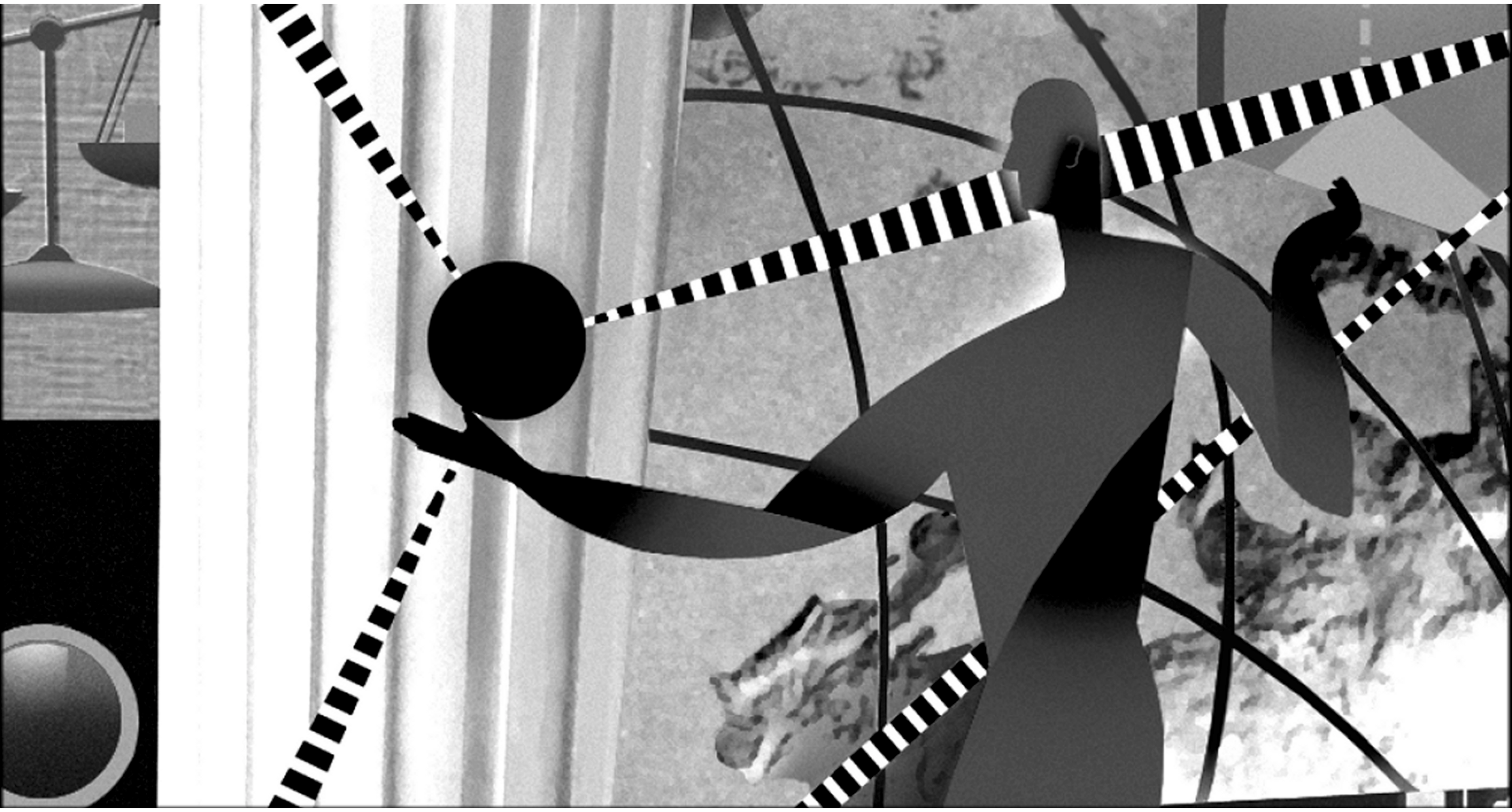
OASYS GLOBAL™
DIRECT



BROKER AND INSTITUTION CONFORMANCE REQUIREMENTS

Version 3.4.2

THOMSON FINANCIAL
ESG

Thomson Financial ESG
22 Thomson Place
Boston, MA 02210.

Printing: 2/2/01

OASYS Global *Direct* Broker and Institution Conformance Requirements

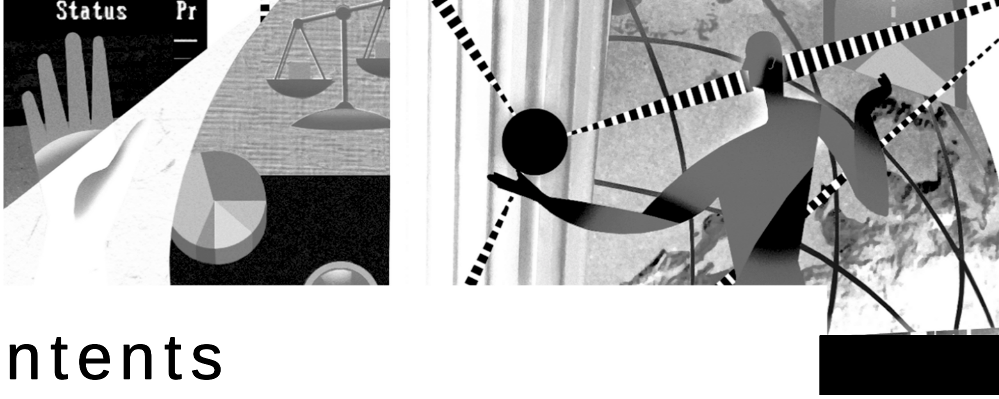
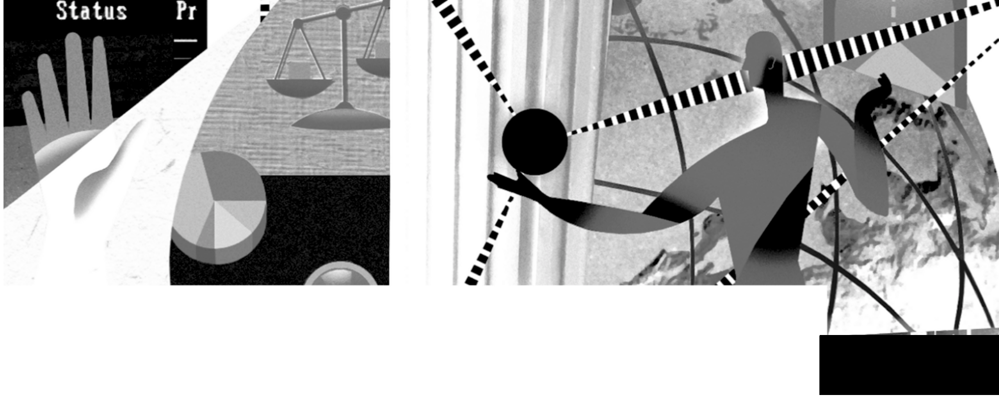


Table of Contents



Preface

This guide provides you with a set of criteria to certify your applications’ compatibility with the OASYS Global *Direct* (OGD) production environment. You must successfully complete the certification process before Thomson ESG can implement your application in the production environment. The certification process consists of a series of message-based and communication-based tests, and it serves as the basis by which Thomson ESG confirms that your application environment is ready for production.

The following sections outline the tests that you need to perform with the aid of the OGD project team. Where applicable, the project team will perform the task(s) required to produce specific scenario(s) to successfully complete a particular test. The team will document each test step-by-step and will include the expected outcome. If your application fails a test, you will need to correct the point of failure and retest. Once all the tests are complete with the expected results, the OGD project team will configure your application to interact with the OGD production environment.

Intended Audience

This document is directed toward your systems analysts, and programmers, and others involved in implementing the link between your internal systems and OGD.

How This Manual Is Organized

This manual contains the following chapters and appendices:

- Chapter 1, “The Certification Process,” introduces the certification process.
- Chapter 2, “Message Flow Testing,” outlines tests involved in broker-side and institution-side testing.
- Chapter 3, “Verification of Minimum, Maximum, and Default Values,” outlines tests involved in confirmation that your application can handle minimum and maximum values for each message type.
- Chapter 4, “Other Tests,” outlines miscellaneous other tests.
- Chapter 5, “Support Procedures,” outlines verification of internal support procedures and hours of availability for support.
- Appendix A, “Client Information Forms,” includes copies of client information forms used during conformance testing.
- Appendix B, “MOA Return/NAK Codes,” includes information on MOA error codes and NAK codes.

Typographic Conventions

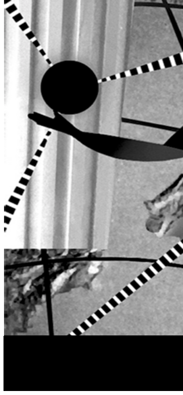
Unless otherwise noted in the text, this manual uses the following typographic conventions:

`Courier`	Commands, printed text examples, function names and parameters, constants, variables, field names, literal values, return values, arguments, transaction names, configuration parameters, default values, format strings, MT511 tags and assigned values, path variables and paths, and C code samples; for example: OASYS LOG REPORT
`Courier Bold`	Data format specifications. For example, **`dd-mmm-yy`** date format.
UPPERCASE	Electronic Trade Confirmation acronyms (such as ETC and API), and message types (such as BLIM).
Mixed-case Italics	Trade and message statuses (for example *Reject*, *Affirm*, *Cancel*).
UPPERCASE ITALICS	MT511 message types (for example, *AE*, *CN* (*CNA CNB*), and *TA*), and return codes (for example, *SUCCESS*, *FAILURE*).
Bold	File names (such as **import.dat** and **trans.map**), and library names (such as **wsock32.dll** and **moa.lib**).

Related Documents

These are other Thomson ESG documents related to this publication:

- OASYS Global Direct MT511 Messaging Specification
- OASYS Global Direct MT511 Parser API Programmer's Guide and Reference
- OASYS Global Direct Message Delivery System TCP/IP API Programmer's Guide
- OASYS Global Direct Migration Guide for OASYS Global Automated Workstation Clients
- OASYS Global Direct Sample MT511 Data - Block and Contract Level Data Flow Examples
- OASYS Global Direct Overview
- OASYS Global Direct Release Notes, Version 3.4.2



Thomson Financial ESG

Thomson Financial ESG is the worldwide leader in providing technology-based workflow solutions to the global investment community. Thomson Financial ESG is committed to partnering with its clients to dramatically reduce the risk and cost of processing trades, and thereby improving their investment performance. Thomson Financial ESG services are used in 37 countries for domestic and cross-border trading. Thomson Financial ESG supports approximately 3,450 clients from 18 offices located in major financial centers across the globe.

Thomson Financial ESG is part of Thomson Financial (TF), a leading provider of information services and work solutions to the worldwide financial community. TF employs more than 7,000 people in more than 40 locations dedicated to the success of our clients and is part of The Thomson Corporation (TTC). With annual revenues approaching US$6 billion, TTC is one of the world's leading information and publishing companies. TTC's common shares are traded on the Toronto, Montreal and London stock exchanges

Thomson Financial ESG’s products and services include ALERT®, ALERT*Direct*™, OASYS Global™, OASYS Global *Direct*™, OASYS™, OASYS *Direct*™, MarketMatch™, Thomson Report™, and AutoMatch™.

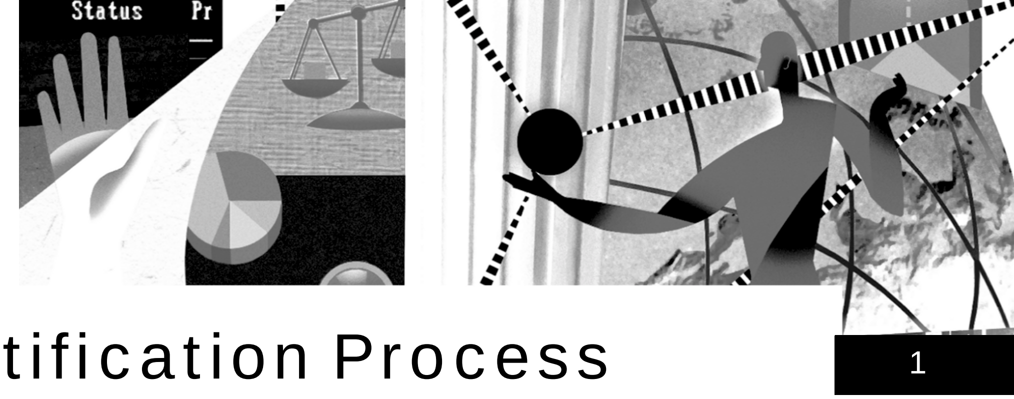
Sales Executives

Call your sales executives to learn more about other Thomson Financial ESG services that can automate your post-trade operations, including ALERT®, ALERT*Direct*™, AutoMatch™, OASYS Global™, OASYS Global *Direct*™, OASYS™, OASYS *Direct*™, MarketMatch™, and Thomson Report™. For assistance call the following sales offices.

Boston	1-800-407-4264	Paris	33-1-4453-7878
Chicago	1-312-629-0957	San Francisco	1-415-989-9797
Edinburgh	44-131-220-8417	Sao Paulo	5511-816-5022
Frankfurt	49-69-971-75200	Singapore	65-295-6383
Hong Kong	852-2533-5415	Stockholm	46-8-587-67160
London	44-207-369-7349	Sydney	612-9225-3119
Mexico City	525-535-6070	Thailand	001-800-65-1555
New York	1-212-612-9604	Tokyo	813-5218-6621

Electronic Trade Confirmation Code of Practice

See the “Electronic Trade Confirmation Code of Practice” section in *OASYS Global Direct Overview* for more information about the Electronic Trade Confirmation Code of Practice.



1: The Certification Process

Overview

The OGD certification process ensures that your application can handle the full functionality of the OGD production environment. You must complete all client coding before your ESG consultant can begin the certification process. In addition, you must submit your user acceptance test results for review by ESG.

This document includes the certification plan for both brokers and institutions.

1. Brokers must complete the tests outlined in the “Broker Message Flow Testing” section in Chapter 2, “Message Flow Testing.”
2. Institutions must complete the tests outlined in the “Institution Message Flow Testing” section in Chapter 2, “Message Flow Testing.”

The certification process test scenarios fully test the valid OGD message flows. The minimum and maximum value tests perform stress testing of individual fields. When you are confident that your application meets the functional requirements and is robust enough for installation in a production environment, Thomson ESG allocates time for conformance testing. To pass conformance testing, you complete the broker or institution tests without interruption. If possible, we recommend that as part of your internal testing you send messages to external counterparties who have a suitable test environment available. Doing this benefits all parties involved when the application migrates to the production environment.

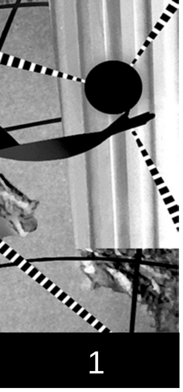
Client Requirements

We strongly encourage you to use this document as part of your internal testing strategy. You must complete all tests internally before ESG allocates time for conformance testing.

To certify the full functionality of the OGD environment, you need to populate all mandatory and optional message fields. If you do not require the full functionality (e.g. the optional field set), certification is still possible with ESG agreement and sign off. If your internal application does not contain certain functionality, you must document this in your profile with the reasons for the exclusion(s). You must also communicate this to an ESG representative before testing. Apart from these specifically agreed instances, ESG will require full testing of both optional and mandatory field sets.

You cannot make functional changes to the application in the production environment; you can make changes only in the test environment. If you add functional enhancements to your application after conformance testing is complete, you must retest the application before the new version migrates to the production environment.

You must complete the conformance test sequentially. If any failed test requires changes to your application(s), you must restart conformance testing from the beginning.

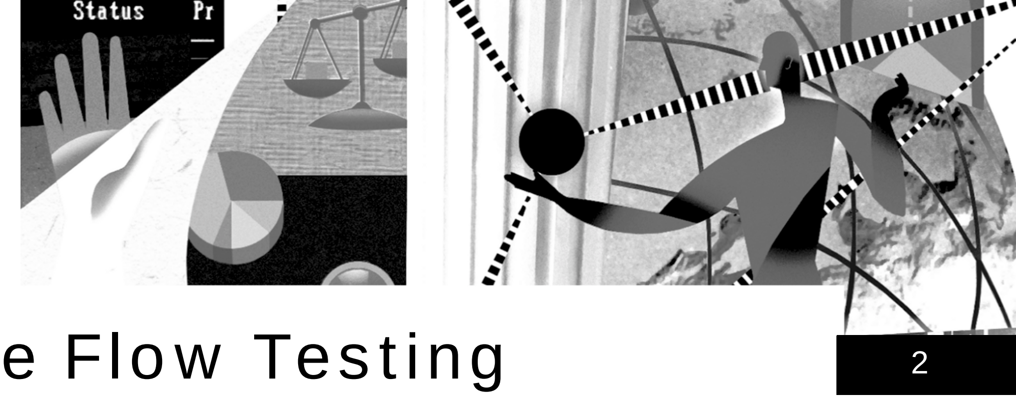


Client Information and Message Data Forms

Before starting the certification procedures, you and ESG must complete the forms located in Appendix A, “Client Information Forms.” These are a set of templates used to agree on the details of each trade before testing begins.

When populating message fields, please observe the following guidelines:

- Consider the range of possible field values and test every value. Test a fair and varied cross section of values as well as minimum and maximum values.
- The test scenarios repeatedly use all message types. This lets you test a variety of different field values. Fully test every field value; ESG can re-run certain tests to incorporate a wider value selection.



2: Message Flow Testing

This chapter outlines tests involved in broker-side and institution-side testing. It contains the following sections:

Note! For all conformance tests in this and subsequent chapters, check the MOA header for each message to ensure that the Message Reference Key (RFK) and Message Transaction Reference Number (TRN) fields are properly formed.

The RFK should be blank-padded to 16 places and constructed of the following fields.

Field	Description
TFH1	3 positions. Always 511,
23A	2 positions. Always *AE*, *TA*, or *CN*.
23B	2 positions. 01 for new, 02 for amend, 03 for cancel.
TF01	1 position. Advice indicator added by MDS/API host.
23M	3 positions. Status/Response code added by MDS/API host.

:

Examples:

511AE01

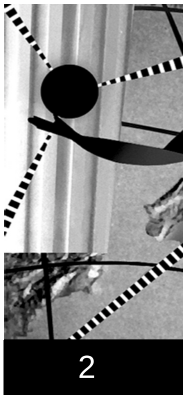
511CN02

The TRN for a new message should be blank-padded to 21 places and constructed of the following fields.

Field	Description
	01 (constant)
TF14	Sender’s reference (FLD_PARTY_REF)
	No version number and no “.”
TF01	1 position. Advice indicator added by MDS/API host.
23M	3 positions. Status/Response code added by MDS/API host.

The TRN for a response to an existing message should be blank-padded to 21 places and constructed of the following fields.

Field	**Description**
20A	2 positions. Always 03.
20B	14 positions. Always OG reference number terminated with “.”
20C	2 positions. Version number (for example, 01, or 02).



Broker Message Flow Testing

Block Trading

Use tests 1 through 8 to test broker-side block trading. These tests validate your application’s ability to send broker-side messages that contain all possible fields. They also validate your application’s ability to receive all possible messages, to receive all fields contained in those messages, and to validate messages from the host and the test institution.

Test 1 — Block Trade Flow

This test verifies that your application can create and send an *AE*, receive allocations from the counterparty, and cancel the *AE*.

1. The broker sends an Advice of Execution (*AE*) message and receives multiple Trade Allocation (*TA*) messages from the institutional counterparty.
2. The broker accepts the allocations.
3. The broker should send an *AE Cancel* just before receiving the *CNB New Affirm*.

Here is the action table for Test 1.

Message	**Action**
AE New	Send
AE New ValueAdd	Receive
AE New Valid	Receive
TA New	Receive
TA New Valid	Send
TA New	Receive
TA New Valid	Send
CNB New	Send
CNB New Valid	Receive
CNB New	Send
CNB New Valid	Receive
AE Cancel	Send
CNB New Affirm	Receive
CNB New Affirm	Receive

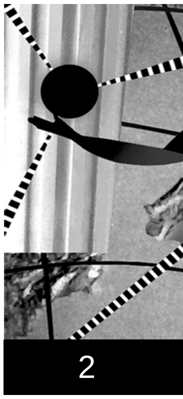
Test 2 — AE and CNB Amend

This test verifies that your application can create, send, and then amend a *CNB New* message.

1. The broker sends an *AE* message. The institution rejects the message.
2. The broker then sends an *AE Amend* message.
3. Upon receiving allocations from the institution, the broker verifies the allocations and sends a *CNB New* message.
4. The institution then rejects the contract note, then the broker sends a *CNB Amend* message.
5. The institution rejects the *CNB Amend*, and the broker again amends the trade.
6. The institution then validates the second amendment, and the broker cancels the block trade.
7. The broker should send its *AE Cancel* before the second *CNB Amend Valid* arrives to test out-of-sequence messages.

Here is the action table for Test 2.

Message	Action
AE New	Send
AE New ValueAdd	Receive
AE New Valid	Receive
AE New Reject	Receive
AE Amend	Send
AE Amend ValueAdd	Receive
AE Amend Valid	Receive
TA New	Receive
TA New Valid	Send
CNB New	Send
CNB New Valid	Receive
CNB New Reject	Receive
CNB Amend	Send
CNB Amend Valid	Receive
CNB Amend Reject	Receive
CNB Amend	Send
CNB Amend Valid	Receive
AE Cancel	Send
AE Canceled	Receive



Test 3 — AE New and Amend Reject and AE Cancel

This test verifies that your application can increment the *AE* version number, and it also verifies proper handling of out-of-sequence messages.

1. The broker sends an *AE* message. The institution rejects the message.
2. The broker then amends the trade.
3. The institution rejects the *AE Amend*, and the broker cancels the trade.
4. The broker should send its *AE Cancel* before the *AE Amend Reject* arrives to test out-of-sequence messages.

Here is the action table for Test 3.

Message	Action	Version
AE New	Send	1
AE New ValueAdd	Receive	
AE New Valid	Receive	
AE New Reject	Receive	
AE Amend	Send	2
AE Amend ValueAdd	Receive	
AE Amend Valid	Receive	
AE Cancel	Send	
AE Canceled	Receive	
AE Amend Reject	Receive	

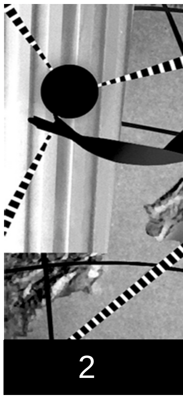
Test 4 — AE New and Amend Reject and AE Cancel

This test verifies that your application can increment the *AE* version number, and it also verifies proper handling of out-of-sequence messages.

1. The broker sends an *AE* message. The institution rejects the message.
2. The broker then amends the trade.
3. The institution rejects the *AE Amend*, and the broker cancels the trade.
4. The broker should send its *AE Cancel* after the *AE Amend Reject* arrives to test out-of-sequence messages.

Here is the action table for Test 4.

Message	Action	Version
AE New	Send	1
AE New ValueAdd	Receive	
AE New Valid	Receive	
AE New Reject	Receive	
AE Amend	Send	2
AE Amend ValueAdd	Receive	
AE Amend Valid	Receive	
AE Amend Reject	Receive	
AE Cancel	Send	
AE Canceled	Receive	



Test 5 — TA and CN Version Increments

This test verifies that your application can increment the Contract Note version number.

1. The broker sends an *AE* message that results in allocations from the institution.
2. The institution amends the allocations (*TA Amend*), and the broker sends contract note (*CNB New*) messages.
3. The institution rejects the contract note messages. The broker sends *CNB Amend* messages.
4. The institution then affirms the contract note messages.

Here is the action table for Test 5.

Message	**Action**	**Version**
AE New	Send	
AE New ValueAdd	Receive	
AE New Valid	Receive	
TA New	Receive	1
TA New Valid	Send	
TA New Reject	Send	1
TA Amend	Receive	2
TA Amend Valid	Send	
CNB New	Send	2
CNB New Valid	Receive	
CNB New Reject	Receive	2
CNB Amend	Send	3
CNB Amend Valid	Receive	
CNB Amend Affirm	Receive	

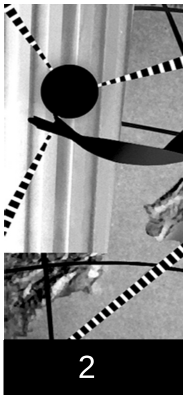
Test 6 — TA Amend and AE Cancel

This test verifies that the *CNB* mirrors the version number of the amended *TA*.

1. The broker sends an *AE* message and the institution allocates it.
2. The broker then rejects the allocations, and the institution amends them.
3. After rejecting the amended allocations, the broker cancels the block trade.

Here is the action table for Test 6.

Message	Action
AE New	Send
AE New ValueAdd	Receive
AE New Valid	Receive
TA New	Receive
TA New Valid	Send
TA New Reject	Send
TA Amend	Receive
TA Amend Valid	Send
TA Amend Reject	Send
TA Amend	Receive
TA Amend Valid	Send
TA Amend Reject	Send
TA Amend	Receive
TA Amend Valid	Send
cn New	Send
CN New Valid	Receive
CN New Affirm	Receive



Test 7 — Message Crossing

This test verifies that your application can handle out-of-sequence messages.

Send *AE Cancel* before receiving *AE New ValueAdd*, then *AE New Valid* and *AE Canceled*.

1. The broker sends an *AE* message.
2. The broker cancels the *AE* before receiving the *AE ValueAdd*.

Here is the action table for Test 7.

Message	**Action**
AE New	Send
AE Cancel	Send
AE New ValueAdd	Receive
AE New Valid	Receive
AE Canceled	Receive

Test 8 — Message Crossing

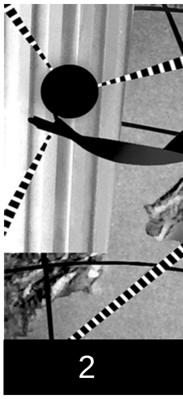
This test verifies that your application can handle out-of-sequence messages.

Send *AE Cancel* after receiving *AE New ValueAdd*, then *AE New Valid* and *AE Canceled*.

1. The broker sends an *AE* message.
2. The broker sends the *Cancel* after the *AE ValueAdd*.

Here is the action table for Test 8.

Message	**Action**
AE New	Send
AE New ValueAdd	Receive
AE Cancel	Send
AE New Valid	Receive
AE Canceled	Receive



Contract Note Trading

Use tests 9 through 13 to test broker-side contract note trading. These tests validate your application’s ability to send broker-side messages that contain all possible fields. They also validate your application’s ability to receive all possible messages and validate those messages from the host and the test institution.

Test 9 — CNA Amend

This test verifies the broker’s ability to amend a *CNA*. The version number should be verified.

1. The broker sends a *CNA New* message that the institution rejects.
2. The broker then sends a *CNA Amend* and receives an affirmation message from the institutional counterparty.

Here is the action table for Test 9.

Message	**Action**
CNA New	Send
CNA New ValueAdd	Receive
CNA New Valid	Receive
CNA New Reject	Receive
CNA Amend	Send
CNA Amend ValueAdd	Receive
CNA Amend Valid	Receive
CNA Amend Affirm	Receive

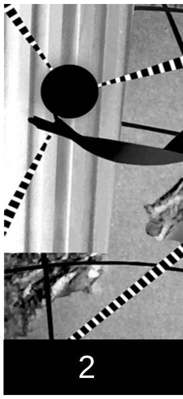
Test 10 — CNA Amend and CNA Cancel

This test verifies that the broker can amend then cancel a *CNA*.

1. The broker sends a *CNA New* message that the institution rejects.
2. The broker then amends the rejected contract note.
3. The institution rejects the *CNA Amend*.
4. The broker then cancels the contract note by sending a *CNA Cancel*.

Here is the action table for Test 10.

Message	Action	Version
CNA New	Send	1
CNA New ValueAdd	Receive	
CNA New Valid	Receive	
CNA New Reject	Receive	
CNA Amend	Send	2
CNA Amend ValueAdd	Receive	
CNA Amend Valid	Receive	
CNA Amend Reject	Receive	
CNA Cancel	Send	
CNA Canceled	Receive	



Test 11 — CNA New Affirm and CNA Cancel

This test verifies that the broker can send a *CNA* and cancel an affirmed *CNA*.

1. The broker sends a *CNA New* message that the institution accepts.
2. The broker then receives affirmation messages from the institutional counterparty.
3. The broker then cancels the contract note.

Here is the action table for Test 11.

Message	**Action**
CNA New	Send
CNA New ValueAdd	Receive
CNA New Valid	Receive
CNA New Affirm	Receive
CNA Cancel	Send
CNA Canceled	Receive

Test 12 — Send CNA Cancel Before Receiving CNA New ValueAdd

This test verifies that your application can handle out-of-sequence messages.

1. The broker sends a *CNA New* message.
2. The broker then cancels the contract before receiving the *CNA ValueAdd*.

Here is the action table for Test 12.

Message	**Action**
CNA New	Send
CNA Cancel	Send
CNA New ValueAdd	Receive
CNA New Valid	Receive
CNA Canceled	Receive

Test 13 — Send CNA Cancel After Receiving CNA New ValueAdd

This test verifies that your application can handle out-of-sequence messages.

1. The broker sends a *CNA New* message.
2. The broker then sends the *CNA Cancel* after *CNA ValueAdd*.

Here is the action table for Test 13.

Message	Action
CNA New	Send
CNA New ValueAdd	Receive
CNA Cancel	Send
CNA New Valid	Receive
CNA Canceled	Receive

Institution Message Flow Testing

Block Level Institution

Use tests 14 through 20 to test block level institution-side block trading. These tests validate your application's ability to send block level institution-side messages that contain all possible fields. They also validate your application's ability to receive all possible messages and validate those messages from the host and a broker.

Test 14 — Block Trade Flow

This test verifies that your application can receive an *AE*, send *TA*s to the broker, and affirm the *CNB*s.

1. The institution receives an *AE* message.
2. The institution then sends *TA* messages to the broker.
3. The institution then receives and affirms the contract notes (*CNB*s).

Here is the action table for Test 14.

Message	**Action**
AE New	Receive
AE New Valid	Send
TA New	Send
TA New	Send
TA New ValueAdd	Receive
TA New ValueAdd	Receive
TA New Valid	Receive
TA New Valid	Receive
CNB New	Receive
CNB New Valid	Send
CNB New Affirm	Send
CNB New	Receive
CNB New Valid	Send
CNB New Affirm	Send

Test 15 — AE Amend With CNB Reject and AE Cancel

This test verifies that your application can reject contract notes.

1. The institution receives an *AE* message. It rejects the message.
2. The broker then amends the trade by sending an *AE Amend* message.
3. After receiving the *AE Amend*, the institution sends a *TA New* message to the broker.
4. The broker creates contract notes. The client application receives and rejects these contract notes.
5. The broker then cancels the block trade.

Here is the action table for Test 15.

Message	**Action**
AE New	Receive
AE New Valid	Send
AE New Reject	Send
AE Amend	Receive
AE Amend Valid	Send
TA New	Send
TA New ValueAdd	Receive
TA New Valid	Receive
CNB New	Receive
CNB New Valid	Send
CNB New Reject	Send
AE Cancel	Receive
AE Canceled	Send

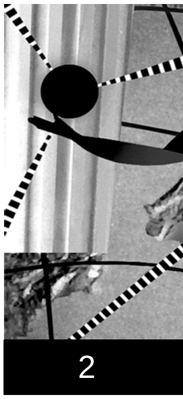
Test 16 — AE Amend and AE Cancel

This test verifies that your application can reject an *AE Amend*.

1. The broker sends an *AE New* message to the institution.
2. The institution sends an *AE New Reject* message to the broker.
3. The broker then sends an *AE Amend* message.
4. The institution rejects the *AE Amend*.
5. The broker sends an *AE Cancel* message to cancel the trade.

Here is the action table for Test 16.

Message	**Action**
AE New	Receive
AE New Valid	Send
AE New Reject	Send
AE Amend	Receive
AE Amend Valid	Send
AE Amend Reject	Send
AE Cancel	Receive
AE Canceled	Send



Test 17 — TA and CNB Amend

This test verifies that your application can affirm amended *CNB*s.

1. The broker sends an *AE New* to the institution.
2. The institution receives the message and sends allocations (*TA New ValueAdd*).
3. The broker then rejects the allocations (*TA New Reject*).
4. The institution amends the allocations (*TA Amend*), and the broker sends contract notes (*CNB New*).
5. The institution rejects the contract notes (*CNB New Reject*), and the broker amends the contract notes (*CNB Amend*).
6. The institution affirms the amended contract notes (*CNB Amend Affirm*).

Here is the action table for Test 17.

Message	**Action**	**Version**
AE New	Receive	
AE New Valid	Send	
TA New	Send	1
TA New ValueAdd	Receive	
TA New Valid	Receive	
TA New Reject	Receive	
TA Amend	Send	2
TA Amend ValueAdd	Receive	
TA Amend Valid	Receive	
CNB New	Receive	2
CNB New Valid	Send	
CNB New Reject	Send	
CNB Amend	Receive	3
CNB Amend Valid	Send	
CNB Amend Affirm	Send	

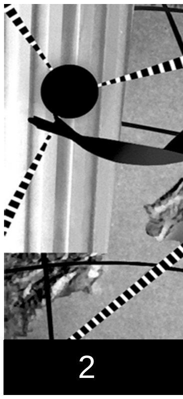
Test 18 — TA Reject, TA Cancel, and AE Cancel

This test verifies that your application can process *TA Amend* and *TA Cancel* messages.

1. The broker sends an *AE New* to the institution.
2. The institution receives the message and allocates the block trade (*TA New*).
3. The broker rejects the allocation.
4. The institution then cancels the allocation (*TA Cancel*) and sends a new allocation (*TA New*) for the trade.
5. The broker rejects the new allocation.
6. The institution then amends (*TA Amend*) the allocation.
7. After rejecting the amended allocation, the broker cancels the block trade.
8. The institution should also send a *TA Cancel* just before receiving the *AE Cancel*; no *TA Canceled* will arrive.

Here is the action table for Test 18.

Message	**Action**
AE New	Receive
AE New Valid	Send
TA New	Send
TA New ValueAdd	Receive
TA New Valid	Receive
TA New Reject	Receive
TA Cancel	Send
TA Canceled	Receive
TA New	Send
TA New ValueAdd	Receive
TA New Valid	Receive
TA New Reject	Receive
TA Cancel	Send
AE Cancel	Receive
AE Canceled	Send



Test 19 — Send TA Cancel Upon Receiving TA New ValueAdd

This test verifies that your application can handle out-of-sequence messages.

1. The broker sends an *AE New* to the institution.
2. The institution receives the message and allocates the block trade (*TA New*).
3. The institution then cancels the allocation (*TA Cancel*) before receiving the *TA New ValueAdd*.

Here is the action table for Test 19.

Message	**Action**
AE New	Receive
AE New Valid	Send
TA New	Send
TA New ValueAdd	Receive
TA Cancel	Send
TA New Valid	Receive
TA Canceled	Receive

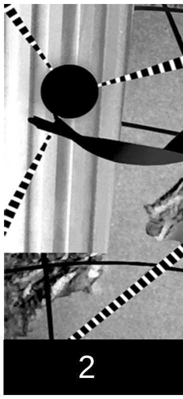
Test 20 — Send TA Cancel After Receiving TA New ValueAdd

This test verifies that your application can handle out-of-sequence messages.

1. The broker sends an *AE New* to the institution.
2. The institution receives the message and allocates the block trade (*TA New*).
3. The institution then cancels the allocation (*TA Cancel*) after receiving the *TA New ValueAdd*.

Here is the action table for Test 20.

Message	Action
AE New	Receive
AE New Valid	Send
TA New	Send
TA New ValueAdd	Receive
TA New Valid	Receive
TA Cancel	Send
TA Canceled	Receive



Confirm Level Institution Message Flow Testing

Use tests 21 through 23 to test institution-side contract note trading. These tests validate your application’s ability to send institution-side messages that contain all possible fields. They also validate your application’s ability to receive all possible messages and validate those messages from the host and a broker.

Test 21 — CNA Amend

1. The broker sends a contract note (*CNA New*) message to the institution.
2. The institution receives the message and rejects it (*CNA New Reject*).
3. The broker then sends a *CNA Amend*, and the institution accepts the change.
4. The institution then affirms the amended contract note (*CNA Amend Affirm*).

Here is the action table for Test 21.

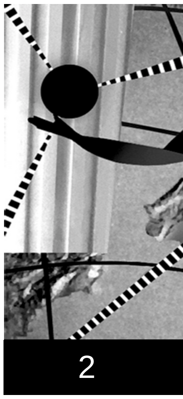
Message	**Action**	**Version**
CNA New	Receive	1
CNA New Valid	Send	
CNA New Reject	Send	
CNA Amend	Receive	2
CNA Amend Valid	Send	
CNA Amend Affirm	Send	

Test 22 — CNA Amend and CNA Cancel

1. The broker sends a contract note (*CNA New*) message to the institution.
2. The institution receives the message and rejects it (*CNA New Reject*).
3. The broker then amends the rejected contract note (*CNA Amend*).
4. The institution rejects the amended contract note (*CNA Amend Reject*).
5. The broker then cancels the contract note (*CNA Cancel*).

Here is the action table for Test 22.

Message	Action	Version
CNA New	Receive	1
CNA New Valid	Send	
CNA New Reject	Send	
CNA Amend	Receive	2
CNA Amend Valid	Send	
CNA Amend Reject	Send	
CNA Cancel	Receive	
CNA Canceled	Send	

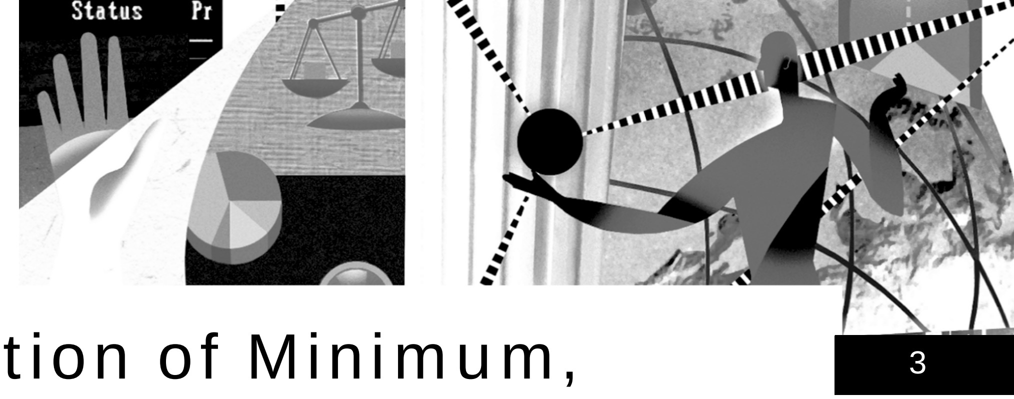


Test 23 — CNA New Affirm and CNA Cancel

1. The broker sends a contract note (*CNA New*) message to the institution.
2. The institution receives the message and sends an affirmation (*CNA New Affirm*).
3. The broker then cancels the contract note (*CNA Cancel*).

Here is the action table for Test 23.

Message	**Action**
CNA New	Receive
CNA New Valid	Send
CNA New Affirm	Send
CNA Cancel	Receive
CNA Canceled	Send



3: Verification of Minimum, Maximum, and Default Values

This chapter outlines tests involved in confirmation that your application can handle minimum and maximum values for each message type. It contains the following sections:

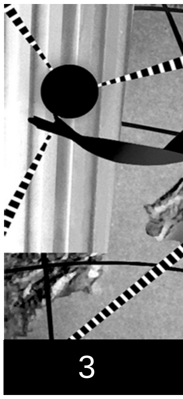
Use tests 24 through 27 to test field value thresholds. These tests confirm that your application can handle minimum and maximum values for message types. Both the broker and the institution need to send messages to the host containing the data values included in the following tables. You must check each field value for accuracy.

Test 24 — Advice of Execution

1. The broker must send two *AE* messages. One must contain fields with the minimum values allowed, and the other must contain fields with the maximum values allowed.
2. The broker must check each field value for accuracy.
3. Send two *AE* messages, one in which you set the quantity to the maximum value and the price to `1`, and the second in which you set the quantity to `1` and the price to the maximum value.

Here is the data table for Test 24.

Data Dict#	Data Element	Field Tag	Sub-field	Min Value	Max Value
23	Quantity of Financial Instrument	35A		SHS0,01	Outbound: SHS999999999999999,98 Inbound: SHS99999999999999,99
27	Transaction Price	33T2		.00000001	Outbound: 99999999999999,98 Inbound: 99999999999999,99
32	Special Concessions	33S		JPY0,	Outbound:JPY99999999999999,98 Inbound: JPY99999999999999,99
35	Accrued Interest (Added) (Subtracted)	34G 34H		0000JPY0, 0000JPY0	9999JPY99999999999999999 9999JPY99999999999999999
40	Financial Instrument Attribute				
	Maturity Date	23F	MDD	19800101	20491231
	Yield	23F	YLD/CALL		Outbound: 9999999999.999998 Inbound: 9999999999,999999
			YLD/CURRENT		Outbound: 9999999999.999998 Inbound: 9999999999,999999
			YLD/FUTURE		Outbound: 9999999999.999998 Inbound: 9999999999,999999
			YLD/MATURE		Outbound: 9999999999.999998 Inbound: 9999999999,999999
			YLD/REP		Outbound: 9999999999.999998 Inbound: 9999999999,999999
	Coupon Rate	23F	CPN	0,	Outbound: 9999999999.999998 Inbound: 9999999999,999999
	Issuer	23F	ISR		30 characters
	Call Date	23F	CLD	19800101	20491231
	Call Price	23F	CLP	JPY0,	JPY99999999,99999999
	Call Type	23F	CLT		17 characters
	Dated Date	23F	DD	19800101	20491231
	Odd First Coupon Date	23F	OFCD	19800101	20491231
	Book Entry Only	23F	BE	`Y` or `N`	`Y` or `N`
	Alternative Minimum Tax	23F	AMTX	`Y` or `N`	`Y` or `N`
	Federal Tax	23F	FTX		`Y` or `N`
	Rating	23F	RT		Outbound: AAAAAAAA Inbound: MOODY AAAAAAAA Outbound: BBBBBBBB Inbound: SP BBBBBBBB
	Amortized Factor	23F	FCT	0,	99999999,99999999
	Original Face Amount	23F	ORG	JPY0,	Outbound: PY99999999999999.98 Inbound: 99999999999999,99
	Current Face Value	23F	CFV	0,	Outbound: 99999999999999.98 Inbound: 99999999999999,99
41	Transaction Condition	23J			
43	Party Type	23K		IMGR	IMGR
45	Party Identification	80J			
45a	Party's Reference to the Transaction	TF14			
42	Override Instructions	23O			



Data Dict#	Data Element	Field Tag	Sub-field	Min Value	Max Value
43	Party Type	23K		EXEC	EXEC
45	Party Identification	80J			
45a	Party's Reference to the Transaction	TF14			
46	Further Info for Party Identified Alert Country Code Alert Method Type Alert Security Type Alert Delivery Name	 72B 72B 72B 72B	 ALCC ALMT ALSC ALDN		
95	Settlement Instruction Field Code	TF10			
96	Settlement Instruction Field Value	TF11			
58	Deal Amount	32M		JPY0,	Outbound: JPY99999999999999,98 Inbound: SHS99999999999999,99
83	Sender to Receiver Information	72			Test with 5 lines by 35 characters.

To certify fixed income trades, test the minimum and maximum values for each fixed income field.

To certify ALERT lookups via OGD, test appropriate values for field 72B `Further Information for Party Identified`.

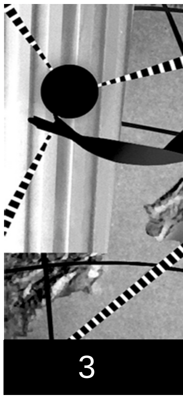
Test 25 — Trade Allocation

1. The institution must send two *TA* messages. One must contain fields with the minimum values allowed, and the other must contain fields with the maximum values allowed.
2. The institution must check each field value for accuracy.

Here is the data table for Test 25.

Data Dict#	Data Element	Field Tag	Sub -field	Min Value	Max Value
23	Quantity of Financial Instrument	35A		SHS0,01	Outbound: SHS999999999999999,98 Inbound: SHS99999999999999,99
27	Transaction Price	33T2		.00000001	Outbound: 99999999999999,98 Inbound: 99999999999999,99
42	Standing Instruction Override Indicator	23O			
43	Party Type	23K		IMGR	IMGR
45	Party Identification	80J			
45a	Party’s Reference to the Transaction	TF14			
46	Further Info for Party Identified Alert Access Code Alert Country Code Alert Method Type Alert Security Type Alert Delivery Name	 72B 72B 72B 72B 72B	 ALAC ALCC ALMT ALSC ALDN		
95	Settlement Instruction Field Code	TF10			
96	Settlement Instruction Field Value	TF11			
43	Party Type	23K		EXEC	EXEC
45	Party Identification	80J			
45a	Party’s Reference to the Transaction	TF14			
43	Party Type	23K		FUND	FUND
46	Further Info for Party Identified Party Name	 72B	 NAM		Test with 5 lines by 35 characters.
51	Type of Commission Sharing Arrangement	23Q			
58	Deal Amount	32M		JPY0,	Outbound: JPY99999999999999,98 Inbound: SHS99999999999999,99
59	Exchange Rate	36I		0,	99999999,99999999
83	Sender to Receiver Information	72			Test with 5 lines by 35 characters.

Test 26 — Contract Note Block Trading

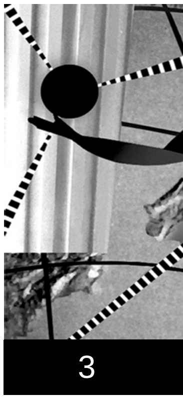


1. The broker must send two *CNB* block trading messages. One must contain fields with the minimum values allowed, and the other must contain fields with the maximum values allowed.
2. The broker must check each field value for accuracy.

Here is the data table for Test 26.

Data Dict#	Data Element	Field Tag	Sub -field	Min Value	Max Value
32	Special Concessions	33S		JPY0,	Outbound:JPY99999999999999,98 Inbound: JPY99999999999999,99
35	Accrued Interest (Added) (Subtracted)	34G 34H		0000JPY0, 0000JPY0	9999JPY99999999999999999 9999JPY99999999999999999
43	Party Type	23K		IMGR	IMGR
45	Party Identification	80J			
45a	Party’s Reference to the Transaction	TF14			
42	Override Instructions	23O			
43	Party Type	23K		EXEC	EXEC
45	Party Identification	80J			
45a	Party’s Reference to the Transaction	TF14			
46	Further Info for Party Identified Account Reference	72B	ACCTREF		
95	Settlement Instruction Field Code	TF10			
96	Settlement Instruction Field Value	TF11			
58	Deal Amount	32M		JPY0,	Outbound: JPY99999999999999,98 Inbound: SHS99999999999999,99
61	Charge/Tax Type	71B1		BROK	BROK
64	Charge/Tax Amount	71B3		JPY0,	Outbound: JPY999999999999999, Inbound: JPY9999999999999999,
61	Charge/Tax Type	71B1		FEES	FEES
64	Charge/Tax Amount	71B3		JPY0,	Outbound: JPY999999999999999, Inbound: JPY9999999999999999,
61	Charge/Tax Type	71B1		MISC	MISC
64	Charge/Tax Amount	71B3		JPY0,	Outbound: JPY999999999999999, Inbound: JPY9999999999999999,
61	Charge/Tax Type	71B1		TTAX	TTAX
64	Charge/Tax Amount	71B3		JPY0,	Outbound: JPY999999999999999, Inbound: JPY9999999999999999,
78	Net Proceeds (without date)	34B		JPY0,	Outbound: JPY9999999999999998

Data Dict#	Data Element	Field Tag	Sub -field	Min Value	Max Value
82	Reporting Detail	23P		LSM LLR LLS LCN LBE	LSM LLR LLS LCN LBE
83	Sender to Receiver Information	72			Test with 5 lines by 35 characters.



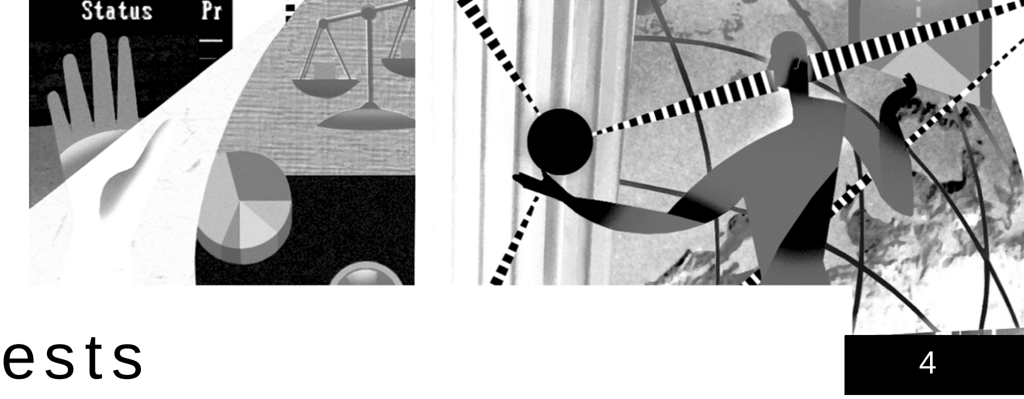
Test 27 — Contract Note Trading

1. The broker must send two *CNA* messages. One must contain fields with the minimum values allowed, and the other must contain fields with the maximum values allowed.
2. The broker must check each field value for accuracy.
3. Send two *CNA* messages, one in which you set the quantity to the maximum value and the price to `1`, and the second in which you set the quantity to `1` and the price to the maximum value.

Here is the data table for Test 27

Data Dict#	Data Element	Field Tag	Sub -field	Min Value	Max Value
23	Quantity of Financial Instrument	35A		SHS0,01	Outbound: SHS999999999999999,98 Inbound: SHS99999999999999,99
27	Transaction Price	33T2		.00000001	Outbound: 99999999999999,98 Inbound: 99999999999999,99
27a	Lot Size	TF15		1	9999
32	Special Concessions	33S		JPY0,	Outbound:JPY99999999999999,98 Inbound: JPY99999999999999,99
35	Accrued Interest (Added) (Subtracted)	34G 34H		0000JPY0, 0000JPY0	9999JPY99999999999999999 9999JPY99999999999999999
40	Financial Instrument Attribute				
	Maturity Date	23F	MDD	19800101	20491231
	Yield	23F	YLD/CALL		Outbound: 9999999999.999998 Inbound: 9999999999,999999
			YLD/CURRENT		Outbound: 9999999999.999998 Inbound: 9999999999,999999
			YLD/FUTURE		Outbound: 9999999999.999998 Inbound: 9999999999,999999
			YLD/MATURE		Outbound: 9999999999.999998 Inbound: 9999999999,999999
			YLD/REP		Outbound: 9999999999.999998 Inbound: 9999999999,999999
	Coupon Rate	23F	CPN	0,	Outbound: 9999999999.999998 Inbound: 9999999999,999999
	Issuer	23F	ISR		30 characters
	Call Date	23F	CLD	19800101	20491231
	Call Price	23F	CLP	JPY0,	JPY99999999,99999999
	Call Type	23F	CLT		17 characters
	Dated Date	23F	DD	19800101	20491231
	Odd First Coupon Date	23F	OFCD	19800101	20491231
	Book Entry Only	23F	BE	Y or N	Y or N
	Alternative Minimum Tax	23F	AMTX	Y or N	Y or N
	Federal Tax	23F	FTX		Y or N
	Rating	23F	RT		Outbound: AAAAAAAA Inbound: MOODY AAAAAAAA Outbound: BBBBBBBB Inbound: SP BBBBBBBB
	Amortized Factor	23F	FCT	0,	99999999,99999999
	Original Face Amount	23F	ORG	JPY0,	Outbound: PY99999999999999.98 Inbound: 99999999999999,99
	Current Face Value	23F	CFV	0,	Outbound: 99999999999999.98 Inbound: 99999999999999,99
42	Override Instructions	23O			
43	Party Type	23K		IMGR	IMGR
45	Party Identification	80J			
45a	Party’s Reference to the Transaction	TF14			

Data Dict#	Data Element	Field Tag	Sub -field	Min Value	Max Value
46	Further Info for Party Identified Alert Access Code Alert Country Code Alert Method Type Alert Security Type Alert Delivery Name	72B 72B 72B 72B 72B	ALAC ALCC ALMT ALSC ALDN		
95	Settlement Instruction Field Code	TF10			
96	Settlement Instruction Field Value	TF11			
42	Override Instructions	23O			
43	Party Type	23K		EXEC	EXEC
45	Party Identification	80J			
45a	Party’s Reference to the Transaction	TF14			
46	Further Info for Party Identified Account Reference Alert Country Code Alert Method Type Alert Security Type Alert Delivery Name	72B 72B 72B 72B 72B	ACCTREF ALCC ALMT ALSC ALDN		
43	Party Type	23K		FUND	FUND
46	Further Info for Party Identified Party Name	72B	NAM		
58	Deal Amount	32M		JPY0,	Outbound: JPY99999999999999,98 Inbound: SHS99999999999999,99
61	Charge/Tax Type	71B1		BROK	BROK
64	Charge/Tax Amount	71B3		JPY0,	Outbound: JPY999999999999999, Inbound: JPY9999999999999999,
61	Charge/Tax Type	71B1		FEES	FEES
64	Charge/Tax Amount	71B3		JPY0,	Outbound: JPY999999999999999, Inbound: JPY9999999999999999,
61	Charge/Tax Type	71B1		MISC	MISC
64	Charge/Tax Amount	71B3		JPY0,	Outbound: JPY999999999999999, Inbound: JPY9999999999999999,
61	Charge/Tax Type	71B1		TTAX	TTAX
64	Charge/Tax Amount	71B3		JPY0,	Outbound: JPY999999999999999, Inbound: JPY9999999999999999,
78	Net Proceeds (without date)	34B		JPY0,	Outbound: JPY9999999999999998
82	Reporting Detail	23P		LSM LLR LLS LCN LBE	LSM LLR LLS LCN LBE
83	Sender to Receiver Information	72			Test with 5 lines by 35 characters.



4: Other Tests

This chapter outlines miscellaneous other tests. It contains the following sections:

Test 28 — Verification of Entire Character Set in Any Free Formatted Text Field

This test verifies inclusion of the entire character set in free formatted text fields (field 72 `Sender to Receiver Information`, field TF11 `Settlement Instruction Field Value`, field 72A `Status/Response Narrative`, and field 72C `Reason Narrative`). Use this test to validate.

1. The following characters or groups of characters:

```
Example  ~ ! @ $ % & * ( ) _ + | ` - = \ { } ″ ′ < > ? ,
         /  . : ; [ ] ^ # £
```

2. A message with a # (hash sign) which is not in the first position and a message with a # (hash sign) which is in the first position.
3. A message including the British Pound symbol (£), octal 243.

Example: Perform a test by sending an *AE New* block trade with fields 72 (`Sender to Receiver Information`) and TF11(`Settlement Instruction Field Value`) containing the following characters:

```
Example  ~ ! @ $ % & * ( ) _ + | ` - = \ { } ″ ′ < > ? ,
         /  . : ; [ ] ^ # £
```

See the *OASYS Global Direct MT511 Messaging Specification* for trades containing these fields.

Test 29 — Verification of Send/Receive of Messages to/from Multiple Institutions

Thomson ESG must verify that brokers using multiple acronyms can send and receive messages for multiple broker acronyms. Thomson ESG will set up this testing with four institutions, two block level and two confirm level.

Test 30 — Message Crossing

Any status message (*Valid*, *Reject*, *Affirm*) can arrive after a counterparty has sent a *Cancel*, even though the *Cancel* message supersedes all the status messages. This applies to all sub-message types (*AE*, *TA*, *CNA*, *CNB*) and may also be the case for *Cancel* messages of child messages. For example, your application needs to handle these scenarios:

Scenario A:

1. The broker sends a *CNB* message after receiving a *TA* message from the institution.
2. After sending the *TA* message, the institution sends a *TA Cancel*.
3. Both parties must recognize the status of the trade as a *TA Cancel*, and the broker must send a *TA Canceled*.

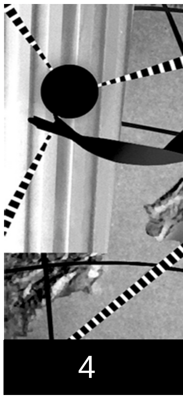
Scenario B:

1. The institution sends a *CNA Affirm* after receiving a *CNA* message from the broker.
2. After sending the *CNA* message, the broker sends a *CNA Cancel*.
3. Both parties must recognize the status of the trade as a *CNA Cancel*, and the institution must send a *CNA Canceled*.

Test 31 — Table Lookup Tests

These test trades are necessary to ensure that the system correctly performs the following table lookups:

1. Currency Table, field 33T2 `Transaction Price`
2. Agency Table, field 83R `Agency/Principal/Cross Trade Indicator`
3. Buy Sell Table, field 23C `Bought/Sold Indicator`
4. Charge Type Table, field 71B1 `Charge/Tax Type`
5. Party Type Table, fields 23K `Party Type` and 80J `Party Identification`
6. Quantity Type Table, field 35A `Quantity of Financial Instrument`
7. Rating Type Table, field 23F/RT/ `Financial Instrument Attribute Rating`
8. Reference Type Table, field 20A `Type of Sub-Message Reference`
9. Report Detail Table, field 23P `Reporting Detail`
10. Yield Code Table, field 23F/YLD/ `Financial Instrument Attribute Yield`



See the *OASYS Global Direct MT511 Messaging Specification* for trades containing these fields.

Test 32— Receipt of Invalid Messages

You must test your application's handling of multiple received *Invalid* messages. This includes identification of *Invalid* messages and any automated or manual procedures that require execution. The test must include at least 100 messages that specify a receiver that is unknown to the host. You will have to modify your local tables to add a dummy receiver so that the messages pass local validation but fail validation at the OG host. Since the host does not have the receiver acronym, it will generate *Invalid* messages back to you.

Test 33— Volume Test

You must validate the ability of your application to handle large volumes of messages. You and ESG will agree on the precise number of messages.

Test 34— ALERT Tests

Your system must complete the following tests, applicable to ALERT lookup:

1. Entering ALERT keys - CNA (broker)
2. Receiving ALERT keys and DIs - CNA (broker and institution)
3. Entering manual DIs - AE (broker)
4. Receiving manual DIs - AE (institution)
5. Entering manual DIs - TA (institution)
6. Receiving manual DIs - TA (broker)
7. Entering IDI ALERT keys (TA institution) with BIA numbers - CNB (broker)
8. Receiving IDI ALERT keys with BIA numbers - CNB (institution)

Test 35 — ISIN Tests

Your system must complete the following tests, applicable to ISIN lookup:

Brokers Only:

1. Enter type, code, and name to verify it is performed correctly.
2. Verify that the broker can process the ISIN information contained in the *ValueAdd* message.

Institutions Only:

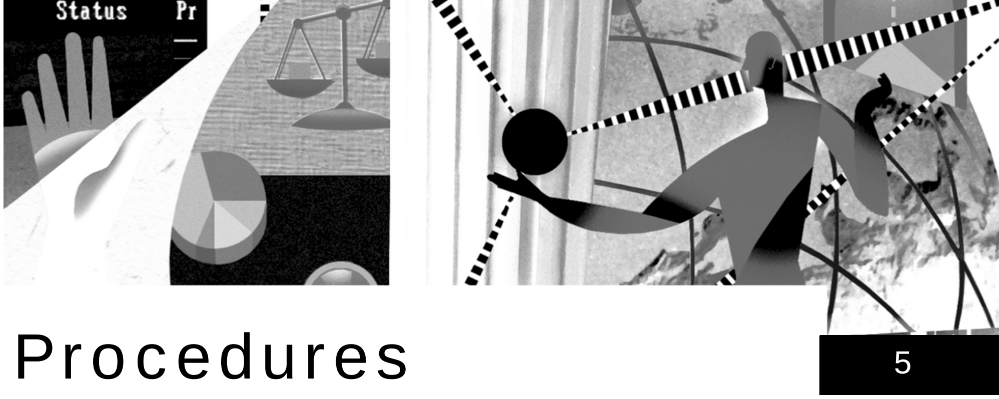
1. Extract the desired code and name from the correct 35B group.
2. Extract the ISIN code.

Test 36— Message Flow Testing

Additional message flow testing on the test system is necessary before your application can run in the OASYS Global *Direct* production environment. You will reuse the message flow tests to test the application in the test environment. These tests will also incorporate the ALERT and ISIN lookup tests.

Test 37— Receipt of Duplicates

Sometimes, due to communications challenges, the Thomson host may send duplicate trade-related messages. The host was designed to ensure that messages sent to your system are received and processed, and if there is no verification of whether a message has been received, it is sent and processed again. Therefore, you must test your application to ensure that it can properly handle duplicate trade messages from the Thomson host.



5: Support Procedures

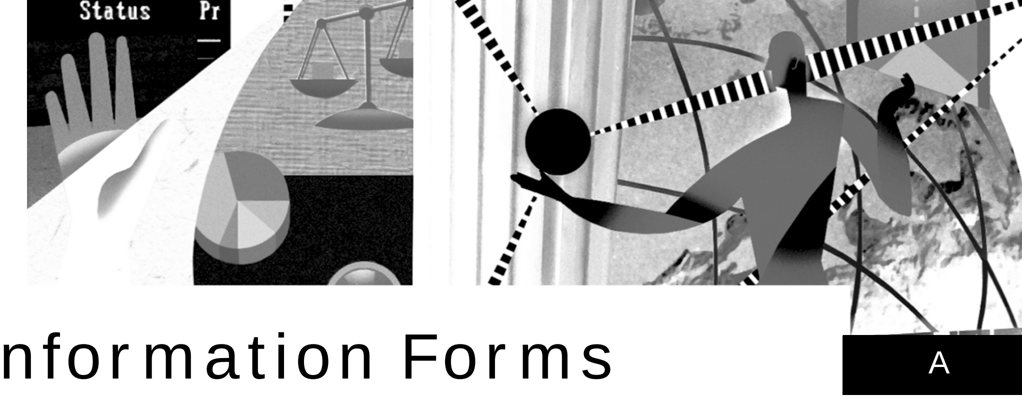
Verify Client Procedures for Handling Support Internally

You must demonstrate that your organization has procedures in place for:

1. Client escalation
2. Application documentation/hand over from client developers to client operations

Primary Contact and Hours of Availability for Support

You and ESG must furnish this information to each other as indicated in Appendix A, “Client Information Forms.”



A: Client Information Forms

This appendix includes copies of client information forms used during conformance testing. It contains the following forms:

Client Information Sheet

Client:	Client Type (BKRO, INSA, INSB, BKRA):

Address:

Production	Disaster Recovery

Contacts:

Client Contacts	Name	Telephone	E-Mail	Fax
Project Manager				
Programmer				
Network Manager				
Network Tech.				
Business Manager				
IT Business Manager				
Thomson ESG Contacts	Name	Telephone	E-Mail	Fax
Project Manager				
TF NetOps				
Account Manager				

Live Date:

OASYS GlobalDirect Contract Signed:	
Live Date: (Plan)	

APIs:

API	Platform
MT511	
MDS	
Client Application Name	

OASYS Global *Direct*/MINT Configuration:

	Production		Development	
	Inbound	Outbound	Inbound	Outbound
CommLink				
IP Address				
Port				
Sender Alias				
Receiver Alias				
OG Account No.				

Client Host Information**:**

	Production	Disaster Recovery	Development
Client IP Address			
Client SubNet			
Host Name			
Platform			

TF Router:

	Production		Disaster Recovery	
	Live	Backup	Live	Backup
Cisco Type				
Location				
E0				
S0				
Line Type				
Line Bandwidth				
Line Provider				
Circuit				

Client Escalation Process	
Network Diagram	
Specific SLA/SLO	

Notes:

Documentation Control:

Date	Version	Description	Author

Client Profile Sheet

Constraints:

This page will define your constraints that will affect testing.

Test Check List

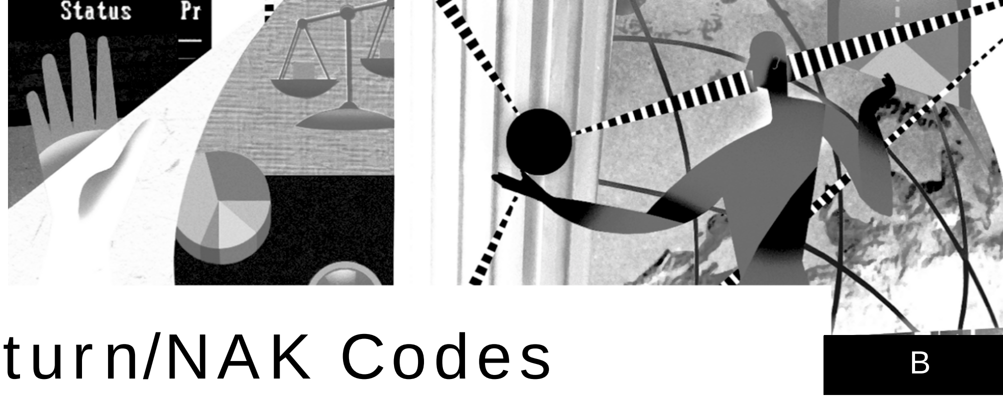
Test	**Pass**	**Fail**	**Not Applicable**	**Trade Reference Number (TRN)**
Broker Message Flow Testing				
Block Trading				
1. Block Trade Flow				
2. *AE* and *CNB Amend*				
3. *AE New* and *Amend Reject* and *AE Cancel*				
4. *AE New* and *Amend Reject* and *AE Cancel*				
5. *TA* and *CN* Version Increments				
6. *TA Amend* and *AE Cancel*				
7. Message Crossing				
8. Message Crossing				
Contract Note Trading				
9. *CNA Amend*				
10. *CNA Amend* and *CNA Cancel*				
11. *CNA New Affirm* and *CNA Cancel*				
12. Send *CNA Cancel* Before Receiving *CNA New ValueAdd*				
13. Send *CNA Cancel* After Receiving *CNA New ValueAdd*				

Test	**Pass**	**Fail**	**Not Applicable**	**Trade Reference Number (TRN)**
Institution Message Flow Testing				
Block Level Institution				
14. Block Trade Flow				
15. *AE Amend* with *CNB Reject* and *AE Cancel*				
16. *AE Amend* and *AE Cancel*				
17. *TA* and *CNB Amend*				
18. *TA Reject*, *TA Cancel*, and *AE Cancel*				
19. Send *TA Cancel* Upon Receiving *TA New ValueAdd*				
20. Send *TA Cancel* After Receiving *TA New ValueAdd*				
Confirm Level Institution				
21. *CNA Amend*				
22. *CNA Amend* and *CNA Cancel*				
23. *CNA New Affirm* and *CNA Cancel*				

Test	**Pass**	**Fail**	**Not Applicable**	**Trade Reference Number (TRN)**
Other Tests				
24. Advice of Execution				
25. Trade Allocation				
26. Contract Note Block Trading				
27. Contract Note Trading				
28. Verification of Entire Character Set in Any Free Formatted Text Field				
29 Verification of Send/Receive of Messages to/from Multiple Institutions				
30. Message Crossing				
31. Table Lookup Tests				
32. Receipt of Invalid Messages				
33. Volume Test				
34. ALERT Tests				
35. ISIN Tests				
36. Message Flow Testing				
37. Receipt of Duplicates				

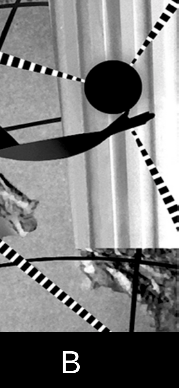
Communications Checklist Sheet

Test	**Pass/Fail (P or F)**	**Description**
Test physical connectivity		Router failover test at client site.
Test connectivity with the MDS		Client must be able to recover from an MDS disconnect at least five times without manual intervention at their end. Client must be able to send/receive information from the MDS after a system failover.
Test ISDN line		Client must be able to connect and communicate through the ISDN line on the MDS. Client must test this three times. A phone call will notify you that the test has begun. ESG will initiate a modem failover by disconnecting the leased line at the router and have you use the ISDN line to connect to the MDS.
Verify automated backup process		Client must make sure the process takes place without manual intervention.
Verify manual backup procedure for major outage		Client must make sure the process works whenever necessary.
Verify open vision Event Manager is working properly.		Production services will monitor this at various points during certification testing.



B: MOA Return/NAK Codes

This chapter includes information about MOA error codes and NAK codes. It contains the following sections:



MOA Return Codes

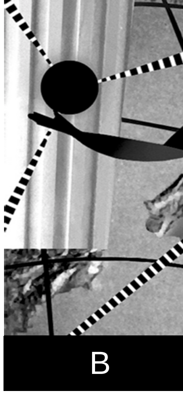
When returning to the application program, MOA provides a return code. If the required operation was executed successfully, the MOA returns 0 (MOA_SUCCESS). If an error occurred, one of the other return codes will indicate the type of error that occurred.

> ***Note!*** In some cases, the return code is supplemented with a FAILURE_REASON field. This field supplies the specific reason regarding the error condition that occurred.

The following table lists the MOA return codes in alphabetical order:

Return Code	Number
MOA_APPL_ERROR	13
MOA_BAD_CONFIG_FILE	8
MOA_BAD_MSG_FORMAT	10
MOA_BAD_SEQ_NUM	9
MOA_COMM_PROBLEM	5
MOA_INV_PARAMS	1
MOA_MAC_FAIL	15
MOA_MAC_REDUNDANT	16
MOA_MSG_NAK	2
MOA_MSG_TOO_LONG	4
MOA_NO_CONFIG_FILE	7
MOA_NO_MAC	14
MOA_NO_MEMORY	6
MOA_NO_MSG	3
MOA_OPEN_ERROR	11
MOA_RECV_TIMEOUT	12
MOA_REPLY_TOO_LONG	17
MOA_SUCCESS	0

During MOA operations, MOA returns the return code number. The MOA return codes are listed on the following pages in numerical order, along with an explanation of the return code, possible supplemental failure reasons (where relevant), and action to be taken (where relevant).



Return Code 0 (MOA_SUCCESS)

The service has been successfully executed.

Return Code 1 (MOA_INV_PARAMS)

An invalid parameter was supplied to the service accessed, or there is a problem in the configuration file. The failure reason indicates which parameter is invalid.

Supplemental Failure Reasons

Reason Name	Number	Explanation	Action Needed
`MOA_COMSERV`	1	Invalid ComServ name was supplied in a non-first MOA service call.	Check the application program for possible corruption of the `moa_work_area`. Check that the correct ComServ name was supplied in the input parameters.
`MOA_SENDER`	4	Mandatory sender field is not present in input parameters structure.	Correct the application program supplying this parameter.
`MOA_RECEIVER`	5	Mandatory receiver field is not present in input parameters structure.	Correct the application program supplying this parameter.
`MOA_NET_COMSERV`	17	Mandatory network Comserv field is not present in the input parameters structure (SIC/SECOM only)	Correct the application program supplying this parameter.
`MOA_NET_MT`	18	Mandatory message type field is not present in input parameters structure.	Correct the application program supplying this parameter.
`MOA_USER_ACT`	21	When using the user/ activity option, a user is supplied without a corresponding activity.	Correct the application program supplying the user/activity pair.
`MOA_INV_AUTHEN_KEY`	22	An invalid authentication key was supplied	Correct the application program supplying this parameter.
`MOA_BIN_AND_AUTH_INVALID`	23	Forbidden combination in the configuration file: `authen_activated = y` and `binary_data = y`	Change the parameter in the configuration file.

Return Code 2 (MOA_MSG_NAK)

The message has been NAKed (`MOA_send_msg` or `MOA_send_stp_msg` call) or the retrieved message was NAKed by MOA due to incorrect sequence number or authentication error (`MOA_retrieve_msg` or `MOA_retrieve_long_msg` or `MOA_rtrv_no_ack` or `MOA_rtrv_long_no_ack` call). The NAK code explains the reason for the error (see the section titled “NAK Codes” for more information.

Return Code 3 (MOA_NO_MSG)

No message was retrieved.

Return Code 4 (MOA_MSG_TOO_LONG)

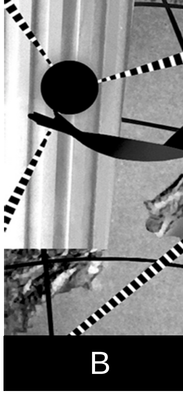
The retrieved message exceeds the length of the message text area. (`MOA_retrieve_msg` or `MOA_rtrv_no_ack` call).

Return Code 5 (MOA_COMM_PROBLEM)

An error occurred when executing a communication related function. Check communication (TCP/IP) definition. Check application program for possible corruption of `moa_work_area`. Check that the communication link, in MINT, is open.

Supplemental Failure Reasons

Reason Name	Number	Explanation
MOA_SOCKET_ALLOC_ERROR	1	Error occurred while allocating a socket in opening the communication link.
MOA_SOCKET_CONNECT_ERROR	2	Error occurred when connecting socket to port when opening the communication link.
MOA_SOCKET_SHUTDOWN_ERROR	3	Error occurred when executing socket shutdown.
MOA_SOCKET_CLOSE_ERROR	4	Error occurred when executing socket net_close.
MOA_LOGIN_SEND_ERROR	5	Error occurred when sending a login message to MINT.
MOA_LOGOUT_SEND_ERROR	6	Error occurred when sending a logout message to MINT.
MOA_MSG_SEND_ERROR	7	Error occurred when sending a message to MINT.
MOA_REQUEST_SEND_ERROR	8	Error occurred when sending a retrieve request to MINT.
MOA_ACK_SEND_ERROR	9	Error occurred when sending a message acknowledgement to MINT.
MOA_LOGIN_RECEIVE_ERROR	10	Error occurred when attempting to receive login response from MINT.
MOA_LOGOUT_RECEIVE_ERROR	11	Error occurred when attempting to receive logout response from MINT.
MOA_ACK_RECEIVE_ERROR	12	Error occurred when attempting to receive a message acknowledgement from MINT.
MOA_RTRV_RECEIVE_ERROR	13	Error occurred when attempting to receive a message from MINT.
MOA_INVALID_DATA	20	Invalid data is present in a response received from MINT.



Return Code 6 (MOA_NO_MEMORY)

Not enough memory for allocations.

Return Code 7 (MOA_NO_CONFIG_FILE)

A configuration file has not been setup for the ComServ.

Return Code 8 (MOA_BAD_CONFIG_FILE)

An error occurred when accessing the ComServ's configuration file. The format of the configuration file is erroneous.

Supplemental Failure Reasons

Reason Name	Number	Explanation	Action Needed
`MOA_READ_ERROR`	1	Read error occurred when attempting to read the configuration file.	Check the ComServ's configuration file for data corruption. If the file is corrupted, activate the `MOA_set` program (with the a-b option) taking care to update the session and sequence numbers with the correct values (taken from MINT).
`MOA_WRITE_ERROR`	2	Write error occurred when attempting to update the configuration file.	Check your application program for possible corruption of the `moa_work_area`. If this is the case, take actions described above.

Return Code 9 (MOA_BAD_SEQ_NUM)

A retrieved message’s sequence number was not as expected.

Return Code 10 (MOA_BAD_MSG_FORMAT)

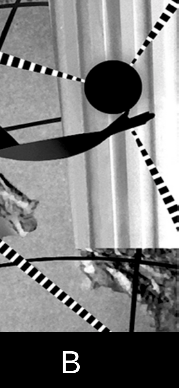
A retrieved message contained a format error.

Supplemental Failure Reasons

Reason Name	Number	Explanation	Action Needed
`MOA_INVALID_SEP`	1	Invalid line separator is received in header of a retrieved message.	Check application program for possible of `moa_work_area`. Check that the separator parameter in the ComServ’s configuration file is the same as that defined in MINT.

Return Code 11 (MOA_OPEN_ERROR)

Negative response for a login request was received from MINT, due to incorrect session number or unknown status in the login response was received from MINT.



Return Code 12 (MOA_RECV_TIMEOUT)

A response was not received from MINT within the timeout period stipulated in the ComServ's configuration file (`MOA_open` or `MOA_close` call), or an ACK/NAK message was not received within the timeout period (`MOA_send_msg` call), or a message was not retrieved within the timeout period (`MOA_retrieve_msg` or `MOA_retrieve_long_msg` or `MOA_rtrv_no_ack` or `MOA_rtrv_long_no_ack` call).

Check the MINT log for reporting of possible errors. Check application program for possible corruption of `moa_work_area`. Check communication (TCP/IP) definition. Check that the communication link, in MINT, is open.

Supplemental Failure Reasons

Reason Name	Number	Explanation
MOA_LOGIN_TIMEOUT	1	Login response not received from MINT.
MOA_LOGOUT_TIMEOUT	2	Logout response not received from MINT.
MOA_ACK_TIMEOUT	3	Acknowledgement message (for a sent message) not received from MINT.
MOA_RTRV_TIMEOUT	4	Message not received from MINT in response to a retrieve request.

Return Code 13 (MOA_APPL_ERROR)

The receiving application could not store or handle the received message, and therefore called the service `MOA_send_nak`.

Return Code 14 (MOA_NO_MAC)

Used internally by MOA routines. This return code is not returned to the calling application.

Return Code 15 (MOA_MAC_FAIL)

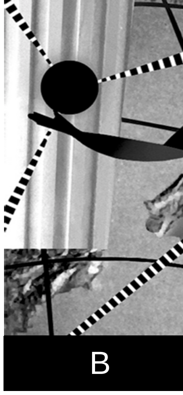
Used internally by MOA routines. This return code is not returned to the calling application.

Return Code 16 (MOA_MAC_REDUNDANT)

Used internally by MOA routines. This return code is not returned to the calling application.

Return Code 17 (MOA_REPLY_TOO_LONG)

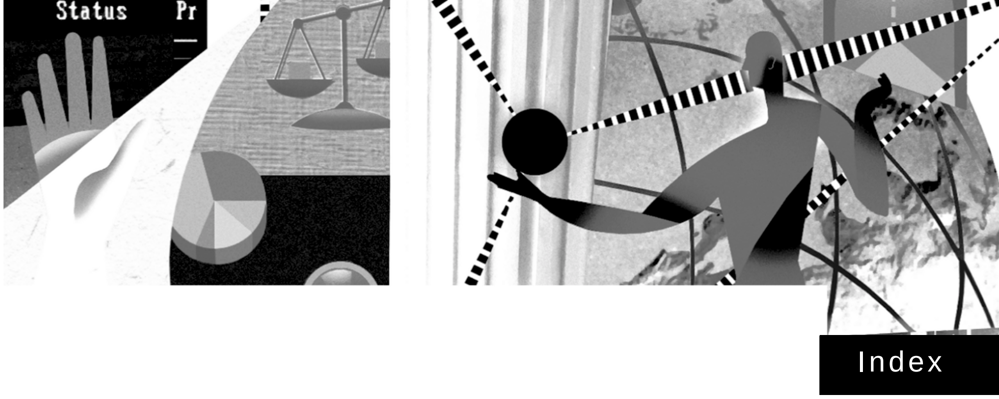
The reply message exceeds the maximum allowed message length [32000 characters] (`MOA_stp_send_reply` call) or the reply message exceeds the length of the reply message text area (`MOA_stp_send_msg` call).



NAK Codes

The following is a list of NAK codes which may be returned by MOA. These codes are returned as strings and so must be compared with `strcmp`.

NAK Code	Meaning
`MOA_NAK_APPL`	NAK sent by the application due to error detected in retrieved messages.
`MOA_NAK_SEQUENCE`	Bad sequence number. Sequencing error.
`MOA_NAK_FORMAT`	Incorrect message format.
`MOA_NAK_NO_RCVR`	No receiver was specified when mandatory for a format type.
`MOA_NAK_SRVR`	Straight Through Processing: Destination NAKed query message.
`MOA_NAK_NO_SRVR`	Straight Through Processing: Destination not available to receive query message.
`MOA_NAK_INV_RCVR`	Receiver unknown.
`MOA_NAK_NO_ROUTE`	The receiver cannot be reached or is disabled. Cannot determine routing for message.
`MOA_NAK_MAC_FAIL`	Authentication error.
`MOA_NAK_NO_MSG`	No message present in call. (i.e., message length was 0 or message pointer was NULL.)
`MOA_NAK_NO_MAC`	MAC not present when expected.
`MOA_NAK_MAC_REDUNDANT`	MAC present when not expected.
`MOA_NAK_INV_RQST`	Invalid request. Straight Through Processing functionality is not allowed.



Index