enable members to retrieve the queue items on demand in addition to the scheduled Bulk SI download.*

Queues correspond to the different dispute stages. Many of these queues are used in the workflows included in other chapters in this document. The VROL queues supported by the RTSI Batch Queues functionality only contain transactions to which the user has work access.

The XML schema elements are based on the queue format, rather than the individual queue. VROL determines the queue format from the BatchQueueType element. This information is detailed in the following table.

For the description of the elements in the respective batch queues, see the Queues tab of the IES.

In the Queues tab of the IES, you can view the elements for a particular batch queue by filtering the "Queue Format" column with the respective batch queue format and the "Function" column with the respective batch queue function. For example, if you want to view the details for "INCOMING\_BQ\_PREARBITRATIONS" batch queue, filter the "Queue Format" column with the queue format "S" and filter the "Function" column with "Batch Queue Request" or "Batch Queue Response". The details for this batch queue are displayed accordingly.

**Table 18-1: Types of Batch Queues**

UI Queue Name	<BatchQueueType>	Queue Format	I/A	For possible workflows, see...
All Incoming Pre-Arb	INCOMING_BQ_PREARBITRATIONS	S		
All Incoming Pre-Comp	INCOMING_BQ_PRECOMPLIANCES	S		

**Table 18-1: Types of Batch Queues (cont'd.)**

UI Queue Name	<BatchQueueType>	Queue Format	I/A	For possible workflows, see...
All Incoming Arb Filings	INCOMING_BQ_ARBITRATIONS	S		<a href="#">Chapter 11, Dispute Case Filings</a>
All Incoming Comp Filings	INCOMING_BQ_COMPLIANCES	S		
All Incoming Copy Requests	INCOMING_BQ_RFC_REQUESTS	T		
All RFC Responses	INCOMING_BQ_RFC_RESPONSES <sup>1</sup>	T		
All Misc. Fees	INCOMING_BQ_MISC_FEES	T		
All Rejects	INCOMING_BQ_REJECTS	R		
All Incoming Adjustments	INCOMING_BQ_ADJUSTMENTS	T	I	<a href="#">Chapter 9, Adjustments</a>
All Awaiting Action Inquiry/GF/ASRR	INCOMING_BQ_COLLABORATION <sup>2</sup>	T		<a href="#">Chapter 22, Collaboration</a>
All Awaiting Action Disputes	AWAITING_ACTION_BQ_DISPUTE <sup>3</sup>	D		<a href="#">Chapter 7, Dispute Questionnaires</a>
All Quick-form Exceptions	INCOMING_BQ_QUICKFORM	K		<a href="#">Chapter 23, Dispute Quick Forms</a>
Recalls <sup>6</sup>	INCOMING_BQ_RECALS	D		<a href="#">Chapter 7, Dispute Questionnaires</a>
All Awaiting Action Pre-Filing <sup>4</sup>	AWAITING_ACTION_BQ_PRE_FILING	D		<a href="#">Chapter 10, Dispute Pre-Filings</a>
All Acceptances Received <sup>5</sup>	INCOMING_BQ_ACCEPTANCES_RECEIVED	D		<a href="#">Chapter 10, Dispute Pre-Filings</a>

<sup>1</sup> RFCS responses are immediately added to the INCOMING\_BQ\_RFC\_RESPONSES batch queue, so the <CPD> field appears blank.

<sup>2</sup> The collaboration batch queue includes only the incoming collaboration transactions, unlike that of the collaboration UI queue.

<sup>3</sup> Includes saved disputes.

<sup>4</sup> Includes dispute Pre-Arbitrations and Pre-Compliances.

<sup>5</sup> Includes accepted dispute responses, Pre-Arbitrations, and Pre-Arbitration responses.

<sup>6</sup> If there is a case without a destination dispute amount in the incoming recalls batch queue, then the case must be considered as a dispute that was recalled before it is delivered to the other side.

**Note:** *If a new BIN is added to a VROL organization, the associated transactions are only included in the queues once VROL recognizes the Org-BIN relationship. If VROL has transactions associated with that BIN prior to this date, user can only retrieve these transactions by using the other RTSI Queue functionality in [Chapter 17, Queues](#) or by using hypersearch in [Chapter 19, Hypersearch Requests](#).*

**Note:** In order to use this feature, the user must select queues in the VROL User Profile associated with the RTSI's User ID attribute.

In general, the batch queues contain values from the time the case or transaction is added to the queue and is not real-time data. So, it is possible that some fields, such as User ID and Last Action Date, may have changed since adding to the queue.

## Batch Queue Process

This section describes the process and XML operations used by an RTSI member to perform the following tasks.

### Retrieve Transactions From a Batch Queue

1. User submits an RTSI request SIGetBatchQueueRequest to retrieve items from a queue.
2. VROL returns up to 400 items per page from the queue.

**Note:** Repeated requests are needed if the queue contains more than 400 transactions and the transactions are returned in more than one page. The user should store the transactions returned locally rather than make repeated queue requests.

User processes the transactions.

### Mark Transactions as Read

#### Important:

To mark batch queue items as read, VROL accepts any of the following items:

- One or multiple BatchQueueItemIDs with no restrictions on the number of BatchQueueItemID provided
  - One or multiple queue type names that will mark all items in the specified queue type as read
1. Once all the transactions are retrieved from the queue and processed by the user, the user sends an RTSI request SIMarkBatchQueueItemAsReadRequest to mark these transactions as "read". The user must provide the identifier (BatchQueueItemSID) for each transaction they want to mark as read.

Once the transactions are marked as "read", these transactions are no longer available in the Bulk SI download.

**Note:** If a transaction is not marked as read, it is removed or purged from the RTSI Batch Queue after 10 days.

2. VROL removes the transactions from the RTSI Batch Queue. This reduces the number of transactions returned for each queue request and the possibility of returning duplicates.

**Note:** The user can trigger the SIAcceptDisputeRequest message to accept liability for a dispute transaction and remove it from the batch queue. For more information, see [SIAcceptDisputeRequest](#).

During the Mark Transactions as Read operation, when any of the following scenarios occur, VROL returns a message, respectively. The following table provides the operation scenarios and the messages returned by VROL.

Mark Transactions as Read Operation Scenarios	VROL Message Returned to User
If the user provided any of the following items: <ul style="list-style-type: none"> <li>• A valid BatchQueueItemIDs</li> <li>• A combination of valid and non-existing BatchQueueItemIDs</li> </ul>	Successfully completed operation <b>Note:</b> VROL returns this message only for the valid BatchQueueItemIDs provided.
If the user provided any of the following items: <ul style="list-style-type: none"> <li>• Invalid BatchQueueItemIDs, or a combination of valid and invalid BatchQueueItemIDs</li> <li>• A combination of invalid and non-existing BatchQueueItemIDs</li> </ul>	MarkBatchQueueItemAsReadRequest operation failed for following batch queue items: Invalid SIDs
If any of the following scenarios occur: <ul style="list-style-type: none"> <li>• The user provided a non-existing BatchQueueItemSID</li> <li>• When a different organization is used for the Mark Transactions as Read operation either for valid or invalid BatchQueueItemSID, or combination of valid or invalid BatchQueueItemIDs</li> </ul>	Invalid BatchQueueItemSID or there are no items for the specified BatchQueueItemSID in the queue

## Batch Queue Operations

RTSI exposes the following batch queue operations:

**SIGetBatchQueueRequest**—Used to retrieve transactions from an RTSI Batch Queue.

**SIMarkBatchQueueItemAsReadRequest**—Used to mark a transaction as read and remove it from the RTSI Batch Queue.

### SIGetBatchQueueRequest (Retrieving Transaction From a Queue)

This function is used to retrieve transactions from an RTSI Batch Queue.

The following table provides information about some XML elements that should be considered when submitting an SIGetBatchQueue request or retrieving a response:

Element	Description
BatchQueueType	This element is required in a request and must specify one of the values shown in <a href="#">Table 18-1</a>
PageNum (request)	Use this element in the request to submit additional requests if more than 400 cases are found.
PageNum TotalPages (response)	Cases are grouped in 'pages' of 400. When a search finds more than 400 cases, RTSI response contains only the first 400, and the TotalPages element indicates the total number of pages found.
BatchQueueItemSID	Identifier for each transaction assigned by VROL. If a transaction must be marked as read, <BatchQueueItemSID> must be provided.

The SIGetBatchQueueRequest operation is used in request messages as shown in the following sample messages.

## Request Message—SOAP-based Services

The request includes the page number for the request, as only 400 items can be returned for each request. Along with the transactions, the response returns the total number of pages and the current page number. The user must send subsequent requests, updating the current page number, to retrieve the later items in the queue. The user should retrieve transactions in order of ascending page.

```
<SIGetBatchQueueRequest>
    <RequestHeader>
        <User id="abcde1234f" type="internalId"/>
        <MemberRole>A</MemberRole>
    </RequestHeader>
    <requestData>
        <BatchQueueType>AWAITING_ACTION_BQ_DISPUTE</BatchQueueType>
        <PageNum>1</PageNum>
    </requestData>
</SIGetBatchQueueRequest>
```

## Response Message—SOAP-based Services

The response includes an identifier, BatchQueueItemSID, which is used to mark the transaction as read, to remove it from the next Batch Queue request.

**Note:** The <User> element is not returned in the queue response as some delivery details are not known in the Batch Queue.

**Note:** It is possible to see multiple entries in the response when a user acts multiple times on the same case. For example, when a user repairs a reject and it rejects again. In this scenario, the user sees two entries in the batch queue results until they explicitly mark the first item as read.

```
<SIGetBatchQueueResponse>
    <Status>
        <Code>I-300000000</Code>
        <Message>Successfully completed Operation</Message>
    </Status>
    <responseData>
        <BatchQueueType>AWAITING_ACTION_BQ_DISPUTE</BatchQueueType>
        <PageInfo>
            <PageNum>1</PageNum>
            <TotalPages>10</TotalPages>
        </PageInfo>
        <Queue>
            <FormatDQueueItem>
                <BatchQueueItemSID>1234567</BatchQueueItemSID>
                <VisaCaseNumber>1000000001</VisaCaseNumber>
                <CaseStatus>Fraud Dispute - Received / 100.00%
Acq Liability</CaseStatus>
                <DateToRespond>2018-11-12</DateToRespond>
                <DisputeAmt currency="554">25.86</DisputeAmt>
                <AccountNumber>410000000000</AccountNumber>
                <ARN>70000000000000000000000000000001</ARN>
                <RetrievalReferenceNumber>800000000000
                </RetrievalReferenceNumber>
                <User>John Doe</User>
                <LastActionDate>2018-09-11T02:45:44+00:00
                </LastActionDate>
                <BIDName>ABC Bank</BIDName>
            </FormatDQueueItem>
        </Queue>
    </responseData>
</SIGetBatchQueueResponse>
```

```
<MerchantName>MERCHANT NAME</MerchantName>
<NetworkID>0002</NetworkID>
<Jurisdiction jurisdictionCd="REG">
    <Jurisdiction>REGIONAL-AP</Jurisdiction>
</Jurisdiction>
<InternalId transactionType="DISPREQ">
    <Id>300005</Id>
</InternalId>
<DCCode>10.4</DCCode>
<DCCodeDescription>Card Absent Environment
</DCCodeDescription>
</FormatDQueueItem>
<FormatDQueueItem>
    <BatchQueueItemSID>1234568</BatchQueueItemSID>
    <VisaCaseNumber>1000000002</VisaCaseNumber>
    <CaseStatus>Fraud Dispute - Received / 100.00%
    Acc Liability</CaseStatus>
    <DateToRespond>2018-11-12</DateToRespond>
    <DisputeAmt currency="554">25.86</DisputeAmt>
    <AccountNumber>410000000001</AccountNumber>
    <ARN>70000000000000000000000000000002</ARN>
    <RetrievalReferenceNumber>800000000001
    </RetrievalReferenceNumber>
    <User>Jane Doe</User>
    <LastactionDate>2018-09-11T02:45:44+00:00
    </LastactionDate>
    <BIDName>ABCD Bank</BIDName>
    <MerchantName>MERCHANT NAME</MerchantName>
    <NetworkID>0002</NetworkID>
    <Jurisdiction jurisdictionCd="REG">
        <Jurisdiction>REGIONAL-AP</Jurisdiction>
    </Jurisdiction>
    <InternalId transactionType="DISPREQ">
        <Id>300006</Id>
    </InternalId>
    <DCCode>10.3</DCCode>
    <DCCodeDescription>Card Present Environment
    </DCCodeDescription>
</FormatDQueueItem>
<FormatDQueueItem>
    <BatchQueueItemSID>1234560</BatchQueueItemSID>
    <VisaCaseNumber>1000000003</VisaCaseNumber>
    <CaseStatus>Processing Error Dispute -
    Received</CaseStatus>
    <DateToRespond>2018-11-03</DateToRespond>
    <DisputeAmt currency="840">11803.74</DisputeAmt>
    <AccountNumber>410000000002</AccountNumber>
    <ARN>70000000000000000000000000000003</ARN>
    <RetrievalReferenceNumber>800000000002
    </RetrievalReferenceNumber>
    <User>Julian Doe</User>
    <LastactionDate>2018-10-03T03:03:23+00:00
    </LastactionDate>
    <BIDName>ABCDE Bank</BIDName>
    <MerchantName>MERCHANT NAME</MerchantName>
    <NetworkID>0002</NetworkID>
    <Jurisdiction jurisdictionCd="DOM">
```

```
        <Jurisdiction>DOMESTIC-US</Jurisdiction>
    </Jurisdiction>
    <InternalId transactionType="DISPREQ">
        <Id>300008</Id>
    </InternalId>
    <DCCode>12.7</DCCode>
    <DCCodeDescription>Invalid Data
    </DCCodeDescription>
</FormatDQueueItem>
    ...
</Queue>
</ResponseData>
</SIGetBatchQueueResponse>
```

## **Response Message—Action—SOAP-based Services**

The following is an example of a response message where the <Action> field is populated with the various states when the case is in a case filing stage such as "Acknowledgement", "Visa Contact Messages", "Response made available to other side", or "Final Decision".

```
<DisputeJurisdiction jurisdictionCd="DOM">
    <Jurisdiction>DOMESTIC-CE</Jurisdiction>
</DisputeJurisdiction>
<DateToRespond>2018-11-12</DateToRespond>
<Action>Acknowledgement</Action>
</FormatSQueueItem>
<FormatSQueueItem>
    <BatchQueueItemSID>2345670</BatchQueueItemSID>
    <VisaCaseNumber>2000000003</VisaCaseNumber>
    <CaseStatus>Arb Filing by Iss -
    Acknowledged by Visa</CaseStatus>
    <ReasonCode>13.3</ReasonCode>
    <ReasonCodeDescription>13.3 Consumer - Not as
    Described</ReasonCodeDescription>
    <CaseAmount currency="840">14208.76</CaseAmount>
    <OriginalTransactionAmount currency="840">14208.76
    </OriginalTransactionAmount>
    <AccountNumber>41000000003</AccountNumber>
    <NetworkID>0002</NetworkID>
    <InternalId transactionType="DISPREQ">
        <Id>4000003</Id>
    </InternalId>
    <MemberCaseNumber>ABCDE</MemberCaseNumber>
    <ARN>20000000000000000000000000000003</ARN>
    <RetrievalReferenceNumber>200000000003
    </RetrievalReferenceNumber>
    <User>James Doe</User>
    <LastactionDate>2019-05-22T04:40:18+00:00
    </LastactionDate>
    <BIDName>EDCBA Bank</BIDName>
    <MerchantName>MERCHANT NAME</MerchantName>
    <DisputeJurisdiction jurisdictionCd="DOM">
        <Jurisdiction>DOMESTIC-US</Jurisdiction>
    </DisputeJurisdiction>
    <DateToRespond>2019-05-29</DateToRespond>
    <Action>Visa Contact Messages</Action>
</FormatSQueueItem>
<FormatSQueueItem>
    <BatchQueueItemSID>2345679</BatchQueueItemSID>
    <VisaCaseNumber>2000000002</VisaCaseNumber>
    <CaseStatus>Arb Filing by Iss -
    Acknowledged by Visa</CaseStatus>
    <ReasonCode>13.3</ReasonCode>
    <ReasonCodeDescription>13.3 Consumer - Not as
    Described</ReasonCodeDescription>
    <CaseAmount currency="840">14208.76</CaseAmount>
    <OriginalTransactionAmount currency="840">14208.76
    </OriginalTransactionAmount>
    <AccountNumber>41000000002</AccountNumber>
    <NetworkID>0002</NetworkID>
    <InternalId transactionType="DISPREQ">
        <Id>4000002</Id>
    </InternalId>
    <ARN>20000000000000000000000000000002</ARN>
    <RetrievalReferenceNumber>200000000002
    </RetrievalReferenceNumber>
    <User>Jules Doe</User>
```

```
<LastActionDate>2019-05-22T04:40:18+00:00
</LastActionDate>
<BIDName>EDCBA Bank</BIDName>
<MerchantName>MERCHANT NAME</MerchantName>
<DisputeJurisdiction jurisdictionCd="DOM">
    <Jurisdiction>DOMESTIC-US</Jurisdiction>
</DisputeJurisdiction>
<DateToRespond>2019-05-29</DateToRespond>
<Action>Response Made Available to Other Side</Action>
</FormatsQueueItem>
...
</Queue>
</responseData>
</SIGetBatchQueueResponse>
```

## Request Message—REST-based Services

```
{  
  "RequestHeader": {  
    "User": {  
      "id": "abcde1234f",  
      "type": "loginId"  
    },  
    "MemberRole": "A"  
  },  
  "requestData": {  
    "BatchQueueType": "AWAITING_ACTION_BQ_DISPUTE",  
    "PageNum": 1  
  }  
}
```

## Response Message—REST-based Services

```
"RetrievalReferenceNumber": "800000000000",
"User": "John Doe",
"LastActionDate": "2018-09-11T03:01:43+00:00",
"BIDName": "ABC Bank",
"MerchantName": "MERCHANT NAME",
"NetworkID": "0002",
"Jurisdiction": {
    "Jurisdiction": "REGIONAL-AP",
    "jurisdictionCd": "REG"
},
"InternalId": {
    "Id": "300005",
    "transactionType": "DISPREQ"
},
"DCCode": "10.3",
"DCCodeDescription": "Card Present Environment"
},
{
    "BatchQueueItemSID": 1234568,
    "VisaCaseNumber": 1000000002,
    "CaseStatus": "Processing Error Dispute - Received",
    "DateToRespond": "2018-11-12",
    "DisputeAmt": {
        "value": 123.89,
        "currency": "554"
    },
    "AccountNumber": "0000000000000008",
    "ARN": "70000000000000000000000000000003",
    "RetrievalReferenceNumber": "700000000008",
    "User": "John Doe",
    "LastActionDate": "2018-09-11T06:10:00+00:00",
    "BIDName": "ABC BANK",
    "MerchantName": "MERCHANT NAME",
    "NetworkID": "0002",
    "Jurisdiction": {
        "Jurisdiction": "DOMESTIC-NZ",
        "jurisdictionCd": "DOM"
    },
    "InternalId": {
        "Id": "394169",
        "transactionType": "DISPREQ"
    },
    "DCCode": "12.4",
    "DCCodeDescription": "Incorrect Account Number",
},
{
    "BatchQueueItemSID": 1234560,
    "VisaCaseNumber": 1000000003,
    "CaseStatus": "Processing Error Dispute - Received",
    "DateToRespond": "2018-11-12",
    "DisputeAmt": {
        "value": 917,
        "currency": "392"
    },
    "AccountNumber": "410000000002",
    "ARN": "70000000000000000000000000000003",
    "RetrievalReferenceNumber": "800000000002",
```

```
        "User": "Julian Doe",
        "LastActionDate": "2018-09-12T03:25:29+00:00",
        "BIDName": "ABCDE Bank",
        "MerchantName": "MERCHANT NAME",
        "NetworkID": "0002",
        "Jurisdiction": {
            "Jurisdiction": "REGIONAL-AP",
            "jurisdictionCd": "REG"
        },
        "InternalId": {
            "Id": "300008",
            "transactionType": "DISPREQ"
        },
        "DCCode": "12.2",
        "DCCodeDescription": "Incorrect Transaction Code"
    },
    ...
]
}
```

### Response Message—Action—REST-based Services

The following is an example of a response message where the <Action> field is populated with the various states when the case is in a case filing stage such as "Acknowledgement", "Visa Contact Messages", "Response made available to other side", or "Final Decision".

```
{
    "Status": [
        {
            "Code": "I-300000000",
            "Message": "Successfully completed Operation."
        }
    ],
    "ResponseData": {
        "BatchQueueType": "INCOMING_BQ_ARBITRATIONS",
        "PageInfo": {
            "PageNum": 1,
            "TotalPages": 1
        },
        "Queue": {
            "FormatSQueueItem": [
                {
                    "BatchQueueItemSID": 2345678,
                    "VisaCaseNumber": 2000000001,
                    "CaseStatus": "Arb Filing by Acq - Acknowledged by Visa",
                    "ReasonCode": "10.4",
                    "ReasonCodeDescription": "10.4 Fraud - Card Absent Environment",

                    "CaseAmount": {
                        "value": 5000,
                        "currency": "840"
                    },
                    "OriginalTransactionAmount": {
                        "value": 5000,
                        "currency": "840"
                    },
                    "AccountNumber": "41000000000001",
                    ...
                }
            ]
        }
    }
}
```

```
"NetworkID": "0002",
"InternalId": {
    "Id": "2189158",
    "transactionType": "ARBT"
},
"ARN": "20000000000000000000000000000001",
"RetrievalReferenceNumber": "200000000000",
"User": "Jenny Doe",
"LastActionDate": "2018-09-12T10:33:38+00:00",
"BIDName": "CBA Bank",
"MerchantName": "MERCHANT NAME",
"DisputeJurisdiction": {
    "Jurisdiction": "DOMESTIC-CE",
    "jurisdictionCd": "DOM"
},
"DateToRespond": "2018-09-19"
"Action": "Acknowledgement"
},
{
"BatchQueueItemSID": 2345670,
"VisaCaseNumber": 2000000003,
"CaseStatus": "Arb Filing by Iss - Acknowledged by Visa",
"ReasonCode": "13.3",
"ReasonCodeDescription": "13.3 Consumer - Not as Described",
"CaseAmount": {
    "value": 14208.76,
    "currency": "840"
},
"OriginalTransactionAmount": {
    "value": 14208.76,
    "currency": "840"
},
"AccountNumber": "410000000003",
"NetworkID": "0002",
"InternalId": {
    "Id": "4000003",
    "transactionType": "ARBT"
},
"MemberCaseNumber": "ABCDE",
"ARN": "20000000000000000000000000000003",
"RetrievalReferenceNumber": "200000000003",
"User": "James Doe",
"LastActionDate": "2019-05-22T04:40:18+00:00",
"BIDName": "EDCBA Bank",
"MerchantName": "MERCHANT NAME",
"DisputeJurisdiction": {
    "Jurisdiction": "DOMESTIC-US",
    "jurisdictionCd": "DOM"
},
"DateToRespond": "2019-05-29",
"Action": "Visa Contact Messages"
}
{
"BatchQueueItemSID": 2345679,
"VisaCaseNumber": 2000000002,
"CaseStatus": "Arb Filing by Iss - Acknowledged by Visa",
"ReasonCode": "13.3",
```

```
        "ReasonCodeDescription": "13.3 Consumer - Not as Described",
        "CaseAmount": {
            "value": 14208.76,
            "currency": "840"
        },
        "OriginalTransactionAmount": {
            "value": 14208.76,
            "currency": "840"
        },
        "AccountNumber": "410000000002",
        "NetworkID": "0002",
        "InternalId": {
            "Id": "4000002",
            "transactionType": "ARBT"
        },
        "MemberCaseNumber": "ABCDE",
        "ARN": "20000000000000000000000000000002",
        "RetrievalReferenceNumber": "200000000002",
        "User": "Jules Doe",
        "LastActionDate": "2019-05-22T04:40:18+00:00",
        "BIDName": "EDCBA Bank",
        "MerchantName": "MERCHANT NAME",
        "DisputeJurisdiction": {
            "Jurisdiction": "DOMESTIC-US",
            "jurisdictionCd": "DOM"
        },
        "DateToRespond": "2019-05-29",
        "Action": "Response Made Available to Other Side"
    },
    ...
]
}
```

## SIMarkBatchQueueItemAsReadRequest (Marking a Transaction as Read)

This function is used to mark a transaction as read in order to remove it from the RTSI Batch Queue.

The following table provides information about some XML elements that should be considered when submitting an SIMarkBatchQueueItemAsReadrequest or retrieving a response:

Element	Description
BatchQueueItemSID	Identifier for each transaction assigned by VROL. If a transaction must be marked as read, either <BatchQueueItemSID> or <BatchQueueType> must be provided.
BatchQueueType	If a transaction must be marked as read, either <BatchQueueType> or <BatchQueueItemSID> must be provided.

A maximum of 400 items is allowed in the SIMarkBatchQueueItemAsReadRequest.

**Note:** This maximum number is configurable in VROL.

As the items in a queue are returned using paging, it is recommended that the RTSI user retrieves all items from the queue before marking any items as read.

If a transaction is not marked as read, it is removed or purged from the RTSI Batch Queue after 10 days. In case this happens, the user can still retrieve the transaction by using the other RTSI Queue functionality in [Chapter 17, Queues](#) or by using hypersearch as mentioned in [Chapter 19, Hypersearch Requests](#).

At times a duplicate may arise if the process of marking a transaction as read fails, or if an invalid BatchQueueItemSID is sent in the request. The user must be able to deal with duplicates even if they mark a transaction as read.

If a member shares the BIN with another member or VROL organization, marking the item as read removes it only from the current user's VROL organization and not from any other VROL organization.

The SIMarkBatchQueueItemAsReadRequest operation is used in request messages as shown in the following sample messages.

### Request Message—SOAP-based Services

For multiple transactions, a single request can be made with multiple <BatchQueueItemSID> elements. The Batch Queue does not need to be identified, as the <BatchQueueItemSID> is unique across all queues.

```
<SIMarkBatchQueueItemAsReadRequest>
  <RequestHeader>
    <User id="abcde1234f" type="internalId"/>
    <MemberRole>A</MemberRole>
  </RequestHeader>
  <requestData>
    <BatchQueueItemSID>1234567</BatchQueueItemSID>
    <BatchQueueItemSID>1234678</BatchQueueItemSID>
    <BatchQueueType>INCOMING_BQ_REJECTS</BatchQueueType>
  </requestData>
</SIMarkBatchQueueItemAsReadRequest>
```

### Response Message—SOAP-based Services

The response returns a success or error status. Success means that the items were successfully removed from the Batch Queue.

Following is the response when all items are successful:

```
<Status>
  <Code>I-300000000</Code>
  <Message>Successfully completed Operation</Message>
</Status>
  <responseData>
    <BatchQueueItemSID>1234589</BatchQueueItemSID>
    <BatchQueueItemSID>1234690</BatchQueueItemSID>
  </responseData>
</SIMarkBatchQueueItemAsReadResponse>
```

Following is the response when one or more items failed. In the sample below, 1234567 & 1234678 are valid IDs that failed.

**Note:** VROL will not return failed IDs that are invalid IDs in this list.

```
<SIMarkBatchQueueItemAsReadResponse>
  <Status>
    <Code>E-121005204</Code>
    <Message><MarkBatchQueueItemAsReadRequest> operation failed for
following batch items</Message>
    <ErrorIn>String</ErrorIn>
  </Status>
  <responseData>
    <BatchQueueItemSID>1234567</BatchQueueItemSID>
    <BatchQueueItemSID>1234678 </BatchQueueItemSID>
  </responseData>
</SIMarkBatchQueueItemAsReadResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": {
    "User": {
      "id": "abcde1234f",
      "type": "internalId"
    },
    "MemberRole": "A"
  },
  "requestData": {
    "BatchQueueItemSID": [1234567]
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": [
    {
      "Code": "I-300000000",
      "Message": "Successfully completed Operation."
    }
  ],
  "responseData": {
    "BatchQueueItemSID": [1234567]
  }
}
```

THIS PAGE INTENTIONALLY LEFT BLANK.

# Hypersearch Requests

19

*Hypersearch requests* allow users to perform searches for single or multiple cases maintained in VROL. An RTSI hypersearch considers no match, a single match, or multiple matches to existing cases as valid responses; it returns a status code and message with a list of any matching cases (with minimal case information for each one) in the response.

Hypersearch requests are commonly used to obtain information in VROL about cases in dispute. For example:

- An issuer can perform a hypersearch request to determine if the acquirer has responded with a fulfillment or nonfulfillment to a Request for Copy (RFC).
- Members can use hypersearch to receive information about a transaction inquiry (TI) and financial.
- Members can obtain information on fraud reports and exception file listings.

Hypersearch also allows users to search for cases that include message transactions, even after the case or message thread is closed.

All hypersearch requests and responses are audited. Both member users who log on to the system and those who actually initiate the action are tracked for auditing purpose.

The methods used to obtain information in a hypersearch first reveals basic information such as the FraudReportID, ExceptionFileID, RFCID, and CaseFilingItemID that are then used to find more detailed information about a case (see [Table 19-1](#)). A sequence number is needed for those identifiers that could refer to multiple items.

Error conditions may be encountered during message validation or when processing the request. In both cases, the response message describes the error condition in the FaultString element of the SOAP message. No other elements appear in the SOAP body.

## Hypersearch Process

The process to submit a hypersearch follows.

1. User submits a hypersearch SIHyperSearchRequest to VROL with at least one of the following parameters:
  - Card/Account Number (<AccountNumber>)

**Note:** For tokenized transactions (Visa Token Service), acquirers can put token in this field. In hypersearch responses, the acquirer, similar to the issuer, will receive the card/account number in the AccountNumber field and Token in the Token field

- Token
- ARN
- Tran Date
- Tran Amount
- Transaction ID + MCSN (if available)
- Retrieval Request ID (any RRID in the case folder)
- VROL Case Number

**Note:** If VROL Case number is provided, all other search criteria are ignored.

- Member Case Number
- RRN
- STAN
- Acquirer BIN/ID

**Note:** VROL uses every matching field provided to perform a match. If a case matches some but not all of those fields, it is not considered a match.

2. VROL groups hypersearch results in 'pages' of 200 cases each. If more than 200 cases are found, the initial response contains the first 200 cases.

To obtain the remaining cases, user submits additional requests containing the same case matching parameters and sets the PageNum element in the request to the page number requested.

## Hypersearch Operation

Hypersearch exposes one operation.

**SIHyperSearchRequest** —Used by the Web Service client to submit a hypersearch request to RTSI. This operation accepts case matching parameters and returns information about all cases found.

The following table provides information about some XML elements that should be considered when submitting a hypersearch request or retrieving a response. For complete details about XML elements, consult the IES, the *Visa Resolve Online RTSI Technical Reference*, and the annotations in the XML schema.

Element	Description
All	While no specific case matching element is required, at least one must be specified. Only cases that match all request parameters are returned.
PageNumTotalPages (response)	Cases are grouped in 'pages' of 200. When a search finds more than 200 cases, RTSI response contains only the first 200, and the TotalPages element indicates the total number of pages found.
PageNum (request)	Use in the request to submit additional requests if more than 200 cases are found. This element defaults to the first page if it is empty.

Element	Description
CaseViewIndicators (response)	Contains indicators for individual transactions in the case as described in <a href="#">Table 19-1</a> . If the hypersearch retrieves no CaseViewIndicators, that is, actions clients can perform, this element is not present in the response.
MemberRole	If the submitting member's organization type is issuer and acquirer (dual organization), then this value is required. The value should contain 'I' when submitting member is acting as an issuer and 'A' when acting as an acquirer.  If submitting member is not a dual organization, then <MemberRole> is not required.

The SIHyperSearchRequest operation accepts case matching parameters and returns information about all cases found as shown in the following sample messages.

## Request Message—SOAP-based Services

The following example shows a sample message submitting a hypersearch request.

```
<SIHyperSearchRequest>
  <RequestHeader>
    <User id="abcde1234f" type="internalId"/>
    <MemberRole>A</MemberRole>
  </RequestHeader>
  <requestData>
    <CaseMatchingInfo>
      <VisaCaseNumber>1000000000</VisaCaseNumber>
    </CaseMatchingInfo>
  </requestData>
</SIHyperSearchRequest>
```

## Response Message—SOAP-based Services

The following example shows a sample hypersearch response returned by RTSI.

```
<SIHyperSearchResponse>
  <Status>
    <Code>I-300000000</Code>
    <Message>Successfully completed Operation.</Message>
  </Status>
  <responseData>
    <Case>
      <VisaCaseNumber>1000000000</VisaCaseNumber>
      <MemberCaseNumber>555555555555</MemberCaseNumber>
      <AccountNumber>4000000000000008</AccountNumber>
      <ARN>200000000000000000000001</ARN>
      <Transaction id="3000000000000001">
        <TranDate>2018-01-15</TranDate>
        <Amount currency="840">22.04</Amount>
      </Transaction>
      <MerchantName>Merchant Name</MerchantName>
      <DisputeAmount currency="840">13.00</DisputeAmount>
      <StageStateDesc>Processing Error Dispute -
      Response Pending Advice</StageStateDesc>
      <RetrievalReferenceNumber>100000000000
      </RetrievalReferenceNumber>
      <SystemTraceAuditNumber>0000001</SystemTraceAuditNumber>
      <AcquirerBIN>100000</AcquirerBIN>
      <NetworkID>0002</NetworkID>
      <CaseResolutionStatus>RESLVD</CaseResolutionStatus>
      <LastactionDate>2018-07-30T04:01:34+00:00</LastactionDate>
      <CaseViewIndicators>
        <DocumentIndicators>
          <DocId>7000006</DocId>
        </DocumentIndicators>
        <DisputeIndicators>
          <DisputeId LastactionDate="2018-04-
          05T07:29:22+00:00">300009</DisputeId>
          <DisputeResponseId LastactionDate="2018-04-
          30T22:28:47+00:00">300005</DisputeResponseId>
          <DisputeFinancial>
            <DisputeFinId LastactionDate="2018-04-
            05T07:29:22+00:00">200001</DisputeFinId>
            <DisputeStatus>F1</DisputeStatus>
          </DisputeFinancial>
          <DisputeFinancial>
            <DisputeFinId LastactionDate="2018-04-
            30T22:28:47+00:00">300001</DisputeFinId>
            <DisputeStatus>P1</DisputeStatus>
          </DisputeFinancial>
        </DisputeIndicators>
      </CaseViewIndicators>
    </Case>
    <PageInfo>
      <PageNum>1</PageNum>
      <TotalPages>1</TotalPages>
    </PageInfo>
  </responseData>
</SIHyperSearchResponse>
```

## Request Message—REST-based Services

```
{  
    "RequestHeader": {  
        "User": {  
            "id": "abcde1234f",  
            "type": "loginId"  
        },  
        "MemberRole": "A"  
    },  
    "requestData": {  
        "CaseMatchingInfo": {  
  
            "VisaCaseNumber": "1000000000"  
        }  
    }  
}
```

## Response Message—REST-based Services

```
{  
    "Status": [ {  
        "Code": "I-300000000",  
        "Message": "Successfully completed Operation."  
    } ],  
    "responseData": {  
        "Case": [ {  
            "VisaCaseNumber": 1000000000,  
            "MemberCaseNumber": "555555555555",  
            "AccountNumber": "4000000000000001",  
            "ARN": "20000000000000000000000001",  
            "Transaction": {  
                "TranDate": "2018-01-15",  
                "Amount": {  
                    "value": 22.04,  
                    "currency": "840"  
                },  
                "id": "3000000000000001"  
            },  
            "MerchantName": "Merchant Name",  
            "DisputeAmount": {  
                "value": 13,  
                "currency": "840"  
            },  
            "StageStateDesc": "Processing Error Dispute -  
Response Pending Advice",  
            "RetrievalReferenceNumber": "100000000000",  
            "SystemTraceAuditNumber": "0000001",  
            "AcquirerBIN": "100000",  
            "NetworkID": "002",  
            "CaseResolutionStatus": "RESLVD",  
            "LastActionDate": "2018-07-30T04:01:34+00:00",  
            "CaseViewIndicators": {  
                "DocumentIndicators": {"DocId": [7000006]},  
                "DisputeIndicators": {  
                    "DisputeId": {  
                        "value": 13,  
                        "currency": "840"  
                    }  
                }  
            }  
        } ]  
    }  
}
```

```
        "value": 300009,
        "LastActionDate": "2018-04-05T07:29:22+00:00"
    },
    "DisputeResponseId": {
        "value": 300005,
        "LastActionDate": "2018-04-30T22:28:47+00:00"
    },
    "DisputeFinancial": [
        {
            "DisputeFinId": {
                "value": 200001,
                "LastActionDate": "2018-04-05T07:29:22+00:00"
            },
            "DisputeStatus": "F1"
        },
        {
            "DisputeFinId": {
                "value": 300001,
                "LastActionDate": "2018-04-30T22:28:47+00:00"
            },
            "DisputeStatus": "P1"
        }
    ]
},
"PageInfo": {
    "PageNum": 1,
    "TotalPages": 1
}
}
```

# Hypersearch Response Elements

Response elements that are returned by a hypersearch include those found in the following table. Refer to the IES for a complete list of response elements.

**Table 19–1: Hypersearch Response Elements**

Element	Descriptions
AdjustmentID AdjustmentID (LastActionDate attribute)	Uniquely identifies an adjustment transaction. Member can use this to access the adjustment information in a subsequent request.  The LastActionDate provides the date and time associated with the last action for the transaction for the member.
AccountingEntryID AccountingEntryID (LastActionDate attribute)	Uniquely identifies an Accounting Entry Memo (AEM) transaction. Member can use this to access the AEM information in a subsequent request.  The LastActionDate provides the date and time associated with the last action for the transaction for the member.

**Table 19–1: Hypersearch Response Elements (cont'd.)**

<b>Element</b>	<b>Descriptions</b>
ExceptionFileID ExceptionFileType ExceptionFileID (LastActionDate attribute)	<p>ExceptionFileID uniquely identifies an exception file listing (EFL). Member can use this to access the EFL information in a subsequent request.</p> <p>ExceptionFileType defines the following domain values:</p> <ul style="list-style-type: none"> <li>• EFNEG=Negative Exception File Listing</li> <li>• EFVIP=Set VIP Limit</li> <li>• EFDEL=Delete Account Listing</li> <li>• EFINQ=Inquiry</li> </ul> <p>The LastActionDate provides the date and time associated with the last action for the transaction for the member.</p>
FraudReportID FraudReportID (LastActionDate attribute)	<p>Uniquely identifies a fraud report. Member can use this to access the fraud report information in a subsequent request.</p> <p>The LastActionDate provides the date and time associated with the last action for the transaction for the member.</p>
MiscFeeID MiscFeeType MiscFeeID (LastActionDate attribute)	<p>MiscFeeID uniquely identifies a miscellaneous fees transaction. Member can use this to access the miscellaneous fee information in a subsequent request.</p> <p>MiscFeeType defines the following domain values:</p> <ul style="list-style-type: none"> <li>• FeeCollection</li> <li>• FundDisbursement</li> </ul> <p>The LastActionDate provides the date and time associated with the last action for the transaction for the member.</p>
RFCID HasRfcFulfillment HasRfcNonFulfillment HasRfcAdvice RFCID (LastActionDate attribute)	<p>Uniquely identifies an RFC fulfillment, nonfulfillment, or advice transaction. Member can use this to access the RFC information in a subsequent request.</p> <p>The LastActionDate provides the date and time associated with the last action for the transaction for the member.</p>
StopPaymentID StopPaymentID (LastActionDate attribute)	<p>Uniquely identifies a stop payment (PPCS) transaction.</p> <p>The LastActionDate provides the date and time associated with the last action for the transaction for the member.</p>
RfiEventId RfiEventId (LastActionDate attribute)	<p>Uniquely identifies a TI transaction.</p> <p>RfiEventId indicates that a transaction inquiry has been submitted. Member can use this element to access the TI transaction detail.</p> <p>The LastActionDate provides the date and time associated with the last action for the transaction for the member.</p>
StageStateDesc	<p>Text description of the case's stage and state. For example, RFI - Fulfilled</p>

**Table 19–1: Hypersearch Response Elements (cont'd.)**

Element	Descriptions
CaseResolutionStatus	<p>Value assigned by VROL when creating the case.</p> <ul style="list-style-type: none"> <li>• UNASGN=Unassigned</li> <li>• INPROC=In Progress</li> <li>• PNDCLS=Pending Closure</li> <li>• CANCEL=Close - Cancelled</li> <li>• DENIED=Close - Denied</li> <li>• RESLVD=Closed - Resolved</li> <li>• CLOSED=Closed</li> <li>• PENDING=Pending</li> </ul>
RetrievalReferenceNumber	<p>For the request, at least one of the following fields is required: MemberCaseNumber, VisaCaseNumber, AccountNumber, ARN, TranDate, Amount, TransactionID, RetrievalRequestID, RetrievalReferenceNumber, STAN, and AcquirerBIN.</p>
DocID	<p>Uniquely identifies an attached document.</p>
ActiveMessageID MessageType MessageIndicatorType ActiveMessageID (LastActionDate attribute)	<p>Uniquely identifies an active RPP message in a message thread. Member can use this to access the message thread information and history in a subsequent request.</p> <p>MessageType domain value=RPP.</p> <p>MessageIndicatorType defines the following domain values:</p> <ul style="list-style-type: none"> <li>• RPPMessage</li> <li>• RPPMessageResp</li> </ul> <p>The LastActionDate provides the date and time associated with the last action for the transaction for the member.</p>
PurchaseInquiryId PurchaseInquiryId (LastActionDate attribute) PurchaseInquiryId (PurchaseInquiryResponseStatus attribute)	<p>PurchaseInquiryId uniquely identifies a Purchase Inquiry. Member can use this to access the purchase information in a SIGetPurchaseInquiry request.</p> <p>The LastActionDate provides the date and time associated with the transaction.</p> <p>The PurchaseInquiryResponseStatus attribute is the status associated with the PurchaseInquiryId (for example, successful, no response).</p>
PurchaseInquiryEligibleInd	<p>PurchaseInquiryEligibleInd indicates the PI eligibility of the case. The PurchaseInquiryEligibleInd indicator is not displayed when value=false or unknown.</p>

**Table 19–1: Hypersearch Response Elements (cont'd.)**

Element	Descriptions
CollaborationId CollaborationType CollaborationId (LastActionDate attribute) CollaborationActiveInd SubmittedBy	<p>CollaborationId uniquely identifies a collaboration transaction. Member can use this to access the collaboration information in a SIGetCollaborationDetails request.</p> <p>CollaborationType defines the following domain values:</p> <ul style="list-style-type: none"> <li>• IQ=Inquiry</li> <li>• GF=Good Faith</li> <li>• ASRR=Auto-Substantiation Receipt Request</li> </ul> <p>The LastActionDate provides the date and time associated with the transaction.</p> <p>CollaborationActiveInd identifies which CollaborationId is for an "active" collaboration transaction in the case.</p> <p>SubmittedBy identifies whether the transaction was sent by the issuer or acquirer, and is useful because either side can initiate a collaboration thread.</p>
MerchantNotificationId MerchantNotificationId (LastActionDate attribute) MerchantNotificationId (NotificationResponseStatus attribute)	<p>MerchantNotificationId uniquely identifies a Merchant Notification. Member can use this to access the merchant notification information in a SIGetMerchantNotificationDetails request.</p> <p>The LastActionDate provides the date and time associated with the transaction.</p> <p>The NotificationResponseStatus attribute is the status associated with the MerchantNotificationId (for example, successful, no response).</p>
DisputeId DisputeId (LastActionDate attribute)	<p>DisputeId uniquely identifies a dispute.</p> <p>Member can use this to access the dispute information in SIGetDisputeDetailsRequest.</p> <p>The LastActionDate provides the date and time associated with the transaction.</p>
AssociatedTransGroupId AssociatedTransGroupId (LastActionDate attribute)	<p>AssociatedTransGroupId uniquely identifies an associated transaction.</p> <p>Member can use this to access the associated transaction information in SIGetAssociatedTransactionListRequest.</p> <p>The LastActionDate provides the date and time associated with the transaction.</p>
DisputeResponseId DisputeResponseId (LastActionDate attribute)	<p>DisputeResponseId uniquely identifies a response to a dispute.</p> <p>Member can use this to access the dispute response information in SIGetDisputeResponseDetailsRequest.</p> <p>The LastActionDate provides the date and time associated with the transaction.</p>

**Table 19–1: Hypersearch Response Elements (cont'd.)**

Element	Descriptions
DisputePreArbId DisputePreArbId (LastActionDate attribute)	<p>DisputePreArbId uniquely identifies a dispute Pre-Arbitration transaction.</p> <p>Member can use this to access the dispute Pre-Arbitration information in SIGetDisputePreArbDetailsRequest.</p> <p>The LastActionDate provides the date and time associated with the transaction.</p>
DisputePreArbResponseId DisputePreArbResponseId (LastActionDate attribute)	<p>DisputePreArbResponseId uniquely identifies a response to a dispute Pre-Arbitration.</p> <p>Member can use this to access the dispute Pre-Arbitration response information in SIGetDisputePreArbResponseDetailsRequest.</p> <p>The LastActionDate provides the date and time associated with the transaction.</p>
DisputePreComplId DisputePreComplId (LastActionDate attribute)	<p>DisputePreComplId uniquely identifies a dispute Pre-Compliance transaction.</p> <p>Member can use this to access the dispute Pre-Compliance information in SIGetDisputePreCompDetailsRequest.</p> <p>The LastActionDate provides the date and time associated with the transaction.</p>
DisputePreCompResponseId DisputePreCompResponseId (LastActionDate attribute)	<p>DisputePreCompResponseId uniquely identifies a response to a dispute Pre-Compliance.</p> <p>Member can use this to access the dispute Pre-Compliance response information in SIGetDisputePreCompResponseDetailsRequest.</p> <p>The LastActionDate provides the date and time associated with the transaction.</p>
DisputeFilingItemId DisputeFilingItemId (LastActionDate attribute) DisputeFilingItemType	<p>DisputeFilingItemId uniquely identifies a dispute filing transaction.</p> <p>Member can use this to access the dispute filing information in SIGetDisputeFilingDetailsRequest.</p> <p>The LastActionDate provides the date and time associated with the transaction.</p> <p>DisputeFilingItemType defines the following domain values:</p> <ul style="list-style-type: none"> <li>• Arbitration</li> <li>• Compliance</li> <li>• FilingResponse</li> <li>• AcknowledgementLetter</li> <li>• RejectionLetter</li> <li>• DecisionLetter</li> </ul>

**Table 19–1: Hypersearch Response Elements (cont'd.)**

Element	Descriptions
ContactMessageId ContactMessageId (LastActionDate attribute)	ContactMessageId uniquely identifies a contact message. Member can use this to access the incoming contact message request information in SIGetContactMessageDetailsRequest. The LastActionDate provides the date and time associated with the transaction.
ContactMessageResponseId ContactMessageResponseId (LastActionDate attribute)	ContactMessageResponseId uniquely identifies a response to a contact message. Member can use this to access the contact message response information in SIGetContactMessageResponseDetailsRequest. The LastActionDate provides the date and time associated with the transaction.
DisputeAdviceId DisputeAdviceId (LastActionDate attribute)	DisputeAdviceId uniquely identifies a dispute advice. Member can use this to access the dispute advice information in SIGetDisputeFinancialAdviceDetailsRequest. The LastActionDate provides the date and time associated with the transaction.
DisputeFinId DisputeFinId (LastActionDate attribute)	DisputeFinId uniquely identifies a dispute financial. Member can use this to access the dispute financial information in SIGetDisputeFinancialDetailsRequest. The LastActionDate provides the date and time associated with the transaction.
VROLFinancialID VROLFinancialID (LastActionDate attribute)	VROLFinancialID identifies a dispute financial and is generated using a common algorithm shared between VROL and VIP. VROL sends this Id to VIP if the dispute financial is generated within VROL; otherwise, the member is expected to send this Id to VIP if the dispute financial is generated outside of VROL. VROLFinancialID is present in the response of services mentioned in <a href="#">Financial Fields Returned in Immediate Response</a> . The LastActionDate provides the date and time associated with the transaction.

**Table 19–1: Hypersearch Response Elements (cont'd.)**

Element	Descriptions
QuickFormID QuickFormID (LastActionDate attribute) QuickFormType	QuickFormID uniquely identifies a dispute quick form. The LastActionDate provides the date and time associated with the transaction. QuickFormType domain value=Dispute.
DisputeStatus	DisputeStatus defines the following domain values: <ul style="list-style-type: none"><li>• F1=Dispute Financial</li><li>• R1=Dispute Reversal – Recall</li><li>• R2=Dispute Reversal – Pre-Arbitration Acceptance</li><li>• R3=Dispute Reversal – Arbitration Decision</li><li>• P1=Dispute Response</li><li>• L1=Dispute Response Reversal – Recall</li><li>• L2=Dispute Response Reversal – Pre-Arbitration Acceptance</li><li>• L3=Dispute Response Reversal – Arbitration Decision</li></ul>

# Messages

20

*Messages* are non-financial transactions that can be exchanged between issuers and acquirers. Messages are internal to VROL and are only valid between BINs active in VROL. VROL supports the Request Proof of Posting (RPP) Message type.

RPP Messages are used for Original Credit Transactions for late or non-posting of funds disputes. The RPP Message functionality is available for all jurisdictions.

**Note:** *For issuers, a PIN-Authenticated Visa Debit transaction has a Reimbursement Attribute of Y, Z, or 8 on network 0002. For acquirers, a PIN-Authenticated Visa Debit transaction has a Reimbursement Attribute of 8 on network 0002. These transactions are available only for US domestic.*

With the message service, members can:

- Initiate a message
- Update one's own message
- Retrieve a message
- Respond to a message

The member that initiates message processing is considered the initiator. The other member is considered the recipient. Both, issuers and acquirers can initiate message processing.

Users can identify new messages in the RPP message queues and take the required action.

UI and RTSI support messages for single transactions only.

**Note:** *<MessageDetailsResponse> includes a history of previous Message transactions in the "thread", that is, with the same MessageType in the case. These are listed in order with the most recent first.*

## Message Process

The messages are added to an existing case. Often this will be a case with a TI but there are no restrictions on other transactions in the case.

There are two possible actions for messages:

- **Initiate:** Either the issuer or acquirer can initiate. If the issuer initiates, the acquirer is the recipient. If the acquirer initiates, the issuer is the recipient.
- **Respond:** Either side can respond to a message thread in a case. A member can respond to their own previous initiate/respond message. Each response includes a history of previous message initiate/responses.

Possible scenarios are shown in table below: (single message thread)

ID	Acquirer	Issuer	Acquirer	Issuer	Comments
1	Initiate				Next action: either side can respond
2	Initiate	Respond			Next action: either side can respond
3	Initiate		Respond		Next action: either side can respond
1		Initiate			Next action: either side can respond
2		Initiate	Respond		Next action: either side can respond
3		Initiate		Respond	Next action: either side can respond

## Send Message Initiate

1. Prior to sending a message initiate, there are a number of workflows the issuer or acquirer may follow. The following options are examples of possible workflows.
  - Sending a message based on TI request. See [Chapter 4, Transaction Inquiries](#) to perform this TI search and retrieve VROL Case Number.
  - Sending a message based on an existing case found in a user's workflow. See [Chapter 17, Queues](#). VROL sends response with VROL Case Number.
  - Creating a case, if no case exists prior to the message creation. See [Chapter 5, Case Creation](#). VROL sends the response with VROL Case Number.

### Important:

Once transaction information is established in a case, there is no need to provide it in subsequent actions. In addition, <TransactionInfo> is no longer required if a case is created from a TI result.

2. To create a message initiate request, user sends the initiate message request SICreateMessageRequest with (RPP Message) <RPPMessage>. If images are being attached, the message must be pended.  
VROL sends response with <MessageID> and success status.
3. To attach images to the message request, user sends request SIUploadMessageImageRequest with image information, using <MessageID>. The image is sent as encoded data with the request.  
VROL sends response with success status.
4. If images were attached, user submits pended message request SICreateMessageRequest with the <MessageID>.  
VROL sends response with success status.

## Retrieve Message and Message Responses

Prior to sending a message respond request, user retrieves the incoming message transaction.

1. To retrieve the message transactions that are awaiting action from the associated RPP message queues, user sends queue request `SIGetQueueRequest`. See [Chapter 17, Queues](#), for more information.  
VROL returns a summary of the message transactions in the queue including their VROL Case Numbers and Message IDs. As RPP message queues are Format T queues, the queue response includes the Message ID in the InternalId as follows:  
`<InternalId transactionType="MSG"> <ID>123456</ID> </InternalId>`.
2. To retrieve message details, user sends request `SIGetMessageRequest` using VROL Case Number and Message ID. The Message ID and VROL Case Number can be retrieved from the `SIGetQueueResponse`.  
VROL sends response with details of message transaction.
3. To retrieve images with the message, issuer sends request `SIGetMessageImageRequest` using VROL Case Number and Doc ID.  
VROL sends response with image details.
4. To close the message and remove it from the queue, user sends close message request `SICloseMessageRequest` or `SICloseTransactionRequest` with CloseItem Type as RPP or OTHER. See [Chapter 17, Queues](#) for more information.

**Note:** *`SICloseTransactionRequest` closes the transaction and removes it from its respective queue but does not close the case. For more information, see [SICloseTransactionRequest](#).*

VROL sends response with success status.

**Note:** *If a message is retrieved using Hypersearch (See [Chapter 19, Hypersearch Requests](#)), the Message ID returned (`<ActiveMessageID>`) is the ID of the "active message". The details of this active message will have the history of previous messages in the message thread.*

## Send Message Respond

**Note:** *Prior to submitting a response, an RTSI user should retrieve the latest prior message in the message thread to get the latest history. Otherwise, there is a risk that a response may be made by the opposing member, for example, between pend/SFR and submit. The RTSI user will get a warning if such a response occurred; however the warning is returned with the submit success status, after the response is sent.*

1. To create a message response request, user sends request `SICreateMessageRequest` with (RPP Message) `<RPPMessageResponse>`. If images are being attached, the message must be pended.  
VROL sends response with success status.
2. To attach images to the message request, user sends request `SIUploadMessageImageRequest` with image information, using `<MessageID>`. The image is sent as encoded data with the request.  
VROL sends response with success status.
3. If images were attached, user submits pended message request `SICreateMessageRequest` with the `<MessageID>`.  
VROL sends response with success status.

## Message Operations

RTSI exposes the following operations:

**SICreateMessageRequest**—Used by user to send an RPP message or response.

**SIGetMessageRequest**—Used by user to request details of a RPP message or response.

### SICreateMessageRequest

Refer to the table in [Chapter 3, RTSI Request Operations](#) for XML elements to consider when submitting a message transaction.

**Important:**

Once transaction information is established in a case, there is no need to provide it in subsequent actions. In addition, <TransactionInfo> is no longer required if a case is created from a TI result.

#### Request Message—Initiate RPP Message—SOAP-based Services

```
<SICreateMessageRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>0000000000</VisaCaseNumber> <- use if updating an existing case
        <MessageID>123456</MessageID> <- use only when updating a previously pended Message
        <MessageDescriptor id="1" operation="Add" action="Submit">
            <TransactionInfo>
                ...
            </TransactionInfo>
            <RPPMessage>
                <MessageBodyText>message text</MessageBodyText>
                <DaysToRespond>55</DaysToRespond>
                <ContactName>a</ContactName>
                <ContactPhone>000000</ContactPhone>
                <ContactFax>000000</ContactFax>
                <ContactEmail>anyemail@xyz.com</ContactEmail>
            </RPPMessage>
        </MessageDescriptor>
    </requestData>
</SICreateMessageRequest>
```

#### Response Message—Initiate RPP Message—SOAP-based Services

```
<SICreateMessageResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <MessageID>1234567</MessageID>
        <MessageType>RPP</MessageType>
    </responseData>
</SICreateMessageResponse>
```

### Request Message—Initiate RPP Message—REST-based Services

```
{  
    "RequestHeader": "...",  
    "requestData": {  
        "VisaCaseNumber": "0000000000", <- use if  
            updating an existing case  
        "MessageID": "0000000", <- use only when  
            updating a previously pended Message  
        "MessageDescriptor": {  
            "id": 1,  
            "operation": "Add",  
            "action": "Submit",  
            "TransactionInfo": {  
                "AccountNumber": "12345678901",  
                "Token": "1234567890001",  
                "TokenAssuranceLevel": "a",  
                ...  
            },  
            "RPPMessage": {  
                "MessageBodyText": "message text",  
                "DaysToRespond": 55,  
                "ContactName": "a",  
                "ContactPhone": "000000<",  
                "ContactFax": "000000<",  
                "ContactEmail": "anyemail@xyz.com",  
                ...  
            },  
        }  
    }  
}
```

### Response Message—Initiate RPP Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "VisaCaseNumber": "0000000000",  
        "MessageID": "0000000",  
        "MessageType": "RPP"  
    }  
}
```

### Request Message—Respond to RPP Message—SOAP-based Services

**Note:** Some fields below are applicable only for the recipient.

```
<SICreateMessageRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <MessageID>123456</MessageID>
    <MessageDescriptor id="1" operation="Add" action="Submit">
      <TransactionInfo>
        ...
      </TransactionInfo>
    <RPPMessageResponse>
      <MessageBodyText>message text</MessageBodyText>
      <DaysToRespond>5</DaysToRespond>
      <ContactName>a</ContactName>
      <ContactPhone>000000</ContactPhone>
      <ContactFax>000000</ContactFax>
      <ContactEmail>anyemail@xyz.com</ContactEmail>
      <PostingInd>true</PostingInd>
      <PostedDate>1967-08-13</PostedDate>
      <AmountPosted currency="000">0.0</AmountPosted>
      <LineInfo>line info text</LineInfo>
    </RPPMessageResponse>
  </MessageDescriptor>
  <requestData>
</SICreateMessageRequest>
```

### Response Message—Respond to RPP Message—SOAP-based Services

```
<SICreateMessageResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <MessageID>1234567</MessageID>
    <MessageType>RPP</MessageType>
  </responseData>
</SICreateMessageResponse>
```

### Request Message—Respond to RPP Message—REST-based Services

```
{  
    "RequestHeader": "...",  
    "requestData":{  
        "VisaCaseNumber": 0000000000, <- use if  
            updating an existing case  
        "MessageID": "0000000", <- use only when  
            updating a previously pended Message  
        "MessageDescriptor":{  
            "id": 1,  
            "operation": "Add",  
            "action": "Submit",  
            "TransactionInfo":{  
                "AccountNumber": "12345678901",  
                "Token": "1234567890001",  
                "TokenAssuranceLevel": "a",  
                ...  
            },  
            "RPPMessageResponse":{  
                "MessageBodyText": "message text",  
                "DaysToRespond": 5,  
                "ContactName": "a",  
                "ContactPhone": "000000",  
                "ContactFax": "000000",  
                "ContactEmail": "anyemail@xyz.com",  
                "PostingInd": true,  
                "PostedDate": "2016-01-01",  
                "AmountPosted":{  
                    "currency": "111",  
                    "Text": 1  
                },  
                "LineInfo": "line info text"  
            }  
        }  
    }  
}
```

### Response Message—Respond to RPP Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData":{  
        "VisaCaseNumber": 0000000000,  
        "MessageID": "0000000",  
        "MessageType": "RPP"  
    }  
}
```

## SIGetMessageRequest

### Request Message—SOAP-based Services

```
<SIGetMessageRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <MessageID>1234567</MessageID>
    </requestData>
</SIGetMessageRequest>
```

### Response Message—Initiate RPP Message—SOAP-based Services

```
<SIGetMessageResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <TransactionInfo>
            ...
        </TransactionInfo>
        <RPPMessage>
            <MessageBodyText>message text</MessageBodyText>
            <DaysToRespond>5</DaysToRespond>
            <ContactName>Any name</ContactName>
            <ContactPhone>000000</ContactPhone>
            <ContactFax>000000</ContactFax>
            <ContactEmail>anyemail@xyz.com</ContactEmail>
            <SubmittedBy>Issuer</SubmittedBy>
            <EventDate>2014-08-04T06:46:45.977</EventDate>
        </RPPMessage>
    </responseData>
</SIGetMessageResponse>
```

### **Response Message—Respond to RPP Message, including the history of the Message thread—SOAP-based Services**

```
<SIGetMessageResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <TransactionInfo>
            ...
        </TransactionInfo>
        <RPPMessageResponse>
            <MessageBodyText>response text</MessageBodyText>
            <DaysToRespond>5</DaysToRespond>
            <ContactName>Any name</ContactName>
            <ContactPhone>2222222222</ContactPhone>
            <ContactFax>2222222222</ContactFax>
            <ContactEmail>abc@asfd.com</ContactEmail>
            <SubmittedBy>Issuer</SubmittedBy>
            <EventDate>2012-12-20T20:47:51+00:00</EventDate>
            <PostingInd>true</PostingInd>
            <PostedDate>2012-07-01</PostedDate>
            <AmountPosted currency="840">23</AmountPosted>
            <LineInfo>Description</LineInfo>
        </RPPMessageResponse>
        <RPPMessage> <- History of previous messages &
                    message responses in the thread
            <MessageBodyText>message text</MessageBodyText>
            <DaysToRespond>5</DaysToRespond>
            <ContactName>Any name</ContactName>
            <ContactPhone>000000</ContactPhone>
            <ContactFax>000000</ContactFax>
            <ContactEmail>anyemail@xyz.com</ContactEmail>
            <SubmittedBy>Acquirer</SubmittedBy>
            <EventDate>2014-08-18T08:29:19.267</EventDate>
        </RPPMessage>
    </responseData>
</SIGetMessageResponse>
```

### **Request Message—REST-based Services**

```
{
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": "0000000000",
        "MessageID": 1234567
    }
}
```

**Response Message—Initiate RPP Message—REST-based Services**

```
{  
  "Status": "...",  
  "responseData": {  
    "VisaCaseNumber": "0000000000",  
    "TransactionInfo": {  
      ...  
    },  
    "RPPMessage": {  
      "MessageBodyText": "message text",  
      "DaysToRespond": -999,  
      "ContactName": "a",  
      "ContactPhone": "000000",  
      "ContactFax": "000000",  
      "ContactEmail": "anyemail@xyz.com",  
      "SubmittedBy": "Issuer",  
      "EventDate": "2016-01-01T19:31:47Z"  
    }  
  }  
}
```

### Response Message—Respond to RPP Message, including the history of the Message thread—REST-based Services

```
{  
    "Status": "...",  
    "responseData":{  
        "VisaCaseNumber": "0000000000",  
        "TransactionInfo":{  
            ...  
        },  
        "RPPMessageResponse":{  
            "MessageBodyText": "message text",  
            "DaysToRespond": -999,  
            "ContactName": "a",  
            "ContactPhone": "000000",  
            "ContactFax": "000000",  
            "ContactEmail": "anyemail@xyz.com",  
            "SubmittedBy": "Issuer",  
            "EventDate": "2016-01-01T19:31:47Z",  
            "PostingInd": true,  
            "PostedDate": "2016-01-01",  
            "AmountPosted":{  
                "currency": "111",  
                "Text": 1  
            },  
            "LineInfo": "Description"  
        }  
        "RPPMessage":{ <- History of previous messages &  
            message responses in the thread  
            "MessageBodyText": "message text",  
            "DaysToRespond": -999,  
            "ContactName": "a",  
            "ContactPhone": "000000",  
            "ContactFax": "000000",  
            "ContactEmail": "anyemail@xyz.com",  
            "SubmittedBy": "Issuer",  
            "EventDate": "2016-01-01T19:31:47Z"  
        }  
    }  
}
```

## Add Image to Message Transaction

Users can add one or multiple documents to a message transaction from a local or network drive by scanning a physical document or by inputting text.

Supporting documentation can be added and retrieved as needed using the following operations:

**SIUploadMessageImageRequest**—Used to add an image to a message transaction.

**SIGetMessageImageRequest**—Used to retrieve an image from a message transaction.

## SIUploadMessageImageRequest

The message transaction must be in pend status in order to attach an image. The VROL Case Number and Message ID are required and the image file must be attached.

**Important:**

SIUploadMessageImageRequest only allows a single image to be uploaded. To upload multiple images, this Web service should be used multiple times, once for each image.

The following table provides information about some XML elements that should be considered when attaching an image to a message transaction.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.
MessageID	Uniquely identifies the message thread within a case to which the image will be attached. The ID can be retrieved from the SICreateMessageResponse.

### Request Message—SOAP-based Services

```
<SIUploadMessageImageRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <MessageID>000000</MessageID>
    <AttachmentDescriptor>
      <Attachment id="1">
        <Comment>String</Comment>
        <ImageData content-Type="image/tiff">binary data</ImageData>
      </Attachment>
    </AttachmentDescriptor>
    <DocType>CALET</DocType>
  </requestData>
</SIUploadMessageImageRequest>
```

### Response Message—SOAP-based Services

```
<SIUploadMessageImageResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>0000000000</VisaCaseNumber>
    <MessageID>000000</MessageID>
    <DocId>000000</DocId>
  </responseData>
</SIUploadMessageImageResponse>
```

## Request Message—REST-based Services

```
{  
    "RequestHeader": "...",  
    "requestData": {  
        "VisaCaseNumber": "0000000000",  
        "MessageID": "0000000",  
        "AttachmentDescriptor": {  
            "AttachType": "Mime",  
            "Attachment": [  
                {  
                    "ContentType": "image/tiff",  
                    "Comment": "a",  
                    "ImageData": "image1"  
                    "id": "123456",  
                }  
            ]  
        },  
        "DocType": "MELET"  
    }  
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "VisaCaseNumber": "0000000000",  
        "MessageID": "0000000",  
        "DocId": 123456  
    }  
}
```

## SIGetMessageImageRequest

In order to retrieve an image attached to a message transaction, SIGetMessageImageRequest requires the VROL Case Number and the Doc ID.

The following table provides information about some XML elements that should be considered when retrieving an image from a message transaction.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.
DocId	Uniquely identifies a document (image) added to a case.  For the member sending the image, the ID can be retrieved from the SIUploadMessageImageResponse.  For the member retrieving the image from an opposing member, the ID can be retrieved from Hypersearch or Queue response.

### Request Message—SOAP-based Services

```
<SIGetMessageImageRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <DocId>0000000</DocId>
    </requestData>
</SIGetMessageImageRequest>
```

### Response Message—SOAP-based Services

```
<SIGetMessageImageResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <DocId>0000000</DocId>
        <AttachmentDescriptor>
            <Attachment id="String">
                <ContentType>image/tiff</ContentType>
                <Comment>String</Comment>
                <ImageData>binary data</ImageData>
            </Attachment>
        </AttachmentDescriptor>
        <DocType>MELET</DocType>
    </responseData>
</SIGetMessageImageResponse>
```

### Request Message—REST-based Services

```
{
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": "0000000000",
        "DocId": 123456
    }
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "VisaCaseNumber": "0000000000",  
        "DocId": 123456,  
        "AttachmentDescriptor": {  
            "Attachment": [  
                {  
                    "ContentType": "image/tiff",  
                    "ImageData": "api/getMessage Image/1111111111/  
                    123456?memberRole=I&user=test",  
                    "id": 123456  
                }  
            ]  
        },  
        "DocType": "MELET"  
    }  
}
```

THIS PAGE INTENTIONALLY LEFT BLANK.

# Purchase Inquiries

21

The Purchase Inquiry (PI) transaction provides relevant purchase information from participating merchants to avoid invalid disputes, accelerate resolution, and increase consumer satisfaction. For example, a purchase inquiry can be initiated if the cardholder does not recognize a TI transaction to provide additional information that is not provided in the TI details. The purchase inquiry details may be sent to the cardholder from VROL to remind the cardholder of their transaction.

The Purchase Inquiry details will be returned as a PDF file in the RTSI response and may have merchant-specific fields in the PDF.

**Important:**

VROL users can initiate a purchase inquiry request to the merchant only if the selected transaction in a case is eligible for Purchase Inquiry. A case or TI transaction is eligible for PI when the field <PurchaseInquiryEligibleInd> =true in SIGetTransDetailsResponse, SIGetTransactionsFromCaseResponse, SIHyperSearchResponse, SIGetCaseDetailsResponse, SIGetTransInquiryResultsResponse, or SISubmitTranInquiryResponse.

**Note:** *The PI transactions are available to both initiator and opposer. They are available to the opposer only when some other action in the case makes the case available to the opposer. An opposer can retrieve the PI fields using the unique identifier that is returned in a hypersearch response.*

## Merchant Notification

VROL provides a Merchant Notification feature that notifies merchants of fraud reports, stop payments, negative exception file listings (EFL), or disputes submitted from VROL against the merchant transactions. If a fraud, stop payment, negative EFL, or dispute is initiated from the case, merchants will be notified about the submitted fraud, stop payment, negative EFL, or dispute.

For Purchase Inquiry, the trigger for stop payment, negative EFL, and dispute initiated notification are as follows:

- Submission of a PPCS (R0, R1, or R2) triggers a stop payment notification
- Submission of a negative EFL triggers a negative EFL notification
- Evaluation of a dispute category triggers a dispute initiated notification

**Note:** Updating or deleting an existing stop payment or negative EFL, or recalling an existing dispute questionnaire from Case Details may trigger a stop payment, negative EFL, or dispute initiated notification.

Merchants can send the purchase inquiry details along with the Merchant Notification response. The merchant notification details will be returned together with the purchase inquiry data in the RTSI response. The purchase inquiry data is intended to indicate actions that the merchant may have taken as a result of knowledge of the fraud.

The Merchant Notification feature is available only if the merchant supports this feature, and only if the case is eligible for Purchase Inquiry.

## Purchase Inquiry Process

Users can request the latest Purchase Inquiry (PI) data from an existing case or from a transaction inquiry (TI) result if no case exists. In addition, users can also retrieve historical PI data in an existing case. The most recent PI is usually the most useful, but access to historical PIs may be necessary.

The information returned when making a PI request may change with time. For example, the data may be more complete or may have changed, for example, delivery time might change. Also, a "no response" may be updated with response. For this reason, multiple PIs may exist in a case, each with a unique date-time stamp.

### Request Latest PI Data from Existing Case

Users can request the latest PI data from an existing case by supplying the VROL Case Number in the request.

1. To retrieve the VROL Case Number, user sends queue request SISGetQueueRequest. The VROL Case Number is returned in the response. See [Chapter 17, Queues](#) for more information.

Some members opt to receive responses using Bulk SI Download files.

2. To request the latest Purchase Inquiry data from an existing case, user sends request SISSubmitPurchaseInquiryRequest using the VROL Case Number.

VROL sends response SISSubmitPurchaseInquiryResponse with the latest Purchase Inquiry data in a PDF file.

**Note:** Some fields are suppressed if the member accessing the Purchase Inquiry on the transaction is an acquirer.

**Note:** The Purchase Inquiry may be returned as an image (TIFF/PDF), in addition to XML, as part of this response message (if optional <IncludeResponseAsImageInd>=true is included in the request). However, any images attached to the Purchase Inquiry must be retrieved separately (step 3 below).

3. To retrieve the image data for the attached images, user sends request SISGetImageRequest using VROL Case Number and Doc ID.

VROL sends response with image data.

### Request Latest PI Data from TI Result if No Case Exists

Users can request the latest PI data from a TI result if no case exists by supplying the RolTransactionId in the request.

1. To retrieve the RolTransactionId, user sends TI request SISubmitTranInquiryRequest. The RolTransactionId is returned in the response. See [Chapter 4, Transaction Inquiries](#) for more information.
2. To request the latest PI data from a TI result if no case exists, user sends request SISubmitPurchaseInquiryRequest using the RolTransactionId.  
VROL sends response SISubmitPurchaseInquiryResponse with the latest Purchase Inquiry data in a PDF file.

**Note:** *If any Purchase Inquiry (PI) has been created for the TI prior to case creation, the last PI associated with the TI will be added to the case with SICreateCaseFromTransactionRequest.*

**Note:** *If this request is used for a selected TI in an existing case, it will return the updated data and not the historical data applicable at the time of the selected TI.*

**Note:** *Some fields are suppressed if the member accessing the Purchase Inquiry on the transaction is an acquirer.*

**Note:** *The Purchase Inquiry may be returned as an image (TIFF/PDF), in addition to XML, as part of this response message (if optional <IncludeResponseAsImageInd>=true is included in the request). However, any images attached to the Purchase Inquiry must be retrieved separately (step 3 below).*
3. To retrieve the image data for the attached images, user sends request SIGetImageRequest using VROL Case Number and Doc ID.  
VROL sends response with image data.

## Retrieve Historical PI Data in Existing Case

Users can retrieve historical PI data in an existing case by supplying both VROL Case Number and Purchase Inquiry Id in the request.

1. To retrieve the VROL Case Number, user sends queue request SIGetQueueRequest. The VROL Case Number is returned in the response. See [Chapter 17, Queues](#) for more information.  
Some members opt to receive responses using Bulk SI Download files.
2. To retrieve the Purchase Inquiry Id, issuer sends a Hypersearch request SIHyperSearchRequest. See [Chapter 19, Hypersearch Requests](#) for more information.
3. To retrieve historical PI data in an existing case, user sends request SIGetPurchaseInquiryRequest using the VROL Case Number and Purchase Inquiry Id that uniquely identifies the PI.  
VROL sends response SIGetPurchaseInquiryResponse with the historical Purchase Inquiry data in a PDF file.

**Note:** *The Purchase Inquiry may be returned as an image (TIFF/PDF), in addition to XML, as part of this response message (if optional <IncludeResponseAsImageInd>=true is included in the request). However, any images attached to the Purchase Inquiry must be retrieved separately (step 4 below).*

4. To retrieve the image data for the attached images, user sends request SIGetImageRequest using VROL Case Number and Doc ID.

VROL sends response with image data.

## Retrieve Merchant Notification Details

Users can retrieve the details of a merchant notification transaction by supplying both VROL Case Number and Merchant Notification Id in the request.

1. To retrieve the Merchant Notification Id, user sends a Hypersearch request SIHyperSearchRequest. See [Chapter 19, Hypersearch Requests](#) for more information.
2. To retrieve merchant notification details for each transaction, user sends request SIGetMerchantNotificationDetailsRequest using the VROL Case Number and MerchantNotificationId.

VROL sends response SIGetMerchantNotificationDetailsResponse with the merchant notification details together with the purchase inquiry data in a PDF file.

**Note:** *The Merchant Notification may be returned as an image (TIFF/PDF), in addition to XML, as part of this response message (if optional <IncludeResponseAsImageInd>=true is included in the request). However, any images attached to the Merchant Notification must be retrieved separately (step 3 below).*

3. To retrieve the image data for the attached images, user sends request SIGetImageRequest using VROL Case Number and Doc ID.

VROL sends response with image data.

## Purchase Inquiry Operations

RTSI exposes the following Purchase Inquiry operations:

**SISubmitPurchaseInquiryRequest**—Used by user to request the latest Purchase Inquiry data.

**SIGetPurchaseInquiryRequest**—Used by user to retrieve the historical Purchase Inquiry data.

**SIGetMerchantNotificationDetailsRequest**—Used by user to retrieve details of a merchant notification transaction.

### SISubmitPurchaseInquiryRequest

The SISubmitPurchaseInquiryRequest operation enables a user to request the latest PI data from an existing case by using the VROL Case Number, or from a TI result if no case exists by using the RolTransactionId.

The following table provides information about some XML elements that should be considered when submitting a Purchase Inquiry data request or retrieving a response.

Element	Description
VisaCaseNumber	Case tracking number assigned by VROL to uniquely identify a case in VROL.  <b>Note:</b> <i>In the schema, the VROL Case Number is &lt;VisaCaseNumber&gt;.</i>
RolTransactionId	Uniquely identifies the TI transaction to which the PI is associated.

Element	Description
PurchaseInquiryId	Uniquely identifies a Purchase Inquiry and is used to access the purchase information in a subsequent request.
PurchaseInquiryResponseStatus	Status associated with the PurchaseInquiryId (for example, successful, no response).

The SISubmitPurchaseInquiryRequest operation contains request and response messages as shown in the following sample messages. The latest PI data will be returned as a PDF file in the SISubmitPurchaseInquiryResponse.

### Request Message—SOAP-based Services

```
<SISubmitPurchaseInquiryRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>0000000000</VisaCaseNumber> (if there
is an existing case)
        <IncludeResponseAsImageInd>true</IncludeResponseAsImageInd>
    </requestData>
</SISubmitPurchaseInquiryRequest>
```

## Response Message—SOAP-based Services

```

<SISubmitPurchaseInquiryResponse>
    <Status>...</Status>
    <responseData>
        <PurchaseInquiryAttachmentDescriptor>
            <ContentType>application/pdf</ContentType>
            <ImageData>...</ImageData>
        </PurchaseInquiryAttachmentDescriptor>
        <PurchaseInquiryInformationDetails>
            <PurchaseCategory>S</PurchaseCategory>
            <CardholderSignatureInd>true</CardholderSignatureInd>
            <EmailAccount>a</EmailAccount>
            <Customer>
                <FirstName>String</FirstName>
                <LastName>String</LastName>
            </Customer>
            <AuthenticationConducted>a</AuthenticationConducted>
            <DeliveryAddress>...</DeliveryAddress>
            <RapidDisputeResolution>
                <CreditIntent>true</CreditIntent>
            </RapidDisputeResolution>
            <DigitalReceipt>...</DigitalReceipt>
            <DownloadDateTime>2017-12-17T09:30:47Z</DownloadDateTime>
            <PasswordLastUpdatedDateTime>2017-12-17T09:30:47Z
            </PasswordLastUpdatedDateTime>
            <RecipientCustomerName>a</RecipientCustomerName>
            <DeviceType>a</DeviceType>
            <TrackingNumber>a</TrackingNumber>
            <DeliveryService>a</DeliveryService>
            <DeliveryStatus>C</DeliveryStatus>
            <ShippingInfo>...</ShippingInfo>
            <CreditReimbursementInformationList>...
            </CreditReimbursementInformationList>
            <ContactUs>a</ContactUs>
            <WebsiteLink>a</WebsiteLink>
            <PackingSlip>a</PackingSlip>
            <ConditionofGoods>a</ConditionofGoods>
            <Address>...</Address>
            ...
        </PurchaseInquiryInformationDetails>
        <PurchaseInquiryId>00000</PurchaseInquiryId>
        <PurchaseInquiryResponseStatus>SUCCESS
        </PurchaseInquiryResponseStatus>
    </responseData>
</SISubmitPurchaseInquiryResponse>

```

## Request Message—REST-based Services

```
{
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": "0000000000"
    }
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "PurchaseInquiryAttachmentDescriptor": {  
            "ContentType": "application/pdf",  
            "ImageData": "..."  
        },  
        "PurchaseInquiryInformationDetails": {  
            "PurchaseCategory": "S",  
            "CardholderSignatureInd": "true",  
            "EmailAccount": "a",  
            "Customer": {  
                "FirstName": "test",  
                "LastName": "test"  
            },  
            "AuthenticationConducted": "a",  
            "DeliveryAddress": "...",  
            "RapidDisputeResolution": {  
                "CreditIntent": "true"  
            },  
            "DigitalReceipt": "...",  
            "DownloadDateTime": "2017-12-17T09:30:47Z",  
            "PasswordLastUpdatedDateTime": "2017-12-17T09:30:47Z",  
            "RecipientCustomerName": "a",  
            "DeviceType": "a",  
            "TrackingNumber": "a",  
            "DeliveryService": "a",  
            "DeliveryStatus": "C",  
            "ShippingInfo": "...",  
            "CreditReimbursementInformationList": "...",  
            "ContactUs": "a",  
            "WebsiteLink": "a",  
            "PackingSlip": "a",  
            "ConditionofGoods": "a",  
            "Address": "...",  
            ...  
        },  
        "PurchaseInquiryId": "00000",  
        "PurchaseInquiryResponseStatus": "SUCCESS"  
    }  
}
```

## SIGetPurchaseInquiryRequest

The SIGetPurchaseInquiryRequest operation enables a user to retrieve historical PI data in an existing case by using both VROL Case Number and Purchase Inquiry Id. This operation contains request and response messages as shown in the following sample messages. The historical PI data will be returned as a PDF file in the SIGetPurchaseInquiryResponse.

### Request Message—SOAP-based Services

```
<SIGetPurchaseInquiryRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>0000000000</VisaCaseNumber>
        <PurchaseInquiryId>00000</PurchaseInquiryId>
        <IncludeResponseAsImageInd>true</IncludeResponseAsImageInd>
    </requestData>
</SIGetPurchaseInquiryRequest>
```

### Response Message—SOAP-based Services

```
<SIGetPurchaseInquiryResponse>
    <Status>...</Status>
    <responseData>
        <PurchaseInquiryInformationDetails>
            <PurchaseCategory>S</PurchaseCategory>
            <CardholderSignatureInd>true</CardholderSignatureInd>
            <EmailAccount>a</EmailAccount>
            <Customer>
                <FirstName>String</FirstName>
                <LastName>String</LastName>
            </Customer>
            <AuthenticationConducted>a</AuthenticationConducted>
            <DeliveryAddress>...</DeliveryAddress>
            <RapidDisputeResolution>
                <CreditIntent>true</CreditIntent>
            </RapidDisputeResolution>
            ...
        </PurchaseInquiryInformationDetails>
        <PurchaseInquiryAttachmentDescriptor>
            <ContentType>image/jpeg</ContentType>
            <ImageData>...</ImageData>
        </PurchaseInquiryAttachmentDescriptor>
        <PurchaseInquiryId>0</PurchaseInquiryId>
        <PurchaseInquiryResponseStatus>String
        </PurchaseInquiryResponseStatus>
    </responseData>
</SIGetPurchaseInquiryResponse>
```

### Request Message—REST-based Services

```
{
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": "0000000000",
        "PurchaseInquiryId": "00000"
    }
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "PurchaseInquiryAttachmentDescriptor": {  
            "ContentType": "application/pdf",  
            "ImageData": "..."  
        },  
        "PurchaseInquiryId": "00000",  
        "PurchaseInquiryResponseStatus": "SUCCESS"  
    }  
}
```

## SIGetMerchantNotificationDetailsRequest

SIGetMerchantNotificationDetailsRequest is used in request and response messages as shown in the following sample messages.

## Request Message—SOAP-based Services

```
<SIGetMerchantNotificationDetailsRequest>  
    <RequestHeader>...</RequestHeader>  
    <requestData>  
        <VisaCaseNumber>0000000000</VisaCaseNumber>  
        <MerchantNotificationId>0000000</MerchantNotificationId>  
        <IncludeResponseAsImageInd>true</IncludeResponseAsImageInd>  
    </requestData>  
</SIGetMerchantNotificationDetailsRequest>
```

## Response Message—SOAP-based Services

```

<SIGetMerchantNotificationDetailsResponse>
    <Status>...</Status>
    <responseData>
        <PurchaseInquiryAttachmentDescriptor>
            <ContentType>application/pdf</ContentType>
            <ImageData>...</ImageData>
        </PurchaseInquiryAttachmentDescriptor>
        <PurchaseInquiryInformationDetails>
            <PurchaseCategory>M</PurchaseCategory>
            <CardholderSignatureInd>true</CardholderSignatureInd>
            <EmailAccount>a</EmailAccount>
            <Customer>
                <FirstName>String</FirstName>
                <LastName>String</LastName>
            </Customer>
            <AuthenticationConducted>a</AuthenticationConducted>
            <DeliveryAddress>...</DeliveryAddress>
            <RapidDisputeResolution>
                <CreditIntent>true</CreditIntent>
            </RapidDisputeResolution>
            <DigitalReceipt>...</DigitalReceipt>
            <DownloadDateTime>2017-12-17T09:30:47Z</DownloadDateTime>
            <PasswordLastUpdatedDateTime>2017-12-17T09:30:47Z
            </PasswordLastUpdatedDateTime>
            <RecipientCustomerName>a</RecipientCustomerName>
            <DeviceType>a</DeviceType>
            <TrackingNumber>a</TrackingNumber>
            <DeliveryService>a</DeliveryService>
            <DeliveryStatus>P</DeliveryStatus>
            <ShippingInfo>...</ShippingInfo>
            <CreditReimbursementInformationList>...
            </CreditReimbursementInformationList>
            <ContactUs>a</ContactUs>
            <WebsiteLink>a</WebsiteLink>
            <PackingSlip>a</PackingSlip>
            <ConditionofGoods>a</ConditionofGoods>
            <Address>...</Address>
            ...
        </PurchaseInquiryInformationDetails>
        <MerchantNotificationId>0000000</MerchantNotificationId>
        <CallType>2</CallType>
        <NotificationResponseStatus>SUCCESS</NotificationResponseStatus>
    </responseData>
</SIGetMerchantNotificationDetailsResponse>

```

## Request Message—REST-based Services

```
{
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": "0000000000",
        "MerchantNotificationId": "0000000"
    }
}
```

## Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "PurchaseInquiryAttachmentDescriptor": {  
            "ContentType": "application/pdf",  
            "ImageData": "..."  
        },  
        "PurchaseInquiryInformationDetails": {  
            "PurchaseCategory": "M",  
            "CardholderSignatureInd": "true",  
            "EmailAccount": "a",  
            "Customer": {  
                "FirstName": "test",  
                "LastName": "test"  
            },  
            "AuthenticationConducted": "a",  
            "DeliveryAddress": "...",  
            "RapidDisputeResolution": {  
                "CreditIntent": "true"  
            },  
            "DigitalReceipt": "...",  
            "DownloadDateTime": "2017-12-17T09:30:47Z",  
            "PasswordLastUpdatedDateTime": "2017-12-17T09:30:47Z",  
            "RecipientCustomerName": "a",  
            "DeviceType": "a",  
            "TrackingNumber": "a",  
            "DeliveryService": "a",  
            "DeliveryStatus": "P",  
            "ShippingInfo": "...",  
            "CreditReimbursementInformationList": "...",  
            "ContactUs": "a",  
            "WebsiteLink": "a",  
            "PackingSlip": "a",  
            "ConditionofGoods": "a",  
            "Address": "...",  
            ...  
        },  
        "MerchantNotificationId": "0000000",  
        "CallType": 2,  
        "NotificationResponseStatus": "SUCCESS"  
    }  
}
```

THIS PAGE INTENTIONALLY LEFT BLANK.

# Collaboration

22

Collaboration transactions are non-financial transactions that can be exchanged between issuers and acquirers. Collaboration transactions are internal to VROL and are only valid between BINs active in VROL.

This chapter pertains to standalone collaboration transactions. For dispute collaboration transactions, refer to [Chapter 7, Dispute Questionnaires](#) for more information.

**Important:**

Collaboration transactions must be added to an existing case. If there is no existing case, use the Create Case function described in [Chapter 5, Case Creation](#).

The member that initiates the collaboration is considered the initiator. The other member is considered the recipient. Both issuers and acquirers can initiate collaboration transactions.

When initiating a collaboration, the initiator must give the reason for this request and supply the number of days that the recipient has to respond. The recipient, when receiving the collaboration, sends a collaboration response. Depending on the collaboration type, the collaboration response may provide requested information, accept full or partial liability, or decline liability. The collaboration thread may then be continued, with members providing additional requests and responses. An initial collaboration may be recalled by the initiator only if the opposing member has not responded.

Users can identify incoming collaboration transactions from opposing members in the regular and batch collaboration queues, and take the required action. In addition, these transactions can be identified using Hypersearch.

## Collaboration Types

VROL supports four types of collaboration:

- Inquiry

Inquiry collaboration is used to exchange information with the intent of resolving the inquiry before it becomes a dispute. This type of collaboration is information-exchange only; it does not include any means to accept liability.

- Good Faith

Good Faith collaboration replaces the existing Good Faith function in VROL. It is an informal process for settling disputes between the members. This type of collaboration includes the means to accept full or partial liability.

- Auto-Substantiation Receipt Request (ASRR)

ASRR collaboration replaces the existing ASRR Message function in VROL. It is used for handling inquiries and disputes on healthcare auto-substantiation transactions. It is valid only for US domestic jurisdiction, either Interlink or PIN-Authenticated Visa Debit, that is for which an RFC is not possible since RFC is only supported for Visa.

- Card Recovery

Card Recovery collaboration is used to create manual card recovery and is typically initiated by the acquirer. This type is available in VROL UI and Bulk SI.

## Collaboration Background

- a. No collaboration transactions exist outside a VROL case.
- b. There must be an initial collaboration in case in order to do a collaboration response, that is, no standalone collaboration responses.
- c. Each subsequent collaboration transaction in a thread will have some history of previous collaboration transactions (initiate/continue/response) in that same thread.
- d. Attached images do not replace the need to populate required fields.
- e. VROL will allow users to have one active collaboration thread per case. If the user attempts to initiate a new collaboration in a case that contains an active collaboration thread, VROL will block the user from creating another thread. An active collaboration thread is one where the request has been submitted, and it has not been recalled, responded to with Accept Full, or closed. Performing any of these actions marks the end of an active thread.
- f. Collaboration can be initiated for Visa, Plus, and Interlink networks only.

## Collaboration Process

The collaboration transactions are added to an existing case. This will often be a case with a TI but there are no restrictions on other transactions in the case.

The possible actions for collaboration transactions are as follows:

- **Initiate/Continue:** Either the issuer or acquirer can initiate. If the issuer initiates, the acquirer is the recipient. If the acquirer initiates, the issuer is the recipient.

The initiator can continue an active collaboration thread after a response from the opposing member. The XML used by the initiator when continuing is the same as for initiating, although some fields may no longer be editable.

- **Respond:** The recipient can respond to an "active" collaboration from the initiator.

The recipient can respond again if the initiator continues. The XML used by the recipient in this subsequent response is the same as for the previous response, although some fields may no longer be editable.

- **Recall:** Only the initiator can recall the initial collaboration, and only if the opposing member has not responded.

A user, either the initiator or recipient, can do a followup collaboration prior to the opposing member's response/continue. This allows editable collaboration data to be updated or added, without waiting for the response. VROL checks the collaboration ID

to ensure that the opposing member responds/continues from the latest collaboration transaction in a case.

## Send Collaboration Initiate

1. Prior to sending a collaboration initiate, there are a number of workflows that the issuer or acquirer may follow. The following options are examples of possible workflows:
  - Sending a collaboration based on TI request. See [Chapter 4, Transaction Inquiries](#) to perform this TI search and retrieve VROL Case Number.
  - Sending a collaboration based on an existing case found in a user's workflow. See [Chapter 17, Queues](#).
  - Creating a case if no case exists prior to sending the collaboration. See [Chapter 5, Case Creation](#). VROL sends the response with VROL Case Number.

**Important:**

Once transaction information is established in a case, there is no need to provide it in subsequent actions. For example, <TransactionInfo> is no longer required if a case is created from a TI result.

2. To create a collaboration initiate request, user sends the initiate collaboration request SISubmitCollaborationRequest. This request can include image(s).  
VROL sends response with <CollaborationId> and success status.

## Retrieve Collaboration or Collaboration Response

Prior to responding to or continuing a collaboration, user retrieves the incoming collaboration transaction.

1. To retrieve the incoming collaboration transactions from the associated collaboration queues, user sends queue request SIGetQueueRequest or batch queue request SIGetBatchQueueRequest. See [Chapter 17, Queues](#) for more information.  
VROL returns a summary of the collaboration transactions in the queue including their VROL Case Numbers and collaboration IDs. As collaboration queues are Format T queues, the queue response includes the collaboration ID in the InternalId as follows: <InternalId transactionType="IQ"> <ID>123456</ID> </InternalId>. The <InternalId transactionType> field can have any of the following values, depending on the collaboration type: IQ for Inquiry, GF for Good Faith, or ASRR for Auto-Substantiation Receipt Request collaboration type.
2. To retrieve collaboration details, user sends request SIGetCollaborationDetailsRequest using VROL Case Number and collaboration ID. The collaboration ID and VROL Case Number can be retrieved from the SIGetQueueResponse or batch queue response SIGetBatchQueueRequest.

VROL sends response with details of collaboration transaction, including the AttachmentIds for any attached images.

**Note:** *The collaboration may be returned as an image (TIFF/PDF), in addition to XML, as part of this response (if optional <IncludeCollaborationImageId>=true). The image format will correspond to the member's preferred image format as configured by the Visa Administrator in the Member Preferences Screen. However, any images attached to the collaboration must be retrieved separately (step 3 below).*

3. To retrieve the image data for the attached images, user sends SIGGetImageRequest with the DocId (equivalent to <AttachmentId> returned in SIGGetCollaborationDetailsResponse).

VROL sends response with image data.

4. To close the collaboration and remove it from the queue, user sends close collaboration request SICloseTransactionRequest with CloseItemType as COLLABORATION. See [Chapter 17, Queues](#) for more information.

**Note:** *SICloseTransactionRequest closes the transaction and removes it from its respective queue but does not close the case. For more information, see [SICloseTransactionRequest](#).*

VROL sends response with success status.

## Send Collaboration Response (Recipient) or Continue Collaboration (Initiator)

**Note:** *Prior to submitting a response or continuing a collaboration, an RTSI user should retrieve the latest prior collaboration in the collaboration thread. VROL checks the collaboration ID to ensure that the opposing member responds/continues from the latest collaboration transaction in a case.*

1. To create a collaboration response or to continue a collaboration, user sends request SISubmitCollaborationRequest using the collaboration ID of the previous collaboration in the thread. This request can include image(s).

VROL sends response with <CollaborationId> and success status.

## Send Collaboration Recall (Initiator only)

**Note:** *Only the initiator can recall the initial collaboration, and only if the recipient has not responded.*

1. To recall a collaboration, user sends request SISubmitCollaborationRequest using the collaboration ID of the initial collaboration.

VROL sends response with <CollaborationId> and success status.

# Collaboration Operations

RTSI exposes the following collaboration operations:

**SISubmitCollaborationRequest**—Used to create the initial collaboration and to respond/continue/recall collaboration in an existing collaboration thread.

**SIGetCollaborationDetailsRequest**—Used to retrieve collaboration details.

**SIGetImageRequest**—Used to retrieve an image attached to an existing collaboration.

## SISubmitCollaborationRequest

SISubmitCollaborationRequest allows RTSI users to create the initial collaboration and to respond/continue/recall collaboration in an existing collaboration thread. This request can include image(s).

The SISubmitCollaborationRequest operation contains request and response messages as shown in the following sample messages.

The following table provides information about some required XML elements that should be provided when continuing a collaboration and submitting a collaboration response and recall.

Element	Description
CollaborationId	<p>The CollaborationId uniquely identifies a particular collaboration transaction. It is required for collaboration response, continue (as initiator), and recall. The CollaborationId value is different for the initial collaboration request, and for each continued collaboration, and response.</p> <p>This field is returned in an RTSI SISubmitCollaborationResponse and SIGetCollaborationDetailsResponse. For an RTSI user who retrieves incoming transactions using a Bulk SI download, this field is also sent in the Bulk SI download.</p> <p>This field is required for VROL to identify which submitted transaction an SI user is responding to. This is important if the opposing member does a re-submission prior to a response. VROL returns a hard-stop error if the transaction is not the "active" collaboration.</p>

**Note:** *The image data for attached documents must be included with SISubmitCollaborationRequest, rather than submitted separately. SIUploadImageRequest cannot be used to attach images to a collaboration transaction, as the collaboration transaction cannot be pended.*

### Request Message—Initiate or Continue Collaboration (Initiator)—SOAP-based Services

The following example shows a sample SISubmitCollaborationRequest message used to initiate or continue collaboration as an initiator.

```
<SISubmitCollaborationRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <CollaborationId>123456789</CollaborationId> <- Required if
      this is a continued collaboration
    <MemberCaseNumber>555555555555</MemberCaseNumber>
    <CollaborationDescriptor id="0" operation="Add" action="Submit">
      <TransactionInfo>...</TransactionInfo>
      <Collaboration>
        <CollaborationType>IQ</CollaborationType> <- Domain
          values: IQ, GF, ASRR
        <LastDayToRespond>2015-03-02</LastDayToRespond>
        <Source>C</Source>
        <CollaborationReason>request reason
        </CollaborationReason> <- Required if
          this is an IQ or ASRR collaboration
        <CBRepreTimeframeExpiredInd>true
        </CBRepreTimeframeExpiredInd> <- Required if
          this is a GF collaboration
        <CardholderMerchantNotificationOutsideOfCBRepreTimeframeInd>true
        </CardholderMerchantNotificationOutsideOfCBRepreTimeframeInd>
        <- Required if this is a GF collaboration
        <ProvideInResponse>response</ProvideInResponse>
        <CollaborationAmount currency="840">42.04
        </CollaborationAmount>
        <Notes>any note</Notes>
        <IssuerContactInfo> <- or <AcquirerContactInfo>,
          if sending member is Acquirer
          ... (many fields - see IES)
        </IssuerContactInfo>
        <CardholderContactInfo> <- or <MerchantContactInfo>,
          if sending member is Acquirer
          ... (many fields - see IES)
        </CardholderContactInfo>
      </Collaboration>
    </CollaborationDescriptor>
    <CollaborationAttachment>
      <AttachType>Mime</AttachType>
      <Attachment id="1">
        <ContentType>application/pdf</ContentType>
        <Comment>any comment</Comment>
        <ImageData>image data</ImageData>
        <DocType>UCLET</DocType>
      </Attachment>
      <Attachment id="2">
        <ContentType>application/pdf</ContentType>
        <Comment>any comment</Comment>
        <ImageData>image data</ImageData>
        <DocType>SHREC</DocType>
      </Attachment>
    </CollaborationAttachment>
  </requestData>
</SISubmitCollaborationRequest>
```

## Request Message—Respond to Collaboration (Recipient)—SOAP-based Services

The following examples show a sample SISubmitCollaborationRequest message used to respond to collaboration as a recipient, depending on the collaboration type and response option.

### Response—Accept Full (Good Faith only)

```
<SISubmitCollaborationRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <CollaborationId>123456789</CollaborationId> <- Required
    <MemberCaseNumber>555555555555</MemberCaseNumber>
    <CollaborationDescriptor id="0" operation="Add" action="Submit">
      <TransactionInfo>...</TransactionInfo>
      <CollaborationResp>
        <ResponseTypeCode>AF</ResponseTypeCode>
        <AcceptanceAmount currency="840">123.45
        </AcceptanceAmount>
        <PaymentMethodCode>OTHER</PaymentMethodCode> <- enter
        how amount will be sent
        <AmountSentDate>2015-03-02</AmountSentDate>
        <PaymentInstructions>enter if "how will amount be sent"
        =OTHER</PaymentInstructions>
        <Notes>any note</Notes>
        <AcquirerContactInfo> <- or <IssuerContactInfo>,
        if sending member is Issuer
          ... (many fields - see IES)
        </AcquirerContactInfo>
        <MerchantContactInfo> <- or <CardholderContactInfo>,
        if sending member is Issuer
          (Similar fields as <CardholderContactInfo>,
          including <MerchantURL> - see IES)
        </MerchantContactInfo>
      </CollaborationResp>
    </CollaborationDescriptor>
    <CollaborationAttachment>
      (See 'Request Message—Initiate or Continue Collaboration
      (Initiator)' for fields)
    </CollaborationAttachment>
  </requestData>
</SISubmitCollaborationRequest>
```

**Response—Accept Partial (Good Faith only)**

```
<SISubmitCollaborationRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <CollaborationId>123456789</CollaborationId> <- Required
        <MemberCaseNumber>555555555555</MemberCaseNumber>
        <CollaborationDescriptor id="0" operation="Add" action="Submit">
            <TransactionInfo>...</TransactionInfo>
            <CollaborationResp>
                <ResponseTypeCode>AP</ResponseTypeCode>
                <AcceptanceAmount currency="840">123.45
                </AcceptanceAmount>
                <PaymentMethodCode>OTHER</PaymentMethodCode> <- enter
                how amount will be sent
                <AmountSentDate>2015-03-02T06:09:06-07:00
                </AmountSentDate>
                <PaymentInstructions>enter if "how will amount be sent"
                =OTHER</PaymentInstructions>
                <AcceptPartialReason>accept partial liability reason
                </AcceptPartialReason>
                <Notes>any note</Notes>
                <AcquirerContactInfo> <- or <IssuerContactInfo>,
                if sending member is Issuer
                    ... (many fields - see IES)
                </AcquirerContactInfo>
                <MerchantContactInfo> <- or <CardholderContactInfo>,
                if sending member is Issuer
                    (Similar fields as <CardholderContactInfo>,
                    including <MerchantURL> - see IES)
                </MerchantContactInfo>
            </CollaborationResp>
        </CollaborationDescriptor>
        <CollaborationAttachment>
            (See 'Request Message—Initiate or Continue Collaboration
            (Initiator)' for fields)
        </CollaborationAttachment>
    </requestData>
</SISubmitCollaborationRequest>
```

### Response—Decline (Good Faith only)

```
<SISubmitCollaborationRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <CollaborationId>123456789</CollaborationId> <- Required
    <MemberCaseNumber>555555555555</MemberCaseNumber>
    <CollaborationDescriptor id="0" operation="Add" action="Submit">
      <TransactionInfo>...</TransactionInfo>
      <CollaborationResp>
        <ResponseTypeCode>D</ResponseTypeCode>
        <DeclineReason>decline reason</DeclineReason>
        <Notes>any note</Notes>
        <AcquirerContactInfo> <- or <IssuerContactInfo>,
        if sending member is Issuer
          ... (many fields - see IES)
        </AcquirerContactInfo>
        <MerchantContactInfo> <- or <CardholderContactInfo>,
        if sending member is Issuer
          (Similar fields as <CardholderContactInfo>,
          including <MerchantURL> - see IES)
        </MerchantContactInfo>
      </CollaborationResp>
    </CollaborationDescriptor>
    <CollaborationAttachment>
      (See 'Request Message—Initiate or Continue Collaboration
      (Initiator)' for fields)
    </CollaborationAttachment>
  </requestData>
</SISubmitCollaborationRequest>
```

**Response—Need More Information (Inquiry, ASRR, and Good Faith)**

```
<SISubmitCollaborationRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <CollaborationId>123456789</CollaborationId> <- Required
        <MemberCaseNumber>555555555555</MemberCaseNumber>
        <CollaborationDescriptor id="0" operation="Add" action="Submit">
            <TransactionInfo>...</TransactionInfo>
            <CollaborationResp>
                <ResponseTypeCode>NMI</ResponseTypeCode>
                <AdditionalInfoNeeded>additional info
                </AdditionalInfoNeeded>
                <Notes>any note</Notes>
                <AcquirerContactInfo> <- or <IssuerContactInfo>,
                    if sending member is Issuer
                    ... (many fields - see IES)
                </AcquirerContactInfo>
                <MerchantContactInfo> <- or <CardholderContactInfo>,
                    if sending member is Issuer
                    (Similar fields as <CardholderContactInfo>,
                     including <MerchantURL> - see IES)
                </MerchantContactInfo>
            </CollaborationResp>
        </CollaborationDescriptor>
        <CollaborationAttachment>
            (See 'Request Message—Initiate or Continue Collaboration
             (Initiator)' for fields)
        </CollaborationAttachment>
    </requestData>
</SISubmitCollaborationRequest>
```

### Response—Inquiry Fulfilled (Inquiry and ASRR only)

```
<SISubmitCollaborationRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <CollaborationId>123456789</CollaborationId> <- Required
    <MemberCaseNumber>555555555555</MemberCaseNumber>
    <CollaborationDescriptor id="0" operation="Add" action="Submit">
      <TransactionInfo>...</TransactionInfo>
      <CollaborationResp>
        <ResponseTypeCode>IQF</ResponseTypeCode>
        <FulfillmentInfo>requested information</FulfillmentInfo>
        <Notes>any note</Notes>
        <AcquirerContactInfo> <- or <IssuerContactInfo>,
        if sending member is Issuer
          ... (many fields - see IES)
        </AcquirerContactInfo>
        <MerchantContactInfo> <- or <CardholderContactInfo>,
        if sending member is Issuer
          (Similar fields as <CardholderContactInfo>,
          including <MerchantURL> - see IES)
        </MerchantContactInfo>
      </CollaborationResp>
    </CollaborationDescriptor>
    <CollaborationAttachment>
      (See 'Request Message—Initiate or Continue Collaboration
      (Initiator)' for fields)
    </CollaborationAttachment>
  </requestData>
</SISubmitCollaborationRequest>
```

**Request Message—Initiate or Continue Collaboration (Initiator)—REST-based Services**

```
{  
    "RequestHeader": "...",  
    "requestData": {  
        "VisaCaseNumber": 1000000000,  
        "CollaborationId": 123456789,  
        "MemberCaseNumber": 555555555555,  
        "CollaborationDescriptor": {  
            "id": 0,  
            "operation": "Add",  
            "action": "Submit",  
            "TransactionInfo": "...",  
            "Collaboration": {  
                "CollaborationType": "IQ",  
                "LastDayToRespond": "2015-03-02",  
                "Source": "C",  
                "CollaborationReason": "request reason",  
                "CBRepreTimeframeExpiredInd": "true",  
                "CardholderMerchantNotificationOutsideOfCBRepreTimeframeInd": "true",  
                "ProvideInResponse": "response",  
                "CollaborationAmount": {  
                    "currency": 840,  
                    "Text": "42.04"  
                },  
                "Notes": "any note",  
                "IssuerContactInfo": "... (many fields - see IES)",  
                "CardholderContactInfo": "... (many fields - see IES)"  
            }  
        },  
        "CollaborationAttachment": {  
            "AttachType": "Mime",  
            "Attachment": [  
                {  
                    "id": 1,  
                    "ContentType": "application/pdf",  
                    "Comment": "any comment",  
                    "ImageData": "image data"  
                    "DocType": "UCLET"  
                }, {  
                    "id": 2,  
                    "ContentType": "application/pdf",  
                    "Comment": "any comment",  
                    "ImageData": "image data"  
                    "DocType": "SHREC"  
                }  
            ]  
        }  
    }  
}
```

## Request Message—Respond to Collaboration (Recipient)—REST-based Services

The following examples show a sample SISubmitCollaborationRequest message used to respond to collaboration as a recipient, depending on the collaboration type and response option.

### Response—Accept Full (Good Faith only)

```
{  
    "RequestHeader": "...",  
    "requestData":{  
        "VisaCaseNumber": 1000000000,  
        "CollaborationId": 123456789,  
        "MemberCaseNumber": 555555555555,  
        "CollaborationDescriptor":{  
            "id": 0,  
            "operation": "Add",  
            "action": "Submit",  
            "TransactionInfo": "...",  
            "CollaborationResp":{  
                "ResponseTypeCode": "AF",  
                "AcceptanceAmount":{  
                    "currency": 840,  
                    "Text": "123.45"  
                },  
                "PaymentMethodCode": "OTHER",  
                "AmountSentDate": "2015-03-02",  
                "PaymentInstructions": "enter if \"how will amount be sent\"=OTHER",  
                "Notes": "any note",  
                "AcquirerContactInfo": "... (many fields - see IES)",  
                "MerchantContactInfo": ""  
            }  
        }  
    },  
    "CollaborationAttachment": "(See 'Request Message—Initiate or Continue  
Collaboration (Initiator)' for fields)"  
}
```

**Response—Accept Partial (Good Faith only)**

```
{  
    "RequestHeader": "...",  
    "requestData": {  
        "VisaCaseNumber": 1000000000,  
        "CollaborationId": 123456789,  
        "MemberCaseNumber": 555555555555,  
        "CollaborationDescriptor": {  
            "id": 0,  
            "operation": "Add",  
            "action": "Submit",  
            "TransactionInfo": "...",  
            "CollaborationResp": {  
                "ResponseTypeCode": "AP",  
                "AcceptanceAmount": {  
                    "currency": 840,  
                    "Text": "123.45"  
                },  
                "PaymentMethodCode": "OTHER",  
                "AmountSentDate": "2015-03-02T06:09:06-07:00",  
                "PaymentInstructions": "enter if \"how will amount be sent\"=OTHER",  
                "AcceptPartialReason": "accept partial liability reason",  
                "Notes": "any note",  
                "AcquirerContactInfo": " ... (many fields - see IES)",  
                "MerchantContactInfo": " "  
            }  
        },  
        "CollaborationAttachment": "(See 'Request Message—Initiate or Continue Collaboration (Initiator)' for fields)"  
    }  
}
```

### Response—Decline (Good Faith only)

```
{  
    "RequestHeader": "...",  
    "requestData":{  
        "VisaCaseNumber": 1000000000,  
        "CollaborationId": 123456789,  
        "MemberCaseNumber": 555555555555,  
        "CollaborationDescriptor":{  
            "id": 0,  
            "operation": "Add",  
            "action": "Submit",  
            "TransactionInfo": "...",  
            "CollaborationResp":{  
                "ResponseTypeCode": "D",  
                "DeclineReason": "decline reason",  
                "Notes": "any note",  
                "AcquirerContactInfo": " ... (many fields - see IES)",  
                "MerchantContactInfo": " "  
            }  
        },  
        "CollaborationAttachment": "(See 'Request Message-Initiate or Continue  
        Collaboration (Initiator)' for fields)"  
    }  
}
```

### Response—Need More Information (Inquiry, ASRR, and Good Faith)

```
{  
    "RequestHeader": "...",  
    "requestData":{  
        "VisaCaseNumber": 1000000000,  
        "CollaborationId": 123456789,  
        "MemberCaseNumber": 555555555555,  
        "CollaborationDescriptor":{  
            "id": 0,  
            "operation": "Add",  
            "action": "Submit",  
            "TransactionInfo": "...",  
            "CollaborationResp":{  
                "ResponseTypeCode": "NMI",  
                "AdditionalInfoNeeded": "additional info",  
                "Notes": "any note",  
                "AcquirerContactInfo": " ... (many fields - see IES)",  
                "MerchantContactInfo": " "  
            }  
        },  
        "CollaborationAttachment": "(See 'Request Message-Initiate or Continue  
        Collaboration (Initiator)' for fields)"  
    }  
}
```

**Response—Inquiry Fulfilled (Inquiry and ASRR only)**

```
{
    "RequestHeader": "...",
    "requestData": {
        "VisaCaseNumber": 1000000000,
        "CollaborationId": 123456789,
        "MemberCaseNumber": 555555555555,
        "CollaborationDescriptor": {
            "id": 0,
            "operation": "Add",
            "action": "Submit",
            "TransactionInfo": "...",
            "CollaborationResp": {
                "ResponseTypeCode": "IQF",
                "FulfillmentInfo": "requested information",
                "Notes": "any note",
                "AcquirerContactInfo": "... (many fields - see IES)",
                "MerchantContactInfo": " "
            }
        },
        "CollaborationAttachment": "(See 'Request Message-Initiate or Continue Collaboration (Initiator)' for fields)"
    }
}
```

**Request Message—Recall Collaboration (Initiator only)—SOAP-based Services**

```
<SISubmitCollaborationRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <CollaborationId>123456789</CollaborationId> <- Required
        <MemberCaseNumber>555555555555</MemberCaseNumber>
        <CollaborationDescriptor id="0" operation="Add" action="Submit">
            <TransactionInfo>...</TransactionInfo>
            <Recall>
                <RecallReason>recall reason</RecallReason>
            </Recall>
        </CollaborationDescriptor>
        <CollaborationAttachment>
            (See 'Request Message-Initiate or Continue Collaboration (Initiator)' for fields)
        </CollaborationAttachment>
    </requestData>
</SISubmitCollaborationRequest>
```

**Response Message—SOAP-based Services**

```
<SISubmitCollaborationResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <CollaborationId>123456789</CollaborationId>
    </responseData>
</SISubmitCollaborationResponse>
```

**Request Message—Recall Collaboration (Initiator only)—REST-based Services**

```
{  
    "RequestHeader": "...",  
    "requestData":{  
        "VisaCaseNumber": 1000000000,  
        "CollaborationId": 123456789,  
        "MemberCaseNumber": 555555555555,  
        "CollaborationDescriptor":{  
            "id": 0,  
            "operation": "Add",  
            "action": "Submit",  
            "TransactionInfo": "...",  
            "Recall":{  
                "RecallReason": "recall reason"  
            }  
        },  
        "CollaborationAttachment": "(See 'Request Message—Initiate or Continue  
        Collaboration (Initiator)' for fields)"  
    }  
}
```

**Response Message—REST-based Services**

```
{  
    "Status": "...",  
    "responseData":{  
        "VisaCaseNumber": 1000000000,  
        "CollaborationId": 123456789  
    }  
}
```

**SIGetCollaborationDetailsRequest**

SIGetCollaborationDetailsRequest allows RTSI users to request a read-only collaboration transaction as an image (TIFF or PDF) in addition to the collaboration data in XML. To retrieve the collaboration as an image, the user must set `<IncludeCollaborationImageInd>=true`. The image data will be returned in the `<CollaborationAttachmentDescriptor>` in the SIGetCollaborationDetailsResponse. The image format will correspond to the member's preferred image format as configured by the Visa Administrator in the Member Preferences Screen.

In order to retrieve a collaboration for an existing case, SIGetCollaborationDetailsRequest requires the VROL Case Number and CollaborationId for the collaboration.

The SIGetCollaborationDetailsRequest operation contains request and response messages as shown in the following sample messages.

**Note:** *The only image data returned is for the image associated with `<IncludeCollaborationImageInd>=true`. Any attached documents must be retrieved using the AttachmentId in this response in the DocId field of SIGetImageRequest.*

## Request Message—SOAP-based Services

```
<SIGetCollaborationDetailsRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <CollaborationId>123456789</CollaborationId>
    <IncludeCollaborationImageInd>true
    </IncludeCollaborationImageInd>
  </requestData>
</SIGetCollaborationDetailsRequest>
```

## Response Message—Initiate or Continue Collaboration (Initiator)—SOAP-based Services

**Note:** The Response Message sample below is for the initiate or continue collaboration. For a collaboration response or recall, the section <Collaboration> and associate fields will be replaced by <CollaborationResp> or <Recall> as appropriate. The fields in these sections are the same with the similar <CollaborationResp> or <Recall> sections in [SISubmitCollaborationRequest](#).

```
<SIGetCollaborationDetailsResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <CollaborationId>123456789</CollaborationId>
    <MemberCaseNumber>555555555555</MemberCaseNumber>
    <TransactionInfo>...</TransactionInfo>
    <Collaboration>
      <CollaborationType>IQ</CollaborationType>
      <Notes>any note</Notes>
      <LastDayToRespond>2015-03-02T06:09:06-07:00</LastDayToRespond>
      <Source>C</Source>
      <CollaborationReason>request reason
      </CollaborationReason> <- Required if
      this is an IQ or ASRR collaboration
      <CBRepretime frameExpiredInd>true
      </CBRepretime frameExpiredInd> <- Required if
      this is a GF collaboration
      <CardholderMerchantNotificationOutsideOfCBRepretime frameInd>true
      </CardholderMerchantNotificationOutsideOfCBRepretime frameInd>
      <- Required if this is a GF collaboration
      <ProvideInResponse>response</ProvideInResponse>
      <CollaborationAmount currency="840">42.04
      </CollaborationAmount>
      <IssuerContactInfo> <- or AcquirerContactInfo,
      if sending member is Acquirer
      ... (many fields - see IES)
      </IssuerContactInfo>
      <CardholderContactInfo> <- or <MerchantContactInfo>,
      if sending member is Acquirer
      ... (many fields - see IES)
      </CardholderContactInfo>
    </Collaboration>
    <SubmittedBy>Issuer</SubmittedBy>
    <EventDate>2015-03-02T06:09:06-07:00</EventDate>
    <Image> <- repeated for multiple attached docs
```

```
...
<AttachmentId/> <- used to retrieve attached images using
SIGetImageRequest
...
</Image>
<CollaborationAttachmentDescriptor>
  <ContentType>application/pdf</ContentType>
  <ImageData/>
</CollaborationAttachmentDescriptor>
<CollaborationHistory> <- history of Collaboration
  transactions in thread
    <CollaborationEvent>RQ</CollaborationEvent>
    <- Domain values: RQ (initiate/continue), RP (respond),
      RC (recall)
    <CollaborationId>123456789</CollaborationId>
    <SubmittedBy>Issuer</SubmittedBy>
    <EventDate>2015-02-10T06:09-06-07:00</EventDate>
  </CollaborationHistory>
  <CollaborationHistory>
    <CollaborationEvent>RC</CollaborationEvent>
    <CollaborationId>123456789</CollaborationId>
    <SubmittedBy>Issuer</SubmittedBy>
    <EventDate>2015-02-23T06:09-06-07:00</EventDate>
  </CollaborationHistory>
</ResponseData>
</SIGetCollaborationDetailsResponse>
```

**Response Message—Respond to Collaboration (Recipient)—SOAP-based Services**

```
<SIGetCollaborationDetailsResponse>
    <Status>...</Status>
    <responseData>
        <VisaCaseNumber>1000000000</VisaCaseNumber>
        <CollaborationId>123456789</CollaborationId>
        <MemberCaseNumber>555555555555</MemberCaseNumber>
        <TransactionInfo>...</TransactionInfo>
        <CollaborationResp>
            <ResponseTypeCode>AF</ResponseTypeCode> <- Domain values:
                AF (Accept Full), AP (Accept Partial), D (Decline),
                NMI (Need More Information), or IQF (Inquiry Fulfilled)
                ... (many fields per <ResponseTypeCode> - see IES)
            <Notes>any note</Notes>
            <AcquirerContactInfo> <- or IssuerContactInfo,
                if sending member is Issuer
                ... (many fields - see IES)
            </AcquirerContactInfo>
        </CollaborationResp>
        <SubmittedBy>Issuer</SubmittedBy>
        <EventDate>2015-03-02T06:09:06-07:00</EventDate>
        <Image> <- repeated for multiple attached docs
            ...
            <AttachmentId/> <- used to retrieve attached images
            using SIGetImageRequest
            ...
        </Image>
        <CollaborationAttachmentDescriptor>
            <ContentType>application/pdf</ContentType>
            <ImageData/>
        </CollaborationAttachmentDescriptor>
        <CollaborationHistory> <- history of Collaboration
            transactions in thread
            <CollaborationEvent>RQ</CollaborationEvent>
            <- Domain values: RQ (initiate/continue), RP (respond),
            RC (recall)
            <CollaborationId>123456789</CollaborationId>
            <SubmittedBy>Issuer</SubmittedBy>
            <EventDate>2015-02-10T06:09:06-07:00</EventDate>
        </CollaborationHistory>
        <CollaborationHistory>
            <CollaborationEvent>RP</CollaborationEvent>
            <CollaborationId>123456789</CollaborationId>
            <SubmittedBy>Issuer</SubmittedBy>
            <EventDate>2015-02-23T06:09:06-07:00</EventDate>
        </CollaborationHistory>
    </responseData>
</SIGetCollaborationDetailsResponse>
```

## Response Message—Recall Collaboration—SOAP-based Services

```
<SIGetCollaborationDetailsResponse>
  <Status>...</Status>
  <responseData>
    <VisaCaseNumber>1000000000</VisaCaseNumber>
    <CollaborationId>123456789</CollaborationId>
    <MemberCaseNumber>555555555555</MemberCaseNumber>
    <TransactionInfo>...</TransactionInfo>
    <Recall>
      <RecallReason>enter reason for recall</RecallReason>
    </Recall>
    <SubmittedBy>Issuer</SubmittedBy>
    <EventDate>2015-03-02T06:09:06-07:00</EventDate>
    <CollaborationAttachmentDescriptor>
      <ContentType>application/pdf</ContentType>
      <ImageData/>
    </CollaborationAttachmentDescriptor>
    <CollaborationHistory> <- history of Collaboration
      transactions in thread
      <CollaborationEvent>RQ</CollaborationEvent>
      <- Domain values: RQ (initiate/continue), RP (respond),
      RC (recall)
      <CollaborationId>123456789</CollaborationId>
      <SubmittedBy>Issuer</SubmittedBy>
      <EventDate>2015-02-10T06:09:06-07:00</EventDate>
    </CollaborationHistory>
  </responseData>
</SIGetCollaborationDetailsResponse>
```

## Request Message—REST-based Services

```
{
  "RequestHeader": "...",
  "requestData":{
    "VisaCaseNumber": 1000000000,
    "CollaborationId": 123456789,
    "IncludeCollaborationImageInd": "true"
  }
}
```

**Response Message—Initiate or Continue Collaboration (Initiator)—REST-based Services**

```
{  
    "Status": "...",  
    "responseData":{  
        "VisaCaseNumber": 1000000000,  
        "CollaborationId": 123456789,  
        "MemberCaseNumber": 555555555555,  
        "TransactionInfo": "...",  
        "Collaboration":{  
            "CollaborationType": "IQ",  
            "Notes": "any note",  
            "LastDayToRespond": "2015-03-02T06:09:06-07:00",  
            "Source": "C",  
            "CollaborationReason": "request reason",  
            "CBRepretime frameExpiredInd": "true",  
            "CardholderMerchantNotificationOutsideOfCBRepretime frameInd": "true",  
            "ProvideInResponse": "response",  
            "CollaborationAmount":{  
                "currency": 840,  
                "Text": "42.04"  
            },  
            "IssuerContactInfo": "... (many fields - see IES)",  
            "CardholderContactInfo": "... (many fields - see IES)"  
        },  
        "SubmittedBy": "Issuer",  
        "EventDate": "2015-03-02T06:09:06-07:00",  
        "Image":{  
            "Text": "...",  
            "AttachmentId": "",  
            "Text": "..."  
        },  
        "CollaborationAttachmentDescriptor":{  
            "ContentType": "application/pdf",  
            "ImageData": ""  
        },  
        "CollaborationHistory": [  
            {  
                "CollaborationEvent": "RQ",  
                "CollaborationId": 123456789,  
                "SubmittedBy": "Issuer",  
                "EventDate": "2015-02-10T06:09:06-07:00"  
            }, {  
                "CollaborationEvent": "RC",  
                "CollaborationId": 123456789,  
                "SubmittedBy": "Issuer",  
                "EventDate": "2015-02-23T06:09:06-07:00"  
            }  
        ]  
    }  
}
```

### Response Message—Respond to Collaboration (Recipient)—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "VisaCaseNumber": 1000000000,  
        "CollaborationId": 123456789,  
        "MemberCaseNumber": 555555555555,  
        "TransactionInfo": "...",  
        "CollaborationResp": {  
            "ResponseTypeCode": "AF",  
            "Notes": "any note",  
            "AcquirerContactInfo": " ... (many fields - see IES)"  
        },  
        "SubmittedBy": "Issuer",  
        "EventDate": "2015-03-02T06:09:06-07:00",  
        "Image": {  
            ...,  
            "AttachmentId": "",  
            ...  
        },  
        "CollaborationAttachmentDescriptor": {  
            "ContentType": "application/pdf",  
            "ImageData": ""  
        },  
        "CollaborationHistory": [  
            {  
                "CollaborationEvent": "RQ",  
                "CollaborationId": 123456789,  
                "SubmittedBy": "Issuer",  
                "EventDate": "2015-02-10T06:09:06-07:00"  
            }, {  
                "CollaborationEvent": "RP",  
                "CollaborationId": 123456789,  
                "SubmittedBy": "Issuer",  
                "EventDate": "2015-02-23T06:09:06-07:00"  
            }  
        ]  
    }  
}
```

**Response Message—Recall Collaboration—REST-based Services**

```
{  
    "Status": "...",  
    "responseData": {  
        "VisaCaseNumber": 1000000000,  
        "CollaborationId": 123456789,  
        "MemberCaseNumber": 555555555555,  
        "TransactionInfo": "...",  
        "Recall": {  
            "RecallReason": "enter reason for recall"  
        },  
        "SubmittedBy": "Issuer",  
        "EventDate": "2015-03-02T06:09:06-07:00",  
        "CollaborationAttachmentDescriptor": {  
            "ContentType": "application/pdf",  
            "ImageData": ""  
        },  
        "CollaborationHistory": {  
            "CollaborationEvent": "RQ",  
            "CollaborationId": 123456789,  
            "SubmittedBy": "Issuer",  
            "EventDate": "2015-02-10T06:09:06-07:00"  
        }  
    }  
}
```

# Dispute Quick Forms

23

Dispute quick forms enable issuers to quickly create and submit dispute questionnaires.

When a dispute quick form is submitted, VROL searches first for an existing case using the matching information from the quick form. If a unique matching case is not found, VROL performs a transaction inquiry (TI) to find a matching original transaction, and creates a new case using the results of the TI.

Most of the needed original transaction fields can be pulled from the Visa transaction repository. The efficiency comes from the system performing case lookup and/or case creation steps done automatically in the background based on a minimal set of input data.

**Note:** *VROL does not support quick forms for dispute response questionnaires or dispute-related Pre-Filing/Pre-Filing response questionnaires.*

## Dispute Quick Form Process

### Initiate Dispute Quick Form

1. To initiate a quick form for a dispute questionnaire, the member host system triggers a SISubmitQFDisputeQuestionnaireRequest message supplying the following required information:
  - Transaction data (using either TransactionID or ARN as a required field)
  - Dispute questionnaire data (including the DisputeCategory along with all required dispute questionnaire fields and ancillary items such as Fraud Reports for Fraud disputes, Provisional Credits if required based on VROL Member Configuration, and so on)

SISubmitQFDisputeQuestionnaireRequest can be saved or submitted. VROL returns a SISubmitQFDisputeQuestionnaireResponse message with the QuickFormID and QuickFormType.

- Save  
SISubmitQFDisputeQuestionnaireRequest message can be saved. When it is saved, it generates a QuickFormID.
- Submit

If a SISubmitQFDisputeQuestionnaireRequest message is submitted for a new dispute quick form, a QuickFormID is generated in the response.

If a SISubmitQFDisputeQuestionnaireRequest message is submitted for a previously-saved dispute quick form, there is no new QuickFormID generated in the response.

**Important:**

Once transaction information is established in a case, there is no need to provide it in subsequent actions. In addition, <TransactionInfo> is no longer required if a case is created from a TI result.

2. To attach one or more images to the dispute questionnaire, include the DisputeAttachmentDescriptor group element in the request message.  
Members can also use the Bulk SI Hybrid Image upload functionality to generate DocIDs in bulk which can be supplied in the SISubmitQFDisputeQuestionnaireRequest against the DocId element.  
For details, see [Hybrid Image Uploads](#) and *Visa Resolve Online Bulk Systems Interface Development Guide*.
  3. Using the supplied transaction information, VROL attempts to match the case to an existing one or creates a new case through Transaction Inquiry, as follows:
    - a. First, VROL attempts to find a unique, existing matching case. If a unique case is found, VROL then searches for associated transactions and ancillary items (such as Fraud Reports, Exception File Listings, Stop Payment Orders, and/or Provisional Credits) that are applicable to the dispute category.
    - b. If search (a) is unsuccessful, VROL searches for a single matching TI (settlement original or adjustment, but not authorization). If a matching TI transaction is found, a case is created from the TI result. VROL then searches for associated transactions and ancillary items (such as Fraud Reports, Exception File Listings, Stop Payment Orders, and/or Provisional Credits) that are applicable to the dispute category.
- If both (a) and (b) searches are unsuccessful, or if VROL found associated transactions but could not default the matching selections, the dispute quick form is placed in the MY\_QUICKFORM\_EXCEPTION or ALL\_QUICKFORM\_EXCEPTION queue.
4. VROL applies validation edits and determines if the required conditions are met to submit the dispute questionnaire.
    - If the dispute passes the validation edits, proceed to step 5.
    - If the dispute fails one or more validation edits, VROL places the dispute quick form in the MY\_QUICKFORM\_EXCEPTION or ALL\_QUICKFORM\_EXCEPTION. VROL also places the dispute questionnaire in the MY\_REJECTS or ALL\_REJECTS queue.
  5. VROL creates the dispute questionnaire and submits or saves it as specified in the request message, along with any associated transactions and ancillary items included within the dispute questionnaire.
- VROL returns a SISubmitQFDisputeQuestionnaireResponse message with the QuickFormID and QuickFormType.

## Repair Dispute Quick Form

If VROL encounters any error while processing a dispute quick form, VROL places the dispute quick form in the MY\_QUICKFORM\_EXCEPTION or ALL\_QUICKFORM\_EXCEPTION queue. Users can repair and resubmit dispute quick forms from these queues.

VROL places a dispute quick form in the exception queue during certain scenarios, including but not limited to the following:

- VROL found associated transactions but could not default the matching selections.
- The dispute questionnaire failed one or more validation edits.
- User did not supply a Fraud Report with a Fraud dispute.
- User did not supply an identified duplicate transaction ID with a Processing Error or duplicate dispute.

1. To check for rejected dispute quick forms, the member host system triggers a SIGetQueueRequest message for MY\_QUICKFORM\_EXCEPTION or ALL\_QUICKFORM\_EXCEPTION queue.

VROL returns a SIGetQueueResponse message with the VisaCaseNumber and QuickFormID.

2. To retrieve the Quick Form details using the QuickFormID, the member host system triggers a Hypersearch request. See [Chapter 19, Hypersearch Requests](#), for more information.

VROL returns the QuickFormID along with its QuickFormType.

3. To repair a rejected dispute quick form, the member host system triggers a SISubmitQFDisputeQuestionnaireRequest message supplying the QuickFormID of the rejected dispute quick form along with the updated transaction identification information as appropriate for the reject reason.

VROL returns a SISubmitQFDisputeQuestionnaireResponse message with the success status.

If a dispute quick form encountered a problem during associated transaction matching, VROL makes the Associated Transaction Selection screen available in the UI's quick form exception queue. This allows the user to modify and save the associated transaction selections. This action can be performed only through the UI.

## Dispute Quick Form Operation

RTSI exposes the following dispute quick form operation:

**SISubmitQFDisputeQuestionnaireRequest**—Used to initiate a quick form for a dispute questionnaire.

### SISubmitQFDisputeQuestionnaireRequest

SISubmitQFDisputeQuestionnaireRequest allows RTSI users to initiate a quick form for a dispute questionnaire. This request can include image(s).

The following elements should be considered when initiating a dispute quick form or receiving a response from this service.

Element	Description
DisputeCategory	<p>Indicates the category of the dispute.</p> <p>Domain values:</p> <ul style="list-style-type: none"><li>• 10=Fraud</li><li>• 11=Authorization</li><li>• 12=Processing Error</li><li>• 13=Consumer</li></ul>
QuickFormID	A unique VROL-generated ID for each saved or submitted dispute quick form.

**SISubmitQFDisputeQuestionnaireRequest** contains request and response messages as shown in the following sample messages.

## Request Message—Fraud—SOAP-based Services

```
<SISubmitQFDisputeQuestionnaireRequest>
<RequestHeader>...</RequestHeader>
<requestData>
    <TransactionID>a</TransactionID>
    <ARN>00000000000000000000000000000000</ARN>
    <AccountNumber>000000000000</AccountNumber>
    <DisputeCategory>10</DisputeCategory>
    <MemberCaseNumber>a</MemberCaseNumber>
    <QuickFormID>0</QuickFormID>
    <QuickFormType>Dispute</QuickFormType>
    <CardholderDidNotAuthorizeOrParticipate>true
    </CardholderDidNotAuthorizeOrParticipate>
    <IssuerDisputeReqInfo>
        <DisputeAmount currency="000">0</DisputeAmount>
        <DisputeAmountChangeReason>a</DisputeAmountChangeReason>
        <CreateDisputeFinancial>Y</CreateDisputeFinancial>
        <TransactionMessageFormat>B</TransactionMessageFormat>
    </IssuerDisputeReqInfo>
    <NotesAndDocumentsInfo>
        <Note>a</Note>
        <DisputeAttachmentDescriptor>
            <Attachment id="String">
                <ContentType>image/tiff</ContentType>
                <Comment>a</Comment>
                <ImageData>xxxxxx</ImageData>
                <DocType>PRDEL</DocType>
            </Attachment>
        </DisputeAttachmentDescriptor>
    </NotesAndDocumentsInfo>
    <TransmittalInfo>
        <DisputeFraudInfo>
            <MailDate>2016-08-13</MailDate>
            <MailCity>a</MailCity>
            <MailStateCode>a</MailStateCode>
            <ValidFrom>0100</ValidFrom>
            <MailPostalCode>a</MailPostalCode>
            <FraudType>0</FraudType>
        </DisputeFraudInfo>
    </TransmittalInfo>
</requestData>
</SISubmitQFDisputeQuestionnaireRequest>
```

```
<ElaborationInfo>
    <ExplanationOfCreditPresented>a
    </ExplanationOfCreditPresented>
    <Fraud>
        <FraudType>0</FraudType>
        <CardStatusWhenTransactionHappened>0
        </CardStatusWhenTransactionHappened>
        <FraudOtherExplanation>a</FraudOtherExplanation>
        <PinPreferringCard>N</PinPreferringCard>
        <PINPadPresentWorking>N</PINPadPresentWorking>
        <ChipOnCard>N</ChipOnCard>
        <CardholderDidNotAuthorizeOrParticipate>true
        </CardholderDidNotAuthorizeOrParticipate>
    </Fraud>
</ElaborationInfo>
<Action>Submit</Action>
</requestData>
</SISubmitQFDisputeQuestionnaireRequest>
```

## Request Message—Authorization—SOAP-based Services

```
<SISubmitQFDisputeQuestionnaireRequest>
<RequestHeader>...</RequestHeader>
<requestData>
    <TransactionID>a</TransactionID>
    <ARN>00000000000000000000000000000000</ARN>
    <AccountNumber>00000000000000000000000000000000</AccountNumber>
    <DisputeCategory>11</DisputeCategory>
    <MemberCaseNumber>a</MemberCaseNumber>
    <QuickFormID>0</QuickFormID>
    <QuickFormType>Dispute</QuickFormType>
    <IssuerDisputeReqInfo>
        <DisputeAmount currency="000">0</DisputeAmount>
        <DisputeAmountChangeReason>a</DisputeAmountChangeReason>
        <CreateDisputeFinancial>Y</CreateDisputeFinancial>
        <TransactionMessageFormat>B</TransactionMessageFormat>
    </IssuerDisputeReqInfo>
    <NotesAndDocumentsInfo>
        <Note>a</Note>
        <DisputeAttachmentDescriptor>
            <Attachment id="String">
                <ContentType>image/tiff</ContentType>
                <Comment>a</Comment>
                <ImageData>xxxxxx</ImageData>
                <DocType>PRDEL</DocType>
            </Attachment>
        </DisputeAttachmentDescriptor>
    </NotesAndDocumentsInfo>
    <ElaborationInfo>
        <ExplanationOfCreditPresented>a
        </ExplanationOfCreditPresented>
        <Authorization>
            <EFLDate>2016-08-13</EFLDate>
            <Region>E</Region>
            <WasAuthObtainedUsingInvalidFallbackData>N
            </WasAuthObtainedUsingInvalidFallbackData>
            <ExplainWhatWasInvalid>a</ExplainWhatWasInvalid>
            <ExplanationOfAuthorizationsPresented>I
            </ExplanationOfAuthorizationsPresented>
            <AuthOtherExplanation>a</AuthOtherExplanation>
        </Authorization>
    </ElaborationInfo>
    <Action>Submit</Action>
</requestData>
</SISubmitQFDisputeQuestionnaireRequest>
```

## Request Message—Processing Error—SOAP-based Services

```
<SISubmitQFDisputeQuestionnaireRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <TransactionID>a</TransactionID>
        <ARN>00000000000000000000000000000000</ARN>
        <AccountNumber>000000000000</AccountNumber>
        <DisputeCategory>12</DisputeCategory>
```

```
<MemberCaseNumber>a</MemberCaseNumber>
<QuickFormID>0</QuickFormID>
<QuickFormType>Dispute</QuickFormType>
<IssuerDisputeReqInfo>
    <DisputeAmount currency="000">0</DisputeAmount>
    <DisputeAmountChangeReason>a</DisputeAmountChangeReason>
    <CreateDisputeFinancial>Y</CreateDisputeFinancial>
    <TransactionMessageFormat>B</TransactionMessageFormat>
</IssuerDisputeReqInfo>
<NotesAndDocumentsInfo>
    <Note>a</Note>
    <DisputeAttachmentDescriptor>
        <Attachment id="String">
            <ContentType>image/tiff</ContentType>
            <Comment>a</Comment>
            <ImageData>xxxxxx</ImageData>
            <DocType>PRDEL</DocType>
        </Attachment>
    </DisputeAttachmentDescriptor>
</NotesAndDocumentsInfo>
<ElaborationInfo>
    <ExplanationOfCreditPresented>a
    </ExplanationOfCreditPresented>
    <ProcessingError>
        <WhatIncorrectAboutTransaction>LP
        </WhatIncorrectAboutTransaction>
        <AccountStatus>OF</AccountStatus>
        <CorrectCurrency>840</CorrectCurrency>
        <CardholderAgreeDCCAmtInd>Y</CardholderAgreeDCCAmtInd>
        <WhatWasDCCProblem>CHNA</WhatWasDCCProblem>
        <TransactionCodeIncorrect>DCR
        </TransactionCodeIncorrect>
        <IssuerMasterFileAccountNumberInd>N
        </IssuerMasterFileAccountNumberInd>
        <MatchingAuthTranId>0</MatchingAuthTranId>
        <AuthorizationDate>2016-08-13</AuthorizationDate>
        <CardholderReceiptAccountNumber>String
        </CardholderReceiptAccountNumber>
        <ChipCardContainValidCryptogramInd>Y
        </ChipCardContainValidCryptogramInd>
        <ExplainWhatIsInvalid>a</ExplainWhatIsInvalid>
        <ReceiptAccountNumberMatchInd>N
        </ReceiptAccountNumberMatchInd>
        <CardholderReceiptAmt currency="000">0
        </CardholderReceiptAmt>
        <MerchantDisputePriceDiffInd>N
        </MerchantDisputePriceDiffInd>
        <BothTransactionsOnSameAcctNumberInd>N
        </BothTransactionsOnSameAcctNumberInd>
        <IssuerOrCardholderOtherAcctWithDiffVisaCardInd>Y
        </IssuerOrCardholderOtherAcctWithDiffVisaCardInd>
        ...
    </ProcessingError>
</ElaborationInfo>
<Action>Submit</Action>
</requestData>
</SISubmitQFDisputeQuestionnaireRequest>
```

**Request Message—Consumer—SOAP-based Services**

```
<SISubmitQFDisputeQuestionnaireRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <TransactionID>a</TransactionID>
        <ARN>00000000000000000000000000000000</ARN>
        <AccountNumber>000000000000</AccountNumber>
        <DisputeCategory>13</DisputeCategory>
        <MemberCaseNumber>a</MemberCaseNumber>
        <QuickFormID>0</QuickFormID>
        <QuickFormType>Dispute</QuickFormType>
        <IssuerDisputeReqInfo>
            <DisputeAmount currency="000">0</DisputeAmount>
            <DisputeAmountChangeReason>a</DisputeAmountChangeReason>
            <CreateDisputeFinancial>Y</CreateDisputeFinancial>
            <TransactionMessageFormat>B</TransactionMessageFormat>
        </IssuerDisputeReqInfo>
        <NotesAndDocumentsInfo>
            <Note>a</Note>
            <DisputeAttachmentDescriptor>
                <Attachment id="String">
                    <ContentType>image/tiff</ContentType>
                    <Comment>a</Comment>
                    <ImageData>xxxxxx</ImageData>
                    <DocType>PRDEL</DocType>
                </Attachment>
            </DisputeAttachmentDescriptor>
        </NotesAndDocumentsInfo>
        <ElaborationInfo>
            <ExplanationOfCreditPresented>a
            </ExplanationOfCreditPresented>
            <Consumer>
                <DisputeDueTo>DM</DisputeDueTo>
                <AccountClosureDate>2016-08-13</AccountClosureDate>
                <MerchandiseOrServices>Merchandise
                </MerchandiseOrServices>
                <CardholderReturnMerchandiseInd>Y
                </CardholderReturnMerchandiseInd>
                <MerchandiseNotReturned>String</MerchandiseNotReturned>
                <MerchantRefuseAcceptMerchandiseInd>N
                </MerchantRefuseAcceptMerchandiseInd>
                <MerchantRefuseAdvise>MRRA</MerchantRefuseAdvise>
                <WhatWasPurchased>a</WhatWasPurchased>
                <MerchandiseReceivedDate>2016-08-13
                </MerchandiseReceivedDate>
                <HowMerchandiseOrServiceMisrepresented>a
                </HowMerchandiseOrServiceMisrepresented>
                <DidCardholderCancel>Y</DidCardholderCancel>
                <CancellationMethod>a</CancellationMethod>
                <CardholderFirstNotifiedIssuerDisputeDate>2016-08-13
                </CardholderFirstNotifiedIssuerDisputeDate>
                <CardholderAttemptToResolve>N
                </CardholderAttemptToResolve>
            ...
        </Consumer>
    </requestData>
</SISubmitQFDisputeQuestionnaireRequest>
```

```
</Consumer>
</ElaborationInfo>
<Action>Submit</Action>
</requestData>
</SISubmitQFDisputeQuestionnaireRequest>
```

### **Response Message—SOAP-based Services**

```
<SISubmitQFDisputeQuestionnaireResponse>
  <Status>...</Status>
  <responseData>
    <QuickFormID>0</QuickFormID>
    <QuickFormType>Dispute</QuickFormType>
  </responseData>
</SISubmitQFDisputeQuestionnaireResponse>
```

**Request Message—Fraud—REST-based Services**

```
{  
  "RequestHeader": {  
    "MemberRole": "I"  
  },  
  "requestData": {  
    "TransactionID": "0000000000000000",  
    "ARN": "00000000000000000000000000000000",  
    "AccountNumber": "0000000000000000",  
    "DisputeCategory": {  
      "value": "10"  
    },  
    "MemberCaseNumber": "I0000000000",  
    "QuickFormType": "Dispute",  
    "IssuerDisputeReqInfo": {  
      "DisputeAmount": {  
        "currency": "344",  
        "value": 25000  
      },  
      "DisputeAmountChangeReason": "a",  
      "CreateDisputeFinancial": "Y",  
      "TransactionMessageFormat": "B"  
    },  
    "TransmittalInfo": {  
      "DisputeFraudInfo": {  
        "MailDate": "2017-04-04",  
        "MailCity": "NEW ZEALAND 8",  
        "MailStateCode": "NZ",  
        "ValidFrom": "0100",  
        "MailPostalCode": "00000000",  
        "FraudType": "0"  
      }  
    },  
    "ElaborationInfo": {  
      "ExplanationOfCreditPresented": "a",  
      "Fraud": {  
        "FraudType": "0",  
        "CardStatusWhenTransactionHappened": "O",  
        "FraudOtherExplanation": "a",  
        "PinPreferringCard": "N",  
        "ChipOnCard": "N"  
      }  
    }  
  }  
}
```

### Request Message—Authorization—REST-based Services

```
{  
    "RequestHeader": {  
        "MemberRole": "I"  
    },  
    "requestData": {  
        "TransactionID": "0000000000000000",  
        "ARN": "00000000000000000000000000000000",  
        "AccountNumber": "0000000000000000",  
        "DisputeCategory": {  
            "value": "11"  
        },  
        "MemberCaseNumber": "I0000000000",  
        "QuickFormType": "Dispute",  
        "IssuerDisputeReqInfo": {  
            "DisputeAmount": {  
                "currency": "344",  
                "value": 25000  
            },  
            "DisputeAmountChangeReason": "a",  
            "CreateDisputeFinancial": "Y",  
            "TransactionMessageFormat": "B"  
        },  
        "TransmittalInfo": {  
            "DisputeFraudInfo": {  
                "MailDate": "2017-04-04",  
                "MailCity": "NEW ZEALAND 8",  
                "MailStateCode": "NZ",  
                "ValidFrom": "0100",  
                "MailPostalCode": "00000000",  
                "FraudType": "0"  
            }  
        },  
        "ElaborationInfo": {  
            "ExplanationOfCreditPresented": "a",  
            "Authorization": {  
            }  
        }  
    }  
}
```

**Request Message—Processing Error—REST-based Services**

```
{  
  "RequestHeader": {  
    "MemberRole": "I"  
  },  
  "requestData": {  
    "TransactionID": "0000000000000000",  
    "ARN": "000000000000000000000000000000",  
    "AccountNumber": "0000000000000000",  
    "DisputeCategory": {  
      "value": "12"  
    },  
    "MemberCaseNumber": "I0000000000",  
    "QuickFormType": "Dispute",  
    "IssuerDisputeReqInfo": {  
      "DisputeAmount": {  
        "currency": "344",  
        "value": 25000  
      },  
      "DisputeAmountChangeReason": "a",  
      "CreateDisputeFinancial": "Y",  
      "TransactionMessageFormat": "B"  
    },  
    "TransmittalInfo": {  
      "DisputeFraudInfo": {  
        "MailDate": "2017-04-04",  
        "MailCity": "NEW ZEALAND 8",  
        "MailStateCode": "NZ",  
        "ValidFrom": "0100",  
        "MailPostalCode": "00000000",  
        "FraudType": "0"  
      }  
    },  
    "ElaborationInfo": {  
      "ExplanationOfCreditPresented": "a",  
      "ProcessingError": {  
        "WhatIncorrectAboutTransaction": "ID"  
      }  
    }  
  }  
}
```

## Request Message—Consumer—REST-based Services

```
{  
    "RequestHeader": {  
        "MemberRole": "I"  
    },  
    "requestData": {  
        "TransactionID": "0000000000000000",  
        "ARN": "00000000000000000000000000000000",  
        "AccountNumber": "0000000000000000",  
        "DisputeCategory": {  
            "value": "13"  
        },  
        "MemberCaseNumber": "I0000000000",  
        "QuickFormType": "Dispute",  
        "IssuerDisputeReqInfo": {  
            "DisputeAmount": {  
                "currency": "344",  
                "value": 25000  
            },  
            "DisputeAmountChangeReason": "a",  
            "CreateDisputeFinancial": "Y",  
            "TransactionMessageFormat": "B"  
        },  
        "TransmittalInfo": {  
            "DisputeFraudInfo": {  
                "MailDate": "2017-04-04",  
                "MailCity": "NEW ZEALAND 8",  
                "MailStateCode": "NZ",  
                "ValidFrom": "0100",  
                "MailPostalCode": "00000000",  
                "FraudType": "0"  
            }  
        },  
        "ElaborationInfo": {  
            "ExplanationOfCreditPresented": "a",  
            "Consumer": {  
                "DisputeDueTo": "DM",  
                "CardholderReturnMerchandiseInd": "N",  
                "MerchandiseReceivedDate": "2017-04-04",  
                "DidCardholderCancel": "N",  
                "CardholderAttemptToResolve": "N",  
                "CardholderAttemptReturnMerchandise": "N",  
                "DamagedOrDefectiveOrderInfo": "100000"  
            }  
        }  
    }  
}
```

**Response Message—REST-based Services**

```
{  
  "Status": [ {  
    "Code": "I-0000000000",  
    "Message": "Successfully completed Operation."  
  } ],  
  "responseData": {  
    "QuickFormID": 000000,  
    "QuickFormType": "Dispute"  
  }  
}
```

# RTSI Report Distribution

24

The RTSI Report Distribution provides RTSI members the capability to access the details of their scheduled report or to request a report data download of their completed scheduled reports in CSV format.

## RTSI Report Distribution Process

1. To retrieve the details of all the scheduled reports for the user's organization, the member host system triggers a **SIGetActiveScheduleRequest** message.

Alternatively, to retrieve the details of a particular report, the member host system triggers a **SIGetActiveScheduleRequest** message supplying the **ReportName** with the scheduled report name.

VROL returns a **SIGetActiveScheduleResponse** message with the details of the scheduled report(s) and success status.

2. To retrieve a scheduled report for a particular run date and in CSV format, the member host system triggers a **SIRetrieveReportOutputRequest** message supplying the **ScheduleId**, **RunDate**, and **ReportDataFileType=CSV**.

**Note:** *User can retrieve the scheduled report data for the past 5 particular run dates.*

VROL returns a **SIRetrieveReportOutputResponse** message with the details of the scheduled report(s) for a particular run date in CSV format and success status.

## Report Distribution Operations

RTSI exposes the following Report Distribution operations:

**SIGetActiveScheduleRequest**—Used to retrieve details of all the scheduled reports for the user's organization or for a particular report that the user wants to know.

**SIRetrieveReportOutputRequest**—Used to retrieve the report for a particular run date in CSV format. For download, only the CSV format of the report is supported by this Web service.

## SIGetActiveScheduleRequest

The SIGetActiveScheduleRequest operation enables a user to retrieve details of all the scheduled reports for the user's organization or for a particular report that the user wants to know.

The following elements should be considered when retrieving the details of scheduled report(s) or receiving a response from this service.

Element	Description
ReportName	Contains the name of the scheduled report for the user's organization.
ScheduleInformation	<p>Provides additional details such as the following:</p> <ul style="list-style-type: none"> <li>• Time Zone - Refers to the time zone for which the report was scheduled.</li> <li>• Run Time - Refers to the schedule that the report will be executed.</li> <li>• Start Date - Refers to the date on which the schedule will start.</li> <li>• Frequency - Refers to how much is the frequency of the schedule (daily, weekly, or monthly).</li> <li>• Schedule Name - Refers to the report schedule name.</li> <li>• Day of run - Refers to the day for which the report is scheduled to execute on a weekly or monthly basis.</li> </ul>
ScheduleId	A unique ID for the scheduled report.

### Request Message—SOAP-based Services

```
<SIGetActiveScheduleRequest>
    <RequestHeader>...</RequestHeader>
    <requestData>
        <reportName>ROL161</reportName>
    </requestData>
</SIGetActiveScheduleRequest>
```

### Response Message—SOAP-based Services

```
<SIGetActiveScheduleResponse>
    <status>...</status>
    <responseData>
        <reportScheduleList>
            <scheduleDetail>
                <reportName>ROL161</reportName>
                <scheduleId>100000</scheduleId>
                <scheduleName>String</scheduleName>
                <reportLongName>String</reportLongName>
                <scheduleInformation>String</scheduleInformation>
                <CreatedBy>String</CreatedBy>
                <CreateDate>2016-08-13</CreateDate>
                <LastRunDate>2016-08-13</LastRunDate>
            </scheduleDetail>
        </reportScheduleList>
    </responseData>
</SIGetActiveScheduleResponse>
```

## Request Message—REST-based Services

```
{
  "RequestHeader": {
    "User": {
      "@id": "a",
      "@type": "internalId"
    },
    "MemberRole": "I"
  },
  "requestData": {
    "ReportName": "ROL161"
  }
}
```

## Response Message—REST-based Services

```
{
  "Status": {
    "Code": "a",
    "Message": "a",
    "ErrorIn": "String"
  },
  "responseData": {
    "ReportScheduleList": {
      "ScheduleDetail": {
        "ReportName": "ROL161",
        "ScheduleId": 100000,
        "ScheduleName": "String",
        "ReportLongName": "String",
        "ScheduleInformation": "String",
        "CreatedBy": "String",
        "CreateDate": "2016-08-13",
        "LastRunDate": "2016-08-13"
      }
    }
  }
}
```

## SIRetrieveReportOutputRequest

The SIRetrieveReportOutputRequest operation enables users to retrieve the report for a particular run date in CSV format.

The following elements should be considered when retrieving the report for a particular run date or receiving a response from this service.

Element	Description
ScheduleId	A unique ID for the scheduled report.
RunDate	The date for which the report has ran.  <b>Note:</b> <i>The user can fetch the report data only for the past 5 runs.</i>
ReportDataFileType	Domain value: CSV.

**Request Message—SOAP-based Services**

```
<SIRetrieveReportOutputRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <ScheduleId>100000</ScheduleId>
    <RunDate>2016-08-13</RunDate>
    <ReportDataFileType>CSV</ReportDataFileType>
  </requestData>
</SIRetrieveReportOutputRequest>
```

**Response Message—SOAP-based Services**

```
<SIRetrieveReportOutputResponse>
  <Status>...</Status>
  <responseData>
    <ScheduleId>100000</ScheduleId>
    <RunDate>2016-08-13</RunDate>
    <ReportAttachment>
      <AttachType>Mime</AttachType>
      <Attachment id="String">
        <ContentType>text/csv</ContentType>
        <Comment>any comment</Comment>
        <ReportData>xxxxx</ReportData>
      </Attachment>
    </ReportAttachment>
  </responseData>
</SIRetrieveReportOutputResponse>
```

**Request Message—REST-based Services**

```
{
  "RequestHeader": {
    "User": {
      "@id": "a",
      "@type": "internalId"
    },
    "MemberRole": "A"
  },
  "requestData": {
    "ScheduleId": 100000,
    "RunDate": "2016-08-13",
    "ReportDataFileType": "CSV"
  }
}
```

## Response Message—REST-based Services

```
{  
    "Status": {  
        "Code": "a",  
        "Message": "a",  
        "ErrorIn": "String"  
    },  
    "responseData": {  
        "ScheduleId": 0,  
        "RunDate": "2017-08-13",  
        "ReportAttachment": {  
            "AttachType": "Mime",  
            "Attachment": {  
                "@id": "String",  
                "ContentType": "text/csv",  
                "Comment": "a",  
                "ReportData": "xxxxx"  
            }  
        }  
    }  
}
```

THIS PAGE INTENTIONALLY LEFT BLANK.

# Visa Fraud Monitoring Program

25

The Visa Fraud Monitoring Program (VFMP) identifies fraudulent transactions on a monthly basis, submits fraud reports outside of VROL, and then transmits a list of fraudulent transactions eligible for VFMP Fraud Disputes (10.5) back to VROL. These transactions are delivered to VROL issuers in a monthly transaction report file, with a 120 day dispute timeline starting on the date when the transaction is reported as fraudulent.

**Note:** *For VFMP Fraud Disputes (10.5), in the 10.5 dispute category/condition, "10" is the value that represents the dispute category "Fraud", while "5" is the value that represents the dispute condition (within Fraud) for Visa Fraud Monitoring Program.*

Depending on the member configuration, VROL can automatically create disputes for these transactions. Transactions for which disputes are automatically created also appear in the queue, with a VFMP status of 'Dispute Requested'. Transactions are distributed to the issuer organization queues based on entitlement to the account number of the reported transaction and initialized with a VFMP status of 'Reported'. These transactions appear in the VFMP queue for issuers to view them, allow VROL to generate dispute financials or receive dispute financial advices if the dispute financials are processed outside of VROL, or mark them as 'Ignored'.

VFMP transactions include non-ATM Visa network card transactions for which VFMP Fraud Dispute applies.

**Note:** *Global Merchant Fraud Performance (GMFP) is renamed to Visa Fraud Monitoring Program (VFMP); however, VROL still uses the old RTSI service and element names for GMFP.*

## Auto Disputes on VFMP-Reported Transactions

Issuer or dual org users can be set up in their VROL Member Configuration to allow VROL to automatically generate Fraud Dispute Questionnaires in the VFMP process. Each auto-generated VFMP Fraud Dispute Questionnaire will have associated dispute financial or financial fields, depending on the organization's dispute financial configuration. VROL uses VFMP Fraud Disputes (10.5) for these auto-generated Fraud Dispute questionnaires. Issuers can also choose whether they want the auto-disputes to be saved or submitted. These are non-ATM Visa network transactions.

VROL follows this process:

- Generates a new case with a TI, a pinned transaction, and a VFMP Fraud dispute questionnaire.
  - Searches for associated transactions when the auto-dispute action is initiated. If VROL finds associated transactions but cannot auto-match them, VROL sets the VFMP status to 'Exception'.
- Note:** *VROL does not search for matching Fraud Reports, Exception File Listings, or Provisional Credits. VROL does not require any of these items when the Fraud dispute questionnaire is auto-submitted.*
- Generates a dispute financial or dispute financial fields for VFMP-reported fraudulent transactions, depending on the organization's dispute financial configuration. Submits or saves the dispute financial from the existing matched case, or from the newly created case.
  - Receives a dispute financial advice for VROL-generated dispute financials and dispute financials that are processed outside of VROL.
  - If auto-dispute is successful, changes VFMP transaction status to 'Dispute Requested' and the processed VFMP transaction appears in the VFMP queue. If auto-dispute is unsuccessful due to any reason, then such transactions appear in the VFMP queue with a VFMP status of 'Exception'.

## VFMP Process

Each VFMP process consists of a series of request and response transmissions managed through VROL. The following steps describe the process and XML operations that can be used by issuers to perform the task.

### Retrieve VFMP Transactions

VFMP transactions can only be retrieved from the VFMP queue. This queue is updated monthly and contains transactions for which a dispute questionnaire and financial may be automatically generated. If a member is not set up to process dispute financials through VROL, the queue results can be retrieved and used in the member's system.

To retrieve VFMP transactions from the VFMP queue, the issuer triggers a `SIGetQueueRequest` message. See [Chapter 17, Queues](#).

The VFMP queue request allows filtering of the returned values. The request has two option fields to filter the queue results: `<ReportMonth>` and `<GMFPStatus>`. If `<ReportMonth>` is omitted, all VFMP transactions for the past 12 months (including current month) are returned. The issuer can retrieve all but transactions in 'Ignored' status by including all other status values using `<GMFPStatus>`.

VROL returns a summary of the VFMP transactions in the filtered queue including the GMFPItemIDs for each transaction.

### Retrieve VFMP Transaction Details

1. To retrieve details of each VFMP-reported transaction, and determine if any matching VROL case(s) exists, the issuer triggers a `SIGetGMFPDetailsRequest` message using GMFPItemID.

VROL sends response with VFMP transaction details.

**Note:** Depending on the member configuration, VROL can automatically create disputes (if eligible) for these VFMP-reported transactions (see [Auto Disputes on VFMP-Reported Transactions](#)).

- If a matching case exists and the issuer is configured to have VROL generate dispute financials on their behalf, a new case with a TI, a pinned transaction, and a VFMP Fraud dispute questionnaire are generated within a certain period.
- If the issuer is configured to generate dispute financials outside of VROL, the dispute financial is processed outside of VROL. VROL later receives a dispute financial advice.

VROL searches for associated transactions when the auto-dispute action is initiated. If VROL finds associated transactions but cannot auto-match them, VROL sets the <GMFPStatus> in the SIGetGMFPDetailsResponse message to "GMEX" (Exception).

VROL does not search for matching Fraud Reports, Exception File Listings, or Provisional Credits. VROL does not require any of these items when the Fraud dispute questionnaire is auto-submitted.

**Note:** The Fraud dispute questionnaire is a subset of the normal questionnaire fields for a Fraud dispute. It does not include any fields that require user entry.

**Note:** VFMP transactions cannot be added to a VROL case.

2. Once the issuer has reviewed each VFMP transaction in the queue, the issuer should change the status of the VFMP transaction to 'Ignored' by sending request SIgnoreGMFPRRequest using GMFPItemID.

**Note:** SIgnoreGMFPRRequest is different from other queue "ignore/close" actions as the ignored VFMP transaction can still be retrieved using a queue request.

## VFMP Operations

RTSI exposes the following Report Distribution operation:

**SIGetGMFPDetailsRequest**—Used to retrieve details of a VFMP-reported transaction.

### SIGetGMFPDetailsRequest

Issuers use SIGetGMFPDetailsRequest to retrieve the details for each VFMP-reported transaction and determine if any matching VROL case(s) exists. This request requires the GMFPItemID and returns a SIGetGMFPDetailsResponse from VROL.

#### Request Message—SOAP-based Services

```
<SIGetGMFPDetailsRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <GMFPItemID>0000000</GMFPItemID>
  </requestData>
</SIGetGMFPDetailsRequest>
```

**Response Message—SOAP-based Services**

```

<SIGetGMFPDetailsResponse>
    <Status>...</Status>
    <responseData>
        <GMFPResponse>
            <MerchantName>Any merchant</MerchantName>
            <MerchantCity>Any city</MerchantCity>
            <MerchantCountryCode>AL</MerchantCountryCode>
            <AccountNumber>xxxxxxxxxxxxxx</AccountNumber>
            <Token>000000000000</Token>
            <OrigTranCPD>2017-05-08</OrigTranCPD>
            <TranDate>2017-05-08</TranDate>
            <DestinationAmount currency="840">20.00</DestinationAmount>
            <USDAmount currency="840">20.00</USDAmount>
            <TransactionID>xxxxxxxxxxxxxx</TransactionID>
            <ARN>xxxxxxxxxxxxxxxxxxxxxx</ARN>
            <ECMOTO>N</ECMOTO>
            <PosEntryModeCd>90</PosEntryModeCd>
            <MCC>5542</MCC>
            <FraudPostDate>2017-07-22</FraudPostDate>
            <DaysToAct>92</DaysToAct>
            <MatchedCases>
                <VisaCaseNumber>1000000000</VisaCaseNumber>
                <StageStateDesc>String</StageStateDesc>
                <IssuerCaseNumber>a</IssuerCaseNumber>
            </MatchedCases>
            <GMFPStatus>RPTD</GMFPStatus>
            <ExceptionReason>String</ExceptionReason>
            <IssuerBID>10000000</IssuerBID>
            <IssuerName>Any issuer</IssuerName>
            <IssuerCountryCode>840</IssuerCountryCode>
            <FraudCode>0</FraudCode>
            <DateAdded>2017-07-25</DateAdded>
        </GMFPResponse>
    </responseData>
</SIGetGMFPDetailsResponse>

```

**Request Message—REST-based Services**

```

{
    "RequestHeader": "...",
    "requestData": {
        "GMFPItemID": "0000000"
    }
}

```

### Response Message—REST-based Services

```
{  
    "Status": "...",  
    "responseData": {  
        "GMFPResponse": {  
            "MerchantName": "Any merchant",  
            "MerchantCity": "Any city",  
            "MerchantCountryCode": "AL",  
            "AccountNumber": "111111111111",  
            "Token": "11111111111111",  
            "OrigTranCPD": "2017-01-01",  
            "TranDate": "2017-01-01",  
            "DestinationAmount": {  
                "currency": "111",  
                "$": 1  
            },  
            "USDAmount": {  
                "currency": "111",  
                "$": 1  
            },  
            "TransactionID": "a",  
            "ARN": "111111111111111111111111",  
            "ECMOTO": "A",  
            "PosEntryModeCd": "A",  
            "MCC": "A",  
            "FraudPostDate": "2016-01-01",  
            "DaysToAct": 1,  
        },  
        "MatchedCases": [  
            {  
                "VisaCaseNumber": 1000000000,  
                "StageStateDesc": "String",  
                "IssuerCaseNumber": "a"  
            },  
        ],  
        "GMFPStatus": "ADMCH",  
        "ExceptionReason": "A",  
        "IssuerBID": "11111111",  
        "IssuerName": "A",  
        "IssuerCountryCode": "A",  
        "FraudCode": "A",  
        "DateAdded": "2017-01-01"  
    }  
}
```

**THIS PAGE INTENTIONALLY LEFT BLANK.**

All RTSI requests include a User ID that is validated against the VOL Login ID. For a UI user, this is the Login ID used to log into Visa VOL to access many Visa applications, including VROL UI. Any change to a user's VOL Login ID affects their VROL RTSI Login ID (they will be the same).

There are two types of user ID, the Complex ID (or internalId) and the Simple ID (or loginId). The Complex ID format is aaaaNNNNu, where aaaa is usually the first 3 letters of the user's last name, plus the first 2 digits of the first name. NNNN is a 4 digit numeric sequence. A user may opt to create a Simple ID to overlay on top of their Complex ID.

**Note:** Once a user has created a Simple ID overlay on their Complex ID, the Complex ID will no longer work. The Simple ID form should be provided from that point forward.

In RTSI, the usage is as follows: <User id="..." type=[internalId/loginId]>, where internalId is the default if the type attribute is omitted.

An additional ID used in RTSI is the System ID (see [System ID](#)). This is the ID associated with the SSL client certificate. This is also a VOL ID, but need not be identical to the User ID in the individual RTSI request. An RTSI client can set up their HTTPS connection using the System ID and use different User IDs to identify the different users sending an individual request. When the User ID is associated with an individual RTSI request (with <User>), the user assigned within VROL to a VROL case will be the User ID and not the System ID (when this is different).

## Password Policy

This section provides a general background on Visa Internet security and password policy.

### Encryption

All requests made to the VROL application require the HTTPS protocol. Using this protocol ensures that information in the request is not viewable by unauthorized parties.

### Authentication

HTTP Basic Authentication is used in VROL RTSI. For more information on HTTP Basic Authentication, refer to RFCs 1945, 2616 and 2617. In Basic Authentication, the UserID

and Password are Base 64 encoded in the HTTP Authentication header. But as the entire HTTP packet is encrypted, the password is not exposed in transit.

To prepare the value of the HTTP Authentication header, the MHS should construct a string of the form <userid>: <password>, and then set the value of this header field to the Base64 encoded value of this string.

For example, given a userid of xyzzy1234u and password Plugh#201, the string "xyzzy1234u:Plugh#201" should be Base64 encoded and used for the Authentication header field value.

Authentication: eH16enkxMjM0dTpQbHVnaCMyMDE=

All users of VROL require a unique VOL user ID and password. A security administrator for Visa Online (VOL) creates these user accounts.

The equivalent of a login to VOL is attempted once for each RTSI request submitted with Basic Authentication. If the User ID and Password combination is invalid, an error message is returned. The VOL authentication server will disable a userid after three consecutive invalid login attempts. When this happens, the VOL help desk will need to be contacted to reset the password for that userid.

If a user attempts a login after the account is disabled, one of the following results:

- If the user enters the correct password, a response message is provided indicating that the account is locked.
- If the user enters an invalid password, the account is disabled by Visa Online. The response message indicates that the account is disabled.

## Password Types

The types of password that an RTSI user can use are as follows:

- Visa Online Password – This is a one-time use password that is provided to the user upon receiving the VOL login ID. The user needs to log on to Visa Online and change this to a permanent password.
- Certificate Password – This is a password that the user can set when configuring the certificates.
- JKS Password – This is a password in JKS format for the Visa root and Visa intermediate CA certificates that the user can download from <http://enroll.visaca.com>. These certificates are also distributed in the RTSI Technical Suite. This password is used during RTSI installation provided that the user performs server certificate validation.

## Password Requirements

Passwords for VROL must meet the following criteria:

- Not have been used within the last 12 months
- Be 8 to 14 characters in length
- Must contain at least three of the following four classes within the first eight characters of the password:
  - Upper case letters (A,B,C, ...Z)
  - Lower case letters (a.b.c, ...z)
  - Numerals (0, 1, 2, ...9)
  - Non-alphanumeric special characters (!@#\$%^&\*()+, and so forth)

These requirements help to guard against hacker password guessing schemes.

A password can be only changed once every 24 hours. VROL remembers several passwords from the past. Recent passwords cannot be reused; users must create new passwords.

VOL user passwords must be changed every 30 days and are revoked after 120 days of inactivity.

If you have questions about your VOL password or cannot log on to VOL, contact your regional Customer Service representative.

Members may request that passwords associated with System IDs be marked as non-expiring. An authorized user for an organization may send a request to the VOL help desk to have a System ID password be marked as non-expiring. If, for any reason, the non-expiring password for a particular user gets changed, the new password will revert to the normal expiration policy until a new request is made to mark the new password as non-expiring. Identify and/or request for a VOL ID to be used as a System ID.

## Visa Security Policies

If the member host system (MHS) supports automated integrated security, Visa highly recommends that the MHS follow Visa Information Security policies.

**Note:** *It is also imperative that the MHS adhere to the member's security policy.*

There are three main components of Visa Security Policy that members should implement:

**Each user of the VROL systems Interface is assigned a unique VROL user ID and password.**

VROL has its own user security database and requires that all users be actively registered with it in order to use the VROL Systems Interface through the MHS. The VROL user ID and password should be different from any user ID and password that is assigned by the user's internal security system.

**VROL user IDs and passwords must not be stored or transmitted in clear text.**

Whether in persistent or transient state, VROL user IDs and passwords must never be exposed in clear text. Use of clear text may allow unauthorized personnel to gain access to VROL and act on behalf of an authorized or legitimate user. To guard against this, the user ID and password must be encrypted in transient state (RAM), persistent storage (databases or files), and also when transmitted over a network. When transmitting to VROL, use of the HTTPS protocol satisfies this requirement.

**Users must be responsible for assigning their own passwords.**

Assigning a new password is to be performed by the user, not a member administrator or program. To assign a new VROL password, the user should be presented with a dialog box or form asking for a new password and also a confirmation of that new password.

## RTSI Security

RTSI provides built-in security that performs stringent authentication and authorization tests before processing client requests. This section describes RTSI security and provides information about the client-side code needed to satisfy RTSI security requirements.

## Client Requirements

RTSI security requirements on the client side include:

1. Implement a method of obtaining VROL user IDs and passwords from users. Identify or request for a VOL ID to be used as a System ID.
2. For the System ID, obtain and install a PKI client certificate.
3. Include the client certificate in every RTSI request, by copying to the target system and using part of the JAVA trusted store. Set the certificate before sending the request. Refer to [Appendix C](#) for more information.
4. Ensure Basic Authentication is implemented. Refer to [Authentication](#) for more information.

## System ID

In VROL RTSI, there is a distinction made between the userid credential involved in establishing the HTTPS connection (Basic Authentication + SSL client certificate) and the userid of the person for which the RTSI action should be attributed (Transaction Inquiry, Queue query, and so on.).

The userid used to establish the HTTPS connection must have a Visa issued SSL client certificate, and the RTSI request must include the user credentials (userid, password) and client certificate in each request. This userid is termed as the System ID.

The System ID may be used as a User ID in the RTSI request in place of the User ID (in <User>). However, it is recommended that a separate User ID is used in the RTSI request to distinguish between different users of the VROL client. In this way, multiple User IDs can share the same System ID's certificate, and still be distinguished within VROL.

The password for a VOL ID used as a System ID can be set to be non-expiring. An authorized user may send a request to the VROL help desk providing the userid and the request to make the current password non-expiring. If for any reason the password is later changed, it will revert to the normal expiration schedule and a new request must be made to set the password as non-expiring.

## TrustStore and KeyStore

TrustStore and keyStore are used in context of setting up the SSL connection of the application between client and server. In the SSL handshake, the purpose of trustStore is to verify credentials while the purpose of keyStore is to provide credentials.

keyStore contains private keys and certificates that corresponds to their public keys. It is required when the user is using the SSL Server or when the SSL requires client authentication.

TrustStore stores the certificates from Visa, for which the application communicates or the certificates signed by Certificate Authorities (CA) can be used to identify the provider to Visa. In general, during the SSL handshake process, this trustStore is downloaded to the user's local computer when establishing connection. In case the member's gateway doesn't allow connection, use the following trust store that is available in the RTSI documentation package:

- ..\VisaCertificate.zip\VisaCertificate
- Visatruststore.jks

The trustStore certificate can be listed by using keytool, with the following details:

- keytool -list -keystore visatruststore.jks

- keyStore password

**Note:** The password is available in the RTSI documentation package.

- Keystore type: JKS
- Keystore provider: SUN

The keystore contains the following two entries:

- root, Feb 27, 2014, trustedCertEntry,  
Certificate fingerprint (SHA1):  
5A:4D:0E:8B:5F:DC:F6:4E:72:99:A3:6C:06:0D:B2:22:CA:78:E4
- vica3, Feb 27, 2014, trustedCertEntry,  
Certificate fingerprint (SHA1):  
6F:52:03:2B:6B:C6:8C:17:59:EE:E8:7A:F3:90:34:FD:8B:59:D5:AA

Use the following command if you want to change the format from jks to pfx:

```
keytool -importkeystore -srckeystore visatruststore.jks -destkeystore  
visatruststore.pfx -deststoretype PFX
```

THIS PAGE INTENTIONALLY LEFT BLANK.

# Systems Interface URLs

A

This appendix lists URLs to use for production and Member Test Environment (MTE). Note that all URLs are case-sensitive.

## Systems Interface Request URL

WSDL services use two URL formats—one for TI-related services and another one for all other RTSI services:

- Transaction Inquiry service
  - **SOAP:** `https://<URL_BASE>/websrv_rolti/[mte2]/RTSIService`
  - **REST:** `https://<URL_BASE>/rsrv_rolti-mte2/api/**`
- Other RTSI services
  - **SOAP:** `https://<URL_BASE>/websrv_rolsi/[mte2]/RTSIService`
  - **REST:** `https://<URL_BASE>/rsrv_rolsi-mte2/api/**`

The difference between TI and non-TI is indicated in the “websrv\_rol*X*i” or “rsrv\_rol*X*i” component of the URL, where **X** is either “s” (for SI services) or “t” (for TI services).

**Table A-1: URL Formats**

Environment	SI/TI	URL
MTE2	SI	<b>SOAP:</b> <a href="https://mutualcertservicesgateway.visa.com/websrv_rolsi/mte2/RTSIService">https://mutualcertservicesgateway.visa.com/websrv_rolsi/mte2/RTSIService</a> <a href="https://mte-vrolgateway.visa.com/websrv_rolsi/mte2/RTSIService">https://mte-vrolgateway.visa.com/websrv_rolsi/mte2/RTSIService</a> <b>REST:</b> <a href="https://mutualcertservicesgateway.visa.com/rsrv_rolsi-mte2/api/**">https://mutualcertservicesgateway.visa.com/rsrv_rolsi-mte2/api/**</a> <a href="https://mte-vrolgateway.visa.com/rsrv_rolsi-mte2/api/">https://mte-vrolgateway.visa.com/rsrv_rolsi-mte2/api/</a>
	TI	<b>SOAP:</b> <a href="https://mutualcertservicesgateway.visa.com/websrv_rolti/mte2/RTSIService">https://mutualcertservicesgateway.visa.com/websrv_rolti/mte2/RTSIService</a> <a href="https://mte-vrolgateway.visa.com/websrv_rolti/mte2/RTSIService">https://mte-vrolgateway.visa.com/websrv_rolti/mte2/RTSIService</a> <b>REST:</b> <a href="https://mutualcertservicesgateway.visa.com/rsrv_rolti-mte2/api/**">https://mutualcertservicesgateway.visa.com/rsrv_rolti-mte2/api/**</a> <a href="https://mte-vrolgateway.visa.com/rsrv_rolti-mte2/api/">https://mte-vrolgateway.visa.com/rsrv_rolti-mte2/api/</a>
MTE2 (Alt)	SI	<a href="https://certservicesgateway.visaonline.com:2443/websrv_rolsi/mte2/RTSIService">https://certservicesgateway.visaonline.com:2443/websrv_rolsi/mte2/RTSIService</a>
	TI	<a href="https://certservicesgateway.visaonline.com:2443/websrv_rolti/mte2/RTSIService">https://certservicesgateway.visaonline.com:2443/websrv_rolti/mte2/RTSIService</a>
Production	SI	<b>SOAP:</b> <a href="https://mutualservicesgateway.visa.com/websrv_rolsi/RTSIService">https://mutualservicesgateway.visa.com/websrv_rolsi/RTSIService</a> <a href="https://vrolgateway.visa.com/websrv_rolsi/RTSIService">https://vrolgateway.visa.com/websrv_rolsi/RTSIService</a> <b>REST:</b> <a href="https://vrolgateway.visa.com/rsrv_rolsi/api/">https://vrolgateway.visa.com/rsrv_rolsi/api/</a>
	TI	<b>SOAP:</b> <a href="https://mutualservicesgateway.visa.com/websrv_rolti/RTSIService">https://mutualservicesgateway.visa.com/websrv_rolti/RTSIService</a> <a href="https://vrolgateway.visa.com/websrv_rolti/RTSIService">https://vrolgateway.visa.com/websrv_rolti/RTSIService</a> <b>REST:</b> <a href="https://vrolgateway.visa.com/rsrv_rolti/api/">https://vrolgateway.visa.com/rsrv_rolti/api/</a>

The type of action being requested is inferred from the outer XML element in the request, as in "SISubmitTranInquiryRequest".

URL: [https://certservicesgateway.visaonline.com:2443/websrv\\_rolti/mte2/RTSIService](https://certservicesgateway.visaonline.com:2443/websrv_rolti/mte2/RTSIService)

```
<SISubmitTranInquiryRequest xmlns="http://www.visa.com/ROLSI">
  <RequestHeader>
  </RequestHeader>
  <requestData>
    <CardNumber>1111111111111111</CardNumber>
    <StartDate>2014-06-28</StartDate>
    <EndDate>2014-07-30</EndDate>
    <IncludeInd>Both</IncludeInd>
  </requestData>
</SISubmitTranInquiryRequest>
```

These requests look slightly different with the inclusion of an embedded user and/or a MemberRole.

```
<SISubmitTranInquiryRequest xmlns="http://www.visa.com/ROLSI">
  <RequestHeader>
    <User id="asafe2352c" type="loginId"/>
    <MemberRole>I</MemberRole>
  </RequestHeader>
  <requestData>
    <CardNumber>1111111111111111</CardNumber>
    <StartDate>2014-06-28</StartDate>
    <EndDate>2014-07-30</EndDate>
    <IncludeInd>Both</IncludeInd>
  </requestData>
</SISubmitTranInquiryRequest>
```

THIS PAGE INTENTIONALLY LEFT BLANK.

# Miscellaneous Useful Information

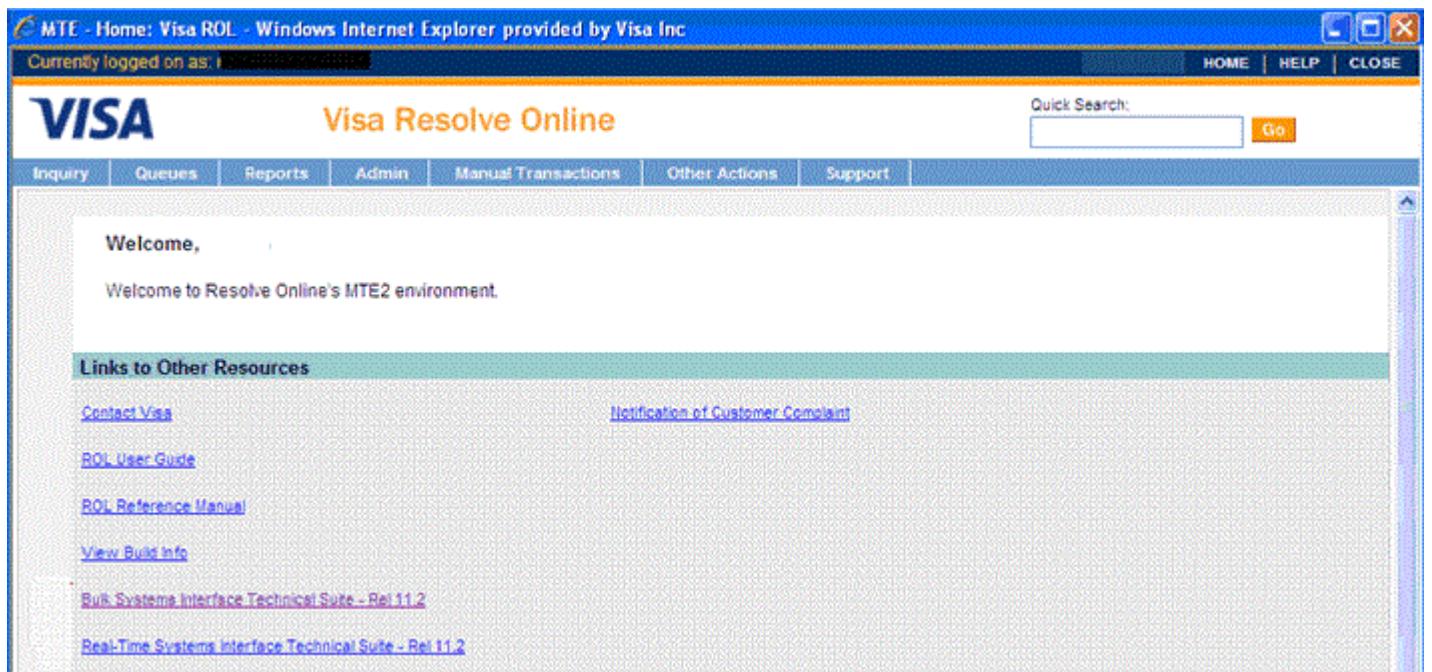
B

This section contains the useful information about the Interface Element Spreadsheet, Member Testing, Locked Fields, common issues with SI fields, handling of Japanese characters, comparison of Web service names, and comparison of TransactionInformation> vs. TI Details XML fields.

## Interface Element Spreadsheet (IES)

The IES refers to the spreadsheet that provides additional information about the SI schema elements to support the SI Member interfaces with VROL. The IES is divided into tabs for the different transaction types, so members only have to use the tabs that are relevant to the transaction types they support.

The IES and schema are available in MTE as part of the SI Technical Suites, accessed through links on the MTE home page. Prior to a release, the Technical Suites are updated monthly, starting 6 months in advance of the release, with a final drop 2 weeks prior to Member Test; the MTE home page has links to the IES and schema for the next release.

**Figure B-1: MTE Home Page**

## Existing Members

Existing members can find the following information useful:

- **Changes for the particular release, in particular mandatory changes:** Mandatory changes include changes to a) fields that a member is required to provide in upload/request, and also, b) download fields that a member must be prepared to handle if an opposing member sends to them. The IES has two tabs, "New Upload Fields" and "New Download Fields", indicating mandatory changes and the Members are also alerted to these in the Technical Letter.
- **Changed fields:** Each tab has a "Last Update" column with the date that the row in the IES was last updated. Filtering on this column is the best way to determine if a change has been made in IES or schema. Column "Release x.x changes" has schema changes or changes to context although some IES-changes are not identified by filtering on this column.
- **Compliance codes and Error messages that are updated every release.** The IES has a tab for "Compliance Codes". Error messages are provided as separate files in the SI Technical Suites.

**Note:** Not all Visa Core Rules and Visa Product and Service Rules changes have associated changes in the IES/schema.

## New Members

New members can find the following interesting:

### Required fields

Each tab in the IES has a column "Required Field in Request/Upload". This column indicates if a field is required in a Bulk SI Upload file or RTSI request a) to pass schema validation, and b) based on a business rule in VROL. In the case of (b), the column often

contains "C" for conditional, and the conditions for requiring this field are given in the "notes" column.

**Note:** *The IES does not indicate if a field processed by another system, for example, VIP/ BASE II, is required by that system. In particular, for fields sent with financials and RFCs, the Member must check relevant Visa manuals to determine which fields are required for transactions processed by other Visa systems. For many fields, VROL will default a field value from TI, if there is a TI in the VROL case.*

**Note:** *It is recommended that a new SI member also have access to the VROL UI, as that gives a visual indication of the relationships between transactions and required/conditionally required fields.*

## Member Testing

The MTE environment in VROL is provided for members to test the coding changes they have made in their code to work with the new VROL release.

VROL provides VCMS BINs and related testing tools to simulate the behavior of an opposing member. These are described in the *Visa Resolve Online Member Implementation Guide* and the *Visa Resolve Online Member Test Scenarios* guide.

For testing support, refer to the Global Client Testing website on Visa Online (<https://secure.visaonline.com/SitePages/Content.aspx?pageid=2.0.3.2.0>).

For testing requests, you need to complete and send the Global Client Testing Questionnaire ([https://secure.visaonline.com/\\_layouts/GVOL/GVOLStream.aspx?filename=\root\content\dam\gvol\en\operations\more-operations\global-client-testing\gct-questionnaire.pdf&contenttype=application/pdf](https://secure.visaonline.com/_layouts/GVOL/GVOLStream.aspx?filename=\root\content\dam\gvol\en\operations\more-operations\global-client-testing\gct-questionnaire.pdf&contenttype=application/pdf)) to your Account Manager or Implementation Manager.

## Common Issues With SI Fields

### Retired fields

Retired fields are visible in legacy transactions and so the SI must still support them in response/download for some period of time. A consideration over the release weekend is a pended questionnaire: a pended questionnaire is considered a legacy transaction, although when submitted, the retired field is no longer valid.

For example, there are many fields that will be retired in release 11.2, and the following convention is followed in the IES and schema, for these fields:

- 11.2 IES: fields marked as download only
- 11.2 schema: No change. Fields are ignored in upload
- 12.1 IES: fields marked as deleted
- 12.1 schema: No change. Fields are ignored in upload
- 12.2 IES: fields are removed completely from IES
- 12.2 schema: The schema is cleaned up & fields are removed completely

**Note:** *"Retired" fields are different from "Replaced" fields that are removed immediately from the IES and schema. Also, there are fields that are deleted from the schema without replacement, for example, as part of cleanup of unused fields that are not in the IES.*

## Locked Fields

For the use of locked fields, see [Locked Fields](#) in [Chapter 3, RTSI Request Operations](#).

**Figure B-2: Locked Fields (1 of 2)**

Case Folder Lockable Fields - Original Transaction Fields										
Trigger	Card Account Number	Tran Amt <sup>18</sup>	Tran Curr Code <sup>19</sup>	Tran Curr (Side performing action)	Tran CPD/ Settlement Date	Account Type <sup>9</sup>	Processing Code	TCQ <sup>21</sup>	Message Type	Tran Type (Sale/Cash/Credit) <sup>22</sup>
Tran Inquiry <sup>2</sup>	X			X						X <sup>1, 11</sup>
Fee Collection, Funds Disbursement – Receipt or Submitted by VROL <sup>3, 14, 19</sup>	X									X X
Fraud Advice submitted on VROL <sup>3, 19</sup>	X	X								X X
Receipt of Fraud Advice Confirmation in VROL <sup>10</sup>										
Exception File Listing submitted on VROL <sup>3, 19</sup>	X									
Exception File Inquiry submitted on VROL <sup>3, 19</sup>	X									
Receipt of RFC or VCRFS advice in VROL	X	X X X X X X							X <sup>1, 11</sup>	X X
RFC Submitted on VROL <sup>13, 19</sup>	X	X X X X X X							X <sup>1, 11</sup>	X X
RFC Response Submitted or Received on VROL <sup>19</sup>	X	X X X X X X							X <sup>1, 11</sup>	X X
Receipt of VDAS Advice <sup>8</sup>	X	X X X X X X								X X X
Pre-filing or Case Filing (sent or received) <sup>16</sup>	X	X X X X X X			X X	X			X X X X X X	X X X X X
Good Faith (sent or received) <sup>16</sup>	X	X X X X X X			X X	X			X X X X X X	X X X X X
Pended, Submitted for Review/Marked as Reviewed QN <sup>6</sup>	X	X X X X X X			X X X	X X				X X X X X
QN Pending Advice with financial <sup>3</sup>	X	X X X X X X			X X X	X X X				X X X X X
Receipt of Chargeback or Repre Financial <sup>7</sup>	X	X X X X X X			X X X	X X X				X X X X X
QN Pending Advice without financial <sup>19</sup>	X	X X X X X X			X X X	X X X				X X X X X
Rejected transaction <sup>9, 15</sup>										
Force Post QN	X	X X X								X X X
Reversals <sup>10, 19</sup>										
Adjustment submitted or received in VROL <sup>19</sup>	X	X		X X				X		X X
CB, 2nd CB or Repre Financial {Submitted, Pended, Submitted for Review, Marked as Reviewed} in VROL <sup>4</sup>	X	X X X X X X	X X X X		X X	X X		X X		X X X X
Stop Payment submitted, submitted for review, marked as reviewed in VROL	X									
Submit Accounting Entry Memo	X									

**Figure B-3: Locked Fields (2 of 2)**

Legend: X = field is locked; blank = field is unchanged from its previous state (stays locked if previously locked, stays unlocked if previously unlocked)

1. For US only. Original source transaction amount will be locked.
2. Fields become locked at point of dispute for both sides.
3. Fields become locked at point of submit for both sides.
4. Fields become locked at point of save (pend, submit for review, mark as reviewed) for other side. (The side that saved might still be able to update and the new fields will be locked for the other side).
5. Fields become locked at point of submit for other side.
6. Locking rules do not apply. Case Folder is updated. Note: TCO portion of Tran Type will not be updated for the "Receipt of VDAS Advice" but will be updated for "Receipt of Chargeback or Repro Financial".
7. This field cannot be changed so that it changes the dispute jurisdiction, unless the jurisdiction change is from VE to CEMEA or CEMEA to VE. Changes between those jurisdictions are allowed and will not trigger a locked field error.
8. If the existing field in the case is null (or spaces or zeros) it will be updated with the incoming value passed by the user and a locked field violation will not occur.
9. Reverts to the current case lock or the opposers action lock, whichever is more restrictive.
10. All locks remain unchanged.
11. This is ALWAYS the Original Transaction Source Amount (Base II) or Local Tran Amount (SMS), hence the opposer's (acquirer's) amount, as the issuer is considered the user 'taking action' in this scenario.
12. The MCC may be updated as long as the category is not changed. The member would not be able to change the MCC such that the category went from Non-ATM to ATM or vice versa.
13. If the RFC rejects and the current lock state is RFC and it is the only thing in the case it will go to unlocked otherwise the current lock state will remain unchanged.
14. If the Misc Fee rejects and the current lock state is Misc Fee and it is the only thing in the case it will go to unlocked; otherwise, the current lock state will remain unchanged. If the account number is not in the Misc. Fee (e.g. if we receive a reason code 140 Fee Collection) then the Account Number is not locked.
15. If the CB rejects and it is the only thing in the case it will continue to be unlocked; otherwise, the current lock state will remain unchanged. This behavior will be the same for a rejected Repro.
16. After a Pre-filing, case filing, or Good Faith, the questionnaire header fields are always "read only" (through the UI or the SI's); therefore, the user will never have an opportunity to change the values of these "lockable" fields. Therefore, there will never be a locked field violation.
17. If a transaction has both issuer and acquirer network IDs, all conditional logic should be based on the acquirer network ID unless the transaction is a representment or adjustment submitted by an acquirer. In that case, the issuer network ID should be used.
18. Depending on who is in context, issuer or acquirer, this may be the transaction source amount or the transaction destination amount. Note that for SMS, the source amount is the local transaction amount, and the destination amount is a composite of cardholder billing amount, settlement amount and transaction amount (whichever is not empty).
19. If the form is Submitted for Review or "Reviewed" the data would be retained and no fields would be locked for either side.
20. Intentionally left blank
21. For SMS only
22. For BASE II only
23. If the user submits a form that has a different ARN than the case but the three equivalent SMS fields match (RRN, STAN, Acq BIN) then it is not considered a locked field violation and VROL will allow the update.

## Japanese Character Handling

Two fields <JapanTCR2MerchantNameKatakana> and <JapanTCR2MerchantNameKanji>, in transaction inquiry, contain Japanese characters.

- In BASE II ITF format <JapanTCR2MerchantNameKatakana> is single byte encoded with IBM's 290 Code Page. It can contain up to 23 symbols encoded in 23 bytes. VROL SI XML will contain this data in xsd:hexBinary format, resulting in a 46 character long string.

*In Java, the following code can be used to convert data sent by VROL into BASE II ITF format and into native to Java representation:*

```
byte[] itfKatakanaBytes =
DatatypeConverter.parseHexBinary(japanTCR2MerchantNameKatakanaXmlValue);
String katakanaSymbols = new String(itfKatakanaBytes, Charset.forName("IBM-290"));
```

*Conversely, the following code can be used to prepare data for VROL from a String containing Katakana symbols or from ITF data:*

```
String katakanaSymbols = ....
byte[] itfKatakanaBytes = katakanaSymbols.getBytes("IBM-290");
String japanTCR2MerchantNameKatakanaXmlValue =
DatatypeConverter.printHexBinary(itfKatakanaBytes);
```

- In BASE II ITF format <JapanTCR2MerchantNameKanji> is double byte encoded with IBM's 300 Code Page. It can contain up to 40 symbols encoded in 80 bytes. VROL SI will contain this data in xsd:hexBinary format, resulting in an 80 character long string.

In Java, the following code can be used to convert data sent by VROL into BASE II ITF format and into native to Java representation:

```
byte[] itfKanjiBytes =
DatatypeConverter.parseHexBinary(japanTCR2MerchantNameKanjiXmlValue);
String kanjiSymbols = new String(itfKanjiBytes, Charset.forName("IBM-300"));
```

Conversely, the following code can be used to prepare data for VROL from a String containing Kanji symbols or from ITF data:

```
String kanjiSymbols = ....
byte[] itfKanjiBytes = katakanaSymbols.getBytes("IBM-300");
String japanTCR2MerchantNameKanjiXmlValue =
DatatypeConverter.printHexBinary(itfKanjiBytes);
```

## Comparison of <TransactionInformation> vs. TI Details XML Fields

The following tables provide the comparison of <TransactionInformation> XML fields in SIGetCaseDetailsResponse vs. UI fields and XML fields in TI details (SIGetTransDetailsResponse).

**Table B-1: <TransactionInformation> vs. TI Details Fields—Manual Transaction Data Entry Form**

UI field: <b>(Manual Transaction Data Entry Form)</b>	XML field: <b>&lt;TransactionInformation&gt;</b>	XML field: <b>TI Details</b>
Card/Acct #	AccountNumber	CardNumber
Network ID	NetworkID	NetworkID
Merchant Category Code	MCC	MCC
Merchant Country Code	MerchantCountryCode	MerchantCountry, MerchantCountryCode
Account Type	AccountType	AccountType1
ARN	ARN	ARN
Acq BIN/Acq Inst ID	AcquirerBIN	AcquirerBIN, AcquirerInstitutionId
Retrieval Reference #	RetrievalReferenceNumber	RetrievalReferenceNumber
Sys Trace Audit #	SystemTraceAuditNumber	SystemTraceAuditNumber
Tran ID – MCSN	Transaction (id attribute) (in <Transaction>)	TransactionID
Transaction Type	TransactionCategory, ProcessingCode	TransactionType, ProcessingCode
Local Tran Date	TranDate (in <Transaction>)	TransactionDateTime, LocalTransactionDateTime
Processing Date	CPD	CPD
Transaction Amount / Currency	Amount, Amount (currency attribute) (in <Transaction>)	TransactionAmount, TransactionAmount (currency attribute)
Reimbursement Attribute	ReimbursementAttribute	ReimbursementAttribute
MOTO/E-Commerce	ECMOTO	ECMOTO, MotoeCommerce

**Table B-1: <TransactionInformation> vs. TI Details Fields—Manual Transaction Data Entry Form (cont'd.)**

UI field: <b>(Manual Transaction Data Entry Form)</b>	XML field: <b>&lt;TransactionInformation&gt;</b>	XML field: <b>TI Details</b>
Special Chargeback Indicator	SpecialConditionInd	RiskInd, RiskDescription
Merchant Name	MerchantName	MerchantName
Merchant City	MerchantCity	MerchantCity
Merchant State / Province	MerchantCountrySubEntityCode	MerchantCountrySubEntityCode
Merchant Postal Code	MerchantPostalCode	N/A
Acquirer BID	AcquirerBID	AcquirerBID
Entry Mode	PosEntryModeCd	EntryMode, PosEntryMode
Terminal Entry Capability	TerminalEntryCapability	PosTerminalCapabilityCode
PIN Entry Capability	PINEntryCapability	PosPinEntryCapabilityCode
Card Acceptor ID	CardAcceptordId	CardAcceptordId
Card Acceptor Terminal ID	TerminalId	TerminalId
Terminal Type (SMS)	CatInd	PosTerminalType (SMS)
CAT Indicator (B2)	CatInd	PosCatInd (B2)
Cardholder ID Method	CardholderIdMethod	CardholderIdMethodCode
Authorization Code	AuthCode	AuthorizationCode
Card Capability	CardCapability	N/A
Auth Characteristics Indicator	AuthCharCd	AuthorizationCharIndicator
Cashback Amount	CashbackAmount	CashbackAmount
Account Type Identification	AccountType1Identification	AccountType1Identification
Token	Token	Token
Token Assurance Level	TokenAssuranceLevel	TokenAssuranceLevel
DCC Indicator	DCCInd, DCCIndDescription	DCCInd, DCCIndDescription
Merchant Verification Value	MerchantVerificationValue	MerchantVerificationValue
Fee Program Indicator	FeeProgramInd	FeeProgramIndicator
Card Expiration Date	CardExpireDate	ExpirationDate
Remote Terminal Indicator	RemoteTerminalInd	OnPremiseOffPremiseInd
Merchant/ATM Location	MerchantLocation	MerchantLocation
Floor Limit Indicator	FloorLimitInd	FloorLimitInd
STIP Indicator	PcasInd	PcasIndicator

**Table B-1: <TransactionInformation> vs. TI Details Fields—Manual Transaction Data Entry Form (cont'd.)**

<b>UI field: (Manual Transaction Data Entry Form)</b>	<b>XML field: &lt;TransactionInformation&gt;</b>	<b>XML field: TI Details</b>
Member Supplied Fee Amount	MemberSuppliedFee	MemberSuppliedFee
Member Supplied Fee Amount Currency	MemberSuppliedFeeIndicator	MemberSuppliedFeeIndicator
Product ID	ProductId	ProductId
CRB/Exception File Indicator	CrbExceptionFileInd	CrbFileInd
CPS Requested Indicator	RequestedPaymentServicesInd	RequestedPaymentServicesInd
International Fee Indicator	InternationalFeeInd	InternationalFeeIndicator
Auth Response Code	AuthRespCode	AuthorizationResponseCode
POS Environment	PosEnvironment	PosEnvironment
Surcharge Fee Amount	SurchargeAmount	SurchargeFeeAmount, SurchargeFeeCHBillCurrAmount
Surcharge [Fee] Amount DR/CR Indicator	SurchargeAmountType	SurchargeFeeAmountDebitCreditInd, SurchargeFeeCHBillCurrAmountDebitCreditInd
National Reimbursement Fee	NationalReimbursementFee	NationalReimbursementFee (Brazil) also incl. curr & CR/DR ind
Prepaid Card Indicator	PrepaidCardInd	PrepaidCardInd
Installment Payment Count	InstallmentPaymentCount	InstallmentPaymentCount (Brazil)
Business Application ID	BusinessAppld	BusinessAppld
Travel Agency ID	TravelAgencyId	TravelAgencyCode (T&E)
Korea NNSS Fields	many fields in <NationalSettlementDraftDataKorea>	many fields in <NationalSettlementDraftDataKorea>
Brazil NNSS Fields	many fields in <NationalSettlementDraftDataBrazil>	many fields in <NationalSettlementDraftDataBrazil>
Mexico NNSS Fields	many fields in <NationalSettlementDraftDataMexico>	many fields in <NationalSettlementDraftDataMexico>
Japan NNSS Fields	many fields in <NationalSettlementDraftDataJapan>	many fields in <NationalSettlementDraftDataJapan>
Sweden NNSS Fields	many fields in <NationalSettlementDraftDataSweden>	many fields in <NationalSettlementDraftDataSweden>
Colombia NNSS Fields	many fields in <NationalSettlementProcessingData>	many fields in <NationalSettlementProcessingData>
Settlement Flag	SettlementFlag	N/A
Market Specific Data Indicator	MarketSpecificInd	MarketSpecificDataInd

**Table B–2: <TransactionInformation> vs. TI Details—Case Details**

UI field: <b>(Case Details)</b>	XML field: <b>&lt;TransactionInformation&gt;</b>	XML field: <b>TI Details</b>
Product Type (part of Case Classification)	VROLProductType	VROLProductType
AVS Result	AvsResponseCode	AvsResponseCode

THIS PAGE INTENTIONALLY LEFT BLANK.

# XML Processing

C

This appendix covers topics associated with the processing of XML documents exchanged using RTSI.

## XML Validation Options

XML validation of an XML instance document against an XSD schema document is often performed in production systems to verify that the incoming documents follow the rules set forth in the schema. These rules would include things such as element sequence, data type, data length, required elements, and so forth.

Validation of the XML requests and responses in the RTSI is optional because VROL always validates incoming requests and displays a system error if the incoming request is not valid. The error message includes specific information that helps track down the cause. The incoming responses from Visa are always valid, so if there is some sort of problem parsing the responses from Visa, it likely means the MHS is not coded to specifications.

Visa recommends that validation be done during development to track down any problems that may occur. Because schema files that are hosted on HTTP servers are publicly accessible, and there is no way to secure XSDs and still make them accessible to parsers, Visa does not host them publicly. Thus, the schemaLocation tag in the XML response header does not point to a real URL.

In order to perform validation on a MHS, the schema files must be kept up-to-date in a location accessible to the client application. This location is usually a directory on the client machine or a network server. Members can request the XSD files from Visa for this purpose.

It is important for members to validate XML requests before sending those requests to Visa. This procedure limits error handling for members.

As long as the schema files are available to the parser, there are three methods that can be used to perform validation.

1. Set the schema location attribute in the parser.

This is only available when using the Xerces parser. The method is `setExternalSchemaLocation`, and the namespace used should be

`"http://www.visa.com/ROLSI file:///xsd/RealTimeXSD/ROLSIRT.xsd"`.

The following sample code has been used successfully at Visa for testing:

```
//The first part of the string is the namespace  
  
CString schemaLocation = "http://www.visa.com/ROLSI";  
//The actual schemalocation is stored in the INI file and is usually something like  
  
//file:///xsd/RealTimeXSD/ROLSIRT.xsd. Add this to the namespace.  
  
schemaLocation += ((CMainFrame*)AfxGetMainWnd())->GetSettings()->getSchemaLocation();  
  
//Set the external schema location in the parser  
  
pXercesParser->setExternalSchemaLocation(XMLString::transcode  
(schemaLocation));
```

## 2. Use a schema cache.

When using either the Xerces parser or the MSXML parser, there is a concept of schema cache, which is a pre-loaded cache of the schema that is used for validation. The schema files are only read from the disk one time, so this has a performance advantage.

## 3. The schemaLocation tag in the XML instance is set to a file location that can be used by the parser at runtime.

Currently, the schemaLocation is set to:

file:///xsd/RealTimeXSD/ROLSIRT.xsd. Thus, the ROL schema can be placed in the directory "XSD/RealTimeXSD" of the runtime directory that the member application is run from. For example, if the member application is run from "C:\Program Files\OurApp\app.exe", the schema would be placed in the directory "C:\Program Files\OurApp\xsd\RealTimeXSD". The parser picks up the schema automatically, using the hint in the schemaLocation attribute.

Alternatively, the schemaLocation tag could be replaced at runtime with a location that is accessible to the client machine on a network share.

## XML Usage

The XML exchanged between the MHS and VROL is validated against the real-time schema (ROLSIRT.XSD).

For more information regarding the XML Schema Language, see:

<http://www.w3.org/XML/Schema>.

## Data Element Sequence

The Real-Time schema requires proper sequencing of elements contained within an element group. The choice to sequence elements was made so that the XML would be more readable for members during development. XML requests received by the systems interface are validated and are rejected with an error message if the elements are not sequenced properly.

### Example (correct):

```
<ContactInfo>
<Address>1313 Mockingbird Lane</Address>
<City>Pittsburgh</City>
</ContactInfo>
```

### Example (incorrect):

```
<ContactInfo>
<City>Pittsburgh</City>
<Address>1313 Mockingbird Lane</Address>
</ContactInfo>
```

## Dates and Timestamps

Dates and timestamps are consistently represented across all the elements in the VROL XML schema. An element may appear in either date format or datetime format.

### Dates

All dates in XML are represented using the format CCYY-MM-DD. The following example represents 22 November 2005.

#### Example

```
<TransactionDate>2005-11-22</TransactionDate>
```

### Timestamps

All timestamps in XML are represented using the format CCYY-MM-DDTHH:mm:ss-<GMT Offset>. The following example represents 22 November 2010 at 3:45pm EST.

#### Example

```
<Timestamp>2010-11-22T15:45:00-04</Timestamp>
```

## More XML Resources

Further information concerning XML is widely available on the Web. Examples include:

- *A Technical Introduction to XML*—Published by XML.com, this site provides a good introduction to XML from a purely technical perspective. See:  
<http://www.xml.com/pub/a/98/10/guide0.html>
- *The XML FAQ*—This FAQ provides thorough answers to commonly asked questions. See:  
<http://www.xml.com/pub/a/98/10/guide0.html>
- *Namespace Myths Exploded*—A discussion of XML namespaces and common fallacies surrounding them. See:  
<http://www.xml.com/pub/a/2000/03/08/namespaces/>

THIS PAGE INTENTIONALLY LEFT BLANK.

# Sample Code

D

## Performing Basic Authentication and Assigning Client Certificate

Before sending the requests, set the certificate. This is just one of the different ways to perform Basic Authentication and Client certificate assignment for JAX-RPC clients. It uses the JAVA\_OPTION to set the certificate

```
((javax.xml.rpc.Stub) stub)._setProperty("javax.xml.rpc.security.auth.username",
session.getAttribute("_vrolusername"));
((javax.xml.rpc.Stub) stub)._setProperty("javax.xml.rpc.security.auth.password",
session.getAttribute("_vrolpassword"));
java.lang.System.setProperty("javax.net.ssl.keyStore", "myuser.jks");
java.lang.System.setProperty("javax.net.ssl.keyStorePassword", "password");
java.lang.System.setProperty("javax.net.ssl.keyStoreType", "JKS");
java.lang.System.setProperty("javax.net.ssl.trustStore", "myuser.jks");
java.lang.System.setProperty("javax.net.ssl.trustStorePassword", "password");
```

**Note:** *The above XML sample is applicable if the reference implementation is JAVA. Property settings needs to be accordingly changed for other implementations.*

## Submitting Images

### SOAP Binding

The SOAPBinding section of the WSDL indicates how to submit multipart attachments.

```
<wsdl:binding name="SubmitImageSoapBinding" type="tns:SubmitImage">
<wsdlsoap:binding style="document" transport=
"http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="SubmitImageOperation">
<wsdlsoap:operation soapAction="SubmitImage"/>
<wsdl:input name="SubmitImageRequest">
<mime:multipartRelated>
<mime:part name="SubmitImageBodyPart">
<wsdlsoap:body parts="SubmitImageRequest" use="literal"/>
</mime:part>
<mime:part name="SubmitImageImagePart">
<mime:content part="image" type="multipart/*"/>
</mime:part>
</mime:multipartRelated>
</wsdl:input>
<wsdl:output name="SubmitImageResponse">
<wsdlsoap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
```

## Remote Service Stub

The following example shows the generated stub for the remote service.

```
public interface SubmitImage extends java.rmi.Remote
{
    public java.lang.String submitImageOperation(java.lang.String submitImageRequest,
        javax.mail.internet.MimeMultipart image) throws java.rmi.RemoteException;
}
//Code snippet for submitting images
//Instantiate the stub
SubmitImage submitImageStub =
    new SubmitImageSoapBindingStub(endpointUrl, rpcService);
//get the submitImage xml
String xml = getSubmitImageXml();
//load images to be submitted
Object image = loadImage();
//invoke submitImageOperation
SubmitImageStub.submitImageOperation(xml, (MimeMultipart) image);
//implementation of loadImage
private Object loadImage(String xml){
    String pathname = null;
    byte[] byteArray = null;
    Object image = null;
    try{
        System.out.println("Inside loadImageAsBytes()");
        //get the pathname from a property file
        pathname = getProperty(IMAGE_PATH_PROPERTY);
        File f = new File(pathname);
        //For multipart/*
        image = getMimeMultiPartImage(f,xml);
    }catch(Exception e){
        e.printStackTrace();
    }
    return image;
}
```

```
/**  
 * Returns MimeMultipart object containing all the images in  
 * the directory as body parts  
 * @parameter f is the directory where images reside  
 */  
private MimeMultipart getMimeMultiPartImage(File f, String xml){  
    DataSource ds = null;  
    MimeMultipart mpart = null;  
    String contentType = "image/tiff";  
    String contentId = null;  
    String pathName = null;  
    try{  
        mpart = new MimeMultipart();  
        if(!f.isDirectory()){  
            System.out.println("pathname: " + f + " has to be a directory");  
            return null;  
        }  
        //Attaches all the files in that particular directory  
        File[] files = f.listFiles();  
        for(int i=0;i<files.length;i++){  
            MimeBodyPart mbp1 = new MimeBodyPart();  
            FileInputStream fis = new FileInputStream(files[i].getAbsolutePath());  
            //Instantiate ImagingDatahandler using the fileinputstream  
            ImagingDatahandler imagedataHandler = new ImagingDatahandler("image",contentType,fis);  
            //set the imagedataHandler as the datahandler for the bodypart  
            mbp1.setDataHandler(imagedataHandler);  
            //get the ContentId from the xml  
            contentId = getContentType(xml);  
            //add the Content-Id header  
            //Need to make sure that this Content-Id is same as in the xml  
            mbp1.addHeader("Content-Id",contentId);  
            mbp1.addHeader("Content-Type",contentType);  
            System.out.println(" bodypart's contenttype: " + mbp1.getContentType());  
            mpart.addBodyPart(mbp1);  
        }  
    }catch(Exception e){  
        e.printStackTrace();  
    }  
    return mpart;  
}  
//Following is implementation of ImagingDatahandler  
public class ImagingDatahandler extends DataHandler implements DataSource{  
    private String name;  
    private String contentType;  
    private InputStream sourceStream;  
    public ImagingDatahandler(String argName, String argContentType,  
        InputStream argStream){  
        super(argStream,argContentType);  
        name = argName;  
        contentType = argContentType;  
        sourceStream = argStream;  
    }  
    /* (non-Javadoc)  
     * @see javax.activation.DataSource#getContent-Type()  
     */  
    public String getContentType() {  
        System.out.println("Inside getContentType and returning: " + contentType);  
    }
```

```
    return contentType;
}
/* (non-Javadoc)
 * @see javax.activation.DataSource#getInputStream()
 */
public InputStream getInputStream() throws IOException {
    System.out.println("Inside getInputStream");
    return sourceStream;
}
public DataSource getDataSource() {
    System.out.println("Inside getDataSource");
    return this;
}
public Object getContent() throws java.io.IOException {
    System.out.println("Inside getContent");
    return sourceStream;
}
/* (non-Javadoc)
 * @see javax.activation.DataSource#getName()
 */
public String getName() {
    System.out.println("Inside getName");
    return name;
}
/* (non-Javadoc)
 * @see javax.activation.DataSource#getOutputStream()
 */
public OutputStream getOutputStream() throws IOException {
    System.out.println("Inside getOutputStream and returning null");
    return null;
}
/**
 * write the bytes to the outputstream
 */
public void writeTo(java.io.OutputStream os) throws java.io.IOException{
    System.out.println("Inside writeTo and writing the byteArray
to the OutputStream: " + os);
    int byteData = -1;
    while( (byteData = sourceStream.read()) != -1) {
        os.write(byteData);
    }
}
```

## Service

```
@WebServiceClient(name = "RTSI-SI-Services", targetNamespace =
"http://www.visa.com/rtsi", wsdlLocation = "../..../VROL-RTSIService.wsdl")
public class RTSISIServices
    extends Service
{
    private final static URL RTSISISERVICES_WSDL_LOCATION;
    static {
        URL url = null;
        try {
            url = new URL("../..../VROL-RTSIService.wsdl");
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
        RTSISISERVICES_WSDL_LOCATION = url;
    }
    public RTSISIServices(URL wsdlLocation, QName serviceName) {
        super(wsdlLocation, serviceName);
    }
    public RTSISIServices() {
        super(RTSISISERVICES_WSDL_LOCATION, new QName("http://www.visa.com/rtsi",
"RTSI-SI-Services"));
    }
    /**
     *
     * @return
     *      returns SIRTSIOperations
     */
    @WebEndpoint(name = "SI-RTSIOperationsPort")
    public SIRTSIOperations getSIRTSIOperationsPort() {
        return (SIRTSIOperations)super.getPort(new QName("http://www.visa.com/rtsi", "SI-
RTSIOperationsPort"), SIRTSIOperations.class);
    }
    /**
     *
     * @param features
     *      A list of {@link javax.xml.ws.WebServiceFeature} to configure on the proxy.
     * Supported features not in the <code>features</code> parameter will have their default
     * values.
     * @return
     *      returns SIRTSIOperations
     */
    @WebEndpoint(name = "SI-RTSIOperationsPort")
    public SIRTSIOperations getSIRTSIOperationsPort(WebServiceFeature... features) {
        return (SIRTSIOperations)super.getPort(new QName("http://www.visa.com/rtsi", "SI-
RTSIOperationsPort"), SIRTSIOperations.class, features);
    }
}
```

## Operations

```
@WebService(name = "SI-RTSIOperations", targetNamespace = "http://www.visa.com/rtsi")
@SOAPBinding(parameterStyle = SOAPBinding.ParameterStyle.BARE)
@XmlSeeAlso({
    ObjectFactory.class
})
```

```
public interface SIRTSIOperations {  
  
    /** *  
     * @param rfcFulfillmentRequest  
     * @return  
     *      returns com.visa.rolsi.SISubmitRFCFulfillmentResponseType  
     */  
    @WebMethod(operationName = "SubmitFulfillmentOperation", action =  
"urn:#SubmitFulfillmentOperation")  
    @WebResult(name = "SISubmitRFCFulfillmentResponse", targetNamespace =  
"http://www.visa.com/ROLSI", partName = "RFCFulfillmentResponse")  
    public SISubmitRFCFulfillmentResponseType submitFulfillmentOperation(  
        @WebParam(name = "SISubmitRFCFulfillmentRequest", targetNamespace =  
"http://www.visa.com/ROLSI", partName = "RFCFulfillmentRequest")  
        SISubmitRFCFulfillmentRequestType rfcFulfillmentRequest);  
  
    /** *  
     * @param getFulfillmentRequest  
     * @return  
     *      returns com.visa.rolsi.SIGetRFCFulfillmentResponseType  
     */  
    @WebMethod(operationName = "GetFulfillmentOperation", action = "downloadImage")  
    @WebResult(name = "SIGetRFCFulfillmentResponse", targetNamespace =  
"http://www.visa.com/ROLSI", partName = "GetFulfillmentResponse")  
    public SIGetRFCFulfillmentResponseType getFulfillmentOperation(  
        @WebParam(name = "SIGetRFCFulfillmentRequest", targetNamespace =  
"http://www.visa.com/ROLSI", partName = "GetFulfillmentRequest")  
        SIGetRFCFulfillmentRequestType getFulfillmentRequest);  
  
    /** *  
     * @param uploadMessageImageRequest  
     * @return  
     *      returns com.visa.rolsi.SIUUploadMessageImageResponseType  
     */  
    @WebMethod(operationName = "UploadMessageImageOperation", action =  
"urn:#UploadMessageImageOperation")  
    @WebResult(name = "SIUploadMessageImageResponse", targetNamespace =  
"http://www.visa.com/ROLSI", partName = "UploadMessageImageResponse")  
    public SIUploadMessageImageResponseType uploadMessageImageOperation(  
        @WebParam(name = "SIUploadMessageImageRequest", targetNamespace =  
"http://www.visa.com/ROLSI", partName = "UploadMessageImageRequest")  
        SIUploadMessageImageRequestType uploadMessageImageRequest);  
  
    /** *  
     * @param getMessageImageRequest  
     * @return  
     *      returns com.visa.rolsi.SIGetMessageImageResponseType  
     */  
    @WebMethod(operationName = "GetMessageImageOperation", action =
```

```
"downloadImage")
    @WebResult(name = "SIGetMessageImageResponse", targetNamespace =
"http://www.visa.com/ROLSI", partName = "GetMessageImageResponse")
    public SIGetMessageImageResponseType getMessageImageOperation(
        @WebParam(name = "SIGetMessageImageRequest", targetNamespace =
"http://www.visa.com/ROLSI", partName = "GetMessageImageRequest")
            SIGetMessageImageRequestType getMessageImageRequest);

    /**
     *
     * @param uploadImageRequest
     * @return
     *      returns com.visa.rolsi.SIUuploadImageResponseType
     */
    @WebMethod(operationName = "UploadImageOperation", action =
"urn:#UploadImageOperation")
    @WebResult(name = "SIUploadImageResponse", targetNamespace =
"http://www.visa.com/ROLSI", partName = "UploadImageResponse")
    public SIUploadImageResponseType uploadImageOperation(
        @WebParam(name = "SIUploadImageRequest", targetNamespace =
"http://www.visa.com/ROLSI", partName = "UploadImageRequest")
            SIUploadImageRequestType uploadImageRequest);

    /**
     *
     * @param getImageRequest
     * @return
     *      returns com.visa.rolsi.SIGetImageResponseType
     */
    @WebMethod(operationName = "GetImageOperation", action = "downloadImage")
    @WebResult(name = "SIGetImageResponse", targetNamespace =
"http://www.visa.com/ROLSI", partName = "GetImageResponse")
    public SIGetImageResponseType getImageOperation(
        @WebParam(name = "SIGetImageRequest", targetNamespace =
"http://www.visa.com/ROLSI", partName = "GetImageRequest")
            SIGetImageRequestType getImageRequest);
}
```

**Client**

```

SIUploadImageRequestType uploadImage=null;
javax.activation.DataHandler dataHandler = new
javax.activation.DataHandler(new FileDataSource("../..//T12345"));
UploadDocType uploadDocType=new UploadDocType();
uploadDocType.setDocType("CALET");
uploadDocType.setVisaCaseNumber(new Long("123456"));
RTSIAjaxDescriptorType value=new RTSIAjaxDescriptorType();
value.setAttachType(SOAPAttachType.MIME);
List<RTSIAjaxAttachmentType> attachList=value.getAttachment();
RTSIAjaxAttachmentType rtsiAttachmentType=new RTSIAjaxAttachmentType();
rtsiAttachmentType.setImageData(dataHandler);
rtsiAttachmentType.setId("1");
rtsiAttachmentType.setContentType("image/tiff");
rtsiAttachmentType.setComment("Comment");
attachList.add(rtsiAttachmentType);
uploadDocType.setAttachmentDescriptor(value);
uploadImage.setRequestData(uploadDocType);
RTSIServices clientSIService=new RTSIServices();
    SIRTSIOperations sirtOperations=
clientSIService.getSirTSIOperationsPort();
    SOAPBinding binding = (SOAPBinding) ((BindingProvider)
sirtOperations).getBinding();
    binding.setMTOMEnabled(true);
    sirtOperations.uploadImageOperation(uploadImage);

```

**Receiving Images****SOAP Binding**

The SOAPBinding section of the WSDL indicates how to receive multipart attachments.

```

<wsdl:binding name="GetImageSoapBinding" type="tns:GetImage">
<wsdlsoap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="GetImageOperation">
    <wsdlsoap:operation soapAction="GetImage"/>
    <wsdl:input name="GetImageRequest">
        <wsdlsoap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="GetImageResponse">
        <mime:multipartRelated>
            <mime:part name="GetImageBodyPart">
                <wsdlsoap:body parts="GetImageResponse" use="literal"/>
            </mime:part>
            <mime:part name="GetImageImagesPart">
                <mime:content part="image" type="multipart/*"/>
            </mime:part>
        </mime:multipartRelated>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>

```

**Bindings**

```

<binding name="SI-RTSIServices" type="tns:SI-RTSIOperations">
    <soap:binding style="document"

```

```
transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="SubmitFulfillmentOperation">
    <soap:operation
        soapAction="urn:#SubmitFulfillmentOperation"/>
    <wsdl:input name="SubmitFulfillmentRequest">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="SubmitFulfillmentResponse">
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetFulfillmentOperation">
    <soap:operation soapAction="downloadImage"/>
    <wsdl:input name="GetFulfillmentRequest">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="GetFulfillmentResponse">
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="UploadMessageImageOperation">
    <soap:operation
        soapAction="urn:#UploadMessageImageOperation"/>
    <wsdl:input name="UploadMessageImageRequest">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="UploadMessageImageResponse">
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetMessageImageOperation">
    <soap:operation soapAction="downloadImage"/>
    <wsdl:input name="GetMessageImageRequest">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="GetMessageImageResponse">
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="UploadImageOperation">
    <soap:operation soapAction="urn:#UploadImageOperation"/>
    <wsdl:input name="UploadImageRequest">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="UploadImageResponse">
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetImageOperation">
    <soap:operation soapAction="downloadImage"/>
    <wsdl:input name="GetImageRequest">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="GetImageResponse">
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
```

```
</binding>
```

## Remote Service Stub

The following example shows the generated stub for the remote service.

```
public interface GetImage extends java.rmi.Remote {  
    public void getImageOperation(java.lang.String getImageRequest,  
        javax.xml.rpc.holders.StringHolder getImageResponse,  
        org.apache.axis.holders.MimeMultipartHolder image)  
        throws java.rmi.RemoteException;  
}  
//Following is a code snippet for receiving images  
//Instantiate the right stub  
GetImage getImageStub = (GetImageSoapBindingStub) stub;  
//get the imageRequest xml  
String xml = getImageRequestXml();  
//Instantiate the StringHolder() to hold the getImage response xml  
StringHolder getImageResponse = new StringHolder();  
//Instantiate MimeMultipartHolder() to hold the images in the response  
MimeMultipartHolder images = new MimeMultipartHolder();  
//invoke getImageOperation  
getImageStub.getImageOperation(xml,getImageResponse,images);  
//get the images from the response  
saveImages(images);  
//Implementation of saveImages(MimeMultipartHolder images)  
private void saveImages(MimeMultipartHolder images) {  
    MimeMultipart mimeMultipart = MimeMultipartHolder.value;  
    for(int i=0;i<mimeMultipart.getCount();i++){  
        MimeBodyPart bodyPart = (MimeBodyPart) getBodyPart(i);  
        InputStream is = bodyPart.getDataHandler().getInputStream();  
        //generate fileName from contentId  
        String fileName = getFilePath() + File.separator + bodyPart.getContentType();  
        //save the image  
        saveImage(is,fileName);  
    }  
}
```

## Service

```
@WebServiceClient(name = "RTSI-SI-Services", targetNamespace =
"http://www.visa.com/rtsi", wsdlLocation = "../..../VROL-RTSIService.wsdl")
public class RTSISIServices
    extends Service
{
    private final static URL RTSISISERVICES_WSDL_LOCATION;
    static {
        URL url = null;
        try {
            url = new URL("../..../VROL-RTSIService.wsdl");
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
        RTSISISERVICES_WSDL_LOCATION = url;
    }
    public RTSISIServices(URL wsdlLocation, QName serviceName) {
        super(wsdlLocation, serviceName);
    }
    public RTSISIServices() {
        super(RTSISISERVICES_WSDL_LOCATION, new QName("http://www.visa.com/rtsi",
"RTSI-SI-Services"));
    }
    /**
     *
     * @return
     *      returns SIRTSIOperations
     */
    @WebEndpoint(name = "SI-RTSIOperationsPort")
    public SIRTSIOperations getSIRTSIOperationsPort() {
        return (SIRTSIOperations)super.getPort(new QName("http://www.visa.com/rtsi", "SI-
RTSIOperationsPort"), SIRTSIOperations.class);
    }
    /**
     *
     * @param features
     *      A list of {@link javax.xml.ws.WebServiceFeature} to configure on the proxy.
     * Supported features not in the <code>features</code> parameter will have their default
     * values.
     * @return
     *      returns SIRTSIOperations
     */
    @WebEndpoint(name = "SI-RTSIOperationsPort")
    public SIRTSIOperations getSIRTSIOperationsPort(WebServiceFeature... features) {
        return (SIRTSIOperations)super.getPort(new QName("http://www.visa.com/rtsi", "SI-
RTSIOperationsPort"), SIRTSIOperations.class, features);
    }
}
```

## Operations

```
@WebService(name = "SI-RTSIOperations", targetNamespace = "http://www.visa.com/rtsi")
@SOAPBinding(parameterStyle = SOAPBinding.ParameterStyle.BARE)
@XmlSeeAlso({
    ObjectFactory.class
})
```

```
public interface SIRTSIOperations {  
  
    /** *  
     * @param rfcFulfillmentRequest  
     * @return  
     *      returns com.visa.rolsi.SISubmitRFCFulfillmentResponseType  
     */  
    @WebMethod(operationName = "SubmitFulfillmentOperation", action =  
    @WebResult(name = "SISubmitRFCFulfillmentResponse", targetNamespace =  
"http://www.visa.com/ROLSI", partName = "RFCFulfillmentResponse")  
    public SISubmitRFCFulfillmentResponseType submitFulfillmentOperation(  
        @WebParam(name = "SISubmitRFCFulfillmentRequest", targetNamespace =  
"http://www.visa.com/ROLSI", partName = "RFCFulfillmentRequest")  
        SISubmitRFCFulfillmentRequestType rfcFulfillmentRequest);  
  
    /** *  
     * @param getFulfillmentRequest  
     * @return  
     *      returns com.visa.rolsi.SIGetRFCFulfillmentResponseType  
     */  
    @WebMethod(operationName = "GetFulfillmentOperation", action = "downloadImage")  
    @WebResult(name = "SIGetRFCFulfillmentResponse", targetNamespace =  
"http://www.visa.com/ROLSI", partName = "GetFulfillmentResponse")  
    public SIGetRFCFulfillmentResponseType getFulfillmentOperation(  
        @WebParam(name = "SIGetRFCFulfillmentRequest", targetNamespace =  
"http://www.visa.com/ROLSI", partName = "GetFulfillmentRequest")  
        SIGetRFCFulfillmentRequestType getFulfillmentRequest);  
  
    /** *  
     * @param uploadMessageImageRequest  
     * @return  
     *      returns com.visa.rolsi.SIUploadMessageImageResponseType  
     */  
    @WebMethod(operationName = "UploadMessageImageOperation", action =  
"urn:#UploadMessageImageOperation")  
    @WebResult(name = "SIUploadMessageImageResponse", targetNamespace =  
"http://www.visa.com/ROLSI", partName = "UploadMessageImageResponse")  
    public SIUploadMessageImageResponseType uploadMessageImageOperation(  
        @WebParam(name = "SIUploadMessageImageRequest", targetNamespace =  
"http://www.visa.com/ROLSI", partName = "UploadMessageImageRequest")  
        SIUploadMessageImageRequestType uploadMessageImageRequest);  
  
    /** *  
     * @param getMessageImageRequest  
     * @return  
     *      returns com.visa.rolsi.SIGetMessageImageResponseType  
     */  
    @WebMethod(operationName = "GetMessageImageOperation", action =  
"downloadImage")
```

```
@WebResult(name = "SIGetMessageImageResponse", targetNamespace =
"http://www.visa.com/ROLSI", partName = "GetMessageImageResponse")
public SIGetMessageImageResponseType getMessageImageOperation(
    @WebParam(name = "SIGetMessageImageRequest", targetNamespace =
"http://www.visa.com/ROLSI", partName = "GetMessageImageRequest")
    SIGetMessageImageRequestType getMessageImageRequest);

/**
 *
 * @param uploadImageRequest
 * @return
 *      returns com.visa.rolsi.SIUuploadImageResponseType
 */
@WebMethod(operationName = "UploadImageOperation", action =
"urn:#UploadImageOperation")
@WebResult(name = "SIUploadImageResponse", targetNamespace =
"http://www.visa.com/ROLSI", partName = "UploadImageResponse")
public SIUploadImageResponseType uploadImageOperation(
    @WebParam(name = "SIUploadImageRequest", targetNamespace =
"http://www.visa.com/ROLSI", partName = "UploadImageRequest")
    SIUploadImageRequestType uploadImageRequest);

/**
 *
 * @param getImageRequest
 * @return
 *      returns com.visa.rolsi.SIGetImageResponseType
 */
@WebMethod(operationName = "GetImageOperation", action = "downloadImage")
@WebResult(name = "SIGetImageResponse", targetNamespace =
"http://www.visa.com/ROLSI", partName = "GetImageResponse")
public SIGetImageResponseType getImageOperation(
    @WebParam(name = "SIGetImageRequest", targetNamespace =
"http://www.visa.com/ROLSI", partName = "GetImageRequest")
    SIGetImageRequestType getImageRequest);
}
```

## Client

```
SIUploadImageResponseType ImageResponse=null;
SIUploadImageRequestType Imagerequest=null;
RTSISIServices clientSIService=new RTSISIServices();
ImageResponse=clientSIService.getSIrtsOperationsPort().
uploadImageOperation(Imagerequest);
UploadDocumentResponseType
uploadDocumentResponse=ImageResponse.getResponseData();
```

## HTTP Headers

To pass the HTTP headers that you want to send in the request, you have to set the `HTTP-Request-Headers` property on the `MessageContext` object. This can be accomplished as follows in any handler, as shown below.

```
public class XXXSoapHandler extends GenericHandler{
private static String HTTP_REQUEST_HEADERS = "HTTP-Request-
Headers";
Map headers = new HashMap();
public final boolean handleRequest(MessageContext context) {
    ...
    //set headers
    headers.put("xyz","abc");
    //set the HTTP_REQUEST_HEADERS property
    context.setProperty(HTTP_REQUEST_HEADERS,headers);
}
```

# Web Service Clients

E

This appendix describes how to write the communications components of a Java-based Web services client. This client was used for RTSI testing.

All components of the client were written using publicly available, non-proprietary, software packages. It permits future expansion while minimizing the amount of modification that would be necessary due to changes in RTSI definitions.

## Tools

The following software packages were used in the construction of the client for Web services.

- Java SE 6 or later—Basic Java package.
- Apache Axis 2.x or later, or JAX WS 2.0 (available in Java SE 6 or later)

## Architecture

The client is implemented using a static architecture, because the services provided by the server are determined at build time by examination of the WSDL files, rather than at run time.

## Building the Client

This section discusses the basic steps necessary to build the static client.

### Convert WSDL Files

The WSDL file is converted in to an associated set of Java classes by the Apache Axis wsdl2java utility.

WSDL Filename	Purpose
VROL-RTSIService.wsdl	Web services for image upload/download and all functions.

## Implement SOAP Handlers

Members may implement SOAP handlers so that pertinent security-related data is transferred between the client and the server for each SOAP request. A basic handler is responsible for inserting pertinent security data to the SOAP and HTTP headers for each request and for capturing data returned by the server in the response.

It is necessary to implement a subclass of this basic handler for any services that need special handling, such as login and RFC attachments (for example, images).

## Enable MTOM/XOP for Image Operations

MTOM needs to be enabled by the client for these services only:

- SIGetRFCFulfillment
- SISubmitRFCFulfillment
- SIGetImage
- SIUploadImage
- SIGetMessageImage
- SIUploadMessageImage
- SIGetDisputeDetails
- SICreateDisputeResponse
- SIGetDisputeResponseDetails
- SICreateDisputePreArb
- SIGetDisputePreArbDetails
- SICreateDisputePreArbResponse
- SIGetDisputePreArbResponseDetails
- SISubmitDisputeQuestionnaire
- SISubmitDisputeFiling
- SIGetDisputeFilingDetails
- SISubmitContactMessageResponse
- SIGetContactMessageResponseDetails
- SIGetContactMessageDetails
- SICreateDisputePreComp
- SICreateDisputePreCompResponse
- SIGetDisputePreCompDetails
- SIGetDisputePreCompResponseDetails
- SISubmitQFDisputeQuestionnaire

There are two ways to configure this in Java, depending upon the toolkit used. In the following example for Axis 2.x:

```
SOAPBinding binding = (SOAPBinding) ((BindingProvider
siOperations).getBinding());
binding.setMTOMEnabled(true);
```

Java JAX WS enables the passing of MTOMFeature as part of the creation of the operations port:

```
SIRTSIOperations sirtOperations=clientSIService.  
getSIRTSIOperationsPort(new MTOMFeature());
```

MTOMFeature can be represented with a specified threshold value, as shown in the following example:

```
SIRTSIOperations sirtOperations= clientSIService.  
getSIRTSIOperationsPort(new MTOMFeature(1000));
```

Where the value provided is the threshold at which the HTTP encoder passes the image data as a binary attachment. By default, the value is 0. Files with a size in bytes under the threshold value are Base64 encoded and sent in-line by the HTTP encoder. The use of Base64 encoding adds a 33% overhead on the transmission of binary data; thus, threshold of about 1000 or so is optimal considering the overhead of packaging a second MIME part in the request. Most files that are uploaded into VROL are well in excess of 1000 bytes so this threshold is not particularly relevant for VROL data. The service will return an error if large files are sent using Base64 encoding instead of as binary attachments.

## HTTP Headers

The standard SOAP handlers do not provide a means to access or modify the contents of HTTP headers. Because security information is transferred in the HTTP header, it is necessary to modify the standard HTTSPort module provided as part of the Apache Axis package to handle this requirement.

## Client With XOP, MTOM Binding

```
SIUploadImageRequestType uploadImage=null;  
javax.activation.DataHandler dataHandler = new  
javax.activation.DataHandler(new FileDataSource("../T12345"));  
UploadDocType uploadDocType=new UploadDocType();  
uploadDocType.setDocType("CALET");  
uploadDocType.setVisaCaseNumber(new Long("123456"));  
RTSIAjaxDescriptorType value=new RTSIAjaxDescriptorType();  
value.setAttachType(SOAPAttachType.MIME);  
List<RTSIAjaxAttachmentType> attachList=value.getAttachment();  
RTSIAjaxAttachmentType rtsiAttachmentType=new RTSIAjaxAttachmentType();  
rtsiAttachmentType.setImageData(dataHandler);  
rtsiAttachmentType.setId("1");  
rtsiAttachmentType.setContentType("image/tiff");  
rtsiAttachmentType.setComment("Comment");  
attachList.add(rtsiAttachmentType);  
uploadDocType.setAttachmentDescriptor(value);  
uploadImage.setRequestData(uploadDocType);  
RTSISIServices clientSIService=new RTSISIServices();  
SIRTSIOperations sirtOperations=  
clientSIService.getSIRTSIOperationsPort();  
SOAPBinding binding = (SOAPBinding) ((BindingProvider)  
sirtOperations).getBinding();  
binding.setMTOMEnabled(true);  
sirtOperations.uploadImageOperation(uploadImage);
```

### Client With download XOP and MTOM Element

```

SIUploadImageResponseType ImageResponse=null;
SIUploadImageRequestType Imagerequest=null;
RTSISIServices clientSIService=new RTSISIServices();
ImageResponse=clientSIService.getSIRTSIOperationsPort().
uploadImageOperation(Imagerequest);
UploadDocumentResponseType
uploadDocumentResponse=ImageResponse.getResponseData();

```

## Submitting RFC Fulfillment Images

The following steps indicate how to develop the code to submit RFC fulfillment images.

1. Generate client stubs for RFCService and port SubmitFulfillment from RTSIServices.wsdl.
  2. Create a test class with a main method.
  3. Instantiate ServiceFactory.
  4. Create a Service for RFCService.
- ```

javax.xml.rpc.Service service = serviceFactory.
createService(wsdl,serviceName);

```
5. Register a handler for the RFC service:
    - Get HandlerClass for port SubmitFulfillment from SOAPHandlerFactory

```

Class handlerClass =
SOAPHandlerFactory.getHandlerClass(portName);

```

    - Create HandlerInfo by passing in the HandlerClass

```

HandlerInfo info = new HandlerInfo(handlerClass, null,
null);

```

    - Get HandlerRegistry from service (Step 4)

```

HandlerRegistry handlerRegistry =
service.getHandlerRegistry();

```

    - Set the handler chain to this HandlerRegistry for a corresponding Port

```

handlerRegistry.setHandlerChain(portQName, handlerList);

```
  6. Instantiate the SubmitFulfillment stub created in step 1.
- ```

SubmitFulfillment submitFulfillmentStub
= (SubmitFulfillmentSoapBindingStub) stub;

```
7. Load image (see "Image Handling Code" below).
  8. Invoke submit FulfillmentOperation on stub and receive response.
- ```

String respXml =
submitFulfillmentStub.submitFulfillmentOperation(xml,
(MimeMultipart) image);

```

## Image Handling Code

The following steps indicate how to develop the code for image handling.

1. Instantiate MimeMultipart.
2. For every image file to be attached, Instantiate a MimeBodyPart.

3. Read the image file, create a Datahandler (either custom or the default) for the image file.
4. Set this datahandler object on the MimeBodyPart.
5. Add any custom headers (Content-Type, Content-Id) to the MimeBodyPart.
6. Add this MimeBodyPart to the MimeMultipart.
7. Repeat steps 2 to 6 for each image file.

```
//instantiate MimeMultipart
mpart = new MimeMultipart();
File[] files = f.listFiles();
for(int i=0;i<files.length;i++){
//instantiate MimeBodyPart
MimeBodyPart mbp1 = new MimeBodyPart();
//read image file
FileInputStream fis = new FileInputStream(files[i].getAbsolutePath());
//create DataHandler for the image
ImagingDatahandler imagedataHandler =
new ImagingDatahandler("image",contentType,fis);
//set DataHandler on MimBodyPart
mbp1.setDataHandler(imagedataHandler);
//add header info (optional)
mbp1.addHeader("Content-Type",contentType);
mbp1.addHeader("Content-Id","abc" + (i+1));
//add MimeBodyPart to MimeMultiPart
mpart.addBodyPart(mbp1);
}
```

8. Pass this MimeMultipart to webservices stub operation (see Step 8 in [Submitting RFC Fulfillment Images](#) above).

THIS PAGE INTENTIONALLY LEFT BLANK.

# Sample Project Plan

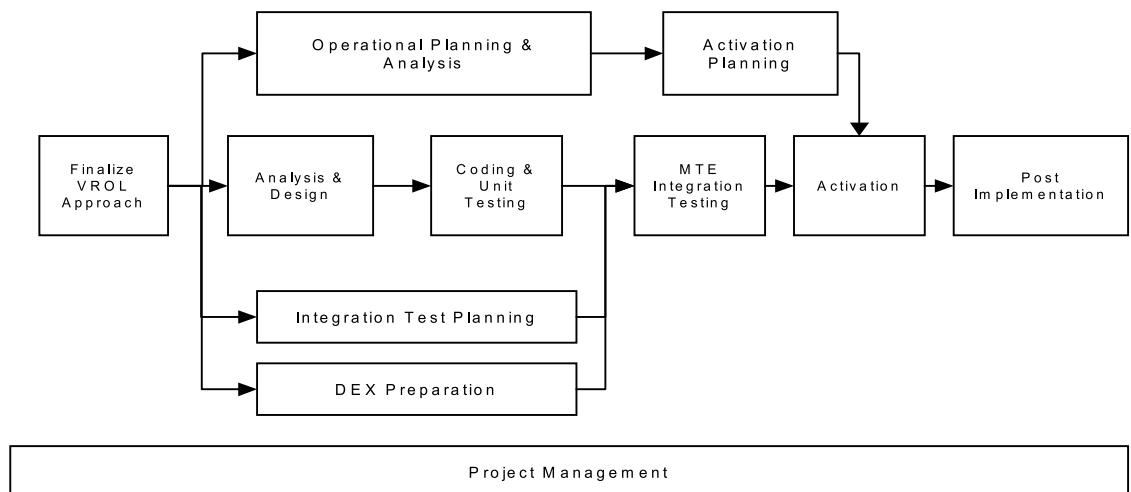
F

This sample project plan outlines the activities associated with the implementation of a Visa Resolve Online systems interface (Real-Time Systems Interface or Bulk Systems Interface).

This information is intended to help members estimate staffing and the effort required to implement a systems interface. Contact Global Product Office (GPO) to obtain the latest version of the Sample Project Plan.

[\*\*Figure F-1\*\*](#) outlines the major phases of the implementation of a Visa Resolve Online systems interface. This high-level approach can be used for either of the systems interfaces. Visa can provide more detailed sample implementation plans as needed.

**Figure F-1: Project Approach**



**Table F-1: Sample Project Plan**

|          |                                            |  | <b>Task Name</b>                                    | <b>Est. Duration</b> | <b>Resource</b>                                    |
|----------|--------------------------------------------|--|-----------------------------------------------------|----------------------|----------------------------------------------------|
| <b>1</b> | <b>Finalize VROL Approach</b>              |  |                                                     |                      |                                                    |
|          | 1.1                                        |  | Confirm approach with business and technology teams |                      | Business Lead, Technology Lead, Visa               |
|          | 1.2                                        |  | Perform cost analysis                               |                      | Business Lead, Technology Lead, Visa               |
|          | 1.3                                        |  | Work with Visa to finalize approach and schedule    |                      | Project Manager, Visa                              |
| <b>2</b> | <b>Operational Planning &amp; Analysis</b> |  |                                                     |                      |                                                    |
|          | 2.1                                        |  | Assess workflow modifications                       |                      | Back Office Operations                             |
|          | 2.2                                        |  | Assess resource requirements and roles              |                      | Back Office Operations                             |
|          | 2.3                                        |  | Determine imaging requirements                      |                      | Back Office Operations                             |
|          | 2.4                                        |  | Assess other business impacts or other system usage |                      | Back Office Operations                             |
|          | 2.5                                        |  | Determine migration path to system interface        |                      | Back Office Operations                             |
|          | 2.6                                        |  | Define detailed business requirements               |                      | Back Office Operations                             |
| <b>3</b> | <b>Analysis &amp; Design</b>               |  |                                                     |                      |                                                    |
|          | 3.1                                        |  | Define functional specifications                    |                      | Development Staff                                  |
|          | 3.2                                        |  | Define data requirements                            |                      | Development Staff                                  |
|          | 3.3                                        |  | Define reporting requirements                       |                      | Development Staff                                  |
|          | 3.4                                        |  | Define technical requirements                       |                      | Development Staff                                  |
|          | 3.5                                        |  | Complete system integration design                  |                      | Development Staff                                  |
| <b>4</b> | <b>Integration Test Planning</b>           |  |                                                     |                      |                                                    |
|          | 4.1                                        |  | Determine scope of integration testing              |                      | Testers, Development Staff, Back Office Operations |
|          | 4.2                                        |  | Develop integration test plan                       |                      | Testers                                            |
|          | 4.3                                        |  | Schedule Member Testing Environment (MTE) with Visa |                      | Project Manager, Visa                              |
|          | 4.4                                        |  | Complete VROL Enrollment Form for MTE               |                      | Project Manager, Visa                              |
|          | 4.5                                        |  | Review integration test plan with Visa              |                      | Project Manager, Visa                              |
|          | 4.6                                        |  | Test new operational                                |                      | n/a                                                |
| <b>5</b> | <b>Complete Development and Unit Test</b>  |  |                                                     |                      |                                                    |

**Table F-1: Sample Project Plan (cont'd.)**

|          |                                |       | <b>Task Name</b>                                          | <b>Est. Duration</b> | <b>Resource</b>          |
|----------|--------------------------------|-------|-----------------------------------------------------------|----------------------|--------------------------|
|          | 5.1                            |       | Code and test logon integration module                    |                      | Development Staff        |
|          | 5.2                            |       | Code and test initiate dispute request/response module    |                      | Development Staff        |
|          | 5.3                            |       | Code and test submit/pend response and case update module |                      | Development Staff        |
|          | 5.4                            |       | Code and test incoming dispute decisioning module         |                      | Development Staff        |
|          | 5.5                            |       | Code and test continue dispute request/response module    |                      | Development Staff        |
|          | 5.6                            |       | Code and test imaging integration module                  |                      | Development Staff        |
|          | 5.7                            |       | Code and test bulk print file handling module             |                      | Development Staff        |
| <b>6</b> | <b>Activation Planning</b>     |       |                                                           |                      |                          |
|          | 6.1                            |       | Develop activation plan                                   |                      | n/a                      |
|          |                                | 6.1.1 | Develop pilot plan                                        |                      | Back Office Operations   |
|          |                                | 6.1.2 | Define success criteria for full-scale roll out           |                      | Back Office Operations   |
|          |                                | 6.1.3 | Define fallback plan                                      |                      | Back Office Operations   |
|          | 6.2                            |       | Develop training plan                                     |                      |                          |
|          |                                | 6.2.1 | Member system training                                    |                      | Member Trainers          |
|          |                                | 6.2.2 | VROL training                                             |                      | Member Trainers, Visa    |
|          | 6.3                            |       | Identify users and assign roles                           |                      | Site Administrator       |
|          | 6.4                            |       | Setup workstation and scanning equipment                  |                      | Back Office Operations   |
|          | 6.5                            |       | Obtain Visa Online (VOL) login user names for all users   |                      | Site Administrator       |
|          | 6.6                            |       | Review activation plan with Visa                          |                      | Project Manager, Visa    |
|          | 6.7                            |       | Complete VROL Enrollment Form for production              |                      | Project Manager          |
| <b>7</b> | <b>MTE Integration Testing</b> |       |                                                           |                      |                          |
|          | 7.1                            |       | Process VROL Enrollment Form for MTE                      | 4 days               | Visa                     |
|          | 7.2                            |       | Promote code to member's test environment                 |                      | Host Application Support |
|          | 7.3                            |       | Assign roles to user names                                |                      | Site Administrator       |

**Table F-1: Sample Project Plan (cont'd.)**

|          |                            |       | <b>Task Name</b>                                                   | <b>Est. Duration</b> | <b>Resource</b>                                  |
|----------|----------------------------|-------|--------------------------------------------------------------------|----------------------|--------------------------------------------------|
|          | 7.4                        |       | Execute integration test plan                                      |                      | Testers                                          |
|          |                            | 7.4.1 | Test logon integration                                             |                      | Testers                                          |
|          |                            | 7.4.2 | Test initiate dispute                                              |                      | Testers                                          |
|          |                            | 7.4.3 | Test continue dispute decisioning                                  |                      | Testers                                          |
|          |                            | 7.4.4 | Test continue dispute                                              |                      | Testers                                          |
|          |                            | 7.4.5 | Test exception handling                                            |                      | Testers                                          |
|          |                            | 7.4.6 | Test imaging integration                                           |                      | Testers                                          |
|          |                            | 7.4.7 | Test other operational processes (reporting, bulk mail, and so on) |                      | Testers                                          |
|          | 7.5                        |       | Review test results                                                |                      | Back Office Operations, Development Staff        |
|          | 7.6                        |       | Confirm activation readiness                                       |                      | Project Manager, Visa                            |
| <b>8</b> | <b>Activation</b>          |       |                                                                    |                      |                                                  |
|          | 8.1                        |       | Process VROL Enrollment Form for production                        | 4 days               | Visa                                             |
|          | 8.2                        |       | Promote code to production system                                  |                      | Host Application Support                         |
|          | 8.3                        |       | Assign roles to user names                                         |                      | Site Administrator                               |
|          | 8.4                        |       | Execute Activation Plan                                            |                      |                                                  |
|          |                            | 8.4.1 | Execute pilot plan                                                 |                      | Back Office Operations                           |
|          |                            | 8.4.2 | Monitor pilot                                                      |                      | Back Office Operations                           |
|          |                            | 8.4.3 | Complete full-scale rollout                                        |                      | Back Office Operations, Host Application Support |
| <b>9</b> | <b>Post-Implementation</b> |       |                                                                    |                      |                                                  |
|          | 9.1                        |       | Continue production monitoring                                     |                      | Back Office Operations                           |
|          | 9.2                        |       | Review and adjust workflow or reporting                            |                      | Back Office Operations                           |
|          | 9.3                        |       | Plan for future releases                                           |                      | Business Lead, Technology Lead                   |
|          | 9.4                        |       | Retire legacy systems or processes                                 |                      | Business Lead, Technology Lead                   |

# System Diagnostic Report

G

The System Diagnostic Report feature can be used to find the status of transactions, which have been submitted through UI, Bulk SI and RTSI and sent to external systems. Users monitoring the Member Testing Environment can find possible environment issues or network/connectivity issues. This report lists the count of specific transactions in various statuses and the count of transactions that do not fall into any of the known statuses (as a discrepancy).

To access this feature, a user must have a VOL ID that can access the VROL UI and must be configured also with a VROL role that has this access, such as the Administrator or the Universal User role.

The user logs into the VROL UI and selects the menu option: **Admin, System Diagnostic Report.**

**Figure G–1: System Diagnostic Report**

Currently logged on as: tesra7005u - Member SMS (A1 - Org D) HOME | CLOSE

**VISA** Visa Resolve Online Quick Search:  Go

Inquiry Queues Reports Admin Manual Transactions Other Actions

**View Switch System Diagnostic Report**

**Date Range (Daily):**  
From: 09/20/11  To: 09/27/11

**Activity Type:**  
 Issuer Activity  Acquirer Activity  All Activity

**Message Direction:**  
 Incoming  Outgoing

| Destination To:                         | Source(Interface):                      | Transaction Type:                         | Transaction Status:                                                   |
|-----------------------------------------|-----------------------------------------|-------------------------------------------|-----------------------------------------------------------------------|
| <input type="checkbox"/> Base II        | <input type="checkbox"/> UI             | <input type="checkbox"/> Chargeback       | <input checked="" type="checkbox"/> Submitted                         |
| <input type="checkbox"/> VIP            | <input type="checkbox"/> RTSI           | <input type="checkbox"/> Representation   | <input type="checkbox"/> Awaiting Network Response                    |
| <input checked="" type="checkbox"/> All | <input type="checkbox"/> Bulk SI        | <input type="checkbox"/> Request For Copy | <input type="checkbox"/> Awaiting Exception Manager Response          |
|                                         | <input checked="" type="checkbox"/> All | <input type="checkbox"/> Adjustment       | <input checked="" type="checkbox"/> Expired Awaiting Network Response |
|                                         |                                         | <input type="checkbox"/> Auto Chargeback  | <input type="checkbox"/> Accepted By Network                          |
|                                         |                                         | <input type="checkbox"/> Misc Fee         | <input type="checkbox"/> Rejected by Network                          |
|                                         |                                         | <input checked="" type="checkbox"/> All   | <input checked="" type="checkbox"/> All                               |

**Figure G-2: System Diagnostic Report — Results**

Currently logged on as: testra7085u - Member SMS IA1 - Org D

HOME | CLOSE

VISA Visa Resolve Online

Quick Search:  Go

Inquiry Queues Reports Admin Manual Transactions Other Actions

View Switch System Diagnostic Report - Results

Sort By: Default Sort | Results 1 - 60

Date Source Transaction Type Received Matched UnMatched New cases Created RTS/Batch Queued Enrolled Out Bulk Download Status Pending/Generated/Sent

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 09/20/11 | VP | Adjustment | 7 | 7 | 0 | 7 | 12 | 0 | 0/0/29 |
| 09/20/11 | Base II | Chargeback Advice | 15 | 13 | 2 | 3 | 0 | 0 | 0/0/0 |
| 09/20/11 | Base II | Chargeback | 10 | 10 | 0 | 0 | 6 | 0 | 0/0/6 |
| 09/20/11 | VP | Chargeback | 69 | 68 | 1 | 1 | 62 | 0 | 0/0/95 |
| 09/20/11 | Base II | Msc Fee | 30 | 30 | 0 | 1 | 20 | 0 | 0/0/29 |
| 09/20/11 | VP | Msc Fee | 12 | 12 | 0 | 0 | 6 | 0 | 0/0/12 |
| 09/20/11 | Base II | Representement Doc | 8 | 8 | 0 | 2 | 0 | 0 | 0/0/0 |
| 09/20/11 | Base II | Representement | 1 | 1 | 0 | 0 | 1 | 0 | 0/0/1 |
| 09/20/11 | VP | Representement | 2 | 1 | 1 | 0 | 3 | 0 | 0/0/4 |
| 09/20/11 | Base II | Request For Copy | 3 | 3 | 0 | 0 | 3 | 0 | 0/0/3 |
| 09/20/11 | VP | Request For Copy | 6 | 6 | 0 | 0 | 12 | 0 | 0/0/14 |
| 09/21/11 | VP | Adjustment | 1 | 1 | 0 | 1 | 0 | 0 | 0/0/6 |
| 09/21/11 | Base II | Chargeback Advice | 14 | 12 | 2 | 4 | 0 | 0 | 0/0/0 |
| 09/21/11 | Base II | Chargeback | 20 | 20 | 0 | 6 | 13 | 0 | 0/0/19 |
| 09/21/11 | VP | Chargeback | 53 | 50 | 3 | 1 | 91 | 0 | 0/0/115 |
| 09/21/11 | Base II | Msc Fee | 37 | 37 | 0 | 2 | 36 | 0 | 0/0/36 |
| 09/21/11 | VP | Msc Fee | 4 | 4 | 0 | 3 | 3 | 0 | 0/0/6 |
| 09/21/11 | Base II | Representement Doc | 8 | 7 | 1 | 0 | 0 | 0 | 0/0/0 |
| 09/21/11 | Base II | Representement | 8 | 6 | 2 | 0 | 5 | 0 | 0/0/7 |
| 09/21/11 | VP | Representement | 8 | 6 | 2 | 0 | 14 | 0 | 0/0/15 |
| 09/21/11 | Base II | Request For Copy | 41 | 41 | 0 | 0 | 115 | 0 | 0/0/113 |
| 09/21/11 | VP | Request For Copy | 11 | 11 | 0 | 0 | 23 | 0 | 0/0/29 |
| 09/22/11 | VP | Adjustment | 0 | 0 | 0 | 0 | 0 | 0 | 0/0/1 |
| 09/22/11 | Base II | Chargeback Advice | 433 | 33 | 400 | 23 | 0 | 0 | 0/0/0 |
| 09/22/11 | Base II | Chargeback | 9 | 9 | 0 | 0 | 5 | 0 | 0/0/5 |
| 09/22/11 | VP | Chargeback | 23 | 23 | 0 | 0 | 61 | 0 | 0/0/68 |
| 09/22/11 | Base II | Msc Fee | 5 | 5 | 0 | 0 | 0 | 0 | 0/0/0 |
| 09/22/11 | VP | Msc Fee | 14 | 14 | 0 | 4 | 7 | 0 | 0/0/4 |

Back Print Close

THIS PAGE INTENTIONALLY LEFT BLANK.

# Alternative Web Services

H

## Alternative Transaction Inquiry Web Services

### Transaction Inquiry Using One or More VisaNet Transaction IDs

#### Overview

The SISubmitMultiTranInquiryRequest/Response Web service is optional and **implementation is not required**. It is an alternative service apart from the required TI Web services described in section. This Web service provides an alternative method to search for transactions using 1 to 10 VisaNet Transaction IDs as the search parameters.

#### Why use this Web service?

When Member systems already have the VisaNet Transaction IDs for the transactions of interest, this Web service may offer a more efficient method for the Member system.

Normally, this Web service executes synchronously. However, if additional search time is required by VROL, it may execute asynchronously by returning a search reference (EventID). Subsequently, the reference is used by the same Web service request to retrieve the search results.

#### Procedure

The detailed steps are as follows:

#### Synchronous

1. The member host system triggers a SISubmitMultiTranInquiryRequest message supplying one or more TransactionIDs in a single request.

**Note:** You can include up to 10 TransactionIDs in the request.

2. VROL returns a SISubmitMultiTranInquiryResponse message with success status. The response includes a Transaction Summary for every transaction that matched one of the given Transaction ID(s).

#### Asynchronous

1. The member host system triggers a SISubmitMultiTranInquiryRequest message supplying one or more TransactionIDs in a single request.

**Note:** You can include up to 10 TransactionIDs in the request.

2. If VROL determines that response is asynchronous, then VROL sends a TIEventID in the response.
3. Subsequently, the member host system triggers a SISubmitMultiTranInquiryRequest message supplying the TIEventID to retrieve the Transaction Summary for every transaction that matched one of the given Transaction ID(s).

**Note:** If VROL does not yet have the results, it returns the I-300300000 message.

### Key Elements and Sample Code for SISubmitMultiTranInquiryRequest

The following table provides information about some XML elements that should be considered when submitting a TI request using one or more TransactionIDs, or retrieving a response.

#### SISubmitMultiTranInquiryRequest

| Element       | Required    | Description                                                                                                                                                                                                                                                                                                                  |
|---------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TransactionID | Yes         | If a Multiple Clearing Sequence Number (MCSN) exists for the transaction, the two-digit MCSN can be added to the end of the transaction ID in the request and will be returned if available in the response. A transaction ID is typically 15 digits; however, the transaction ID will be 17 digits with an additional MCSN. |
| TIEventID     | Conditional | Unique internal ID returned in response. This element is required when retrieving the results of a request that went asynchronous.                                                                                                                                                                                           |

#### SISubmitMultiTranInquiryResponse

| Element                | Description                                                                                                                                                                                                                                                                                                                  |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TIEventID              | Unique internal ID returned in response.                                                                                                                                                                                                                                                                                     |
| TransactionID          | If a Multiple Clearing Sequence Number (MCSN) exists for the transaction, the two-digit MCSN can be added to the end of the transaction ID in the request and will be returned if available in the response. A transaction ID is typically 15 digits; however, the transaction ID will be 17 digits with an additional MCSN. |
| AssociatedTransGroupId | Uniquely identifies a list of associated transactions.<br>Members can use this to retrieve a list of associated transactions or to associate a group of transactions to a case.                                                                                                                                              |

The SISubmitMultiTranInquiry operation contains request and response messages as shown in the following sample messages.

#### Request Message—SOAP-based Services

```
<SISubmitMultiTranInquiryRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <TransactionID>a</TransactionID>
    <TransactionID>a</TransactionID>
    <TransactionID>a</TransactionID>
    <TIEventID>0</TIEventID>
  </requestData>
</SISubmitMultiTranInquiryRequest>
```

### Response Message—SOAP-based Services

```
<SISubmitMultiTranInquiryResponse>
  <Status>...</Status>
  <responseData>
    <TIEventID>0</TIEventID>
    <Repository>VTRS</Repository>
    <TransactionSummary>
      <RolTransactionId>0</RolTransactionId>
      <IdentifiedTransaction>true</IdentifiedTransaction>
      <TransactionType>String</TransactionType>
      <TransactionTypeShortDescription>String
      </TransactionTypeShortDescription>
      <TransactionDateTime>2017-01-01T09:30:47Z</TransactionDateTime>
      <CPD>2017-01-01</CPD>
      <BusinessAppId>aa</BusinessAppId>
      <BusinessAppIdDesc>String</BusinessAppIdDesc>
      <SourceOfFunds>a</SourceOfFunds>
      <TransactionAmount currency="000">0</TransactionAmount>
      <TranAmountDesc>String</TranAmountDesc>
      <MerchantName>a</MerchantName>
      <MerchantCity>a</MerchantCity>
      <MerchantCountrySubEntityCode>a</MerchantCountrySubEntityCode>
      <MerchantCountryCode>AA</MerchantCountryCode>
      <PosEntryMode>String</PosEntryMode>
      <PaymentFacilitatorId>1234</PaymentFacilitatorId>
      <SubMerchantId>1123</SubMerchantId>
      <PosEntryModeDescription>String</PosEntryModeDescription>
      <MoToEciCode>String</MoToEciCode>
      <MoToEciDescription>String</MoToEciDescription>
      <NetworkID>aaa</NetworkID>
      <AuthorizationResponseCode>String</AuthorizationResponseCode>
      <AuthorizationResponseDescription>String
      </AuthorizationResponseDescription>
      <AuthorizationCode>String</AuthorizationCode>
      <MCC>0000</MCC>
      <MCCDescription>String</MCCDescription>
      <TransactionID>String</TransactionID>
      <SpecialMerchantID>1111</SpecialMerchantID>
      <ParceladoPaymentType>String</ParceladoPaymentType>
      <RetrievalReferenceNumber>a</RetrievalReferenceNumber>
      <ARN>00000000000000000000000000000000</ARN>
      <CardNumber>000000000000</CardNumber>
      <Token>String</Token>
      <CVV2Result>String</CVV2Result>
      <CVV2ResultDesc>String</CVV2ResultDesc>
      <PurchaseInquiryEligibleInd>true
      </PurchaseInquiryEligibleInd>
      <RdrEligibilityInd>true</RdrEligibilityInd>
      <AssociatedTransGroupId>0</AssociatedTransGroupId>
    </TransactionSummary>
    <Repository>VTRS</Repository>
    <TransactionSummary>...</TransactionSummary>
    <Repository>VTRS</Repository>
    <TransactionSummary>...</TransactionSummary>
  </responseData>
</SISubmitMultiTranInquiryResponse>
```

**Request Message—REST-based Services**

```
{
  "RequestHeader": {
    "User": {
      "@id": "a",
      "@type": "internalId"
    },
    "MemberRole": "A"
  },
  "requestData": {
    "TransactionID": [
      "a",
      "a",
      "a"
    ],
    "TIEventID": 0
  }
}
```

**Response Message—REST-based Services**

```
{
  "Status": {
    "Code": "a",
    "Message": "a",
    "ErrorIn": "String"
  },
  "responseData": {
    "TIEventID": 0,
    "Repository": [
      "VTRS",
      "VTRS",
      "VTRS"
    ],
    "TransactionSummary": [
      {
        "R01TransactionId": 0,
        "IdentifiedTransaction": "true",
        "TransactionType": "String",
        "TransactionTypeShortDescription": "String",
        "TransactionDateTime": "2017-01-01T09:30:47Z",
        "CPD": "2017-01-01",
        "BusinessAppId": "aa",
        "BusinessAppIdDesc": "String",
        "SourceOfFunds": "a",
        "TransactionAmount": {
          "@currency": "000",
          "$": 0
        },
        "TranAmountDesc": "String",
        "MerchantName": "a",
        "MerchantCity": "a",
        "MerchantCountrySubEntityCode": "a",
        "MerchantCountryCode": "AA",
        "PaymentFacilitatorId": "1234",
        "SubMerchantId": "1123",
        "MerchantAddress": [
          {
            "Line1": "a",
            "Line2": "a",
            "City": "a",
            "State": "a",
            "Zip": "a"
          }
        ]
      }
    ]
  }
}
```

```
"PosEntryMode": "String",
"PosEntryModeDescription": "String",
"MoToEciCode": "String",
"MoToEciDescription": "String",
"NetworkID": "aaa",
"AuthorizationResponseCode": "String",
"AuthorizationResponseDescription": "String",
"AuthorizationCode": "String",
" MCC": "0000",
" MCCDescription": "String",
" TransactionID": "String",
" SpecialMerchantID": "1111",
" ParceladoPaymentType": "String",
" RetrievalReferenceNumber": "a",
" ARN": "00000000000000000000000000",
" CardNumber": "000000000000",
" Token": "String",
" CVV2Result": "String",
" CVV2ResultDesc": "String",
" PurchaseInquiryEligibleInd": "true",
" RdrEligibilityInd": "true",
" AssociatedTransGroupId": 0
}
{
"RolTransactionId": 0,
...
},
{
"RolTransactionId": 0,
...
}
]
```

## Get Transaction Details Using One or More VisaNet Transaction IDs

### Overview

The SIGetAllTransDetailsRequest/Response Web service is optional and **implementation is not required**. It is an alternative service apart from the required TI Web services described in section. This Web service provides an alternative method to retrieve complete Transaction Details for transactions using 1 to 10 VisaNet Transaction IDs as the search parameters.

### Why use this Web service?

When the Member systems already have the Transaction IDs for the transactions of interest, this Web service may offer a more efficient method for the Member system to perform the following actions:

1. Get **complete** Transaction Details without first getting Transaction Summaries using Transaction Inquiry Request,
2. Get Transaction Details using ROL Transaction ID as a reference, and
3. Get Associated Transactions in another separate request.

In other words, one Web service request retrieves the complete Transaction Details.

**Complete** Transaction Details includes the Transaction Details at a minimum. It also includes More Merchant Information (MMI), Associated Transactions Group (ATR), and Visa Cardholder Purchase Inquiry (VCPI) information if requested for inclusion in the response.

As an alternative to using VisaNet Transaction IDs in the request, ROL Transaction IDs may be used instead.

Normally, this Web service executes synchronously. However, if additional search time is required by VROL, it may execute asynchronously by returning a search reference (EventID). Subsequently, the reference is used by the same Web service request to retrieve the complete Transaction Details.

## Procedure

The detailed steps are as follows:

### Synchronous

1. The member host system triggers a SIGetAllTransDetailsRequest message supplying one or more TransactionIDs in a single request.

**Note:**

- You can include up to 10 TransactionIDs in the request.
- ROLTransactionId can be used instead of TransactionID.
- PlInd, ATRInd, and MMInd attributes are included in the request.

2. VROL returns a SIGetAllTransDetailsResponse message with all the transaction details for each transaction, including the list of VCPI information, ATRs, MMI, and additional merchant information.

The response includes a Transaction Summary for every transaction that matched one of the given Transaction ID(s).

### Asynchronous

1. The member host system triggers a SIGetAllTransDetailsRequest message supplying one or more TransactionIDs in a single request.
2. If VROL determines that response is asynchronous, then VROL sends a TIEventID in the response.
3. Subsequently, the member host system triggers a SIGetAllTransDetailsRequest message supplying the TIEventID to retrieve all the transaction details for each transaction, including the list of VCPI information, ATRs, MMI, and additional merchant information.

**Note:** If VROL does not yet have the results, it returns the I-300300000 message.

## Key Elements and Sample Code for SIGetAllTransDetailsRequest

The following XML elements should be considered when submitting a transaction details request using one or more TransactionIDs, or retrieving a response.

### SIGetAllTransDetailsRequest

Element	Required	Description
RolTransactionId	Yes	<p>Uniquely identifies a transaction. Member can use this to access the TI transaction details.</p> <p><b>Note:</b> Either RolTransactionId or TransactionID is required in the request. If RolTransactionId is supplied in the request, then it is not necessary to supply the TransactionID.</p>
TransactionID	Yes	<p>If a Multiple Clearing Sequence Number (MCSN) exists for the transaction, the two-digit MCSN can be added to the end of the transaction ID in the request and will be returned if available in the response. A transaction ID is typically 15 digits; however, the transaction ID will be 17 digits with an additional MCSN.</p> <p><b>Note:</b> Either RolTransactionId or TransactionID is required in the request. If TransactionID is supplied in the request, then it is not necessary to supply the RolTransactionId.</p>
ATRInd attribute	No	<p>If the value of this attribute in the request is "true", it indicates that ATR details are requested along with TI Details. If "false", it indicates that ATR details are not requested along with TI Details. If no value is provided, then its value defaults to "false".</p> <p><b>Note:</b> This attribute is used with RolTransactionId or TransactionID.</p>
PIIInd attribute	No	<p>If the value of this attribute in the request is "true", it indicates that PI details are requested along with TI Details. If "false", it indicates that PI details are not requested along with TI Details. If no value is provided, then its value defaults to "false".</p> <p><b>Note:</b> This attribute is used with RolTransactionId or TransactionID.</p>
MMIInd attribute	No	<p>If the value of this attribute in the request is "true", it indicates that MMI details are requested along with TI Details. If "false", it indicates that MMI details are not requested along with TI Details. If no value is provided, then its value defaults to "false".</p> <p><b>Note:</b> This attribute is used with RolTransactionId or TransactionID.</p>
TIEventID	Conditional	Unique internal ID returned in response. This element is required when retrieving the results of a request that went asynchronous.

### SIGetAllTransDetailsResponse

Element	Description
RolTransactionId	Uniquely identifies a transaction. Member can use this to access the TI transaction details.
PurchaselInquiryId	Unique ID that is associated with Purchase Inquiry.

Element	Description
TIEventID	Unique internal ID returned in response.  <b>Note:</b> This element is used for transactions on which the response became asynchronous.
AssociatedTransGroupId	Uniquely identifies a list of associated transactions.  Members can use this to retrieve a list of associated transactions or to associate a group of transactions to a case.

The SIGetAllTransDetails operation contains request and response messages as shown in the following sample messages.

### Request Message—SOAP-based Services

```
<SIGetAllTransDetailsRequest>
  <RequestHeader>...</RequestHeader>
  <requestData>
    <R0lTransactionId ATRInd="true" PIIInd="true"
      MMIInd="true">0</R0lTransactionId>
    <R0lTransactionId ATRInd="true" PIIInd="true"
      MMIInd="true">0</R0lTransactionId>
    <TransactionID ATRInd="true" PIIInd="true"
      MMIInd="true">String</TransactionID>
    <TransactionID ATRInd="true" PIIInd="true"
      MMIInd="true">String</TransactionID>
    <TIEventID>0</TIEventID>
  </requestData>
</SIGetAllTransDetailsRequest>
```

### Response Message—SOAP-based Services

```
<SIGetAllTransDetailsResponse>
  <Status>...</Status>
  <responseData>
    <TIEventID>0000000</TIEventID>
    <AllTransDetails>
      <Transaction>
        <R0lTransactionId>0</R0lTransactionId>
        <TransactionDetails>
          <Status>...</Status>
          <TransactionDetail>
            <R0lTransactionId>0000001</R0lTransactionId>
            <IdentifiedTransaction>true</IdentifiedTransaction>
            <TransactionType>String</TransactionType>
            <TransactionTypeShortDescription>String
            </TransactionTypeShortDescription>
TI details data (many fields - see IES)
            <Token>String</Token> ->
[for tokenized transactions only]
            <CVV2Result>String</CVV2Result>
            <CVV2ResultDesc>String</CVV2ResultDesc>
            <PurchaseInquiryEligibleInd>true
            </PurchaseInquiryEligibleInd>
            <RdrEligibilityInd>true</RdrEligibilityInd>
TI details data (many fields - see IES)
            <Authentication>
```

```
<InvalidRequestCode>String</InvalidRequestCode>
<InvalidRequestCodeDescription>String
</InvalidRequestCodeDescription>
<ProtocolVersion>String</ProtocolVersion>
<MerchantId>String</MerchantId>
<MerchantURL>a</MerchantURL>
...
<CavvAuthenticationResultsCode>String
</CavvAuthenticationResultsCode>
<CavvAuthenticationResultsCodeDescription>
String</CavvAuthenticationResultsCodeDescription>
<CavvSecondFactorAuthCode>String
</CavvSecondFactorAuthCode>
<CavvSecondFactorAuthCodeDescription>String
</CavvSecondFactorAuthCodeDescription>
<CavvKeyIndicator>String</CavvKeyIndicator>
...
</Authentication>
<IndFor3Dsecure>String</IndFor3Dsecure>
<IndFor3DsecureDescription>String
</IndFor3DsecureDescription>
<PinWasPresentCd>String</PinWasPresentCd>
TI details data (many fields - see IES)
<DisputeJurisdiction jurisdictionCd="REG">
    <Jurisdiction>String</Jurisdiction>
    <IssuerRegion>LAC</IssuerRegion>
    <AcquirerRegion>US</AcquirerRegion>
    <CountryCd>AA</CountryCd>
</DisputeJurisdiction>
TI details data (many fields - see IES)
</TransactionDetail>
<TransactionDetail>...</TransactionDetail>
</TransactionDetails>
<MMIDetails>
    <Status>...</Status>
    <MMIDetail>
        <MerchantAlternateName>String</MerchantAlternateName>
        <MerchantStreetAddress>String</MerchantStreetAddress>
        <MerchantCity>String</MerchantCity>
        <MerchantStateProvince>String</MerchantStateProvince>
        <MerchantZipPostalCode>String</MerchantZipPostalCode>
        <MerchantPhone>String</MerchantPhone>
    </MMIDetail>
    <MMIDetail>...</MMIDetail>
</MMIDetails>
<AssociatedTransactionList>
    <Status>...</Status>
    <AssociatedTransaction>
        <AssociatedTransGroupId>0</AssociatedTransGroupId>
        <RolTransactionId>0</RolTransactionId>
        <TransactionType>String</TransactionType>
        <TransactionAmount currency="000">0</TransactionAmount>
        <TranDate>2017-01-01</TranDate>
        <MerchantName>a</MerchantName>
        <TransactionID>a</TransactionID>
        <AuthCode>a</AuthCode>
        <ARN>00000000000000000000000000000000</ARN>
    </AssociatedTransaction>
</AssociatedTransactionList>
```

```
<RetrievalReferenceNumber>a</RetrievalReferenceNumber>
<MatchingPass>String</MatchingPass>
<MatchingScore>String</MatchingScore>
</AssociatedTransaction>
<AssociatedTransaction>...</AssociatedTransaction>
</AssociatedTransactionList>
<PurchaseInquiryInformationDetails>
<Status>...</Status>
<PurchaseInformation>
    <PurchaseInquiryInformationDetail>
        <PurchaseCategory>S</PurchaseCategory>
        <CardholderSignatureInd>true
        </CardholderSignatureInd>
        <EmailAccount>a</EmailAccount>
        <Customer>
            <FirstName>String</FirstName>
            <LastName>String</LastName>
        </Customer>
        <AuthenticationConducted>a
        </AuthenticationConducted>
        <DeliveryAddress>...</DeliveryAddress>
        <RapidDisputeResolution>
            <CreditIntent>true</CreditIntent>
        </RapidDisputeResolution>
        <DigitalReceipt>
            <OrderDateTime>2017-01-01T09:30:47Z
            </OrderDateTime>
            <OrderNumber>a</OrderNumber>
            <InvoiceNumber>a</InvoiceNumber>
            <TitlesIncludedInOrder>a
            </TitlesIncludedInOrder>
            <SubTotal currency="000">0</SubTotal>
            <ShippingAndHandlingCharge currency="000">0
            </ShippingAndHandlingCharge>
            <Tax currency="000">0</Tax>
            <TaxDescription>a</TaxDescription>
            <OrderTotal currency="000">0</OrderTotal>
            <OrderDetails>...</OrderDetails>
            <PaymentInformation>...</PaymentInformation>
        </DigitalReceipt>
        <DownloadDateTime>2017-01-01T09:30:47Z
        </DownloadDateTime>
        <PasswordLastUpdatedDateTime>2017-01-01T09:30:47Z
        </PasswordLastUpdatedDateTime>
        <RecipientCustomerName>a</RecipientCustomerName>
        <DeviceType>a</DeviceType>
        <TrackingNumber>a</TrackingNumber>
        <DeliveryService>a</DeliveryService>
        <DeliveryStatus>0</DeliveryStatus>
        <ShippingInfo>...</ShippingInfo>
        <CreditReimbursementInformationList>...
        </CreditReimbursementInformationList>
        <ContactUs>a</ContactUs>
        <WebsiteLink>a</WebsiteLink>
        <PackingSlip>a</PackingSlip>
        <ConditionofGoods>a</ConditionofGoods>
    ...

```

```
</PurchaseInquiryInformationDetail>
<PurchaseInquiryId>0</PurchaseInquiryId>
<PurchaseInquiryResponseStatus>String
</PurchaseInquiryResponseStatus>
</PurchaseInformation>
<PurchaseInformation>...</PurchaseInformation>
</PurchaseInquiryInformationDetails>
</Transaction>
</AllTransDetails>
</ResponseData>
</SIGetAllTransDetailsResponse>
```

### Request Message—REST-based Services

```
{
  "RequestHeader": {
    "User": {
      "@id": "a",
      "@type": "loginId"
    },
    "MemberRole": "I"
  },
  "requestData": {
    "R0lTransactionId": {
      "@ATRInd": "true",
      "@PIInd": "true",
      "@MMIInd": "true",
      "$": 0
    },
    "TransactionID": {
      "@ATRInd": "true",
      "@PIInd": "true",
      "@MMIInd": "true",
      "$": "String"
    },
    "TIEventID": 0
  }
}
```

### Response Message—REST-based Services

```
{
  "Status": {
    "Code": "a",
    "Message": "a",
    "ErrorIn": "String"
  },
  "responseData": {
    "TIEventID": 0000000,
    "AllTransDetails": {
      "Transaction": {
        "R0lTransactionId": 0,
        "TransactionDetails": {
          "Status": "...",
          "TransactionDetail": [
            {
              "R0lTransactionId": 0000001,
              "Status": "...",
              "TransactionDetail": [
                {
                  "R0lTransactionId": 0000002,
                  "Status": "...",
                  "TransactionDetail": [
                    {
                      "R0lTransactionId": 0000003,
                      "Status": "...",
                      "TransactionDetail": [
                        {
                          "R0lTransactionId": 0000004,
                          "Status": "...",
                          "TransactionDetail": [
                            {
                              "R0lTransactionId": 0000005,
                              "Status": "...",
                              "TransactionDetail": [
                                {
                                  "R0lTransactionId": 0000006,
                                  "Status": "...",
                                  "TransactionDetail": [
                                    {
                                      "R0lTransactionId": 0000007,
                                      "Status": "...",
                                      "TransactionDetail": [
  {
  "R0lTransactionId": 0000008,
  "Status": "...",
  "TransactionDetail": [
  {
  "R0lTransactionId": 0000009,
  "Status": "...",
  "TransactionDetail": [
  {
  "R0lTransactionId": 0000010,
  "Status": "...",
  "TransactionDetail": [
  {
  "R0lTransactionId": 0000011,
  "Status": "...",
  "TransactionDetail": [
  {
  "R0lTransactionId": 0000012,
  "Status": "...",
  "TransactionDetail": [
  {
  "R0lTransactionId": 0000013,
  "Status": "...",
  "TransactionDetail": [
  {
  "R0lTransactionId": 0000014,
  "Status": "...",
  "TransactionDetail": [
  {
  "R0lTransactionId": 0000015,
  "Status": "...",
  "TransactionDetail": [
  {
  "R0lTransactionId": 0000016,
  "Status": "...",
  "TransactionDetail": [
  {
  "R0lTransactionId": 0000017,
  "Status": "...",
  "TransactionDetail": [
  {
  "R0lTransactionId": 0000018,
  "Status": "...",
  "TransactionDetail": [
  {
  "R0lTransactionId": 0000019,
  "Status": "...",
  "TransactionDetail": [
  {
  "R0lTransactionId": 0000020,
  "Status": "...",
  "TransactionDetail": [
  {
  "R0lTransactionId": 0000021,
  "Status": "...",
  "TransactionDetail": [
  {
   "R0lTransactionId": 0000022,
   "Status": "...",
   "TransactionDetail": [
   {
   "R0lTransactionId": 0000023,
   "Status": "...",
   "TransactionDetail": [
   {
   "R0lTransactionId": 0000024,
   "Status": "...",
   "TransactionDetail": [
   {
   "R0lTransactionId": 0000025,
   "Status": "...",
   "TransactionDetail": [
   {
   "R0lTransactionId": 0000026,
   "Status": "...",
   "TransactionDetail": [
   {
   "R0lTransactionId": 0000027,
   "Status": "...",
   "TransactionDetail": [
   {
   "R0lTransactionId": 0000028,
   
```

```

    "IdentifiedTransaction": "true",
    "TransactionType": "String",
    "TransactionTypeShortDescription": "String",
TI details data (many fields - see IES)
    "Token": "String", [for tokenized transactions only]
    "CVV2Result": "String",
    "CVV2ResultDesc": "String",
    "PurchaseInquiryEligibleInd": "true",
    "RdrEligibilityInd": "true",
TI details data (many fields - see IES)
    "Authentication": {
        "InvalidRequestCode": "String",
        "InvalidRequestCodeDescription": "String",
        "ProtocolVersion": "String",
        "MerchantId": "String",
        "MerchantURL": "a",
        ...
        "CavvAuthenticationResultsCode": "String",
        "CavvAuthenticationResultsCodeDescription": "String",
        "CavvSecondFactorAuthCode": "String",
        "CavvSecondFactorAuthCodeDescription": "String",
        "CavvKeyIndicator": "String",
        ...
    },
    "IndFor3Dsecure": "String",
    "IndFor3DsecureDescription": "String",
    "PinWasPresentCd": "String",
TI details data (many fields - see IES)
    "DisputeJurisdiction": {
        "@jurisdictionCd": "REG",
        "Jurisdiction": "String",
        "IssuerRegion": "LAC",
        "AcquirerRegion": "US",
        "CountryCd": "AA"
    },
TI details data (many fields - see IES)
},
{
...
}
],
},
"MMIDetails": {
    "Status": "...",
    "MMIDetail": [
        {
            "MerchantAlternateName": "String",
            "MerchantStreetAddress": "String",
            "MerchantCity": "String",
            "MerchantStateProvince": "String",
            "MerchantZipPostalCode": "String",
            "MerchantPhone": "String"
        },
        {
            ...
        }
    ]
}
]

```

```
},
"AssociatedTransactionList": {
  "Status": "...",
  "AssociatedTransaction": [
    {
      "AssociatedTransGroupId": 0,
      "RolTransactionId": 0,
      "TransactionType": "String",
      "TransactionAmount": {
        "@currency": "000",
        "$": 0
      },
      "TranDate": "2017-01-01",
      "MerchantName": "a",
      "TransactionID": "a",
      "AuthCode": "a",
      "ARN": "00000000000000000000000000000000",
      "RetrievalReferenceNumber": "a",
      "MatchingPass": "String",
      "MatchingScore": "String"
    },
    {
      ...
    }
  ]
},
"PurchaseInquiryInformationDetails": {
  "Status": "...",
  "PurchaseInformation": [
    {
      "PurchaseInquiryInformationDetail": {
        "PurchaseCategory": "S",
        "CardholderSignatureInd": "true",
        "EmailAccount": "a",
        "Customer": {
          "FirstName": "test",
          "LastName": "test"
        },
        "AuthenticationConducted": "a",
        "DeliveryAddress": "...",
        "RapidDisputeResolution": {
          "CreditIntent": "true"
        },
        ...
      },
      "DigitalReceipt": {
        "OrderDateTime": "2017-01-01T09:30:47Z",
        "OrderNumber": "a",
        "InvoiceNumber": "a",
        "TitlesIncludedInOrder": "a",
        "SubTotal": {
          "@currency": "000",
          "$": 0
        },
        "ShippingAndHandlingCharge": {
          "@currency": "000",
          "$": 0
        }
      }
    }
  ]
}
```

```
        },
        "Tax": {
            "@currency": "000",
            "$": 0
        },
        "TaxDescription": "a",
        "OrderTotal": {
            "@currency": "000",
            "$": 0
        },
        "OrderDetails": {
            ...
        },
        "PaymentInformation": {
            ...
        }
    },
    "DownloadDateTime": "2017-01-01T09:30:47Z",
    "PasswordLastUpdatedDateTime": "2017-01-01T09:30:47Z",
    "RecipientCustomerName": "a",
    "DeviceType": "a",
    "TrackingNumber": "a",
    "DeliveryService": "a",
    "DeliveryStatus": "O",
    "ShippingInfo": {
        ...
    },
    "CreditReimbursementInformationList": {
        ...
    },
    "ContactUs": "a",
    "WebsiteLink": "a",
    "PackingSlip": "a",
    "ConditionofGoods": "a",
    ...
},
    "PurchaseInquiryId": 0,
    "PurchaseInquiryResponseStatus": "String"
},
{
    "PurchaseInquiryInformationDetail": {
        "..."
    }
}
}
```

## Visa Cardholder Purchase Inquiry

### Introduction

This section describes the Visa Cardholder Purchase Inquiry (VCPI) web service.

The problem of Cardholder's not positively recognizing their transactions on their statements or online view of transactions affects card issuing banks with queries to their call centers, initiation of invalid dispute inquiries or groundless chargebacks. It affects acquirers and merchants managing groundless chargebacks because they need to invest in researching and refuting the chargeback with gathering evidence and presenting to the issuing bank.

The result of the cardholder not recognizing valid transaction is added resource and exception transaction fee costs to the issuing bank, acquiring bank, and merchant as well as cardholder dissatisfaction.

The root cause of the Cardholder not recognizing their transaction is not having sufficient, relevant, and timely additional transaction data. The VCPI Web service addresses this root cause and has the following benefits:

- Provides additional information about the merchant including alternative names, type of business conducted, merchant contact information, and merchant Web site address.
- Provides additional information about the type of transaction.
- Includes receipt and invoice-type information that is directly from the merchant.

Cardholders wishing to identify and recognize transactions may require enhanced information about a specific transaction. This enhanced information is over and above what appears on their statements or online transaction view. On behalf of the Cardholder, issuing banks can search Visa's transaction repository using basic transaction parameters for enhanced information about specific transactions. At that point, Cardholders can positively identify transactions. The enhanced data needs to be just enough information and presented in a clear intelligible way for the Cardholder.

Cardholders access Visa's enhanced transaction data using their issuing bank's online banking application as an intermediary to the Visa Cardholder Purchase Inquiry (VCPI) Web service.

VROL aggregates transaction data from several sources and curates it to return just the right information in the response. Unlike existing VROL Web services, to get the information is a single request and the resulting information has only the data useful to a Cardholder.

VROL uses Visa's Global Merchant Repository (GMR) to retrieve merchant demographic data, even if the merchant isn't participating in VCPI. This increases the response "match" rate significantly, and allows issuers to at least access the merchant's full name or address where coverage is available in GMR.

**Note:** *Transaction inquiry data retrieved from the database is deleted after 9-10 days.*

### Visa Cardholder Purchase Inquiry (VCPI) Request and Response

This is an enhancement to the existing Visa Merchant Purchase Inquiry functionality. This service delivers high-level merchant data from a Visa repository. The Visa Merchant Purchase Inquiry data received directly from participating merchants will still be available.

The **request** includes the following search parameters:

**Table H-1: VCPI Search Parameters**

Search Parameter	Mandatory (M), Conditional (C), or Optional (O)
CardNumber	M - primary
TransactionAmount <sup>1</sup>	M - primary
TransactionDateTime <sup>1</sup>	M - primary
TransactionType	M - primary
TransactionID <sup>2</sup>	C - primary
AuthCode <sup>2</sup>	C - primary
ARN <sup>2</sup>	C - primary
RetrievalReferenceNumber	C - primary
Token	O - secondary
AuthorizationDate	O - secondary
AuthorizationTime	
SystemTraceAuditNumber	O - secondary
CardAcceptorID	O - secondary
MCSN	O - secondary
PurchaserIdentifier	O - secondary
ExpirationDate	O - secondary

<sup>1</sup> Apply the same search procedure and parameters as when doing a Creating Transaction Inquiry.

<sup>2</sup> At least 2 of these 3 fields must have values

1. To request additional merchant data, the member host system triggers a SICardholderPurchaseInquiryRequest message supplying the fields as specified in [Table H-1](#).
2. VROL returns a SICardholderPurchaseInquiryResponse message with success status.  
The response includes the details of the transaction, including additional merchant information and VCPI response information.

## Visa Cardholder Purchase Inquiry (VCPI) Unsuccessful Response

As part of responding to a VCPI request, if a unique matching transaction cannot be found on Visa's systems, where either no transaction could be found or no unique transaction could be found in the Visa repository, then VROL returns an unsuccessful outcome of "transaction not found".

## VCPI Operations

**SICardholderPurchaseInquiryRequest**—Used to request additional merchant information regarding a transaction.

**SIGetCardholderPurchaseInquiryRequest**—Used to retrieve details of additional merchant information regarding a transaction.

## SICardholderPurchaseInquiryRequest

SICardholderPurchaseInquiryRequest enables users to request additional merchant information regarding a transaction by supplying the CardNumber.

The following elements should be considered when submitting a request or receiving a response from this service.

Element	Required	Description
CardNumber	Yes	Used for the card account number. It can also be used for 8-digit BIN or extended BIN searches.
TransactionAmount	Yes	Indicates the amount of the transaction.
TransactionDateTime	Yes	Indicates the date and time of the transaction.
TransactionID	Conditional (see Description)	If a Multiple Clearing Sequence Number (MCSN) exists for the transaction, the two-digit MCSN can be added to the end of the transaction ID in the request and will be returned if available in the response. A transaction ID is typically 15 digits; however, the transaction ID will be 17 digits with an additional MCSN.  This field along with either <ARN> or <AuthCode> are required in the request. Additionally, the following required fields should also be provided in the request: <ul style="list-style-type: none"><li>• &lt;CardNumber&gt;</li><li>• &lt;TransactionAmount&gt;</li><li>• &lt;TransactionDateTime&gt;</li></ul>
TransactionType	Yes	Indicates the type of transaction.
ARN	Conditional (see Description)	This field along with either <TransactionID> or <AuthCode> are required in the request. Additionally, the following required fields should also be provided in the request: <ul style="list-style-type: none"><li>• &lt;CardNumber&gt;</li><li>• &lt;TransactionAmount&gt;</li><li>• &lt;TransactionDateTime&gt;</li></ul>
AuthCode	Conditional (see Description)	This field along with either <TransactionID> or <ARN> are required in the request. Additionally, the following required fields should also be provided in the request: <ul style="list-style-type: none"><li>• &lt;CardNumber&gt;</li><li>• &lt;TransactionAmount&gt;</li><li>• &lt;TransactionDateTime&gt;</li></ul>

The SICardholderPurchaseInquiry operation contains request and response messages as shown in the following sample messages.

**Request Message—SOAP-based Services**

```
<SICardholderPurchaseInquiryRequest>
  <RequestHeader>
    <User id="a" type="loginId"/>
    <MemberRole>I</MemberRole>
  </RequestHeader>
  <requestData>
    <CardNumber>1000000000</CardNumber>
    <TransactionAmount currency="000">0</TransactionAmount>
    <TransactionDateTime>2001-12-17T09:30:47Z</TransactionDateTime>
    <TransactionID>a</TransactionID>
    <AuthCode>aaaaaa</AuthCode>
    <ARN>00000000000000000000000000000000</ARN>
    <Token>0000000000000000</Token>
    <AuthorizationDate>2017-08-13</AuthorizationDate>
    <AuthorizationTime>String</AuthorizationTime>
    <RetrievalReferenceNumber>a</RetrievalReferenceNumber>
    <SystemTraceAuditNumber>a</SystemTraceAuditNumber>
    <CardAcceptorId>a</CardAcceptorId>
    <PurchaseIdentifier>String</PurchaseIdentifier>
    <TransactionType>String</TransactionType>
    <ExpirationDate>2017-08-13</ExpirationDate>
  </requestData>
</SICardholderPurchaseInquiryRequest>
```

**Response Message—SOAP-based Services**

```
<SICardholderPurchaseInquiryResponse>
  <Status>
    <Code>a</Code>
    <Message>a</Message>
    <ErrorIn>String</ErrorIn>
  </Status>
  <responseData>
    <TransactionInformationDetails>
      <Status>
        <Code>a</Code>
        <Message>a</Message>
        <ErrorIn>String</ErrorIn>
      </Status>
      <TransactionInformation>
        <CardholderName>a</CardholderName>
        <CPD>2017-08-13</CPD>
        <MerchantName>a</MerchantName>
        <ProcessingCode>aa</ProcessingCode>
        <ProcessingCodeDescription>String
        </ProcessingCodeDescription>
        <ECMOTO>a</ECMOTO>
        <ECMOTODescription>String</ECMOTODescription>
        <EntryMode>String</EntryMode>
        <EntryModeDesc>String</EntryModeDesc>
        <SurchargeFeeAmount currency="000">0</SurchargeFeeAmount>
        <LocalCashBackAmount currency="000">0</LocalCashBackAmount>
        <MerchantGroupDescription>String</MerchantGroupDescription>
        <MerchantCategoryGroup>String</MerchantCategoryGroup>
        <MCC>0000</MCC>
      </TransactionInformation>
    </TransactionInformationDetails>
  </responseData>
</SICardholderPurchaseInquiryResponse>
```

```
<MCCDescription>String</MCCDescription>
<CardNumber>1000000000</CardNumber>
<TransactionDateTime>2001-12-17T09:30:47Z
</TransactionDateTime>
<TransactionID>a</TransactionID>
<TransactionType>String</TransactionType>
<VcpiTransactionId>0</VcpiTransactionId>
<TransactionAmount currency="000">0</TransactionAmount>
</TransactionInformation>
</TransactionInformationDetails>
<PurchaseInquiryInformationDetails>
<Status>
<Code>a</Code>
<Message>a</Message>
<ErrorIn>String</ErrorIn>
</Status>
<PurchaseInquiryInformation>
<PurchaseCategory>M</PurchaseCategory>
<CardholderSignatureInd>true</CardholderSignatureInd>
<EmailAccount>a</EmailAccount>
<Customer>
<FirstName>String</FirstName>
<LastName>String</LastName>
</Customer>
<AuthenticationConducted>a</AuthenticationConducted>
<DeliveryAddress>...</DeliveryAddress>
<RapidDisputeResolution>
<CreditIntent>true</CreditIntent>
</RapidDisputeResolution>
<DigitalReceipt>...</DigitalReceipt>
<DownloadDateTime>2001-12-17T09:30:47Z</DownloadDateTime>
<PasswordLastUpdatedDateTime>2001-12-17T09:30:47Z
</PasswordLastUpdatedDateTime>
<RecipientCustomerName>a</RecipientCustomerName>
<DeviceType>a</DeviceType>
<TrackingNumber>a</TrackingNumber>
<DeliveryService>a</DeliveryService>
<DeliveryStatus>IT</DeliveryStatus>
<ShippingInfo>...</ShippingInfo>
<CreditReimbursementInformationList>...
</CreditReimbursementInformationList>
<ContactUs>a</ContactUs>
<WebsiteLink>a</WebsiteLink>
<PackingSlip>a</PackingSlip>
<ConditionofGoods>a</ConditionofGoods>
<Address>...</Address>
...
</PurchaseInquiryInformation>
</PurchaseInquiryInformationDetails>
</ResponseData>
</SICardholderPurchaseInquiryResponse>
```

**Request Message—REST-based Services**

```
{
  "RequestHeader": {
    "User": {
      "@id": "a",
      "@type": "loginId"
    },
    "MemberRole": "I"
  },
  "requestData": {
    "CardNumber": 1000000000,
    "TransactionAmount": {
      "@currency": "000",
      "$": 0
    },
    "TransactionDateTime": "2001-12-17T09:30:47Z",
    "TransactionID": "a",
    "AuthCode": "aaaaaa",
    "ARN": "00000000000000000000000000000000",
    "Token": "0000000000000000",
    "AuthorizationDate": "2016-08-13",
    "AuthorizationTime": "String",
    "RetrievalReferenceNumber": "a",
    "SystemTraceAuditNumber": "a",
    "CardAcceptorId": "a",
    "PurchaseIdentifier": "String",
    "TransactionType": "String",
    "ExpirationDate": "2017-08-13"
  }
}
```

**Response Message—REST-based Services**

```
{
  "Status": [
    {
      "Code": "a",
      "Message": "a",
      "ErrorIn": "String"
    }
  ],
  "responseData": {
    "TransactionInformationDetails": {
      "Status": [
        {
          "Code": "a",
          "Message": "a",
          "ErrorIn": "String"
        }
      ]
    },
    "TransactionInformation": {
      "CardholderName": "a",
      "CPD": "2017-08-13",
      "MerchantName": "a",
      "ProcessingCode": "aa",
      "ProcessingCodeDescription": "String",
    }
  }
}
```

```
"ECMOTO": "a",
"ECMOTODescription": "String",
"EntryMode": "String",
"EntryModeDesc": "String",
"SurchargeFeeAmount":{
  "@currency": "000",
  "$": 0
},
"LocalCashBackAmount":{
  "@currency": "000",
  "$": 0
},
"MerchantGroupDescription": "String",
"MerchantCategoryGroup": "String",
"MCC": "0000",
"MCCDescription": "String",
"CardNumber": 1000000000,
"TransactionDateTime": "2001-12-17T09:30:47Z",
"TransactionID": "a",
"TransactionType": "String",
"VcpITransactionId": 0
"TransactionAmount":{
  "@currency": "000",
  "$": 0
}
},
},
"PurchaseInquiryInformationDetails":{
>Status": [
{
  "Code": "a",
  "Message": "a",
  "ErrorIn": "String"
},
],
"PurchaseInquiryInformation":{
"PurchaseCategory": "M",
"CardholderSignatureInd": "true",
"EmailAccount": "a",
"Customer":{
  "FirstName": "test",
  "LastName": "test"
},
"AuthenticationConducted": "a",
"DeliveryAddress": "...",
"RapidDisputeResolution":{
  "CreditIntent": "true"
},
"DigitalReceipt": "...",
"DownloadDateTime": "2001-12-17T09:30:47Z",
"PasswordLastUpdatedDateTime": "2001-12-17T09:30:47Z",
"RecipientCustomerName": "a",
"DeviceType": "a",
"TrackingNumber": "a",
"DeliveryService": "a",
"DeliveryStatus": "C",
"ShippingInfo": "..."
}
```

```
"CreditReimbursementInformationList":{  
    "CreditReimbursementInformation": "..."  
},  
"ContactUs": "a",  
"WebsiteLink": "a",  
"PackingSlip": "a",  
"ConditionofGoods": "a",  
"Address": "..."  
.  
.  
.  
}  
}  
}  
}
```

## SIGetCardholderPurchaseInquiryRequest

**SIGetCardholderPurchaseInquiryRequest** allows RTSI users to retrieve additional merchant information regarding a transaction.

The following elements should be considered when submitting a request or receiving a response from this service.

<b>Element</b>	<b>Required</b>	<b>Description</b>
VcpITransactionId	Yes	Uniquely identifies a transaction. Member can use this to access the VCPI response.

The SIGetCardholderPurchaseInquiry operation contains request and response messages as shown in the following sample messages.

## Request Message—SOAP-based Services

```
<SIGetCardholderPurchaseInquiryRequest>
    <RequestHeader>
        <User id="a" type="internalId"/>
        <MemberRole>I</MemberRole>
    </RequestHeader>
    <requestData>
        <VcpTransactioId>0</VcpTransactioId>
    </requestData>
</SIGetCardholderPurchaseInquiryRequest>
```

## Response Message—SOAP-based Services

```
<SIGetCardholderPurchaseInquiryResponse>
    <Status>
        <Code>a</Code>
        <Message>a</Message>
        <ErrorIn>String</ErrorIn>
    </Status>
    <ResponseData>
        <TransactionInformationDetails>
            <Status>
                <Code>a</Code>
                <Message>a</Message>
                <ErrorIn>String</ErrorIn>
            </Status>
        </TransactionInformationDetails>
    </ResponseData>
</SIGetCardholderPurchaseInquiryResponse>
```

```
<TransactionInformation>
    <CardholderName>a</CardholderName>
    <CPD>2017-08-13</CPD>
    <MerchantName>a</MerchantName>
    <ProcessingCode>aa</ProcessingCode>
    <ProcessingCodeDescription>String
    </ProcessingCodeDescription>
    <ECMOTO>a</ECMOTO>
    <ECMOTODescription>String</ECMOTODescription>
    <EntryMode>String</EntryMode>
    <EntryModeDesc>String</EntryModeDesc>
    <SurchargeFeeAmount currency="000">0</SurchargeFeeAmount>
    <LocalCashBackAmount currency="000">0</LocalCashBackAmount>
    <MerchantGroupDescription>String</MerchantGroupDescription>
    <MerchantCategoryGroup>String</MerchantCategoryGroup>
    <MCC>0000</MCC>
    <MCDDescription>String</MCDDescription>
    <CardNumber>1000000000</CardNumber>
    <TransactionDateTime>2001-12-17T09:30:47Z
    </TransactionDateTime>
    <TransactionID>a</TransactionID>
    <TerminalId>a</TerminalId>
    <TransactionType>String</TransactionType>
    <Vcp0</Vcp
        <TransactionAmount currency="000">0</TransactionAmount>
    </TransactionInformation>
</TransactionInformationDetails>
<PurchaseInquiryInformationDetails>
    <Status>
        <Code>a</Code>
        <Message>a</Message>
        <ErrorIn>String</ErrorIn>
    </Status>
    <PurchaseInquiryInformation>
        <PurchaseCategory>S</PurchaseCategory>
        <CardholderSignatureInd>true</CardholderSignatureInd>
        <EmailAccount>a</EmailAccount>
        <Customer>
            <FirstName>String</FirstName>
            <LastName>String</LastName>
        </Customer>
        <AuthenticationConducted>a</AuthenticationConducted>
        <DeliveryAddress>...</DeliveryAddress>
        <RapidDisputeResolution>
            <CreditIntent>true</CreditIntent>
        </RapidDisputeResolution>
        <DigitalReceipt>...</DigitalReceipt>
        <DownloadDateTime>2001-12-17T09:30:47Z
        </DownloadDateTime>
        <>PasswordLastUpdatedDateTime>2001-12-17T09:30:47Z
        </PasswordLastUpdatedDateTime>
        <RecipientCustomerName>a</RecipientCustomerName>
        <DeviceType>a</DeviceType>
        <TrackingNumber>a</TrackingNumber>
        <DeliveryService>a</DeliveryService>
        <DeliveryStatus>C</DeliveryStatus>
        <ShippingInfo>...</ShippingInfo>
```

```

<CreditReimbursementInformationList>
  ...
</CreditReimbursementInformationList>
<ContactUs>a</ContactUs>
<WebsiteLink>a</WebsiteLink>
<PackingSlip>a</PackingSlip>
<ConditionofGoods>a</ConditionofGoods>
<Address>...</Address>
  ...
</PurchaseInquiryInformation>
</PurchaseInquiryInformationDetails>
</ResponseData>
</SIGetCardholderPurchaseInquiryResponse>

```

### Request Message—REST-based Services

```

{
  "RequestHeader": {
    "User": {
      "@id": "a",
      "@type": "internalId"
    },
    "MemberRole": "I"
  },
  "requestData": {
    "VcpITransactionId": 0
  }
}

```

### Response Message—REST-based Services

```

{
  "Status": {
    "Code": "a",
    "Message": "a",
    "ErrorIn": "String"
  }
  "responseData": {
    "TransactionInformationDetails": {
      "Status": {
        "Code": "a",
        "Message": "a",
        "ErrorIn": "String"
      }
      "TransactionInformation": {
        "CardholderName": "a",
        "CPD": "2017-08-13",
        "MerchantName": "a",
        "ProcessingCode": "aa",
        "ProcessingCodeDescription": "String",
        "ECMOTO": "a",
        "ECMOTODescription": "String",
        "EntryMode": "String",
        "EntryModeDesc": "String",
        "SurchargeFeeAmount": {
          "@currency": "000",
          "$": 0
        }
      }
    }
  }
}

```

```
        },
        "LocalCashBackAmount": {
            "@currency": "000",
            "$": 0
        },
        "MerchantGroupDescription": "String",
        "MerchantCategoryGroup": "String",
        "MCC": "0000",
        "MCCDescription": "String",
        "CardNumber": 1000000000,
        "TransactionDateTime": "2001-12-17T09:30:47Z",
        "TransactionID": "a",
        "TerminalId": "a",
        "TransactionType": "String",
        "VcpTransactioId": 0
        "TransactionAmount": {
            "@currency": "000",
            "$": 0
        }
    }
},
"PurchaseInquiryInformationDetails": {
    "Status": {
        "Code": "a",
        "Message": "a",
        "ErrorIn": "String"
    }
    "PurchaseInquiryInformation": {
        "PurchaseCategory": "S",
        "CardholderSignatureInd": "true",
        "EmailAccount": "a",
        "Customer": {
            "FirstName": "test",
            "LastName": "test"
        },
        "AuthenticationConducted": "a",
        "DeliveryAddress": "...",
        "RapidDisputeResolution": {
            "CreditIntent": "true"
        },
        "DigitalReceipt": "...",
        "DownloadDateTime": "2001-12-17T09:30:47Z",
        "PasswordLastUpdatedDateTime": "2001-12-17T09:30:47Z",
        "RecipientCustomerName": "a",
        "DeviceType": "a",
        "TrackingNumber": "a",
        "DeliveryService": "a",
        "DeliveryStatus": "C",
        "ShippingInfo": "..."
    }
    "CreditReimbursementInformationList": {
        "CreditReimbursementInformation": "..."
    },
    "ContactUs": "a",
    "WebsiteLink": "a",
    "PackingSlip": "a",
    "ConditionofGoods": "a",
    "Address": "..."
}
```

```
    ...
    }
  },
}
}
```

# RTSI Implementation and Testing Checklist

I

## Overview

There are seven steps to implement Real Time System Interface (RTSI) with VROL, starting with an assessment if RTSI is the correct solution all the way through implementation and processing disputes real-time.

1. Assess if a system interface solution is appropriate.
2. Prepare and identify internal project teams and resources.
3. Complete the development of your host system.
4. Prepare for Visa testing and validation by reviewing all testing training and documentation.
5. Submit the System Interface Application Form.
6. Coordinate Visa configuration and testing.
7. Implement the new connection and train staff.

### **Important:**

Members must complete steps 1-5 prior to initiating a project with Visa, while a dedicated Visa project team supports the implementation in steps 6-7.

**Note:** This checklist can also be found on  
[https://secure.visaonline.com/\\_layouts/GVOL/GVOLStream.aspx?filename=%5Croot%5Content%5Cdam%5Cgvol%5Cen%5Creference-library%5Cnews-and-communications%5Cvbn-articles%5C2018\\_Q3%5CAI07795\\_RTSI\\_Implementation\\_Checklist.pdf](https://secure.visaonline.com/_layouts/GVOL/GVOLStream.aspx?filename=%5Croot%5Content%5Cdam%5Cgvol%5Cen%5Creference-library%5Cnews-and-communications%5Cvbn-articles%5C2018_Q3%5CAI07795_RTSI_Implementation_Checklist.pdf).

## Checklist

This section guides members through each step of the project.

### **Step 1: Assess if a System Interface Solution Is Appropriate**

#### **Key Considerations:**

Visa recommends that members with less than 5,000 disputes per month continue to use the Visa hosted User Interface solution. Members with more than 5,000 disputes per month should evaluate the need for integration against the cost of developing and maintaining a custom system interface solution.

System interface connections are custom solutions that require considerable upfront development, ongoing commitment of resources, and expertise in API development to interface with Visa Resolve Online. Additionally, there are VisaNet Business Enhancement Releases twice a year (<https://secure.visaonline.com/SitePages/Section.aspx?pageid=2.0.5.0.0>) as well as scheduled VROL maintenance windows that will require ongoing system maintenance of your host system.

Reference materials and additional resources are also available on:  
<https://secure.visaonline.com/SitePages/Content.aspx?pageid=2.1.1.4.0>.

See the *Visa Resolve Online Member Implementation Guide* to know more about the requirements for members and processors deploying VROL.

## Step 2: Prepare and Identify Internal Project Teams and Resources

Members need to make sure that they assign the appropriate personnel to participate in their RTSI implementation. Visa recommends the following member personnel to participate in all implementations meetings.

### **Member Project Team:**

- Project Manager—Coordinates communication, tracks overall project schedule, and identifies affected member organizations.
- Back-Office Operations—Assesses the effect on back-office processes, designs and implements workflow changes, and supports the member's dispute analysts.
- Development Staff—Designs and codes the member host system modifications in preparation for implementation of VROL. Development staff is required only if a member chooses to integrate with VROL using the systems interfaces.
- Implementation Manager—Coordinates communication, tracks overall project schedule, and identifies affected member organizations.
- Internal Testing Authorization and Settlement Teams—Assist with incoming files and authorizations delivered and processed so disputes can be submitted.
- Testers—Coordinate testing connectivity from the member site to VROL.
- Trainers—Lead training sessions if a member decides to conduct VROL training.

## Step 3: Development of Your Host System

Members must complete the detail design, development, and testing effort for the coding changes required in their environment to use RTSI. This design process can include tasks such as data mapping, defining matching algorithms, defining batch processes, and other system design activities.

### **Technical Specifications:**

- *Real Time System Interface Development Guide* available on Visa Resolve Online MTE2
- Technical Specifications (IES and Schema) available on Visa Resolve Online MTE2 or request through email to [VROLGPO@visa.com](mailto:VROLGPO@visa.com)

## Step 4: Prepare For Visa Testing and Validation by Reviewing All Testing Training and Documentation

Members must complete unit testing internally before installing the code in Visa's pre-production or test environment. At that point, members can choose to use the VROL MTE to verify that their applications integrate with VROL.

The testing capabilities supported in MTE are described in Chapter 5: Member Testing of the *Visa Resolve Online Member Implementation Guide*. Members can also refer to the VCR testing training and documentation available on <https://www.visabusinessschool.com/en/vcr2>.

Prior to testing, issuers and acquirers must perform the following actions:

### Issuers:

- Confirm settlement CIB ID and Authorization station ID.
- Members identify their internal settlement and authorization processing team.
- Create or identify test account numbers or BINs to be used during testing.
- Request access to VTS-CS.
- Issuers can test using only VTS-CS (settlement data) or authorizations with settlement and any combination thereof.

### Acquirers:

- Identify settlement CIB and Authorization station ID.
- ACQ submitting authorizations need to identify their VCMS station and ACQ BIN.
  - Required for full SMS members.
  - BASE II members need to identify their CIB and internal process to submit clearing data (TC05).
- Members identify their internal settlement and authorization processing teams.

Refer to the following documentation for additional information:

- *Real Time System Interface Development Guide* available on Visa Resolve Online MTE2 or request through email to [VROLGPO@visa.com](mailto:VROLGPO@visa.com)
- Technical Specifications (IES and Schema) available on Visa Resolve Online MTE2
- Edit Package & VTS-CS Setup Guide for Visa Claims Resolution ([https://secure.visaonline.com/\\_layouts/GVOL/GVOLStream.aspx?filename=%5Croot%5Ccontent%5Cdam%5Cgvol%5Cen%5Copera%5Cdispute-resolution%5Cresolve-online%5Cedit-package-and-vts-cs-setup-guide-for-vcr.pdf&contenttype=application/pdf](https://secure.visaonline.com/_layouts/GVOL/GVOLStream.aspx?filename=%5Croot%5Ccontent%5Cdam%5Cgvol%5Cen%5Copera%5Cdispute-resolution%5Cresolve-online%5Cedit-package-and-vts-cs-setup-guide-for-vcr.pdf&contenttype=application/pdf))
- *Visa Resolve Online Member Test Scenarios*
- *Visa Resolve Online User's Guide*
- *Visa Resolve Online Reference Manual*

## Step 5: Submit the System Interface Application Form

Complete and submit the System Interface Application Form ([https://secure.visaonline.com/\\_layouts/GVOL/GVOLStream.aspx?filename=%5Croot%5Ccontent%5Cdam%5Cgvol%5Cen%5Creferenc%5Cnews-and-communications%5Cvbn-articles%5C2018\\_Q3%5CAI07795\\_RTSI\\_Application\\_Form.docx&contenttype=application/docx](https://secure.visaonline.com/_layouts/GVOL/GVOLStream.aspx?filename=%5Croot%5Ccontent%5Cdam%5Cgvol%5Cen%5Creferenc%5Cnews-and-communications%5Cvbn-articles%5C2018_Q3%5CAI07795_RTSI_Application_Form.docx&contenttype=application/docx)) and the additional documentation referenced below to request a project.

A Visa Project Manager will get in touch with you to confirm project readiness. The kick-off meetings are scheduled from August 2018 onwards once Visa has confirmed the member's project readiness.

#### **Project Readiness Criteria:**

- Completed steps 1 to 5 outlined in [Overview](#).
- Dispute volume is more than 5,000 per month.
- Provide robust documentation to Visa, detailing member's host solution including technical specifications, application infrastructure, and architecture diagrams that define how the host system will interface with VROL.
- Provide a project team structure (see [Step 2: Prepare and Identify Internal Project Teams and Resources](#)) with identified contact details for all resources involved in the project.
- Review the results of member's self-testing activities.
- Provide a detailed test plan to Visa, including acceptance criteria.

### **Step 6: Coordinate Visa Configuration and Testing**

Through VROL, Visa offers the Member Test Environment (MTE) that connects to the VisaNet Certification Management Service (VCMS) environment to support testing of the full dispute life cycle. MTE and VCMS also provide tools to support member testing for generating initial transactions and automating the opposing member response.

The following table outlines the tasks that must be completed one month prior to commencement of VCR testing.

Item	Tasks	Reference
1	Obtain access to the MTE2 test environment	VBN 12 April 2018 (AI07684)  ( <a href="https://secure.visaonline.com/_layouts/15/GVOL/GVOLStream.aspx?filename=%5Croot%5Ccontent%5Cdam%5Cgvol%5Cen%5Creference-library%5Cnews-and-communications%5Cvbn-articles%5Carticles%5C2017%5C11-09%5CAI07184.pdf&amp;contenttype=application/pdf&amp;cid=28649">https://secure.visaonline.com/_layouts/15/GVOL/GVOLStream.aspx?filename=%5Croot%5Ccontent%5Cdam%5Cgvol%5Cen%5Creference-library%5Cnews-and-communications%5Cvbn-articles%5Carticles%5C2017%5C11-09%5CAI07184.pdf&amp;contenttype=application/pdf&amp;cid=28649</a> )
2	Complete and submit the Global Client Information Questionnaire (CIQ), WSI Certificate, and Global Client Testing Questionnaire	<ul style="list-style-type: none"> <li>• Global Client Information Questionnaire (CIQ) (<a href="https://secure.visaonline.com/SitePages/Section.aspx?pageid=2.4.1.0.0">https://secure.visaonline.com/SitePages/Section.aspx?pageid=2.4.1.0.0</a>)</li> <li>• WSI Certificate (request from VROL Helpdesk)</li> <li>• Global Client Testing Questionnaire (<a href="https://secure.visaonline.com/_layouts/GVOL/GVOLStream.aspx?filename=%5Croot%5Ccontent%5Cdam%5Cgvol%5Cen%5Coperaions%5Cmore-operations%5Cglobal-client-testing%5Cgct-questionnaire.pdf&amp;contenttype=application/pdf">https://secure.visaonline.com/_layouts/GVOL/GVOLStream.aspx?filename=%5Croot%5Ccontent%5Cdam%5Cgvol%5Cen%5Coperaions%5Cmore-operations%5Cglobal-client-testing%5Cgct-questionnaire.pdf&amp;contenttype=application/pdf</a>)</li> </ul>
3	Review testing documentation	Links provided in <a href="#">Step 4: Prepare For Visa Testing and Validation by Reviewing All Testing Training and Documentation</a>

Item	Tasks	Reference
4	Prepare edit package setup for testing <ol style="list-style-type: none"> <li>Ensure that the latest Edit Package software version is installed</li> <li>Update Edit Package tables by processing a VCMS Full File Table Replacement (FFR)</li> <li>Setup Test BINs/Change Run Mode = TEST</li> <li>Update run control options               <ul style="list-style-type: none"> <li>– Setup pass options</li> <li>– Setup separate incoming CTF for VCR</li> </ul> </li> </ol>	Edit Package & VTS-CS Setup Guide for Visa Claims Resolution ( <a href="https://secure.visaonline.com/_layouts/GVOL/GVOLStream.aspx?filename=%5Croot%5Ccontent%5Cdam%5Cgvol%5Cen%5Copera%5Cdispute-resolution%5Cresolve-online%5Cedit-package-and-vts-cs-setup-guide-for-vcr.pdf&amp;contenttype=application/pdf">https://secure.visaonline.com/_layouts/GVOL/GVOLStream.aspx?filename=%5Croot%5Ccontent%5Cdam%5Cgvol%5Cen%5Copera%5Cdispute-resolution%5Cresolve-online%5Cedit-package-and-vts-cs-setup-guide-for-vcr.pdf&amp;contenttype=application/pdf</a> )
5	CIB Activation <ul style="list-style-type: none"> <li>• Obtain access to VTS-CS</li> <li>• Activate your Test BIN with Visa</li> </ul>	
6	Prepare Host &/or VTS-VIP for online test data generation <ol style="list-style-type: none"> <li>Setup VTS-VIP</li> <li>Perform station pairing through VTS/3</li> <li>Conduct test host sign on/sign off connectivity test through VCMS</li> </ol>	Visa Claims Resolution VTS-VIP Setup Guide ( <a href="https://secure.visaonline.com/_layouts/GVOL/GVOLStream.aspx?filename=%5Croot%5Ccontent%5Cdam%5Cgvol%5Cen%5Copera%5Cdispute-resolution%5Cresolve-online%5Cvcr-vts-vip-setup-guide.pdf&amp;contenttype=application/pdf">https://secure.visaonline.com/_layouts/GVOL/GVOLStream.aspx?filename=%5Croot%5Ccontent%5Cdam%5Cgvol%5Cen%5Copera%5Cdispute-resolution%5Cresolve-online%5Cvcr-vts-vip-setup-guide.pdf&amp;contenttype=application/pdf</a> )
7	VROL/VCR preparation <ol style="list-style-type: none"> <li>Perform VCR issuer testing</li> <li>Perform VCR acquirer testing</li> </ol>	VROL VCR Setup / Testing Guide ( <a href="https://secure.visaonline.com/_layouts/GVOL/GVOLStream.aspx?filename=%5Croot%5Ccontent%5Cdam%5Cgvol%5Cen%5Copera%5Cdispute-resolution%5Cresolve-online%5Cvrol-vcr-setup-testing-guide.pdf&amp;contenttype=application/pdf">https://secure.visaonline.com/_layouts/GVOL/GVOLStream.aspx?filename=%5Croot%5Ccontent%5Cdam%5Cgvol%5Cen%5Copera%5Cdispute-resolution%5Cresolve-online%5Cvrol-vcr-setup-testing-guide.pdf&amp;contenttype=application/pdf</a> )

## Step 7: Implement the New Connection

Upon completion of successful testing, Visa will coordinate the activation of your RTSI solution.

- Once the configuration within VROL is activated, members using RTSI can integrate their system with VROL. Hence, it is desirable for members to coordinate the modifications of their systems with the activation date. Activating the member's configuration within VROL can be scheduled in advance.
- Download and install certificates—Visa PKI X-509, Visa CA Root, and Visa Intermediate CA certificates. For more details, see Certificates for VROL RTSI available in the documentation suite on MTE2.

Visa can help plan the details of this implementation process to allow members to better manage the complexity associated with large-scale activations using the VROL systems interfaces. Visa is also available for post-implementation technical support

for all services. However, members are responsible for monitoring and tracking their own users and systems.

Visit VCR Learning Resources (<https://secure.visaonline.com/SitePages/Content.aspx?pageid=2.1.2.9.0>) to access the most important training materials for VCR. Members have access to a full catalog of online courses free of charge.

Members can contact the global VROL helpdesks for inquiries and issues:

- AP (apvrolsupport@visa.com)
- Canada, U.S. (rolsupport@visa.com)
- CEMEA (cemeavrol@visa.com)
- Europe (CustomerSupport@visa.com)
- LAC (VCSA CASE must be opened)

# RTSI Optimization Points

J

This appendix describes the additional important points to consider when implementing RTSI as well as the recommended best practices for optimizing RTSI.

## Additional Important Points to Consider in RTSI

The user must consider the following additional important points when implementing RTSI.

### SOAP and REST Web Services

Real-Time Systems Interface (RTSI) is a Web service that supports both SOAP and REST protocols, and synchronous request/response process, for which the client or consumer sends a request message to VROL (provider). VROL then generates an immediate response.

By establishing a synchronous protocol like RTSI, VROL and clients are able to communicate in real-time.

The advantages of REST/JSON are as follows:

- Faster response time
- Able to take higher payload
- Decoupled Architecture
- Simplicity but flexibility

### Error, Warning, and Information Messages

The differences between error messages and warning or information messages are as follows:

- Error messages are displayed for hard stop errors, for which the processing will not continue further when the system encounters an error. During the workflow, error messages are encountered and displayed in the form of E-XXXXXXXXXX.
- Warning or Information messages are not displayed for hard stop errors. They are soft messages for which the processing will continue further when the system encounters a warning or an information message. During the workflow,

warning or information messages are displayed in the form of W-XXXXXXXXXXXX or I-XXXXXXXXXXXX.

## Image Upload Methods

There are two methods that the user can upload images: Hybrid Approach or Non-Hybrid Approach.

### Hybrid Approach

In the Hybrid approach, the user typically makes use of the RTSI Web services along with the Bulk SI upload functions to manage their images.

The three ways to upload images in the Hybrid approach are as follows:

1. Upload images to a central image repository
2. Attach images to an existing dispute questionnaire
3. Add images to an existing case

See [Hybrid Image Uploads](#) for more information.

### Non-Hybrid Approach

In the Non-Hybrid approach, the user makes use of the RTSI Web services to manage their images.

The two ways to upload images in the Non-Hybrid approach are as follows:

1. Attach images to an existing questionnaire
2. Add images to an existing case

See [Attaching Images to Questionnaires or Existing Cases](#) for more information.

The following table describes the scenarios for which the user can use the Hybrid and Non-Hybrid approach.

Hybrid Approach	Non-Hybrid Approach
<ol style="list-style-type: none"> <li>1. The user is a pre-dominantly BulkSI client</li> <li>2. When the images to be added are not readily available at the time of submission</li> </ol>	<ol style="list-style-type: none"> <li>1. The member mainly uses RTSI services for dispute processing</li> <li>2. When the images are readily available</li> </ol>

## Ways to Upload Images

The user can upload images by using one of the following ways:

- Upload images in the standalone upload service
- Upload images in the same service. For example, Dispute Submission service.

## Dispute Submission Service

### When Member Needs to Pull Associated Transactions

As a response to SIIInitiateDisputeFromTransactionOrCaseRequest, if the response received in SIIInitiateDisputeFromTransactionOrCaseResponse for ATRSelectionReqdInd is 'Required', then the member needs to pull the transactions. If the value of ATRSelectionReqdInd is 'Unknown', then the member needs to check again.

**Note:** The value 'Unknown' means that the querying is still in progress.

## When Member Needs to Submit Supporting Functions

The values returned for FraudRptInd, PPCSInd, EFLInd, and AEMInd flags in SInitiateDisputeFromTransactionOrCaseResponse determines the next steps. If it is false, for which the value is 'NotExist', then the member needs to submit Fraud Report, Stop Payment, Exception File Listing, or Accounting Entry Memo in conjunction with dispute submission.

**Note:** Refer to the 'Dispute' tab in Interface Element Specification (IES) for the details of domain values and their interpretations.

## Supporting Services vs Dispute Flow Services

Do not combine supporting functions unless it is needed.

For example, for the case of Fraud standalone service and submit dispute service, if a fraud report has already been submitted for a particular transaction in a card number, and for a case involving the same transaction, then do not submit fraud when submitting the dispute.

## Hypersearch

RTSI users can get the details for ROL cases by using Hypersearch requests. Hypersearch response contains the details of the ROL case, which includes the values of various Ids that are present in the case. The users can use these Ids to retrieve relevant appropriate details. For example, by using the DisputeAdviceld, that is available in the Hypersearch response, the user can retrieve the Dispute Advice details.

See [Chapter 19, Hypersearch Requests](#) for more information.

## Queues vs Download

Queues can either be Web service requests/responses (RTSI queues/General queues) or Batch queues. On the other hand, Download refers to the Bulk system downloads that is equivalent to the Web service response for a request.

The following table describes the scenarios on when queues or downloads should be used.

Queues	Download
1. The user is an RTSI client	1. The user is a BulkSI client
2. Less number of records in output	2. More records (up to 5000 in number)
3. Ideal for low to mid-volume clients	3. Best for high volume clients
4. Synchronous response	4. Asynchronous response

See [Chapter 17, Queues](#) and [Chapter 18, RTSI Batch Queues](#) for more information.

## Batch to RTSI Synchronous Processing Migration

### Important:

The user should spread the load when migrating from batch to RTSI synchronous processing.

The user must consider the following points when migrating from batch to RTSI synchronous processing:

- Throttle API request triggering
  - Limit the concurrent request
  - Set a defined number and make it configurable
  - Start with a lower number, for example: 25
- **Note:** *The increase or decrease of the number depends on the API usage and response processing.*
- API Timeouts
  - Make it configurable or leave the default HTTPs timeout
  - Set the request timeouts to 30 seconds
- Limit the retry counts for API
  - Set 3 as the defined number
  - Make it configurable
  - If all of the 3 calls failed, mark the transaction as failed
  - Open an incident with Visa

## Best Practices

The following sections describe the recommended best practices that the user can consider during dispute submission, and when using queues and batch queues.

### Dispute Submission

The best practices that the user can apply during dispute submission are as follows:

- Avoid bundling
  - From the identified transaction, the optimized way is to create a case and then initiate the dispute rather than directly calling the initiate dispute service.
- Avoid batch processing in RTSI. Instead, call the service whenever the form is ready.
- If opting for Fraud bundling, ensure that you go through the pre-conditions that are required for a successful dispute submission.
- If the value of FraudRptInd, PPCSInd, EFLInd, AEMInd or ATRSelectionReqdInd flag is 'Unknown' in SInitiateDisputeFromTransactionOrCaseResponse, then it is a good practice to wait and check the values in SIGetDisputeDetailsResponse before submitting SISubmitDisputeQuestionnaireRequest.

**Note:** *The value 'Unknown' means that the querying is still in progress.*

See [Chapter 7, Dispute Questionnaires](#) for more information.

### Queues

The best practices that the user can apply when using queues are as follows:

- Transactions or cases must be removed from the queue to reduce the number of transactions returned from VROL for each queue request.

- Once a user has retrieved all the items in a queue, an action should be taken on each transaction. For example, responding to a transaction or using a request to ignore or close the transaction.
- Depending on the downstream usages, the members can choose to implement either the General queues or the Batch queues.
- For Close or Accept dispute items, it is recommended to keep the queue clean and empty by using the Close/Accept functions accordingly.

See [Chapter 17, Queues](#) for more information.

## Batch Queues

- In a batch queue, after completion of work on a particular Batch Item Id, use the Web service to mark the item as read.
- Mark item as read service can be used for individual Item ids or the whole queue depending on the member's needs.
- Keep the following items in mind:
  - An item may appear in multiple batch queues (as applicable) and marking an item as read considering one queue will remove the item from all batch queues for the particular side
  - Batch queues are applicable and will only display the incoming items for the particular side

**Note:** *In a batch queue, the particular item is removed from the queue only when it is marked as read, which means that if the latest status of a particular case has advanced further, it may still be appearing in the batch queue until it is marked as read. However, in the General queues/RTSI queues, a particular case is not returned in the queue if it has advanced further in the workflow.*

See [Chapter 18, RTSI Batch Queues](#) for more information.

THIS PAGE INTENTIONALLY LEFT BLANK.

# Glossary

## **attribute**

A named value or relationship that exists for some or all instances of an entity, in this case a data element, and is directly associated with a given instance.

## **binary**

A system of numbers having 2 as its base and using 0s and 1s for its notation. Computers use binary code because it works well with digital electronics and Boolean algebra.

## **clear Text**

A classification identifying one of various types of data, such as floating-point, date, string, integer, or Boolean, stating the possible values for that type, the operations that can be done on that type, and the way values of that type are stored.

## **data type**

A classification identifying one of various types of data, such as floating-point, date, string, integer, or Boolean, stating the possible values for that type, the operations that can be done on that type, and the way values of that type are stored.

## **extranet**

Part of a company's intranet that is extended to users outside the company.

## **HyperText Markup Language (HTML)**

A cousin of XML, also derived from SGML. HTML has become the de facto language on the Web to facilitate manipulation and display of objects on Web pages. HTML is essentially a display language, while XML is a defining language. Tags in HTML are predefined, whereas in XML, the developer defines the tags, their attributes, and meanings.

## **JPEG**

Joint Photographic Experts Group. JPEG is compression technique for color images and photographs that balances compression against loss of detail in the image. The greater the compression, the more information is lost (this is called lossy compression).

**markup**

Syntactically delimited characters added to the data of a document to represent its structure. There are four different kinds of markup—descriptive markup (tags), references, markup declarations, and processing instructions. Called “markup” because its original use, SGML, was in computerized document preparation, a method of adding information to the text indicating the logical components of a document (for example, paragraph), or instructions for layout of the text (for example, boldface) on the printed page.

**parser**

1. In computer technology, a parser is a program that receives input in the form of sequential source program instructions, interactive online commands, markup tags, or some other defined interface, and breaks them up into parts (for example, the nouns [objects], verbs [methods], and their attributes or options) that can then be managed by other programming (for example, components in a program). A parser may also check to see that all necessary input has been provided.
2. In the XML context, the primary purpose of a parser is to convert data from XML syntax to the internal format of a program, and then pass it to that program. If the data contains syntax errors, it cannot be parsed, and is said to be not well-formed.

**selected transaction**

Before a transaction is selected to dispute, the last transaction viewed by a user or sub draft created is the selected transaction. The selected transaction is used to populate the case header until a transaction is selected to dispute. After a transaction is selected to dispute, that becomes the transaction that populates the header, and viewing or creating sub drafts for any transactions in the case does not change the disputed transaction.

**SGML**

Standard Generalized Markup Language, the forefather of XML. SGML was introduced in the 1980s, and is still popular today, primarily in the arena of document publishing. XML was built using SGML as a foundation, and is, in essence, a simplified version of SGML.

**syntax**

Syntax is the grammar, structure, or order of the elements in a language statement. (“Semantics” is the meaning of these elements.) Syntax applies to computer languages as well as to natural, or spoken, languages. Usually, one thinks of syntax as “word order.”

**tag**

A code, or token, within a data structure that gives instructions for formatting or other actions. XML documents, or files, are set up using tags, which identify each data element. In normal XML syntax, tags always start with < and end with >. The term “tag” is often informally used to mean “element.”

**TIFF**

Tag-based Image File Format. Designed to promote the interchange of digital image data, TIFF is a general purpose data format that is compatible with a wide range of scanners and image processing applications. It is device independent and is used in most operating environments, including Windows and UNIX.

## **XML**

The eXtensible Markup Language, a standard developed by the World Wide Web Consortium that allows developers to create customized tags that allow efficient organization and delivery of content. XML is not, in and of itself, a markup language. Rather it's a "metalinguage", one used to create other specialized languages, of which VML (Vector Markup Language) is an example. As a metalinguage, XML allows for the definition and processing of custom tags. XML is "extensible" because the markup names are unlimited and self-defining (that is, not restricted to a predetermined set of rules).

## **XML payload**

An XML message that is embedded within an HTML page. It is wrapped in a hidden form field called XMLPayload.