

Johannesburg Stock Exchange

Post-trade Services

JSE Services Documentation Volume PT02 – Post-trade EMAPI Clearing

Document Version	2.8.6
RTC Version	1.34.0
Release Date	03 December 2020
Number of Pages	266 (Including Cover Page)

1 DOCUMENT CONTROL

1.1 Table of Contents

1	DOCUMENT CONTROL	2
1.1	Table of Contents	2
1.2	Document Information	5
1.3	Revision History.....	5
1.4	About this Document	9
1.5	Intended Audience.....	10
1.6	Typographical Conventions	10
1.7	Related Documents	11
1.8	Contact Details	12
1.9	Definitions, Acronyms and Abbreviations.....	13
2	OVERVIEW	14
2.1	Functional and Integration Overview	14
2.2	RTC Lifecycle	16
3	PARTICIPANT STRUCTURE	19
4	RTC ACCOUNT SETUP	20
4.1	Position Accounts	21
4.1.1	House Accounts.....	21
4.1.2	Member Client Accounts.....	22
4.2	Collateral Accounts.....	22
5	BROADCAST FLOWS.....	23
6	REFERENCE DATA	25
6.1	Overview.....	25
6.2	Reference Data in RTC	26
6.2.1	Reference Data Flow	26
6.2.2	Summary of Reference Data	26
6.3	Reference Data Maintained by JSE	28
6.3.1	Clearing Member, Trading Member and Member Branch	28
6.3.2	Clearing Member Link.....	29
6.3.3	Clearing House	29
6.3.4	Information Vendor	29
6.3.5	Tripartite Agreement	30
6.3.6	Cash Account.....	30
6.3.7	Market Structure	30
6.3.8	Calendar	32
6.3.9	Currencies.....	33
6.3.10	Country	33
6.3.11	Risk Node	34
6.3.12	Class and Series Spread Group	34
6.3.13	Subscription Groups	34
6.3.14	Corporate Actions	36
6.3.15	Curves.....	36
6.3.16	Pricing and Reference Instruments (Rates and curves)	36
6.3.17	Other Reference Data Setup by JSE	37
6.3.18	Interface Users and Role-based Permissions	37
6.4	Reference Data Maintained by Members.....	38
6.4.1	Member Client.....	38
6.4.2	Add Client sequence diagram.....	40
6.4.3	Cash Account.....	40
6.4.4	Position Account	41
6.4.5	Maintenance (add/update/) of Reference Data	41

7	RISK MANAGEMENT	43
7.1	Risk Nodes	43
7.2	Risk Parameters Maintained by JSE	44
7.3	Risk Parameters Maintained by Members	44
7.3.1	Setting Risk Limits	44
7.3.2	Setting Additional Margin Percentage	45
7.4	Receiving and Interpreting Risk Data	45
7.4.1	Risk Node Event values.....	45
7.4.2	Notional Values.....	46
7.4.3	Collateral Position Values	47
7.4.4	Risk arrays	48
8	TRADE RECEIPT FROM TRADING SYSTEM.....	49
8.1.1	Trade Cancellation.....	50
8.1.2	Trade Cancellation Price Adjust	51
9	TRADE AND POSITION MANAGEMENT	52
9.1	Trade and Position Management	52
9.2	Deal Management Sequence Diagrams	56
9.2.1	Allocate Trade.....	56
9.2.2	Correct Allocation Error	57
9.2.3	Correct Principal	58
9.2.4	Modify Trade Sub Account	59
9.2.5	Accumulate Trade.....	60
9.2.6	Assign Trade.....	61
9.2.7	Tripartite Allocation	62
9.3	Commission Management.....	63
9.3.1	Handling of commission scenarios and reference fields	63
9.4	Trade and Position Admin Functions	66
9.5	Account Position and Account Trade Events Clarification	66
9.6	Removal of Account Position	70
9.7	Manual upload of trades directly to RTC system	72
10	END-OF-DAY PROCESSING	73
10.1	Normal Processing	73
10.2	Re-Run	73
10.3	Intraday Margin Call	73
10.4	End-of-Day Schedule	73
10.5	Start of Day schedule	77
10.6	Account Position Events and Sequence Numbers on Start of Day	78
11	COLLATERAL MANAGEMENT	79
11.1	Overview.....	79
11.2	Set Minimum Cash Limit	79
11.3	Collateral End-of-Day Process	80
11.3.1	Overview	80
11.3.2	Securities Collateral Processing.....	81
11.3.3	Foreign Exchange Collateral Processing	81
11.3.4	Settlement.....	82
11.3.5	EOD Collateral Sequence Diagram	82
11.4	Collateral Intraday Re-balancing	84
11.4.1	Overview	84
11.4.2	Withdrawals Processing	85
11.4.3	Intraday Collateral Re-balancing Sequence Diagram	85
12	CLEARING MEMBER BALANCING	87
12.1	CM Balancing 1	87
12.2	CM Balancing 2	87
12.3	CM Balancing Sequence Diagram	88

13	SETTLEMENT MANAGEMENT FUNCTIONS	89
13.1	Signs of values in daily account summary messages	90
14	MEMBER AND CLIENT TRANSFERS.....	91
	APPENDIX A - MESSAGE FORMATS	95
	Message: UpdateTradeReferenceReq.....	194
	Message: UpdateTradeReferenceRsp.....	195
	APPENDIX B - COMMISSIONS	253
	APPENDIX C – UPDATEABLE POSITION REASONS	266

1.2 Document Information

Drafted By	Post-trade Services
Status	FINAL
Version	2.8.6
Release Date	03 December 2020

1.3 Revision History

Date	Version	Description
11 May 2016	1.0	Initial draft created.
13 Jul 2016	1.1	<p>Note: Below is a summary of the changes introduced in RTC Release 1.7.0 and 1.8.0. Please see <i>EmapiTransactionsRevHistForMember</i> for details on all changes made to the API.</p> <ol style="list-style-type: none"> 1. Added Commissions Management functionality (see Section 9.2) 2. Updated numerous EMAPI field descriptions 3. Removed redundant messages and fields 4. Added several new fields to existing messages
16 Sep 2016	1.2	<p>Note: Below are the key changes introduced in RTC Release 1.9.0 and 1.10.0. Please see <i>EmapiTransactionsRevHistForMember</i> for details on all changes made to the API.</p> <ol style="list-style-type: none"> 1. Market Data broadcast flow is added for clearing and trading members on EMAPI. This also includes the related reference data objects on the reference data flow, for instance curve, surface and curve constituents. This allows clients to obtain this data via EMAPI. 2. Alpha code of Tradable Instrument changed to <i>updatable</i>. 3. ISIN of a tradable Instrument changed to <i>optional</i>. 4. In <i>Section 9.3 - Trade and Position Admin Functions</i>, the messages for finding the history of trades as well the messages to find positions in an account have been deprecated. The QueryTradesReq message provides similar functionality for searching trades; accounting information can be obtained by snapshot subscription to the account. 5. The SetWithdrawalConfirmStateForAllReq message to confirm/reject all withdrawals have been deprecated. <i>Section 11.4.2 Withdrawals Processing</i> updated. 6. Calendar Dates added to cater for public holidays.. 7. Clearing member can perform deal management and position sub account modification activities on behalf of their trading members. 8. The length of externalPositionAccountId in EMAPI object PositionAccount is now decreased from 100 to 10.

Date	Version	Description
		<p>9. Added two new optional fields; firmTradeId and tradeReportId to trades. The new fields are set on incoming trades to RTC,</p> <p>10. Added Trading User field to Position Sub Account Modification</p> <p>11. Added new message, GetRiskArrayReq to get risk arrays from RTC.</p>
28 Oct 2016	1.3	<p>Note: Below are the key changes introduced in RTC Release 1.11.0. Please see <i>EmaApiTransactionsRevHistForMember</i> for details on all changes made to the API.</p> <p>1. Updates to CM Balance 2, Daily Account Summary and Aggregated Summary Details messages to include different types of fees.</p> <p>2. Updated to include information on how clients should process reference data terminations.</p>
9 Dec 2016	1.4	<p>Note: Below are the key changes introduced in RTC Release 1.12.0. Please see <i>EmaApiTransactionsRevHistForMember</i> for details on all changes made to the API.</p> <p>1. The AtmVolatilityTermStructure message was removed as it is no longer published by RTC.</p> <p>2. RTC lifecycle tables in Section 2.2 were updated with revised timings and system functions.</p> <p>3. Included additional information on termination of TradableInstruments (see Section 6.3.7).</p> <p>4. Updated equation for Collateral After Haircut (see Section 7.4.3).</p> <p>5. Calculated Dividend Factors can now be queried with new QueryDividendPaymentFactorsReq message – See Section 10).</p>
20 Jan 2017	1.5	<p>Note: Below are the key changes introduced in RTC Release 1.13.0. Please see <i>EmaApiTransactionsRevHistForMember</i> for details on all changes made to the API.</p> <p>1. New MemberType(=INFORMATION_VENDOR) and ParticipantType(=INFORMATION_VENDOR) added to cater for data vendors wishing to subscribe to the Market Data Flow.</p> <p>2. The proposed changes to the ID Number, Passport Number and BDA Code of the Member message (as detailed in Appendix A) has been de-scoped.</p> <p>3. Minor updates to the Constant Group</p> <p>4. Update to description of message CdUpdateRtcMemberClientReq as some fields, e.g. allowed markets and passport were not updatable.</p> <p>5. Added note on <i>Section 8 – Trade receipt from Trading System</i> around the source of the tradingUserId field required for trade management messages discussed in Section 9.</p>
17 Feb 2017	1.6	<p>Note: Below are the key changes introduced in RTC Release 1.14.0 and Release 1.15.0. Please see <i>EmaApiTransactionsRevHistForMember</i> for details on all changes made to the API.</p>

Date	Version	Description
		<ol style="list-style-type: none"> Updated Section 6.4.1 <i>Member Client</i> to clarify that the <code>member</code> object is only published to the Reference Data Flow once the client is <i>linked</i> to the clearing member and <i>enabled</i>. Added note in Section 11.3.2 <i>Securities Collateral Processing</i> on processing of zero margin exposures. Added note in Section 11.3.3 <i>Foreign Exchange Collateral Processing</i> on processing where FX allowed is zero. Added table in Section - <i>RTC Account Setup</i> detailing how External Position Account IDs are defined. Added note highlighting that clients need to request authorisation from the JSE before they can create client sub accounts (See Section 6.4.3).
07 Mar 2017	1.7	<p>Note: Below are the key changes introduced in RTC Release 1.16.0. Please see <i>EmapTransactionsRevHistForMember</i> for details on all changes made to the API.</p> <ol style="list-style-type: none"> Updated TradableInstrument message to mark some field attributes as mandatory. Added new fields to the GetSettlementInstructionsReq and GetSettlementInstructionsRsp messages to allow bookmarking/pagination for these messages.
02 June 2017	1.8	<ol style="list-style-type: none"> Updated description for several fields in the AccountPositionEvent. IsBuy field added in the GiveUpEvent. Updated table for the Broadcast Flow in Section 5 to include the missing CommissionEvent Added clarification on <i>allowedmarkets</i> update in Section 6.4.1. Deal Management Sequence Diagrams added In Section 9.
30 June 2017	1.9	<ol style="list-style-type: none"> Minor updates to CollateralPositionValue Event Message. Minor updates to CdRequest General Message. Minor updates to RequestMessage General Message.
21 July 2017	2.0	<ol style="list-style-type: none"> Updated description for several fields in the DailyAccountSummaryDetailsEvent. Updated description for several fields in the FxDailyAccountSummaryDetails. Updated description for several fields in the AggregatedSummaryDetails Added updateId fields to the ExerciseOptionPositionReq, AbandonOptionPositionReq and AddCommissionReq messages Updated description for allowedMarkets field in the Member Object and CdAddRtcMemberClientReq message. Updated mandatory field for allowedMarkets to required in the CdAddRtcMemberClientReq message Added section 13.1 <i>Signs of values in daily account summary messages</i> to clarify signs in messages. Minor updates to section 9.1 <i>Trade and Position Management</i>.

Date	Version	Description
14 Oct 2017	2.1	Updates for RTC Release 1.20: <ol style="list-style-type: none"> 1. Field description changes for the TaxSnapshotSubscribeRsp message 2. Field description changes for the Currency message. 3. Field description changes for the RtcTradeExternalData message. 4. Field description changes for the TradeRes message.
22 Jan 2018	2.2	Updates for RTC Release 1.21: <ol style="list-style-type: none"> 1. Field description changes for the TaxEndSnapshot message. 2. Field description changes for the DailyAccountSummaryDetailsEvent. 3. Field description changes for the FxDailyAccountSummaryDetails internal message. 4. Field description changes for the AggregatedSummaryDetails internal message. 5. Field description changes for the QueryTradesReq message. 6. Field description changes for the PositionReason constant group. 7. Added section 10.5 <i>Start of Day schedule</i> 8. Added section 10.6 <i>Account Position Events and Sequence Numbers on Start of Day</i>
05 Mar 2018	2.3	Updates for RTC Release 1.22: <ol style="list-style-type: none"> 1. Field description changes for the AddCommissionReq message.
03 Apr 2018	2.4	Only version number updates
13 Jul 2018	2.5	Updates for RTC Release 1.24: <ol style="list-style-type: none"> 1. Field description changes for the RiskNodeEvent message. 2. Field description changes for the CommissionEvent message. 3. Field description changes for the CommissionEvent message. 4. Field description changes for the RtcTradeExternalData message. 5. Field description changes for the ModifyPositionSubAccountReq message. 6. Field description changes for the TradeRes message. 7. Added section 8.1.1 <i>Trade Cancellation</i> 8. Added section 8.1.2 <i>Trade Cancellation Price Adjust</i> 9. Added section 9.3.1 <i>Handling of commission scenarios and reference fields</i> 10. Added Appendix B
13 Aug 2018	2.6	Updates for RTC Release 1.25: <ol style="list-style-type: none"> 1. Correction on Section 5 <i>Broadcast Flows</i> to indicate that Market Data Event Flow is not Replay Supported. 2. Added Section 5.1 <i>Account Position and Account Trade Events Clarification</i> to clarify how the two events are used with different subscription types.
04 Sept 2018	2.7	Updates for RTC Release 1.26: <ol style="list-style-type: none"> 1. Field description changes for the TradableInstrument message. 2. Correction in Section 5.1 <i>Account Position and Account Trade Events</i>
25 Sept 2018	2.7.1	1. Added Section 9.6 <i>Removal of Account Position</i>
11 Oct 2018	2.7.2	1. Added Section 6.4.2 <i>Add Client sequence diagram</i>

Date	Version	Description
		2. Minor updates to Section 6.4.1 <i>Member Client</i>
15 NOV 2018	2.7.3	3. Field description changes for the <i>DailyAccountSummaryDetailsEvent</i> message
06 December 2018	2.7.4	1. Updated Section 9.6 Removal of Account Position.
21 December 2018	2.7.5	2. Updated Section 9.3 Commissions Management and APPENDIX B-Commissions
21 December 2018	2.7.6	1. Updated Section 9.5 <i>Account Position and Account Trade Events Clarification</i> 2. Updated Section 14 <i>Member and Client Transfers</i>
22 Feb 2019	2.7.7	RTC version update in Appendix A to 1.28.1
05 Jun 2019	2.7.8	1. RTC version update 1.29.0
27 August 2019	2.7.9	Updates for RTC Release 1.30.0: 1. Added updated Trade management message: <i>ModifyPositionSubAccountReq</i> 2. Added updated Reference Data Messages- External Message: <i>CdUpdateRtcMemberClientReq</i>
17 September 2019	2.8.0	1. Updated Section 14 <i>Member and Client Transfers</i> to clarify the identification of the active Clearing Member Link after a Trading Member transfer
29 November 2019	2.8.1	1. RTC version update 1.31.0
12 February 2020	2.8.2	1. Updated section 10.4 <i>End-of-Day Schedule</i> so that calculation of dividends happens before processing of Corporate Actions
07 August 2020	2.8.3	1. Added section 9.7 for Manual upload of trades
24 August 2020	2.8.4	1. RTC version 1.34.0 update, which includes version 1.33.0 2. Updated section 9.3 <i>Commission Management</i> , which has new functionality to accept commissions 1. Updated section 9.1 <i>Trade and Position Management</i> , which has new functionality to update trade references
17 November 2020	2.8.5	1. Added Appendix C <i>Updateable Position Reasons</i> to indicate <i>PositionReasons</i> linked to trades that can be updated using <i>UpdateTradeReferenceReq</i> message
<u>03 December 2020</u>	<u>2.8.6</u>	<u>2. Updated section 10.4 <i>End-of-Day Schedule</i> to clarify when events are published as a result of a Corporate Action event</u>

1.4 About this Document

This document describes the business/clearing or application aspects of the EMAPI protocol available to external participants (trading and clearing members). It relies on the common protocol aspects described in the related document *Volume PT01- Post-trade EMAPI Common*.

The purpose of this document and its related documents (See 1.7) is to serve as a description of the EMAPI protocol when implementing EMAPI client applications or backend systems to integrate with the JSE's real-time clearing (RTC) system.

1.5 Intended Audience

The information in this document is intended for software developers writing EMAPI interfaces to RTC.

1.6 Typographical Conventions

EMAPI messages or enumerations are shown in upper camel case using the `courier new` font and are hyperlinked to their detailed definitions in Appendix A. For example: [AllocateTradeReq](#) or [PositionAccountType](#).

EMAPI fields are shown using the `courier new` font in lower camel case. For example: `positionType` or `strike`.

Note: For ease of navigation of the document using the hyperlinks, please ensure you have the *Previous View (Alt + Left Arrow)* and *Next View (Alt + Right Arrow)* buttons enabled on the page navigation toolbar of Adobe Acrobat or the equivalent in other PDF viewers (if available).

1.7 Related Documents

Note: The documents in the table below are published on the ITaC website:
<https://www.jse.co.za/services/itac>

Name	Description
Volume PT00 – Post-trade Services Overview.pdf	Provides an overview of the deal management and clearing/settlement functionality available for the Equity Derivatives and Currency Derivatives markets.
Volume PT01 – Post-trade EMAPI Common.pdf	Describes the semantics and syntax of the common or session/admin EMAPI protocol messages.
Volume PT03 – Post-trade Margin Methodology Specifications.pdf	Describes in detail the JSE margin methodology including all calculations used in deriving margins.
Post-trade EMAPI TagWire	Describes the syntax of the TagWire encoding of EMAPI message body.
EmapiTransactionsForMember.html	The syntax of all EMAPI protocol messages in HTML format for trading and clearing members.
EmapiTransactionsForMember.xml	The XML definition of all EMAPI protocol messages for trading and clearing members.
EmapiTransactions.xsd	The XML schema that EmapiTransactionsForMember.xml conforms to.

Note: The IDP3 documents in the table below are published on the ITaC website:
<https://www.jse.co.za/services/itac>

Name	Description
Currency Derivatives Non-Live Data Products Specifications.pdf	Provide users with relevant and useful reference and statistical data on the day's trading activity on the currency derivatives market. The data is published on IDP.
Equity Derivatives Non-Live Data Products Specifications.pdf	Provide users with relevant and useful reference and statistical data on the day's trading activity on the equity derivatives market. The data is published on IDP.

1.8 Contact Details

JSE Limited One Exchange Square Gwen Lane, Sandown South Africa Tel: +27 11 520 7000 www.jse.co.za	Post Trade and Information Services ITAC Queries Email: CustomerSupport@jse.co.za
Clearing specifications disclaimer Disclaimer: All rights in this document vests in the JSE Limited ("JSE") and Cinnober Financial Technology AB (publ) ("Cinnober"). Please note that this document contains confidential and sensitive information of the JSE and Cinnober and as such should be treated as strictly confidential and proprietary and with the same degree of care with which you protect your own confidential information of like importance. This document must only be used by you for the purpose for which it is disclosed. Neither this document nor its contents may be disclosed to a third party, nor may it be copied, without the JSE's prior written consent. The JSE endeavours to ensure that the information in this document is correct and complete but do not, whether expressly, tacitly or implicitly, represent, warrant or in any way guarantee the accuracy or completeness of the information. The JSE, its officers and/or employees accept no liability for (or in respect of) any direct, indirect, incidental or consequential loss or damage of any kind or nature, howsoever arising, from the use of, or reliance on, this information.	

1.9 Definitions, Acronyms and Abbreviations

EMAPI	External Messaging API. EMAPI is the API used to integrate a client application or backend system with the RTC Clearing System.
External participant	A clearing member or trading member.
cCran	An administration front-end for the RTC clearing system.
cDew	A monitoring front-end for the RTC clearing system.
CM	A Clearing Member.
Clearing House	The JSE acts as the clearing house for JSE equity derivatives and currency derivatives.
Client	A client that connects to RTC Servers using the EMAPI protocol.
EOD	End of Day. An RTC system state.
IDP	Information Delivery Portal. An FTP site provided by the JSE for members to download reference data at end-of-day.
Member	A Clearing Member or Trading Member.
Member Client	A client of a trading member or clearing member.
Nostro Account	An account denominated in a foreign currency established through the local bank at a bank in the respective country of the currency desired.
RTC	Real-time Clearing. The JSE's implementation of the Cinnober TradeExpress™ clearing system.
Server	RTC server that supports the EMAPI protocol. For example, the TAX (Trading Application Multiplexer) server.
SDA	A Segregated Depository Account (SDA) held at Strate.
Strate	A Central Securities Depository (CSD). Strate provides electronic settlement of equities and bonds transactions concluded on the JSE. It also settles transactions in money market securities and offers collateral management services.
SWIFT	The Society for Worldwide Interbank Financial Telecommunication provides a network that enables financial institutions worldwide to send and receive information about financial transactions in a secure, standardised and reliable environment.
TM	A trading member.

2 OVERVIEW

The JSE Real Time Clearing (RTC) solution provides services related to clearing, settlement and deal management pertaining to the equity derivatives and currency derivatives markets¹.

Note: Please refer to [Volume PT00 – Post-trade Services Overview](#) (See 1.7) for an overview of the deal management, clearing and settlement functionality available for the Equity Derivatives and Currency Derivatives markets.

RTC provides the following functions to members via the EMAPI interface:

- Client and Account Loading
- Reference Data Subscription
- Broadcast or Event Flow Subscriptions (e.g. Risk Events, Account Events, etc.)
- Trade and Position Management
- Collateral Management
- Margining
- Clearing and Settlement
- Post-trade Risk Monitoring
- Corporate Actions and Transfers
- Commissions

This document provides guidance on how to use EMAPI to access the above services; it describes the semantics of the EMAPI protocol and the syntax of the *application* or *business/clearing*-related EMAPI messages and services available to members integrating with RTC.

Note: *Volume PT01 – Post-trade EMAPI Common* (See 1.7) describes the semantics of the EMAPI protocol and the syntax of the *common* or *admin/session*-related EMAPI messages and should be read prior to this document.

Note: The EMAPI message specifications are provided as separate HTML and XML files (See 1.7 for more details). The HTML specifications are also included in this document for convenience under Appendix A; where appropriate, links to these message definitions are provided in this document.

Note: At the time of writing this document, there are a few upcoming changes to the specifications that are currently under development. These changes will be published in a future update to the specifications. Please see full list of changes in Appendix A.

2.1 Functional and Integration Overview

The following diagram below depicts a functional overview of RTC and its integration with other systems and market participants.

¹ Future releases of RTC will support other derivatives markets as well as the equities market.

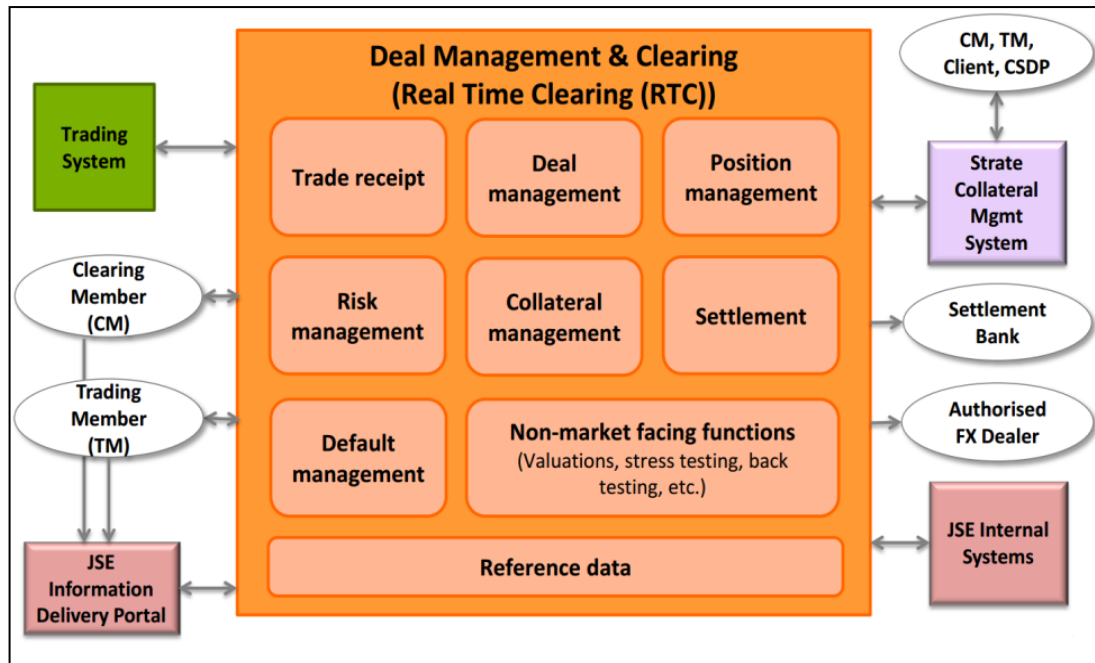


Figure 1 - RTC Functional Overview

The main RTC functional components include: Trade Receipt, Deal Management, Position Management, Risk Management, Collateral Management, Settlement, Default Management and Reference Data.

RTC has other internal, non-market facing components (e.g. for valuations and stress testing), but their functionality is not directly accessible via the EMAPI interface by external participants. However, some of the result of processing by these components are published to members via broadcast flows (See Section 5 for more details) or as end-of-day products on the JSE's information delivery portal (IDP) (See Section 6 for more details).

RTC integrates with the trading system to get trades as well other internal systems (e.g. CBS for billing information and MDS for reference data), Strate for collateral management and IDP for public reference data.

The Clearing Members (CM), Trading Members(TM), Settlement Banks and Authorised FX Dealers also interface with RTC - the CMs and TMs interface to RTC using EMAPI while the Settlement Banks and Authorised FX Dealers interface to RTC via JSE Integration Layer (not shown) using SWIFT messages.

2.2 RTC Lifecycle

During the course of a normal business day, RTC can be in the following states:

RTC State	Start Time	End Time	Description
OPEN	08:00	18:30	Open for new trades and trade management and resultant updates.
END_OF_TRADE_MANAGEMENT	18:30	18:30	Trade management is no longer allowed
END_OF_DAY	18:30	20:00	End of Day process started
POST_END_OF_DAY	20:00	08:00 (next day)	End of Day process completed

The following table describes the key clearing functions provided by RTC during a normal business day as well as other supporting functions such as reference data dissemination via the JSE's Information Delivery Portal (IDP) FTP site:

Start time	System Events	Functionality Available
POST_END_OF_DAY (20:00 – 8:00 next day):		
05:00	<ul style="list-style-type: none"> • RTC is available for logon • Reference data and transactional data published • Re-run of EOD from previous day (if required) • IDP end-of-day reference data published to FTP site. • (See 1.7 for details on IDP documentation) 	<ul style="list-style-type: none"> • EOD reference data available on IDP site • Reference data downloads • Client maintenance
OPEN (8:00 – 18:30) :		
08:00	<ul style="list-style-type: none"> • Business date is set • Trades from the trading system are received by RTC <p>Note: Refer to Trading Specifications for more details (See 1.7)</p>	<ul style="list-style-type: none"> • Reference data downloads • Client maintenance • Trade management • Commissions Management • Position and risk updates • New tradable instruments
11:00	<ul style="list-style-type: none"> • Collateral intraday re-balancing • Benchmark rates published 	
15:00	<ul style="list-style-type: none"> • Early valuations prices published 	<ul style="list-style-type: none"> • Reference data downloads
17:00	<ul style="list-style-type: none"> • No more on-book trades from the trading system are inserted, only trade management can be done 	<ul style="list-style-type: none"> • Client maintenance • Trade management • Commissions Management • Position and risk updates
END_OF_TRADE_MANAGEMENT (18:30)		
18:30	<ul style="list-style-type: none"> • Trade management state ends • Remove pending give-up instructions 	<ul style="list-style-type: none"> • Reference data downloads
END_OF_DAY (18:30 – 20:00):		

18:30	EOD procedures begin, including <ul style="list-style-type: none"> • Option and future expiration • Initial margin , variation margin • Collateral • Stress testing • CM balancing • Settlement • Member transfers • Corporate actions • etc. 	<ul style="list-style-type: none"> • Client loading • Reference data downloads • Position and risk updates
POST_END_OF_DAY (20:00 – 8:00 next day):		
20:00	<ul style="list-style-type: none"> • Client management 	<ul style="list-style-type: none"> • Client transfers
21:00	<ul style="list-style-type: none"> • IDP publishes EOD reference data 	<ul style="list-style-type: none"> • Reference data downloads • EOD reference data available on IDP site
22:00	<ul style="list-style-type: none"> • System maintenance 	<ul style="list-style-type: none"> • RTC offline

3 PARTICIPANT STRUCTURE

This section describes the objects that define the member/user structure used in the system.

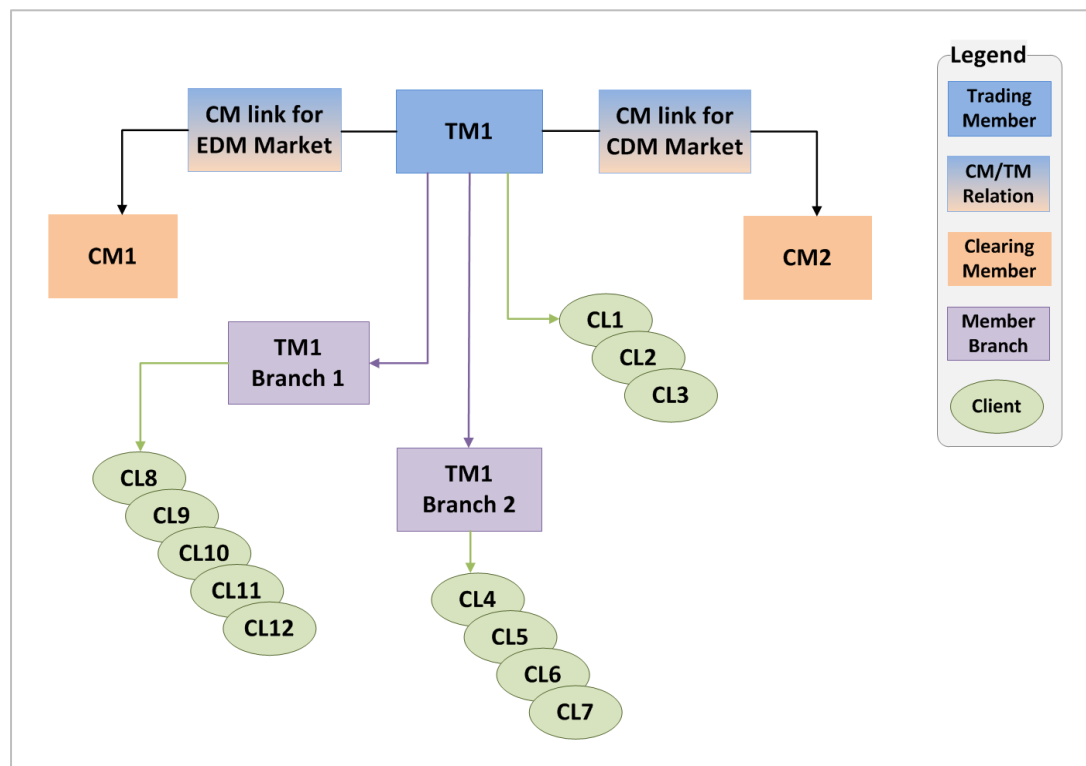


Figure 2 - Participant structure

Clearing members, trading members and branches are uploaded into RTC on a daily basis from the JSE Master Data System (MDS). This will be done last thing at end-of-day (EOD), except for tripartite agreements maintenance, which will take place intraday. Please note that RTC does not prevent any reference data updates intraday. The member data will be ready to use as soon as it has been loaded into RTC.

The member structure is built up by [Member](#) objects with different types in a tree structure. This includes Clearing Member (CM), Trading Member(TM), Member Branch and Client. These are all [Member](#) objects with different [MemberType](#) and [ParticipantUnitType](#) in the reference data.

A TM has one CM link for each Market that the TM is allowed to trade. The CM link points out the CM, and is defined in reference data from MDS. The TM may have a number of Branches. Both the TM and Branches may have a number of Clients. The branches inherit the CM/TM link defined for each market.

Clients, client main and sub accounts and TM/Branch additional house sub accounts and clients suspense accounts from the 99999 client suspense account are managed by the TM/branch (please see Section 4 for more details on account setup).

The member reference data exists only in one instance, separate from the risk structure. This means that if a Client attribute is changed, it will only need to be done once, not per Client cloned to different CMs (as per CM link) in the risk structure.

4 RTC ACCOUNT SETUP

Clearing members do not have position accounts in RTC. The [PositionAccounts](#) are created for trading members, member branches and member clients. Separate accounts are set up for each clearing member-trading member link.

The diagram below describes the account structure in RTC. The position accounts are shown in green in the diagram.

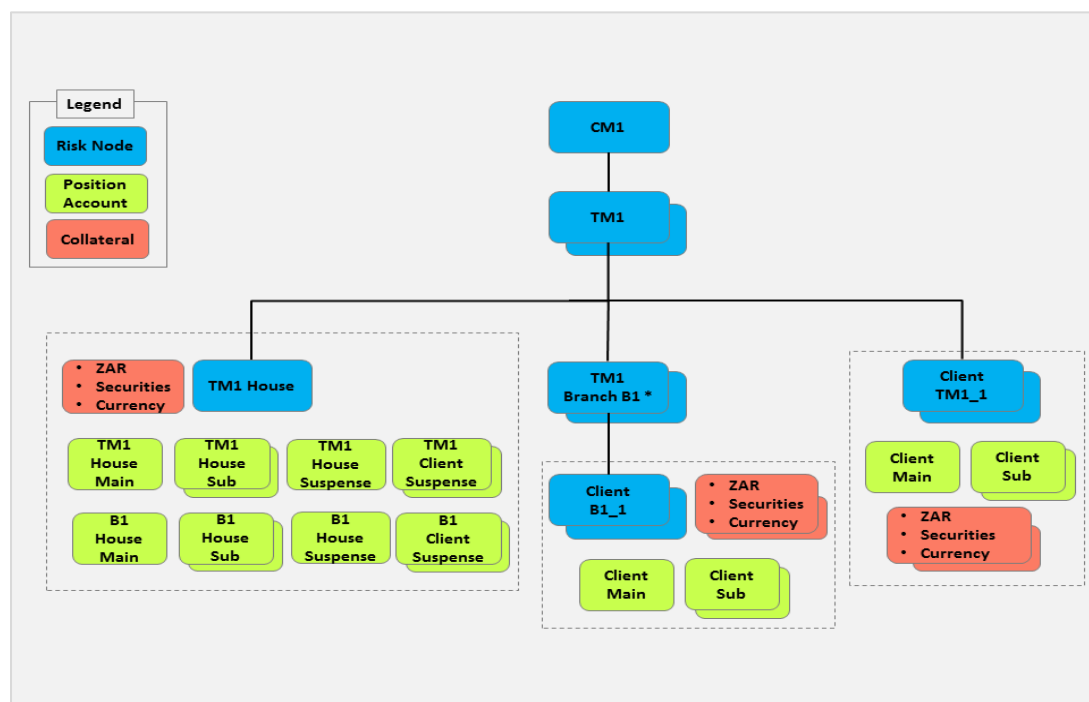


Figure 3 - RTC account structure

Note: RTC calculates risk at Trading Member Branches level for illustrative purposes only but is rolled up to the main member level for final risk and settlement purposes.

Below are the types of position accounts that are applicable to a member.

Type of account	Account Type	Account Sub type	Automatically created	Additional accounts can be added
House main account	HOUSE	MAIN	Yes	No
House sub account	HOUSE	SUB	No	Yes
House suspense account	HOUSE	SUSPENSE	Yes	No
Client main account	CLIENT	MAIN	Yes	No
Client sub account	CLIENT	SUB	No	Yes
Client suspense account	CLIENT	SUSPENSE	Yes	Yes

The Account Type can be identified in the `positionAccountType` field of the [PositionAccount](#) object. The Account Sub Type is provided in the `positionAccountSubType` field.

The External Position Account IDs are defined as follows:

Account Type	Source	Uniqueness	Maximum Length
House Main	Generated by RTC	Global	5
Branch Main	Generated by RTC	Global	5
Client Main	Set by RTC to Client ID. The Client ID is generated by RTC; the first 5 characters make the ID unique, this is followed by a 6 th character	Global	6
House Sub	Set by Member	Per TM	9
Client Sub	Generated by RTC. Client ID + NNN	Global	9
House Suspense	Set by RTC to 40006	Per TM	5
Client Suspense for TM	Set by RTC to 99999	Per TM	5
Clients Suspense for Branch	Set by RTC to 99BNN, where NN is numbered 01-99 within the TM	Per TM	5
Clients Suspense, additional	Set by Member	Per TM	10

4.1 Position Accounts

4.1.1 House Accounts

Trading member's main accounts

Each trading member has a separate house main account that is created automatically when the trading member is linked to a clearing member. The `externalPositionAccountId` of the `PositionAccount` is the same as the trading member id. The account is included in the trading member's risk node. The trading member's risk node is included in the clearing member's risk node.

Member branch's main accounts

Each trading member branch has a separate house main account that is automatically created when the trading member branch is created. The `externalPositionAccountId` of the `PositionAccount` is the same as the trading member branch id. The account is included in the trading member's risk node (i.e. not in the branch's risk node). This account is also linked to the clearing member's risk node.

Note: It is not possible to add additional house main accounts.

Sub accounts

A trading member and a member branch can add house sub accounts (see section section 6.4.3 for more details). The house sub accounts are included in the same risk structure as the main accounts.

Note: Clients need to first request authorisation from the JSE to enable sub accounts. Once authorised, the JSE will enable the client on RTC to be able to add sub accounts via EMAPI. The process to request this authorisation will be communicated to clients in due course.

House suspense accounts

Each trading member has a house suspense account that is created automatically when the trading member or branch is created. The ID of the automatically created suspense account is "40006". The house suspense accounts are included in the same risk structure as the main accounts.

Client suspense accounts

Each trading member and trading member branch has a clients' suspense account that is created automatically when the trading member or branch is created. The ID of the automatically created clients' suspense account is "99999" or "99B[Branch number]"(generated by RTC).

A trading member can add additional clients' suspense accounts (See section 6.4.4 for more details). The client suspense accounts are included in the same risk structure as the main accounts.

4.1.2 Member Client Accounts

Main accounts

Each member client has a client main account that is automatically created when the client is created. The `externalPositionAccountId` of the `PositionAccount` is the same as the client's id. This is the account number that is required when submitting the order or trade to the trading system. The account is included in the client's risk node. This risk node is included in trading member's risk node.

Sub accounts

A member or branch can add client sub accounts (See section 6.4.3 for more details). The client sub account is assigned an ID by RTC. The ID format is Client ID + a three-digit number.

Client sub accounts are included in same risk structure as the main account.

4.2 Collateral Accounts

Collateral accounts are created automatically by the system for trading members and member clients. A `CollateralAccount` is connected to a risk node, and created at the same time as the risk node. Risk nodes for Trading Member Branches do not have collateral accounts; collateral for these risk nodes are located in the member's risk nodes.

Member clients of trading members or branches have their collateral accounts included in the client's risk node, which is included in the main member's risk node.

5 BROADCAST FLOWS

In order to get the full picture of the current market state, the client needs to process both the standard request/response calls and event messages. Whenever a business event occurs, such a deal being allocated or collateral being pledged², RTC generates event messages. These event messages are grouped into a concept called Broadcast Flows.

Broadcast Flows can be seen as a channel for broadcasts of the same category. RTC has several different broadcast flows for the various types of events that occur. In order to receive broadcast messages, the client needs to set up a number of subscriptions, one for each flow as described in detail in Section 7 of *Volume PT01 – Post-trade EMAPI Common* (See 1.7)

Note: The EMAPI broadcast flows are NOT to be confused with UDP-type multicast broadcasts. The events published on EMAPI broadcast flows are guaranteed as they are sent via the TCP/IP protocol and not UDP.

Broadcast Flow	Replay Supported	Description and messages sent	Subscription groups used
PUBLIC GLOBAL REFERENCE DATA FLOW	No ³	Reference data events	Reference data messages have no subscription group. Use subscription group 1 when setting up the subscription.
ACCOUNT EVENT FLOW	Yes	Positions and trades. <ul style="list-style-type: none"> AccountPositionEvent AccountTradeEvent CommissionEvent 	Events are published on the subscription group for the account (can be any subscription group except 1). AccountTradeEvent is only used in a current value snapshot. In other cases, the trade is included in the AccountPositionEvent
RISK EVENT FLOW	Yes	Risk calculation results <ul style="list-style-type: none"> RiskNodeEvent 	Event is published on the RiskNode's subscription group.
GIVEUP EVENT FLOW	Yes	Assign and Tripartite work flow. <ul style="list-style-type: none"> GiveUpEvent 	Event is published on the subscription group used for the account.
SETTLEMENT EVENT FLOW	Yes	Collateral and payment information <ul style="list-style-type: none"> DailyAccountSummaryDetailsEvent AggregatedSummaryClearingMemberEvent AggregatedSummaryTradingMemberEvent CmBalancing1Event CmBalancing2Event 	DailyAccountSummaryDetailsEvent is published on the RiskNode's subscription group. AggregatedSummaryTradingMemberEvent is published on the subscription group for the CM-TM combination. Other events are published on the subscription group for the CM's risk node.

² See *Volume PT02 – Post-trade EMAPI Clearing* for more details.

³ Current value and subscription only

Broadcast Flow	Replay Supported	Description and messages sent	Subscription groups used
		<ul style="list-style-type: none"> ReadyConfirmAvailableFXEvent WithdrawalNotificationEvent 	
MARKET DATA EVENT FLOW	No ⁴	Prices and other market data. <ul style="list-style-type: none"> PriceEvent CurveEvent SurfaceEvent DividendEvent OptionDataEvent ATMVolatilityEvent YieldEvent 	All messages published on subscription group 1.

⁴ Current value only

6.1 Overview

Reference data is the set of static data that does not change with every trade management⁵ or other transaction that is processed. Reference data is for example the definition of members, instruments and risk parameters.

All reference data required by clients to interface with RTC is published via the Reference Data broadcast flow using the EMAPI protocol.⁶

Note: Public clearing reference data is also available from the JSE's Information Delivery Portal (IDP) for those customers who do not need to connect to EMAPI.. Please refer to the IDP specifications for more details (See 1.7 Related Documents).

The following diagram illustrates the reference data components and data flows:

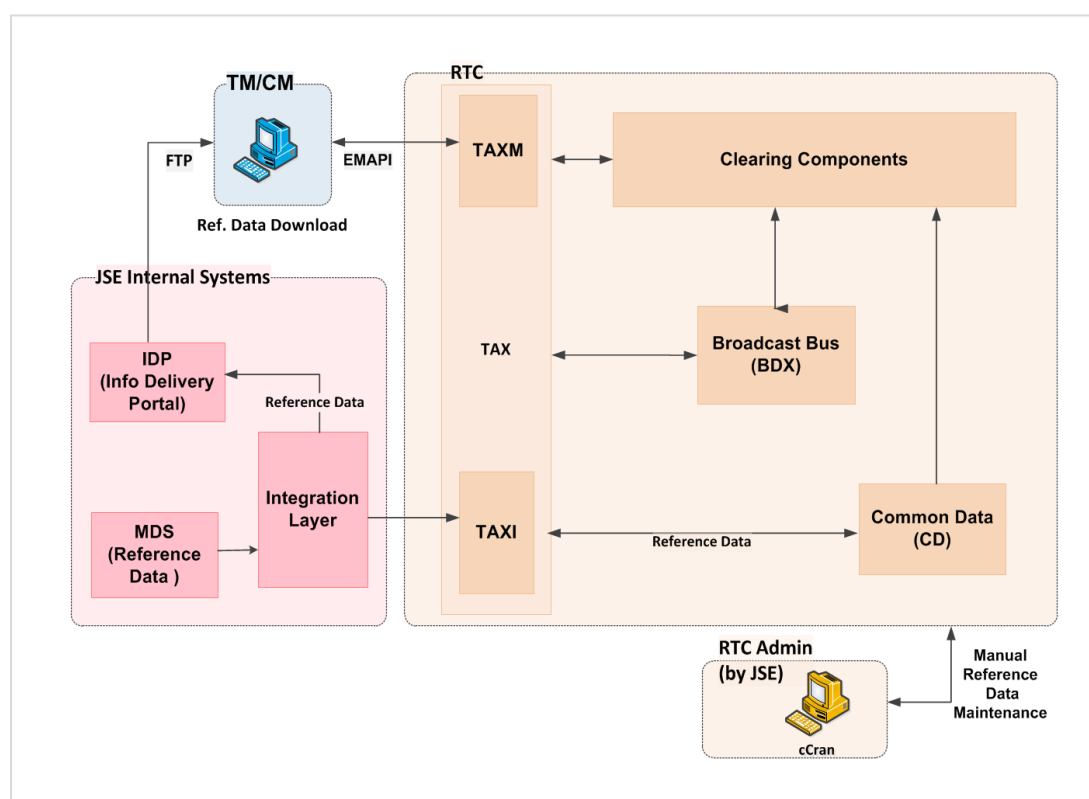


Figure 4 - Reference Data Dissemination

The JSE uploads reference data to RTC from its reference data system (MDS) during both intraday and end-of-day. The JSE can also manually maintain reference data in RTC using the system's administration front-end. Reference data uploaded to RTC is published to members via the EMAPI interface.

Note: Reference data will be published via EMAPI as a complete set of data, not an incremental update. Therefore, each day's reference data will be a complete set, and clients are expected to download and refresh all reference data daily to ready their systems for the trading day.

Note: New reference data published intraday via EMAPI should be added to client's reference data. Any intraday updates to reference data should replace the previously published reference data.

⁵ Trade management is also known as deal management.

⁶ From RTC Release 1.10.0.

Note: Please see Volume PT01 Section 7.6 *Building a copy of the reference data cache* for information on how to set up a reference data cache.

6.2 Reference Data in RTC

Reference data is inserted in RTC from different sources:

- Initial reference data that is supplied with the RTC installation
- JSE's reference data system (MDS)
- Reference data upload by members using EMAPI
- Manually input by JSE using RTC's user interface
- Automatically created within RTC based on other reference data

The reference data in RTC is available for download by members using EMAPI. Members also use the EMAPI to maintain their client and account reference data.

6.2.1 Reference Data Flow

EMAPI clients receive reference by subscribing to the `Reference Data Flow`. This broadcast flow supports `SubscriptionRequestType=CURRENT_VALUE`; this means that the flow delivers a snapshot of the latest version of all reference data objects. Subsequent changes to reference data objects are published to all sessions that have an ongoing subscription. The reference data published in this flow is summarised in Table 1 on the next page.

Note: Refer to Section 7 of Volume PT01 – Post-Trade EMAPI Common (See 1.7) for details on how to subscribe to broadcast data flows.

6.2.2 Summary of Reference Data

All reference data added to RTC can be downloaded by sending a request into the system. The resulting `CdResponse` message is returned for most reference data request messages, unless otherwise specified. The 'code' field in the response message indicates whether the request was successful or not. The `StatusCodes` enumeration lists the status codes that will be returned in the response message.

Table 1 on the next page lists all the reference data objects available to clients. These objects are filtered both on type and data based on the logged in user role. A clearing member for instance has only access to trading members and position accounts that are connected to the clearing member.

Member clients, cash accounts and position accounts are maintained by members; all other reference data is maintained by the JSE.

Table 1 - Summary of reference data available via EMAPI

#	Reference Data Entity	Reference Data Flow Object	JSE to Maintain (Yes/No)	Members to Maintain (Yes/No)
1)	Clearing Member	Member	Yes	No
2)	Trading Member	Member	Yes	No
3)	Member Branch	Member	Yes	No
4)	Member Client	Member	No	Yes
5)	Clearing Member Link	ClearingMemberLink	Yes	No
6)	Tripartite Agreement	TripartiteAgreement	Yes	No
7)	Cash Account	CashAccount	Yes	Yes
8)	Market	Market	Yes	No
9)	Market List	MarketList	Yes	No
10)	Segment	Segment	Yes	No
11)	Instrument	Instrument	Yes	No
12)	Tradable Instrument	TradableInstrument	Yes	No
13)	Calendar	RtcCalendar	Yes	No
14)	Currencies	Currency	Yes	No
15)	Country	Country	Yes	No
16)	Risk Node	RiskNode	Yes	No
17)	Series Spread Group	SeriesSpreadGroup	Yes	No
18)	Class Spread Group	ClassSpreadGroup	Yes	No
19)	Subscription Group	SubscriptionGroup	Yes	No
20)	Position Account	PositionAccount	Yes	Yes
21)	Collateral Account	CollateralAccount	Yes	No
22)	Access Group	AccessGroup	Yes	No
23)	Settlement Account	SettlementAccount	Yes	No
24)	Current System State	CurrentSystemState	Yes	No
25)	Eligible Currency	EligibleCurrency	Yes	No
26)	Eligible Security	EligibleSecurity	Yes	No
27)	Corporate Action	CorporateAction	Yes	No
28)	Calendar Date	CalendarDate	Yes	No
29)	Curve	Curve	Yes	No
30)	Deposit	Deposit	Yes	No
31)	InterestRateSwap	InterestRateSwap	Yes	No
32)	ForwardRateAgreement	ForwardRateAgreement	Yes	No
33)	Surface	Surface	Yes	No

6.3 Reference Data Maintained by JSE

This section describes the reference data that is maintained by the JSE using the appropriate scenarios for the entity.

All reference data is published through the [PUBLIC_GLOBAL_REFERENCE_DATA_FLOW](#).

Note: Please refer section 7.6 of *Volume PT01 – Post-trade EMAPI Common* for details on how reference data termination should be processed by clients.

6.3.1 Clearing Member, Trading Member and Member Branch

Clearing Members are maintained in RTC according to the following scenarios:

Entity	Scenario	Published Objects	Notes
Clearing Members	Add	Member SubscriptionGroup PositionAccount CollateralAccount RiskNode AccessGroup CashAccount SettlementAccount	participantUnitType = CLEARING_MEMBER RiskNodes are automatically created by the system when new CMs are added. PositionAccount and RiskNode include a reference to the new AccessGroup , which in turn has a reference to the new SubscriptionGroup . This means that AccountPositionEvents and RiskNodeEvents can be published on the new SubscriptionGroup . CollateralAccount and SettlementAccount also refer to the AccessGroup , but there are no published events concerning these objects.
	Update	Member	Only the member object is published on updates.
	Enable/ Disable	Member	isDisabled flag is updated in the republished Member object.

Trading Members are maintained in RTC according to the following scenarios:

Entity	Scenario	Published Objects	Notes
Trading Members	Add	Member	participantUnitType = TRADING_MEMBER No SubscriptionGroup or risk structure is created for the new trading member. These will be created when a ClearingMemberLink is created, connecting the TM to one of the existing CMs.
	Update	Member	Only the member object is published on updates.
	Enable/ Disable	Member	isDisabled flag is updated in the republished Member object.

Member Branches are maintained in RTC according to the following scenarios:

Entity	Scenario	Published Objects	Notes
Member Branches	Add	Member	<code>participantUnitType</code> = TRADING_MEMBER_BRANCH
	Update	Member	Only the member object is published on updates.
	Enable/Disable	Member	<code>isDisabled</code> flag is updated in the republished Member object.

6.3.2 Clearing Member Link

A Clearing Member Link defines the clearing member that a trading member can use for a market. The link is maintained by the JSE.

Links are maintained in RTC according to the following scenarios:

Entity	Scenario	Published Objects	Notes
Clearing Member Link	Add	ClearingMemberLink RiskNode PositionAccount CollateralAccount SettlementAccount AccessGroup SubscriptionGroup	Risk nodes are automatically created by the system when new links are added. The PositionAccount includes Main and Suspense Accounts.
	Update	Same as Add scenario	In order to update the link, the existing Clearing Member Trading Member Link will be removed, and then new Clearing Member Trading Member Link added in RTC.

6.3.3 Clearing House

The Clearing House is also defined as a Member in the system, with `MemberType=MARKETPLACE` and `memberId= JSE`. Only one clearing house member can be defined; it is pre-configured when the RTC system is installed and will not be updated or deleted.

6.3.4 Information Vendor

Information Vendors are maintained in RTC according to the following messages:

Entity	Scenario	Published Objects	Notes
Information Vendor	Add	Member	<code>participantUnitType</code> = INFORMATION_VENDOR
	Update	Member	<code>memberId</code> must exist in RTC
	Disable/Enable	Member	The flag <code>isDisabled</code> is set to True/False.

When an information vendor is updated, only the [Member](#) object is published on the Reference Data Flow.

Note: There is no `SubscriptionGroup` or risk structure created for the new information vendor. When setting up a subscription to the Market Data Flow or Reference Data Flow, information vendors should use the default `SubscriptionGroup = 1`.

6.3.5 Tripartite Agreement

Tripartite agreements are maintained in RTC according to the following scenarios:

Entity	Scenario	Published Objects	Notes
Tripartite Agreement	Add/ Update	TripartiteAgreement	

6.3.6 Cash Account

The JSE can add Cash Accounts for clearing members, trading members and branches. Cash Accounts for clients are added by the members. Both members and clients can only have one Cash Account per currency.

Cash Accounts are maintained in RTC according to the following scenarios:

Entity	Scenario	Published Objects	Notes
Cash Account	Add/ Update	CashAccount	Cash Account maintenance can also be done by members (See 6.4.2)

6.3.7 Market Structure

Market, Market List, Segment, Instrument, Tradable Instrument

Market, Market List and Segment is a tree structure with Market as the top node and Segment as the leaf.

Tradable Instrument is linked to a parent generic Instrument object; the generic Instrument belongs to a Segment; and a Segment belongs to a MarketList and a Market.

Example:

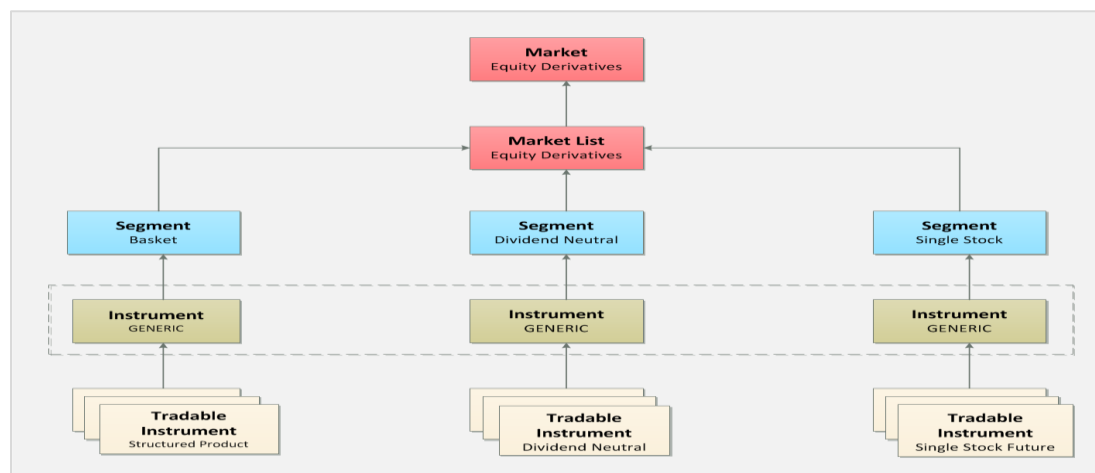


Figure 5 - Market Structure

Market, market list, segment, instrument and tradable instrument hierarchy is maintained in RTC according to the following scenarios:

Entity	Scenario	Published Objects	Notes
Market	Add	Market	This is a once-off setup as part of the configuration of the system.
	Update	Market	Updated entity is republished.
MarketList	Add	MarketList	This is a once-off setup as part of the configuration of the system. <code>parentInternalId</code> must be an existing Market in RTC.
	Update	MarketList	Updated entity is republished.
Segment	Add	Segment	This is a once-off setup as part of the configuration of the system. <code>parentInternalId</code> must be an existing MarketList in RTC.
	Update	Segment	Updated entity is republished.
Instrument	Add	Instrument	This is a once-off setup as part of the configuration of the system.
	Update	Instrument	Updated entity is republished.
Tradable Instrument	Add	TradableInstrument	<code>parentInternalId</code> must be an existing instrument in RTC.
	Update	TradableInstrument	Updated entity is republished.
	Delete	TradableInstrument	Removed automatically in RTC after configured number of days after <code>validTo</code> has passed. After the tradable instrument has been automatically removed, it will not be included in the reference data snapshot on EMAPI.

Instruments are parent to tradable instruments. This reflects the underlying level. Each tradable instrument must be connected to an instrument. The instrument specifies the Position Manager partition that all its tradable instruments use. This is a way of grouping all tradable instruments for one underlying in the same partition. [Instrument](#) will be pre-configured in RTC.

Example:

Instrument INSTR_PARTITION1 has these tradable instruments connected to it:

- AGL (Equity)
- AGL-20151128 (Future)
- AGL-20151125-114P (Put Option)
- J200 (Index)

Instrument types

The RTC [InstrumentType](#) corresponds to the JSE Instrument Class. The RTC [InstrumentSubType](#) corresponds to the JSE Instrument Type. A [TradableInstrument](#) may have additional attributes that are only applicable for a certain instrument type.

InstrumentType (JSE Instrument Class)	Valid Instrument Sub Types (JSE Instrument Type)
SPOT	Equity Index Forex Pair
FUTURE	Single Stock Index Forex Dividend Neutral
OPTION	Single Stock Index Forex Dividend Neutral
CFD	CFD
BOND	Bond

6.3.8 Calendar

Calendars in RTC are used to define dates (holidays) when the system is closed. A calendar consists of an [RtcCalendar](#) object and a number of [CalendarDates](#).

[CalendarDates](#) can be of three types:

- Full day
- Half day
- Closed day

The date type is used to identify a schedule for the daily operations. There will typically be at least two schedules: one “full day” schedule with normal hours and one “closed day” schedule where no scheduled events take place. Each day, the system identifies which schedule to run based on the current date and the configuration in the [RtcCalendar](#) object.

If no [CalendarDate](#) is found for the current date, the “full day” schedule is assumed.

The calendar defines the time zone to use when interpreting the configured times for scheduled events.

The calendar is also used for calculating settlement dates taking holidays into account.

If the calendar attribute *saturdaysSundaysClosed* is set to “true”, all Saturdays and Sundays will be treated as closed days, i.e. there is no need to specify them as closed [CalendarDate](#) objects.

Example:

RtcCalendar:

Field	Value
rtcCalendarId	"1"
calendarDateCalendarId	"1"
timeZone	"Africa/Johannesburg"
saturdaysSundaysClosed	"true"

CalendarDate:

Field	Value
date	"2015-01-01"
dateType	"CLOSED"
calendarId	"1"

CalendarDate:

Field	Value
date	"2015-12-25"
dateType	"CLOSED"
calendarId	"1"

In this example, the two dates are both of type CLOSED and belong to calendar 1. On these dates (and every Saturday and Sunday), schedule 3 (closed day schedule) will be used. On all other dates, schedule 1 (full day schedule) will be used.

When RTCalendar or CalendarDates are maintained (i.e. added, updated or removed), the corresponding reference data object will be published on the `Public Global Reference Data Flow`.

6.3.9 Currencies

The `Currency` object defines the currencies available in RTC. For cash settlement positions, the `externalInstrumentId` is not a tradable instrument, but a currency.

Currencies are maintained in RTC according to the following scenarios:

Entity	Scenario	Published Objects	Notes
Currency	Add	<code>Currency</code>	This is a once-off setup as part of the configuration of the system.

6.3.10 Country

The country code is used for validation purposes and specifically used to identify foreign clients, e.g. when a new Client is added.

Countries are maintained in RTC according to the following scenarios:

Entity	Scenario	Published Objects	Notes
Country	Add	Country	This is a once-off setup as part of the configuration of the system.

6.3.11 Risk Node

Risk nodes are created automatically when members, branches, clients or links are added.

Risk nodes are maintained according to the following scenarios:

Entity	Scenario	Published Objects	Notes
Risk Node	Add	RiskNode	Created automatically by RTC.

Note: Please see Section 7 for additional info on risk management and risk parameters setup.

6.3.12 Class and Series Spread Group

The Series Spread Group (SSG) and Class Spread Group (CSG) parameters are maintained in RTC according to the following scenarios:

Entity	Scenario	Published Objects	Notes
Series Spread Group	Add	SeriesSpreadGroup	The new entity is published.
	Update	SeriesSpreadGroup	The updated entity is re-published.
Class Spread Group	Add	ClassSpreadGroup	The new entity is published.
	Update	ClassSpreadGroup	The updated entity is re-published.

Note: Please see Section 7 for additional info on risk management and risk parameters setup.

6.3.13 Subscription Groups

A subscription group defines the filter used for outgoing messages on broadcast flows.

Subscription Groups are created by the system for the combination of clearing member, trading member and member branch. They are published as [SubscriptionGroup](#) objects on the Public Global Reference Data Flow.

Member model

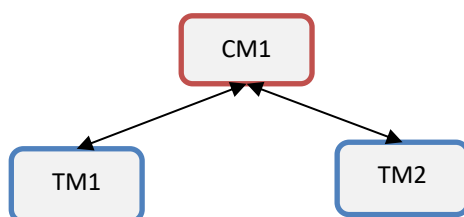
A Clearing Member (CM) has its own subscription group for information that only concerns the clearing member. This subscription group can only be accessed by the Clearing Member. A Trading Member (TM) can be linked to one or more Clearing Members. When this link is created, the relevant accounts, risk nodes etc. are also created.

Separate subscription group is created for each CM-TM link, enabling the Trading Member's accounts and risk nodes to be visible to both parties. This means that the CM can receive information about all its trading members, but not about the Trading Members' links with other CMs.

The accounts and risk nodes objects are available on the same subscription group.

Example 1:

Two Trading Members (TM1 and TM2) are using the same Clearing Member (CM1).



There will be three subscription groups:

- “CM1”, for accounts, risk nodes etc. that only belong to CM1
- “CM1_TM1” for TM1’s accounts, risk nodes etc.
- “CM1_TM2” for TM2’s accounts, risk nodes etc.

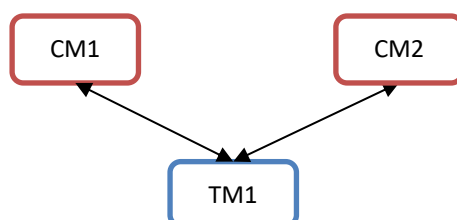
CM1 is authorised to view subscription groups CM1, CM1_TM1 and CM1_TM2.

TM1 is authorised to view subscription group CM1_TM1.

TM2 is authorised to view subscription group CM1_TM2.

Example 2:

A Trading Member (TM1) is using two different Clearing Members (CM1 and CM2).



There will be four subscription groups:

- “CM1” for accounts, risk nodes etc. that only belong to CM1
- “CM2” for accounts, risk nodes etc. that only belong to CM2
- “CM1_TM1” for TM1’s accounts, risk nodes etc. with CM1
- “CM2_TM1” for TM1’s accounts, risk nodes etc. with CM2

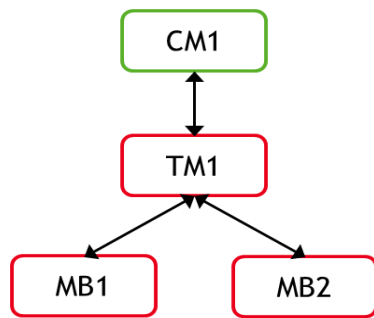
CM1 is authorised to view subscription groups CM1 and CM1_TM1.

CM2 is authorised to view subscription groups CM2 and CM2_TM1.

TM1 is authorised to view subscription group CM1_TM1 and CM2_TM1.

Example 3:

Two Member Branches (MB1 and MB2) belonging to the same Trading Member (TM1) and are using the same Clearing Member (CM1).



There will be four subscription groups:

- “CM1”, for accounts, risk nodes etc. that only belong to CM1
- “CM1_TM1” for TM1’s accounts, risk nodes etc.
- “CM1_TM1_MB1” for MB1’s accounts, risk nodes etc.
- “CM1_TM1_MB2” for MB2’s accounts, risk nodes etc.

CM1 is authorised to view subscription groups CM1, CM1_TM1, CM1_TM1_MB1 and CM1_TM1_MB2.

TM1 is authorised to view subscription group CM1_TM1, CM1_TM1_MB1 and CM1_TM1_MB2.

MB1 is authorised to view subscription group CM1_TM1_MB1.

MB2 is authorised to view subscription group CM1_TM1_MB2.

6.3.14 Corporate Actions

A [CorporateAction](#) defines how positions related to an instrument will be affected when the corporate action is performed. Positions for one instrument can be changed into positions for another instrument, with the quantity multiplied by a position factor.

When the corporate action is added or updated, a [CorporateAction](#) reference data object will be published on the [Public Global Reference Data Flow](#)

6.3.15 Curves

A [Curve](#) consists of a number of [CurveConstituents](#). Each [CurveConstituent](#) refers to one of the following instruments:

- [ForwardRateAgreement](#)
- [InterestRateSwap](#)
- [Deposit](#)

The Curve id is used as the key for a [CurveEvent](#) on the [Market Data Event Flow](#).

6.3.16 Pricing and Reference Instruments (Rates and curves)

Reference data for valuations are published on the [Reference Data Flow](#).

When each of the following entities are maintained (i.e. added, updated or removed), the corresponding reference data object will be published on the [Public Global Reference Data Flow](#).

- [ForwardRateAgreement](#)
- [InterestRateSwap](#)
- [Deposit](#)
- [Curve](#)
- [Surface](#)

6.3.17 Other Reference Data Setup by JSE

The reference data in the following is created by initial RTC configuration or automatically by RTC based on other reference data.

Reference Data Entity	Description	Published Object	Created by
Position Account	Position account for position keeping. <i>See Section 4.</i>	PositionAccount	Automatically by RTC or by member (See 6.4.4).
Collateral Account	Account used for collateral. <i>See Section 4.2</i>	CollateralAccount	Automatically by RTC
Access Group	Defines access for RTC data. <i>See Section 7.3.1 in Volume PT01 – Post-Trade EMAPI Common</i>	AccessGroup	Automatically by RTC
Settlement Account	Account holding settlement positions (payments).	SettlementAccount	Automatically by RTC
Current System State	Used to communicate RTC System State.	CurrentSystemState	Automatically by RTC
Eligible Currency	Currency allowed for collateral. <i>See section 11.</i>	EligibleCurrency	Ad hoc by JSE
Eligible Security	Security allowed for collateral. <i>See section 11.</i>	EligibleSecurity	Ad hoc by JSE

6.3.18 Interface Users and Role-based Permissions

Users are added or updated by the JSE. Users log in to the system using the EMAPI interface connections have dedicated User IDs.

Users belong to a member and are identified with the combination of member ID and user id. All member types except clients can have users.

A user is assigned one or more roles, which defines the functionality in the system that the user can access.

The following user roles are available for member users:

Member Type	Role
Clearing Member	A clearing member user has access to all accounts that he or she is CM for and to no other accounts. Note: See Section 4 & 6.4.4 for details on account setup.
Trading Member	A trading member user belonging to a TM has access to the following accounts and to no other accounts:

Member Type	Role
	<ul style="list-style-type: none"> House accounts owned by the TM and by Branches of the TM Accounts owned by Clients of the TM Accounts owned by Branches of the TM
Trading Member Branch	<p>A trading member user belonging to a Branch has access to the following accounts and to no other accounts:</p> <ul style="list-style-type: none"> House accounts owned by the Branch Client accounts owned by the Branch

6.4 Reference Data Maintained by Members

6.4.1 Member Client

Member's clients are maintained in RTC by Members and Branches according to the following messages:

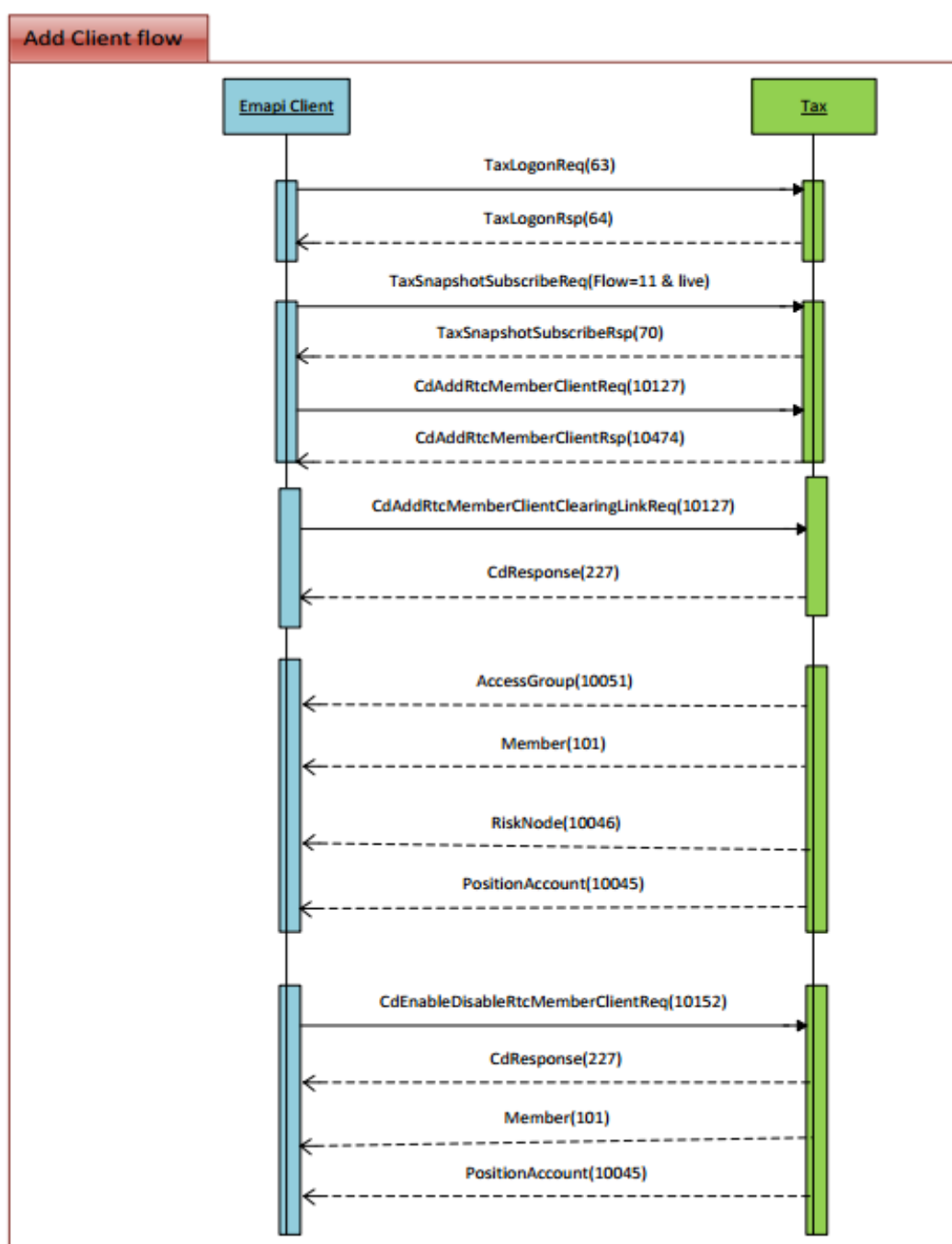
Scenario	Request Message	Published Objects	Key criteria	Notes
Add (new Member Client in RTC)	CdAddRtcMemberClientReq	When a new member client is added, it must then be linked to the correct clearing member and enabled before the Member object is published. See hierarchy for maintenance at end of this section.	MemberType = MEMBER_UNIT participantUnitType = CLIENT	The generated client ID is included in the response CdAddRtcMemberClientRsp . After the client has been created, the allowed markets can only be updated by the Clearing House (JSE). Please contact Customer Support.
Link (a new member client to a clearing member in RTC)	CdAddRtcMemberClientClearingLinkReq	Member PositionAccount CollateralAccount RiskNode		The field clearingMemberId required by this message can be obtained from the ClearingMemberLink message.
Enable/Disable (existing Member Client in RTC)	CdEnableDisableRtcMemberClientReq	Member PositionAccount	memberId must exist in RTC	A member client cannot be terminated, but can be disabled.
Update (an existing Member Client in RTC)	CdUpdateRtcMemberClientReq	Member	memberId must exist in RTC	See message specification for list of fields that can be updated by members.

A response message is returned for all reference data request messages. The `code` field in the response message indicates whether the request was successful or not. Refer to [Status Codes](#) (in HTML and XML specifications), for a detailed list of status codes that will be returned in the response message. When adding a new member client, the response is a [CdAddRtcMemberClientRsp](#) message; for the other requests the response is a [CdResponse](#) message.

The hierarchy for maintenance of Member Clients on RTC must occur in the following order of events:

#	Event Sequence	Messages
1	CdAddRtcMemberClientReq	
2	CdAddRtcMemberClientClearingLinkReq	<p>Member event (this event is created after the #2 with the <code>isDisabled</code> field = T (member is disabled))</p> <p>Position Account event1 (this event is created after the #2 with the <code>isEnabled</code> field = F (position is disabled))</p>
3	CdEnableDisableRtcMemberClientReq	<p>Member event (this event is created after the #3 with the <code>isDisabled</code> field = F (member is enabled))</p> <p>Position Account event2 (this event is created after the #3 with the <code>isEnabled</code> field = T (position is enabled))</p>

6.4.2 Add Client sequence diagram



6.4.3 Cash Account

Cash Accounts for foreign clients are added by the members. Only one Cash Account per currency per client is allowed.

Cash Accounts are maintained in RTC according to the following messages:

Entity	Scenario	Request Message	Key criteria	Notes
Cash Account	Add	CdAddCashAccountReq		
	Update	CdUpdateCashAccountReq		

When a new Cash Account is added or updated, the [CashAccount](#) reference data entity will be published on the Public Global Reference Data Flow.

Note: Cash account can also be maintained by the JSE. See 6.3.6.

6.4.4 Position Account

Position accounts cannot be explicitly created by JSE. The position accounts that exist for each member, member branch or member client are created automatically by RTC when the entity is created or linked to a clearing member.

Trading members can maintain position accounts using the following messages:

Entity	Scenario	Request Message	Key criteria	Notes
Position Account	Add	CdAddRtcPositionAccountReq	<code>positionAccountExternalId</code> must be unique within the member and be up to a maximum of 9 characters.	<p>The position account is automatically enabled when it is created. The generated account ID (<code>positionAccountExternalId</code>) is included in the response CdAddRtcPositionAccountResp.</p> <p>Note: Clients need to first request authorisation from the JSE to enable sub accounts. Once authorised, the JSE will enable the client on RTC to be able to add sub accounts via EMAPI. The process to request this authorisation will be communicated to clients in due course</p>
	Enable/Disable	CdEnableDisableRtcPositionAccountReq	<code>positionAccount</code> Id must exist in RTC	<p>No trade can be registered to the disabled position account.</p> <p>New trades for the account will be registered on the House Suspense account of the TM or Branch (if applicable) on the trade.</p> <p>A disabled account can be enabled again using the same request.</p>

When accounts are maintained on RTC, a [PositionAccount](#) message is published on the Public Global Reference Data Flow.

6.4.5 Maintenance (add/update/) of Reference Data

Updates to reference data objects should be sent after addition of new reference data of the same type.

Note: When a client sends a requests to add reference data the “action” field should be set to ADD (1).

To keep track of the order of updates on reference data there is the `stateSequenceNumber` (SSN). SSN is a global sequence number for the reference data. Every update to RTC's reference data increases the `stateSequenceNumber`.

If an EMAPI client requests an addition or update of reference data, the resulting SSN is returned in the response to the add/update request. The SSN is also sent with the event on the reference data flow. This way the client can know in which event on the reference data flow the add/update is published.

7 RISK MANAGEMENT

7.1 Risk Nodes

Risk nodes are created by the system for the different levels in the member structure. A risk node contains position accounts and collateral accounts. The positions on the accounts are aggregated when risk calculations are performed for the risk node. A position account instance in RTC belongs to one risk node only.

Risk nodes are created automatically when members, branches, clients or links are added. Risk nodes cannot be explicitly added. The position account and collateral account reference data objects contain a link to the risk node.

The following table provides a description of the various types of risk nodes:

Risk Node	Description
Clearing Member	Contains the risk for all trading members that are linked to that clearing member.
Trading Member	Used for calculating the risk for the trading member's house accounts, its branches' house accounts as well as the clients' suspense accounts.
Trading Member Branch	Branch risk nodes do not have collateral accounts. The risk node values are aggregated to the parent (trading member) risk nodes.
Member Client	Used for calculating the risk for the client's main and sub accounts.

The diagram below describes the risk structure in RTC. The risk nodes are shown in blue in the diagram:

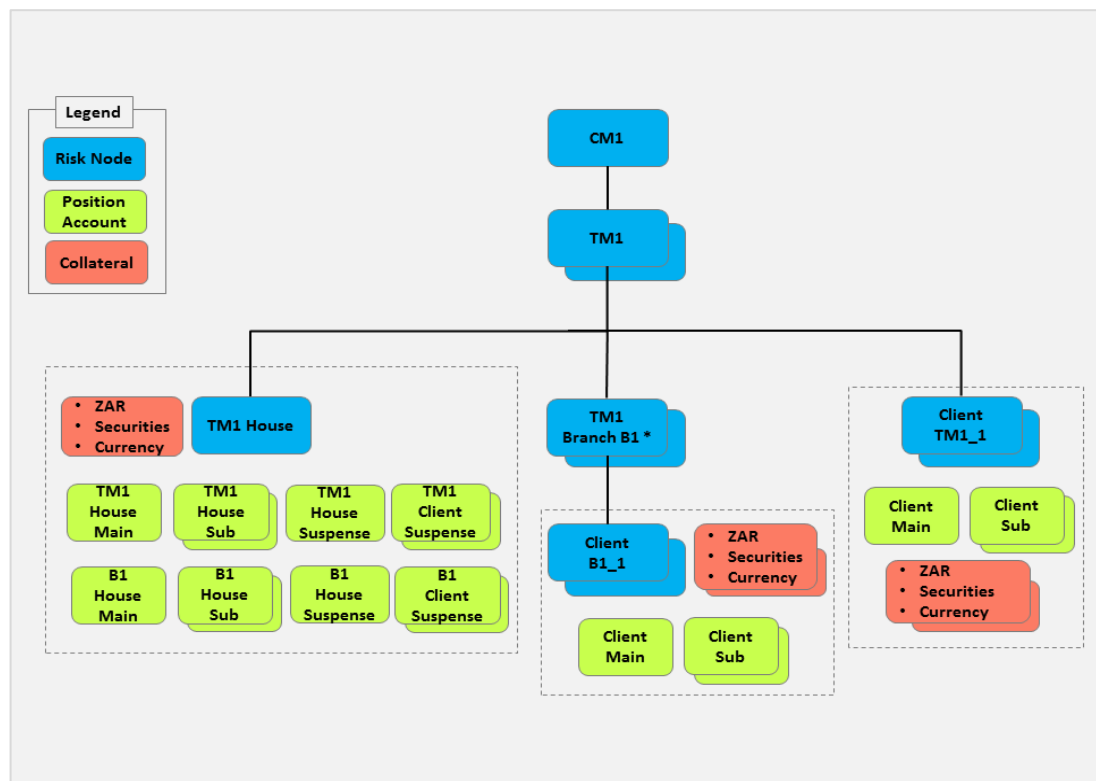


Figure 6 - RTC risk structure

7.2 Risk Parameters Maintained by JSE

The JSE maintains the risk parameters [SeriesSpreadGroup \(SSG\)](#) and [ClassSpreadGroup \(CSG\)](#). The ClassSpreadGroup contains the SSMR (Series Spread Margin Requirement) for the CSG within the SSG. These risk parameters are published on the Reference Data Flow.

The [Curves](#), [Deposits](#), [InterestRateSwaps](#), [ForwardRateAgreements](#) and [Surfaces](#) are also published on the Reference Data Flow.

At the [Market](#) level, several J-SPAN parameters (see *Volume PT03 – Post-trade Margin Methodology Specifications* for more details) such as Look Back Period, Volatility Lookback Period, Max Scale Up, Max Scale Down, Confidence Percentile, IMR Statistics Period and Daily Maximum Participation Factor. See 7.4.4 for information on how to obtain the Risk Arrays used as input to J-Span.

In addition, several risk parameters are defined at the [TradableInstrument](#) level; for example, for a futures contract, the IMR (Initial Margin Requirement), VSR (Volatility Scanning Range) as well as the CSMR (Class Spread Margin Requirement of the future within the CSG) is provided.

The JSE also publishes risk parameters and risk reference data on IDP (Information Delivery Portal) for those clients. For example, the Daily Margin Report, Anchor Volatility, Margin Requirements, Volatility Skews, Rates, Risk Arrays, etc.

Note: The JSE also provides risk parameters and reference data via FTP. Please refer to IDP documentation (See 1.7) for additional information on the products available from IDP and how to download this information.

7.3 Risk Parameters Maintained by Members

7.3.1 Setting Risk Limits

External participants can maintain the risk limits to be alerted via a [RiskNodeEvent](#) if the following condition is met:

$$(Initial\ Margin + Additional\ Margin) - (Variation\ Margin + Collateral) > Risk\ Limit$$

Risk limits can be maintained in the following scenarios:

Field	Scenario	Request Message	Key criteria	Notes
Risk limit	Set risk limit for TM	CdSetTradingMemberRiskLimitReq		CMs may set a more strict (lower) value than the global value on the risk nodes of their TMs (all TMs or a specific TM).
	Set risk limit for client	CdSetClientRiskLimitReq		TMs may set a more strict (lower) value than the value set by the CM on the risk nodes of their Clients (all Clients or a specific Client).

7.3.2 Setting Additional Margin Percentage

External participants can maintain Additional Margin Percentage for their trading members and clients. Clearing members sets the value for the Trading Members and Trading Members set the values for clients.

Entity	Scenario	Request Message	Key criteria	Notes
Additional Margin Percentage	Set Additional Margin percentage for TM	CdSetTradingMemberAMPercentageReq		CMs may set a more strict (higher) value than the global value on their TMs.
	Set Additional Margin for client	CdSetClientAMPercentageReq		TMs may set a more strict (higher) value than the global value.

7.4 Receiving and Interpreting Risk Data

The JSE will perform risk calculations throughout the day based on latest price updates and/or trading, deal management and position management activities.

Risk results are published by RTC as [RiskNodeEvent](#) objects on the Risk Event Flow. The flow uses the same subscription groups as the Account Event Flow. This means that a member may need to set up subscriptions for several subscription groups to get a complete view of their risk.

7.4.1 Risk Node Event values

Risk results are published as [RiskNodeEvent](#) objects on the Risk Event Flow.

EMAPI Field Number	Field Name	Description
31	<code>variationMargin</code>	Any profit or loss given the current market value compared to the previous mark-to-market value (or trade value).
32	<code>portfolioRisk</code>	The mark-to-market loss on the portfolio over a given time horizon (and probability level). A positive value indicates a risk.
33	<code>collateralValue</code>	Total collateral value of all collateral positions in the collateral account for the risk node.
35	<code>liquidationAddOn</code>	The liquidation period add-on is an amount that gets added to the margin calculated by J-SPAN. The Liquidation Period add-on increases the margin requirement when the client's notional exposure in a particular underlying forms a significant portion of the value that gets traded in the market on a daily basis.
36	<code>largePositionAddOn</code>	The system calculates additional Initial Margin to compensate for large positions or concentration risk. This calculation takes into account position size thresholds which are defined by the Clearing House.
37	<code>J-SPANValue</code>	J-SPAN Initial Margin based on the positions and the Contract Scenario Exposure (CSE) risk arrays. The

EMAPI Field Number	Field Name	Description
		system calls the J-SPAN algorithm with the netted positions of all the accounts under the risk node. The J-SPAN algorithm is defined according to the official <i>Volume PT03 – Post-trade Margin Methodology Specifications</i> (See 1.7).
38	additionalMargin	Additional Margin is a margin that is added on top of IM and calculated as a percentage of IM. Different members and clients can be set up for different additional margin percentages. $AM = IM * AM \text{ percentage (defined for a risk node)}$.
39	additionalMarginPercentage	Additional Margin Percentage used in the Additional Margin Calculation.
40	riskLimit	The clearing house may set a global Risk Limit in ZAR. By default, 0 will be set. Clearing Members may set a more strict (lower) value than the global value on the risk nodes of their Trading Members (for all or specific Trading Members). Trading Members may set a more strict (lower) value than the value set by the Clearing Member on the risk nodes of their Clients (for all or a specific client).
41	valueAgainstLimit	The current risk of the risk node. Calculated as follows: $(IM + AM) - (VM + Collateral)$
42	Alert	Risk alert status is YES or NO: YES if $(IM + AM - VM - Collateral) > Risk \text{ Limit}$, NO otherwise
43	settlementMargin	For risk nodes with settlement positions for physical delivery positions: $SM = \text{official SMR} * \text{quantity (netted on risk node)}$.
44	equityNotionalValue	The system adds the Notional Exposure per Underlying (with sign) to the Notional Exposure for the asset class of the underlying. This effectively adds up the Notional Exposure per Underlying for all underlyings within the asset class per underlying.
45	fxNotionalValue	The system adds the Notional Exposure per Underlying (with sign) to the Notional Exposure for the asset class of the underlying. This effectively adds up the Notional Exposure per underlying for all underlyings within the asset class per underlying.

7.4.2 Notional Values

Selected fields in the embedded message `NotionalValue` within the `RiskNodeEvent` are defined as follows:

EMAPI Field Number	Field Name	Description
2	netNotionalValue	<p>Notional exposure per underlying and asset class.</p> <p>- <i>Future</i>: Notional = Price of the future x 1 x position qty (with sign) x contract size</p> <p>- <i>Option</i>: Notional = Price of the underlying future x delta of option (with sign) x position qty (with sign) x contract size</p> <p>Note: The price is either a settlement price if the margin calculation is part of a margin call calculation OR an indicative price.</p> <p>Notional Exposure per Underlying = Aggregated notional exposure (with sign) of all positions in tradable instruments with the same underlying spot.</p>
3	grossNotionalValue	Gross Notional Exposure = sum of the absolute notional exposure per position in instruments with the same underlying spot.

7.4.3 Collateral Position Values

Selected fields in the embedded message `CollateralPositionValue` within the `RiskNodeEvent` are defined as follows:

EMAPI Field Number	Field Name	Description
2	isCash	True if a cash position.
3	quantity	The collateral position quantity.
4	marketValue	Market value of the position.
5	valueAfterHaircut	<p>Collateral position value after haircut:</p> <p>Collateral value (FX or security) = $\text{Min}(\text{MtM price} * \text{qty} * 1 / (1 + \text{haircut}\%); \text{maxAmount})$, where maxAmount and haircut % are from the eligible instrument.</p>
6	externalInstrumentId	<p>The JSE Master ID of the collateral security, or the currency if the collateral position is in ZAR or FX. When FX or ZAR is posted, the currency-value (USD, EUR, GBP, ZAR) will be populated in the externalInstrumentId field. When securities are pledged, the equity's InstrumentMasterID will be populated in the externalInstrumentId field.</p>

7.4.4 Risk arrays

Risk arrays are calculated once per day during the end-of-day batch run. Clients can request the latest risk arrays by submitting a `GetRiskArrayReq` message to RTC.

Entity	Scenario	Request Message	Key criteria	Notes
Risk arrays	Retrieve generated risk arrays	<code>GetRiskArrayReq</code>		

8 TRADE RECEIPT FROM TRADING SYSTEM

Trades from the Trading System are sent to RTC during the trading day. The following trade types are allowed into the clearing system:

- On Book Trades (automated central order book trades)
- Off Book Trades (reported trades)
- Trade cancellations (of automatic or reported trades)
- Price Adjust

The following events are published in broadcast flows when trades are uploaded to RTC:

Event Flow	Event	Notes
Account Event Flow	AccountPositionEvent	<p>The inserted trade will trigger two AccountPositionEvents: one for the buy account and one for the sell account.</p> <p>These events will be published on the subscription groups related to the accounts.</p> <p>The AccountPositionEvents also include the position updates as a result of the trade.</p>
Risk Event Flow	RiskNodeEvent	<p>An inserted trade will trigger a risk calculation for each of the member risk nodes.</p>

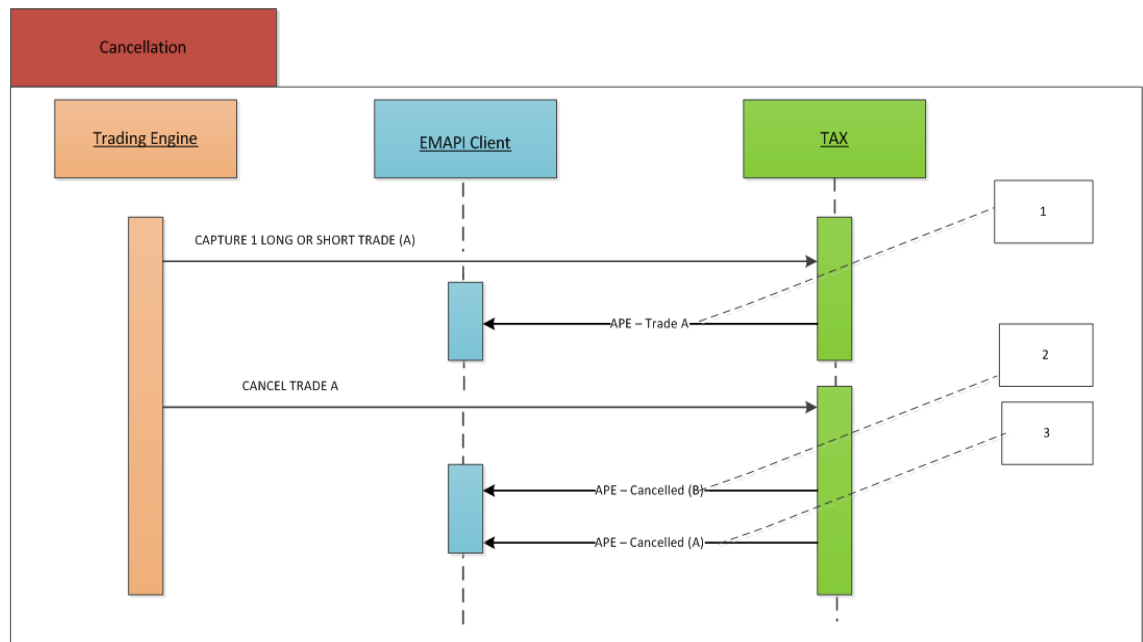
Note: An uploaded trade will affect the public Open Interest of the instrument but this data is not published in EMAPI. Please refer to Market Data Gateway specifications on the trading system for details on how to receive this data (See 1.7).

The trade field the [AccountPositionEvent](#) contains the [Trade](#) that triggered the account position event. The [Trade](#) object represent one side of the matched trade: the buyer's or the seller's.

The original information from the trading system is embeded "as-is" in the [Trade](#) message as a [RtcTradeExternalData](#) object. The trade from trading system is identified by the `tradingSystemMatchId` field.

8.1.1 Trade Cancellation

Explanation on how the trade cancellations work between the Trading System and RTC.



Label 1:

Trade (A) - Original Trade

Label 2:

Cancelled Trade (B)

New equal and opposite trade with position reason "cancelled" to off-set the position. The originalTradingSystemMatch ID is populated with the TradingSystemMatchID on Trade A to link the cancellation trade with the original trade

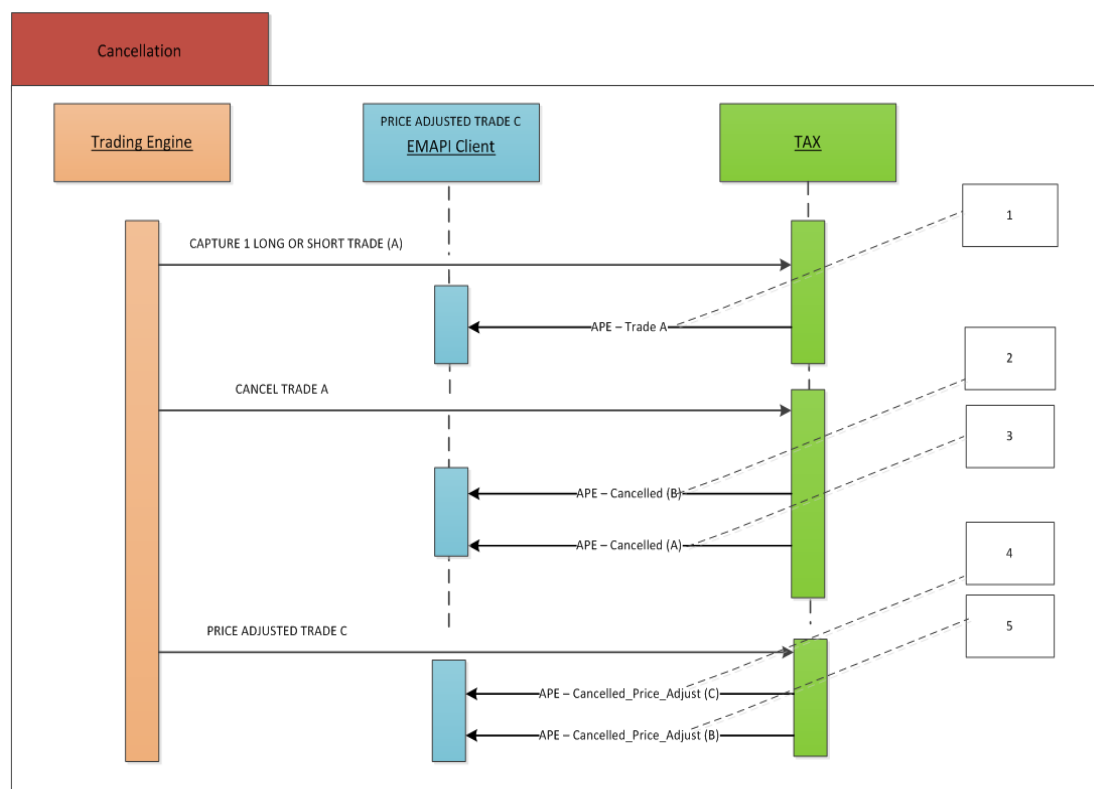
Label 3:

Cancelled (A)

Original Trade with position reason updated to Cancelled

8.1.2 Trade Cancellation Price Adjust

Explanation on how the price adjusts works between the Trading System and RTC.



Label 1:

Trade (A) - Original Trade

Label 2:

Cancelled Trade (B)

New equal and opposite trade with position reason "cancelled" to off-set the position. The originalTradingSystemMatch ID is populated with the TradingSystemMatchID on Trade A to link the cancellation trade with the original trade

Label 3:

Cancelled (A)

Original Trade with position reason updated to Cancelled

Label 4:

Cancelled_Price_Adjust (C)

New trade booked at the correct price with the original quantity

Label 5:

Cancelled_Price_Adjust (B)

Cancelled Trade's (B) position reason updated to Cancelled_Price_Adjust

9 TRADE AND POSITION MANAGEMENT

9.1 Trade and Position Management

Trade and position management is performed by members on trades for the current trading day. These activities must be performed before the clearing system reaches the `END_OF_TRADE_MANAGEMENT` state.

Note: The JSE can in some specific cases perform trade and position management functions on behalf of members from the RTC front-end.

Note: Clearing Members can also perform trade and position management on behalf of their trading members. CMs do not need to change how they connect to RTC in order to carry out on-behalf-of trade and position management.

Trade management requests must include the `tradeId` assigned by the clearing system. This `tradeId` can be obtained from the `AccountPositionEvent`.

Note: Clients should include the `tradingUserId` in the trade management messages in the table below. The `tradingUserId` will be provided by the JSE as part of the taken-on process.

External participants can subscribe to the results of trade and position management activities by setting up a snapshot or replay subscription to the `Account Event Flow`. In order to receive all account events, a client must set up subscriptions to all subscription groups it can access. Note the following when subscribing to the broadcast flow:

- When a trade is entered or modified, an `AccountPositionEvent` is published, reflecting the updated position. The trade is included in the `AccountPositionEvent`.
- The trade attribute in `AccountPositionEvent` is always empty in a snapshot of current position as the trade is sent separately in an `AccountTradeEvent`.
- On a replay subscription, the `AccountPositionEvent` includes the trade, and no `AccountTradeEvent` is published.

The `AccountPositionEvent` message can include settlement positions and collateral positions. Settlement positions have `positionType=SETTLEMENT` and a settlement date. Collateral positions are positions for a collateral account.

The following trade management functions are allowed for external participants:

#	Scenario	Request/Response Message	Notes
1	Allocate a trade	<code>AllocateTradeReq</code> <code>AllocateTradeRsp</code>	<p>The purpose of trade allocation is to allocate or split a trade from a house account to an own client account.</p> <p>Allocation is performed by making an opposite trade on the original account and an equal trade on the receiving account.</p>

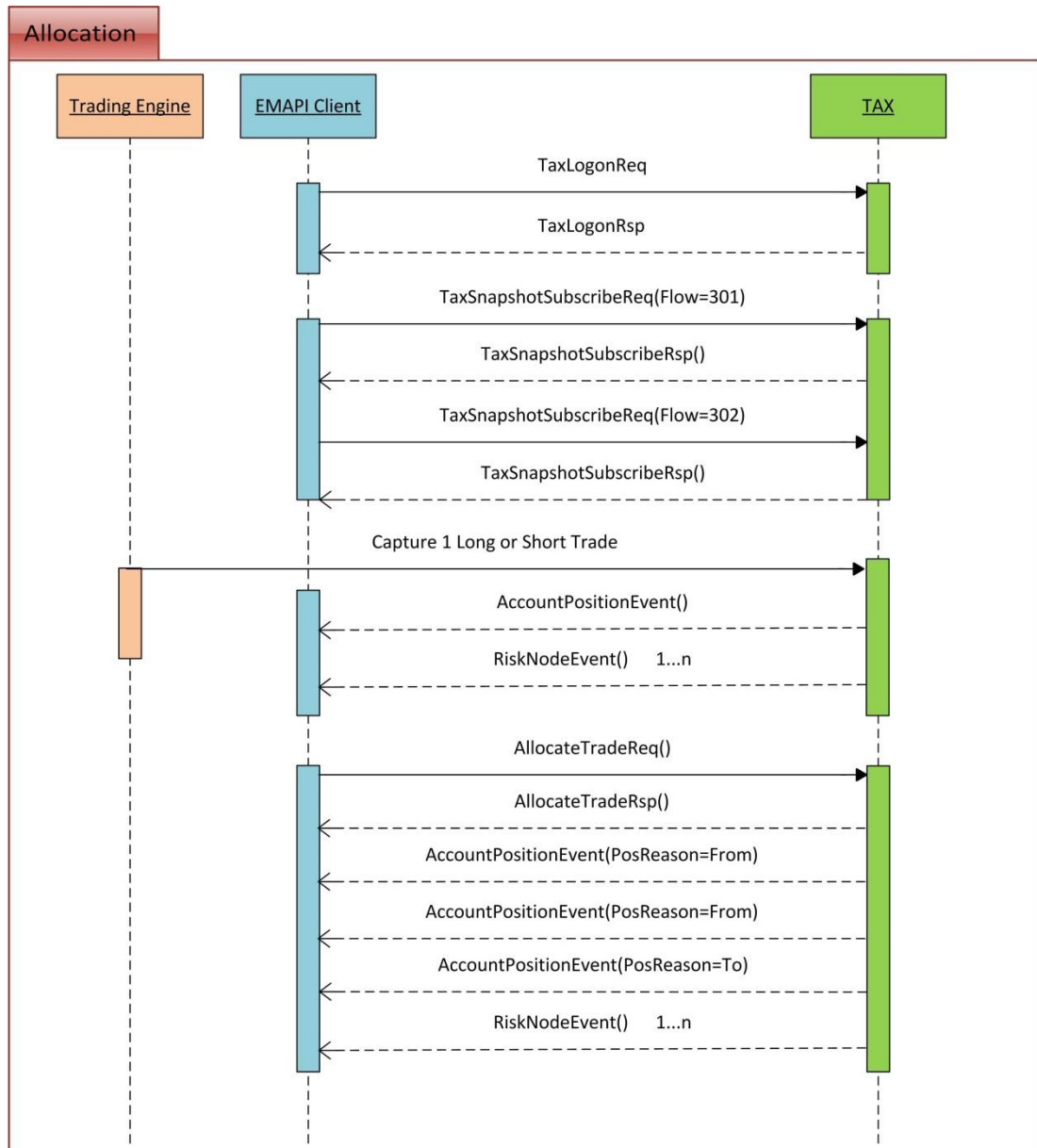
#	Scenario	Request/Response Message	Notes
2.	Correct allocation error	CorrectAllocationErrorReq CorrectAllocationErrorRsp	<p>To correct allocation when a trade has erroneously been allocated to wrong client, i.e. to move the trade from one client to another.</p> <p>The system will create an opposite trade in the original account (to even out the original position) and a new trade in the new account. The original trade will be updated to have an active quantity of zero.</p>
3.	Correct principal	CorrectPrincipalReq CorrectPrincipalRsp	<p>To move a trade from a client account to a member main or sub account.</p> <p>The system will create an opposite trade in the original account (to even out the original position) and a new trade in the new account. The original trade will be updated to have an active quantity of zero.</p>
4.	Modify trade sub account	ModifyTradeSubAccountReq ModifyTradeSubAccountRsp	<p>To move a trade from house accounts to other house accounts.</p> <p>The system will create an opposite trade in the original account (to even out the original position) and a new trade in the new account. The original trade will be updated to have an active quantity of zero.</p>
5.	Modify position sub account	ModifyPositionSubAccountReq ModifyPositionSubAccountRsp	<p>Request to move a position from member house main account to member sub account or client suspense account.</p> <p>The system will create an opposite trade in the original account (to even out the original position) and a new trade in the new account. The original trade will be updated to have an active quantity of zero.</p>
6.	Accumulate trades	AggregateTradesReq AggregateTradesRsp	<p>A number of trades can be accumulated into a single trade with a volume weighted average price. The trades need to be on the same account, the same contract and the same side (only buy or only sell) from current day.</p> <p>The original trades are updated to have a zero quantity, and a new trade is created with the sum of the quantities of the accumulated trades.</p>

#	Scenario	Request/Response Message	Notes
7.	Assign trade (initiator)	AssignTradeReq ResponseMessage	<p>A trading member can assign a trade to another trading member. The trade is margined with the new member once the assign is approved. The appointed trading member can chose to reject the assign request and the trade will continue to be the responsibility of the original trading member.</p> <p>Assign requests that are not handled during the day are removed by the system.</p>
8.	Allocate trade (tripartite) (initiator)	TripartiteAllocationReq TripartiteAllocationRsp	<p>A tripartite agreement is an agreement between two members and a client where one member is allowed to report or allocate trades for another member's client. The position holding member must accept the trade before the risk is moved to him.</p> <p>Tripartite requests that are not handled during the day are removed by the system.</p>
9.	Cancel assigned or tripartite trade (initiator)	CancelGiveUpReq ResponseMessage	The recipient of the GiveUpEvent will then be notified with a new GiveUpEvent where the state is set to CANCELLED.
10.	Accept assigned or tripartite trade (recipient)	ApproveGiveUpReq ResponseMessage	The recipient will be notified by a GiveUpEvent on the GiveUp Event Flow.
11.	Reject assigned or tripartite trade (recipient)	RejectGiveUpReq ResponseMessage	The recipient will be notified by a GiveUpEvent on the GiveUp Event Flow.
12.	Request early exercise of an option position	ExerciseOptionPositionReq ExerciseOptionPositionRsp	<p>For American style options, this is allowed at any time during the contract's lifetime. For European style options, this can only be done on the expiry day.</p> <p>Please refer to contract specifications for the cut-off times for option exercise.</p>
13.	Request abandon of an option position	AbandonOptionPositionReq AbandonOptionPositionRsp	<p>For American style options, this is allowed at any time during the contract's lifetime. For European style options, this can only be done on the expiry day.</p> <p>Please refer to contract specifications for the cut-off times for option exercise.</p>

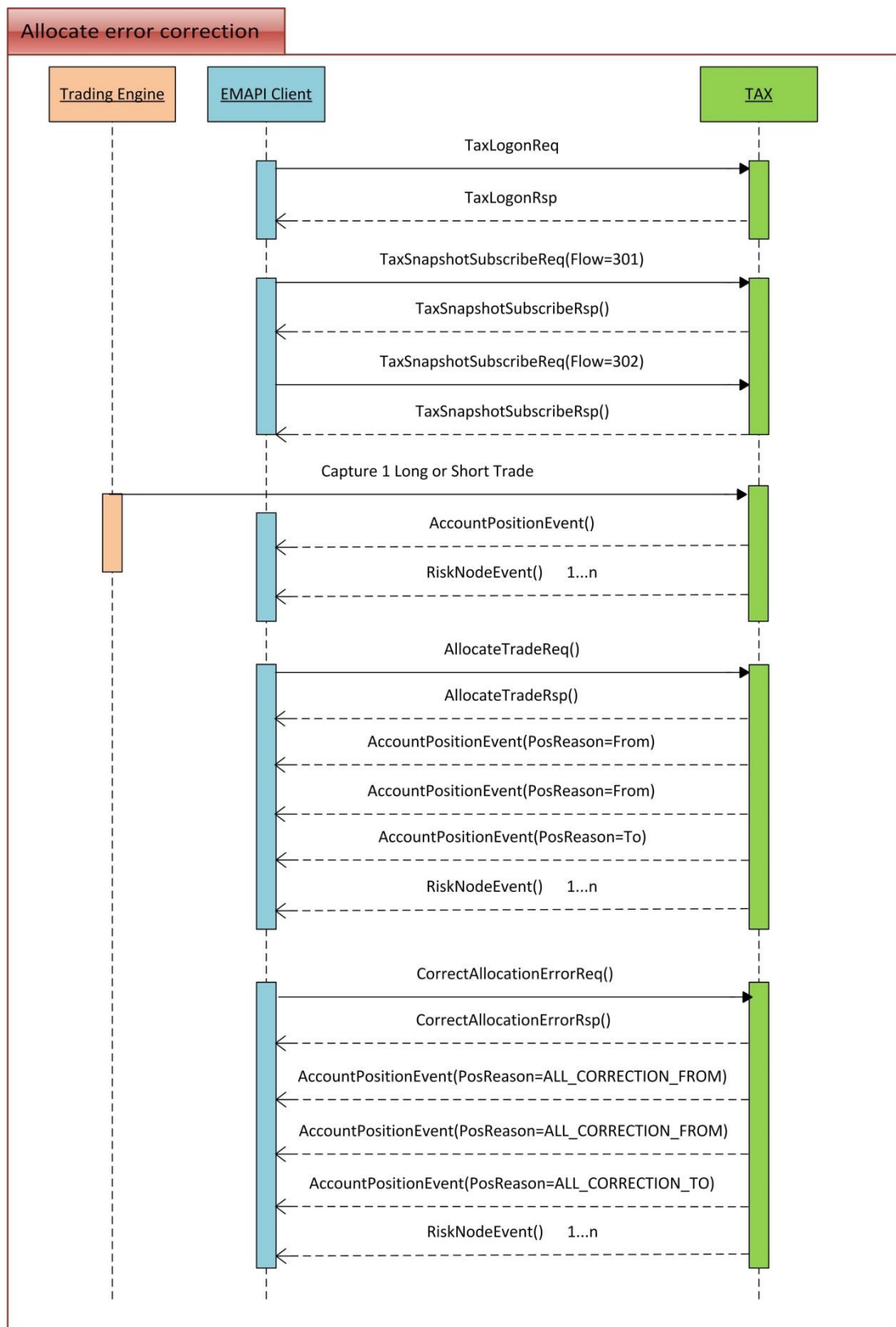
#	Scenario	Request/Response Message	Notes
14.	Update the reference for a trade	UpdateTradeReferenceReq UpdateTradeReferenceRsp	A new AccountPositionEvent will be published, containing the trade with the updated reference.

9.2 Deal Management Sequence Diagrams

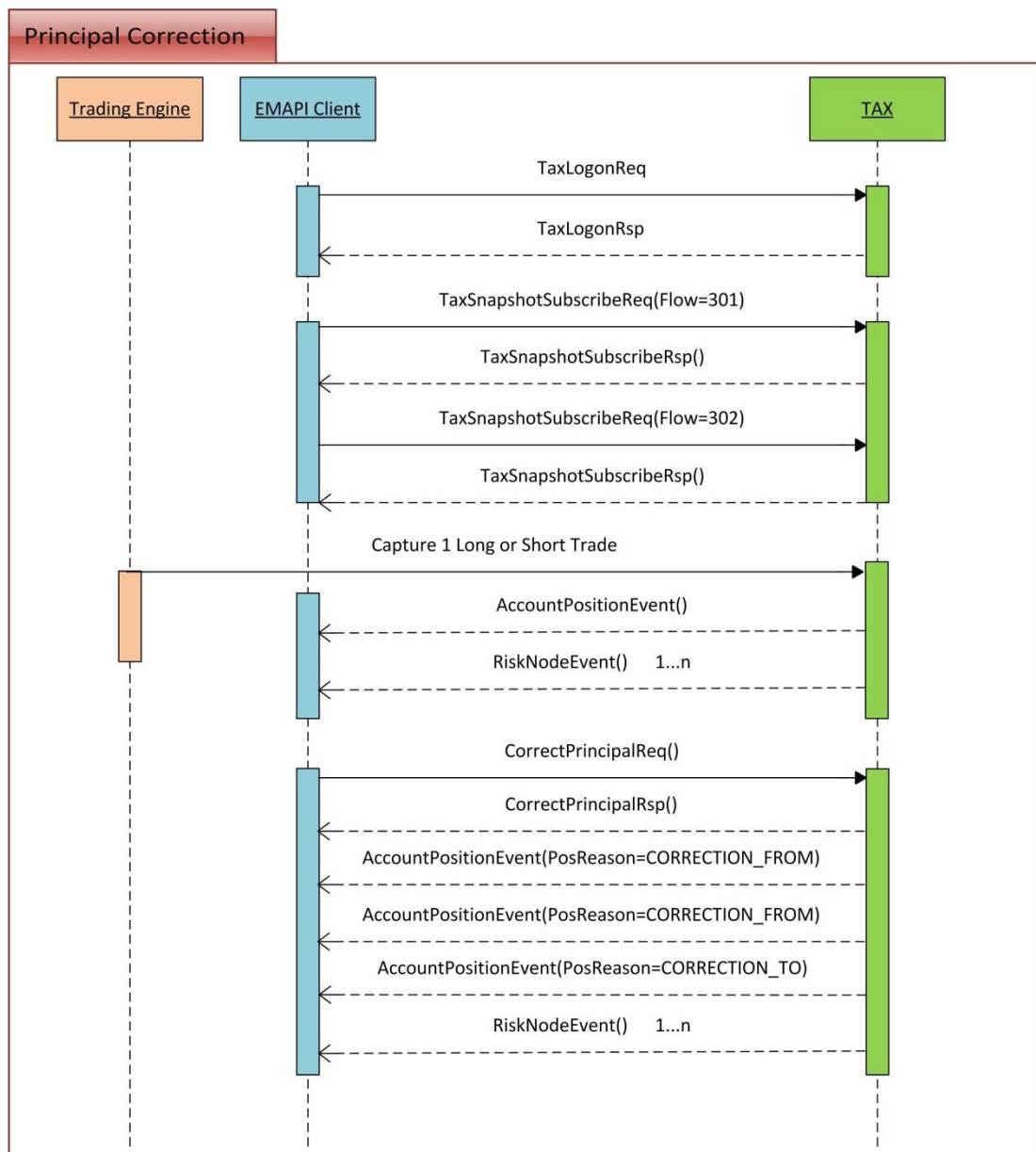
9.2.1 Allocate Trade



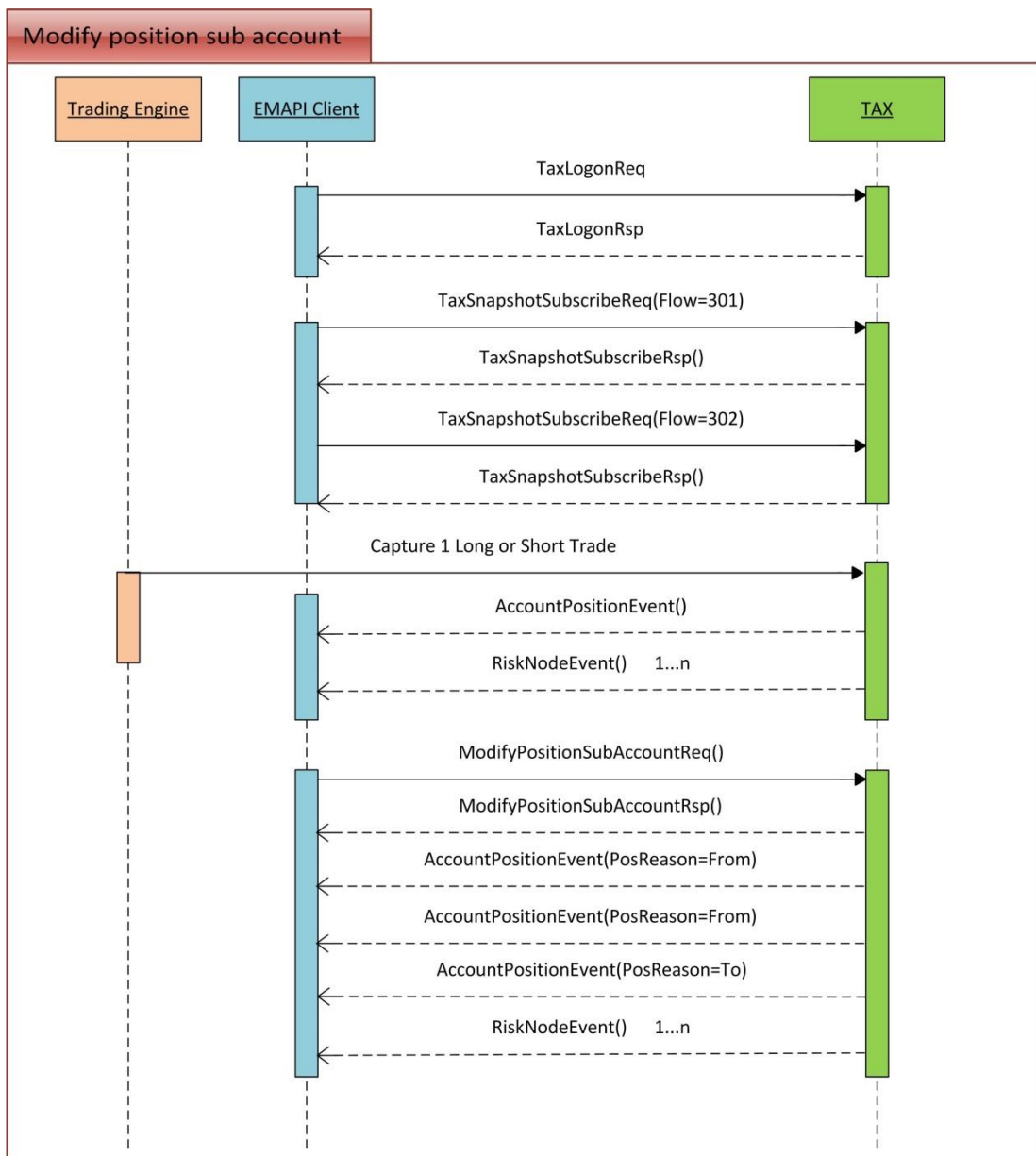
9.2.2 Correct Allocation Error



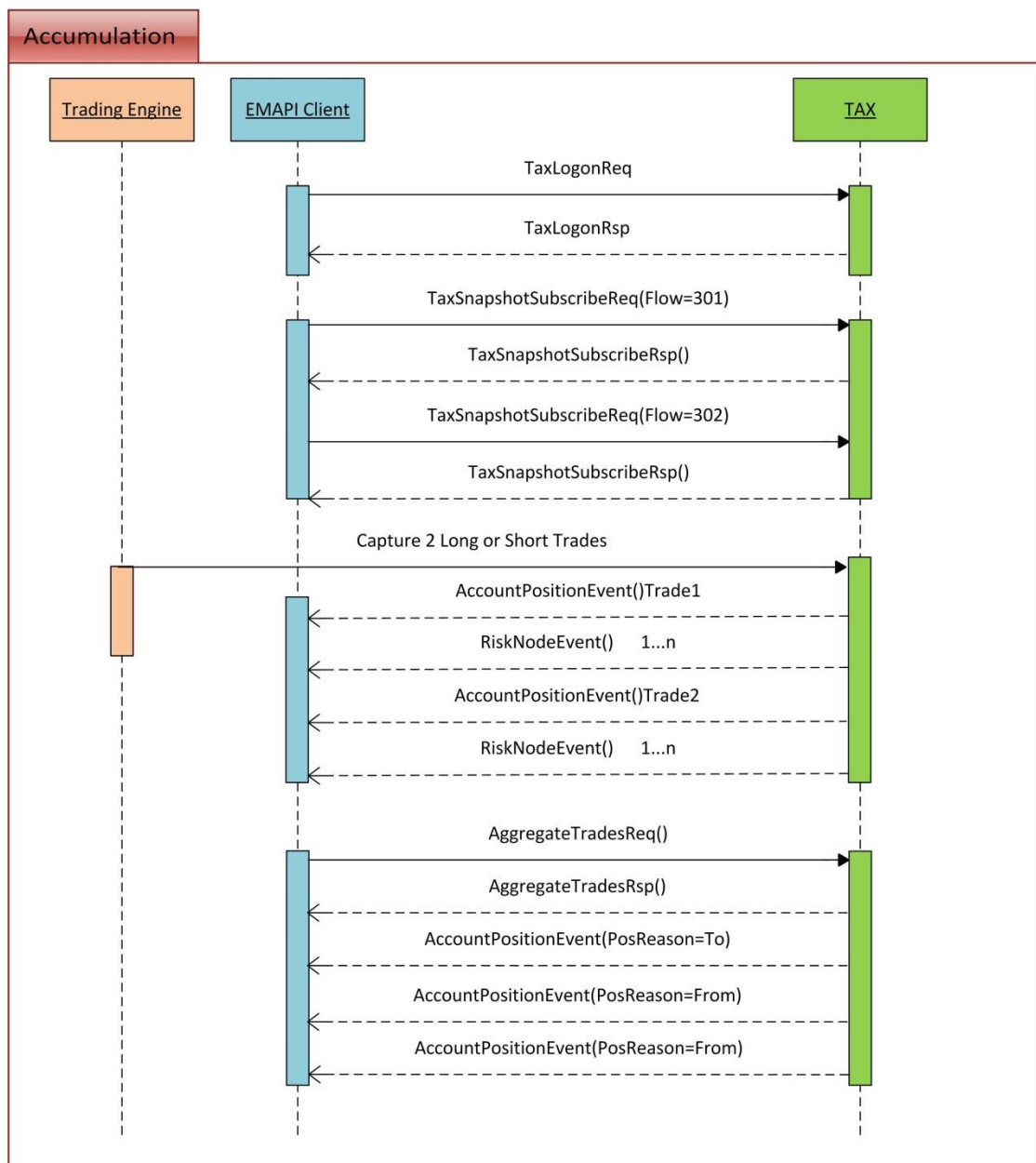
9.2.3 Correct Principal



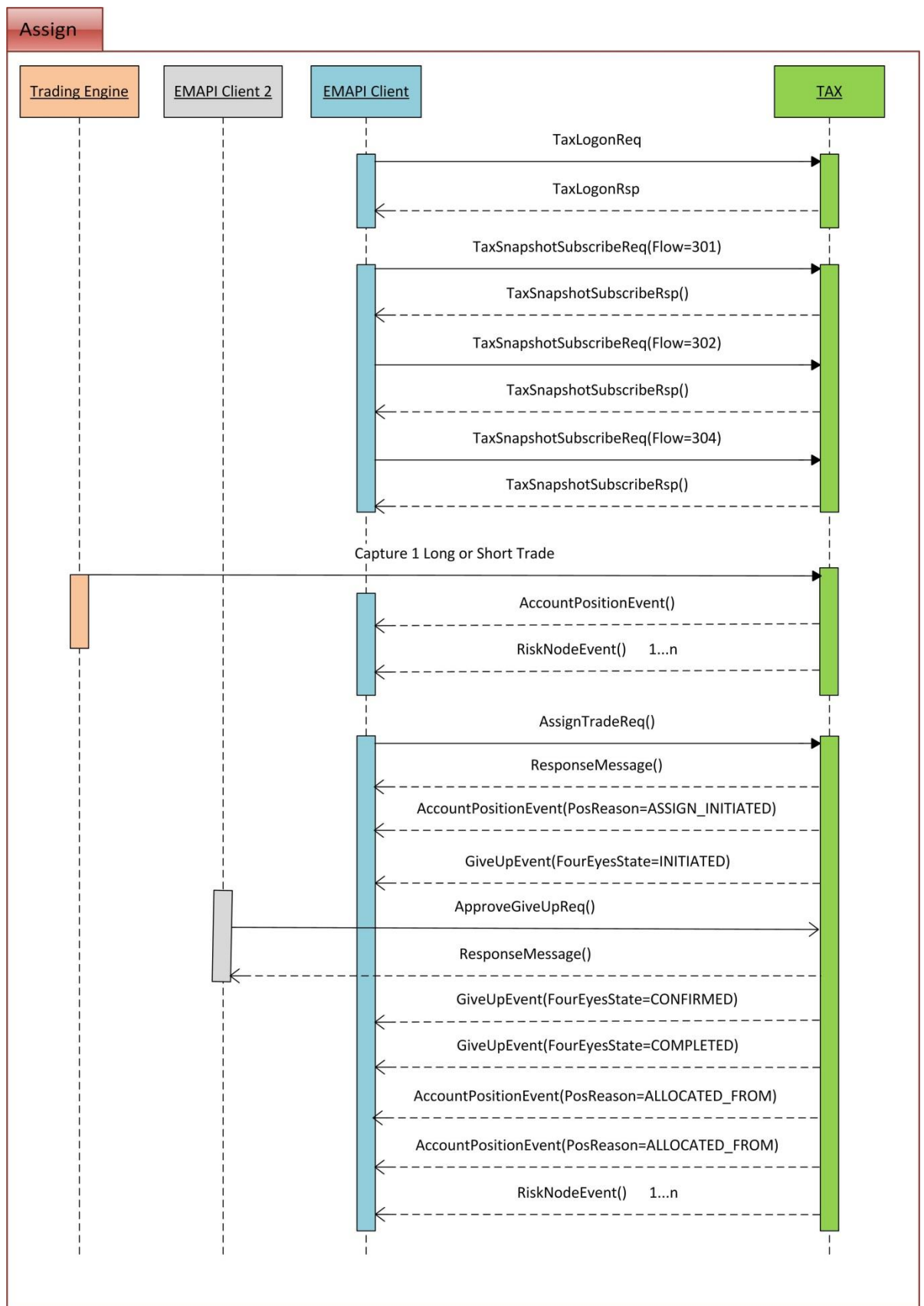
9.2.4 Modify Trade Sub Account



9.2.5 Accumulate Trade



9.2.6 Assign Trade



9.2.7 Tripartite Allocation



9.3 Commission Management

Commissions is sent in separately from trades. They are handled as separate objects, but can reference trades and clients using the reference fields in the requests.

Commission can be added, accepted, rejected and cancelled by the recipient and cancelled if the system is in OPEN state. No changes are allowed when End of Day has started.

Commission functions are open both for Trading Members and Clearing Members. Payments are created accordingly and included in the netted clearing member payment in the End of Day run. Daily Account Summary statements include the netted amount in a separate `Commission` field.

A `CommissionEvent` is published to both participants on the `ACCOUNT_EVENT_FLOW` for every new or updated commission object. The events are included in snapshot and replay on current business day.

#	Scenario	Request/Response Message	Notes
1.	Add Commission	AddCommissionReq	<p>Add commission from between the initiating Trading Member and destination Trading Member. Initiating Trading Member can add commission on behalf of its own branch Members.</p> <p>If the initiating TM/Branch is not the same as destination TM/Branch, the commission gets the status PENDING. It will then await an accept request from the destination TM/Branch. No settlement position is updated/created at this point.</p> <p>If the initiating TM/Branch is the same as destination TM/Branch, the commission gets the status NEW and the settlement positions are updated accordingly.</p>
2.	Accept Commission	AcceptCommissionReq	Destination TM/Branch can accept a PENDING commission sent by an initiator within the same business day. When accepted the settlement positions are updated accordingly.
3.	Cancel Commission	CancelCommissionReq	Initiating Trading Member can cancel a commission sent earlier same day. Initiating Trading Member can cancel commission on behalf of its own branch Members.
4.	Reject Commission	RejectCommissionReq	Receiving Trading Member (destination) can reject a commission sent by an initiator within same business day. Destination Trading Member can add commission on behalf of its Branch Members.

9.3.1 Handling of commission scenarios and reference fields

In order to ensure an efficient, working commissions process it is necessary that the front end solutions used by different trading members are aligned in certain aspects, namely the population of the `clientReference` and `commissionReference` fields of the commission message (`commissionReference` will contain the reference to the deal associated with the commission). The purpose of Appendix B is to:

- Provide an indication of how front ends can manage the process for creating pending commissions and submitting these into the JSE Clearing System.

Note: This aspect of the appendix provides some indicative options for managing this process but each software provider is to determine how their solution can best manage this process.

- Specify how the two reference fields on the commission entry message must be populated to allow the counterparty to recognise and process the commission and allow Clearing Members to effectively process commissions at EOD.

Note: This aspect of the appendix must be fully adhered to in order to ensure the necessary alignment across all participants' solutions

Summary of how the two reference fields must be populated in the relevant trading and deal management scenarios:

#	Scenarios	ClientReference	CommissionReference
1.	Assign	TM/Branch code	'NextTradeIDs' field (15) of Trade sub-message (10015), which is field# 72 of AccountPositionEvent (10032) – deal with position reason of 'Assign From'
	Tripartite Allocation	Tripartite Client ID	'NextTradeIDs' field (15) of Trade sub-message (10015), which is field# 72 of AccountPositionEvent (10032) – deal with position reason of 'Tripartite From'
2.	Allocations	'accountID' of client	TradeID field (1) of Trade sub-message (10015), which is field# 72 of AccountPositionEvent (10032) – deal with position reason of 'Allocate To'
3.	Trades directly on client account	'accountID' of client	TradeID field (1) of Trade sub-message (10015), which is field# 72 of AccountPositionEvent (10032) – deal with position reason of 'Trade'
4.	Adhoc commissions	As above depending on destination entity	CommissionReference is not prescribed. They can be used for bulk commissions i.e. a commission charged to a client for execution of multiple trades or to affect a commission adjustment for an erroneous commission from a previous day.
5.	Allocation Correction	'accountID' of client	TradeID field (1) of Trade sub-message (10015), which is field# 72 of AccountPositionEvent (10032) – deal with position reason of 'Allocate To' Note: Ensure commission associated with the initial incorrect client is cancelled.
6.	Principal Correction	N/A	N/A Note: Ensure Commission associated with cancelled trade is Cancelled.
7.	Trade Cancellation	N/A	N/A Note: Ensure Commission associated with cancelled trade is Cancelled.
8.	Price Adjust	'accountID' of client	TradeID field (1) of Trade sub-message (10015), which is field# 72 of AccountPositionEvent (10032) – deal with position reason of 'Price_Adjust' Note: Ensure Commission associated with cancelled trade is Cancelled.

9.	Reported Trades (Charge Counterparty)	TM/Branch code	TradingSystemMatchID Field (10) of RTCTradeExternalData sub-message (10085) of Trade sub message (10015) of AccountPositionEvent (10032). or Trade Capture Report (AE) message i.e. Trade Capture Report for the confirmation of the acceptance of the Trade. Field: TradeID (Tag 1003)
----	---------------------------------------	----------------	---

Note: Appendix B provides details and examples for the above scenarios.

9.4 Trade and Position Admin Functions

The following trade management administration functions are allowed for members:

#	Scenario	Request/Response Message	Notes
15.	Finding trades from previous dates	QueryTradesReq QueryTradesRsp	If the flag <code>hasMore</code> is set in the response, there are too many trades matching the search criteria. The EMAPI client should specify narrower criteria and submit the query again.

9.5 Account Position and Account Trade Events Clarification

Account Position Event (APE) and Account Trade event (ATE) are used with different subscription types. Trade events are sent only for current value subscriptions and position events are sent for all subscription types but will not include trades for current values and certain replay scenarios (described below).

Assume that the following trades are entered into the system during the day:

Trade number	Account ID	Instrument ID	Reason
1	H0001	1000531	Trade
2	H0001	1000531	Trade
3	H0001	1000531	Trade
4	H0001	1000444	Trade

Example 1: ACCOUNT_EVENT_FLOW with TYPE = SUBSCRIPTION

If the user sets up a subscription on the account event flow before any of the trades were entered, the user will receive the following events once the trades are entered:

Trade number	Event	Comment
1	APE (with trade)	Trade within the APE populated with trade detail
2	APE (with trade)	Trade within the APE populated with trade detail

3	APE (with trade)	Trade within the APE populated with trade detail
4	APE (with trade)	Trade within the APE populated with trade detail

Example 2: ACCOUNT_EVENT_FLOW with TYPE = CURRENT

If the user sets up a subscription on the account event flow with the subscription type set to current after all the trades were entered, the user will receive the following events:

Trade number	Event	Comment
For 1 to 3	APE (No trade)	This will give you the event of what the position on the instrument and account is at the moment of the subscription (snapshot of position). This is the result of trade 1 to 3.
For 4	APE (No trade)	This will give you the event of what the position on the instrument and account is at the moment of the subscription (snapshot of position). This is the result of trade 4
1	ATE	
2	ATE	
3	ATE	
4	ATE	

Example 3: ACCOUNT_EVENT_FLOW with TYPE = CURRENT + SUBSCRIPTION

If the user sets up subscription on the account event flow with the subscription type set to current, after trade 1 and 2 have been entered but not yet trade 3 and 4, the user will receive the following events:

Trade number	Event	Comment
For 1 and 2	APE (No trade)	This will give you the event of what the position on the instrument and account is at the moment of the subscription (snapshot of position). This is the result of trade 1 and 2.
1	ATE	
2	ATE	
3	APE (with trade)	Trade within the APE populated with trade detail
4	APE (with trade)	Trade within the APE populated with trade detail

Example 4: ACCOUNT_EVENT_FLOW with REPLAY

After inserting one trade in the system

Trade number	Account ID	Instrument ID	Reason
1	H0001	1000531	Trade

If the user sends a replay request on the account event flow, the user will receive the following events:

Trade number	Event	Comment
1	APE (with trade)	Trade within the APE populated with trade detail

Example 4.1: ACCOUNT_EVENT_FLOW with REPLAY_SUBSCRIPTION

The user sends a replay request with request type as `REPLAY_SUBSCRIPTION` before entering the trades the user will receive the following events:

Trade number	Event	Comment
1	APE (with trade)	Trade within the APE populated with trade detail

After inserting another two trades in the system

Trade number	Account ID	Instrument ID	Reason
1	H0001	1000531	Trade
2	H0001	1000531	Trade

The user will receive the following events in the subscription:

Trade number	Event	Comment
2	APE (with trade)	Trade within the APE populated with trade detail
3	APE (with trade)	Trade within the APE populated with trade detail

Example 4.2: ACCOUNT_EVENT_FLOW with REPLAY_SUBSCRIPTION (Disconnection)

If the replay subscription disconnected and a new subscription set up, the user will receive the following events:

Trade number	Event	Comment
1	APE (with trade)	Trade within the APE populated with trade detail
2	APE (with trade)	Trade within the APE populated with trade detail
3	APE (with trade)	Trade within the APE populated with trade detail

Example 4.3: ACCOUNT_EVENT_FLOW with REPLAY (Corporate Actions)

Corporate actions and transfer trades executed during EOD for the above trades which results new trades effective from next business day.

If the replay subscription is setup after EOD finished and before system shutdown.

The user will receive the following events:

Trade number	Event	Comment
1	APE (with trade)	Trade within the APE populated with trade detail created by the Corporate Action
2	APE (with trade)	Trade within the APE populated with trade detail created by the Corporate Action
3	APE (with trade)	Trade within the APE populated with trade detail created by the Corporate Action

Example 4.4: ACCOUNT_EVENT_FLOW with REPLAY (System Start Up Before roll over)

If the user sends a replay request on the account event flow after the system starts up in the morning before business day roll over and the system still in POST EOD state, the user will receive the following events:

Trade number	Event	Comment
1	APE (No trade)	This will give you the event of what the position on the instrument and account is at the moment of the subscription (snapshot of position). This is the result of trade 1.
2	APE (No trade)	This will give you the event of what the position on the instrument and account is at the moment of the subscription (snapshot of position). This is the result of trade 2.
3	APE (No trade)	This will give you the event of what the position on the instrument and account is at the moment of the subscription (snapshot of position). This is the result of trade 3.
1	ATE	Contains the transferred trade details.
2	ATE	Contains the transferred trade details.
3	ATE	Contains the transferred trade details.

Note: The replay subscription will also publish APEs with no trades for all other positions in the system for that member

Example 4.5: ACCOUNT_EVENT_FLOW with REPLAY (System Start Up After roll over)

If the user sends a replay request on the account event flow after the system starts up in the morning after business day rolled, the user will receive the following events:

Trade number	Event	Comment
1	APE (No trade)	This will give you the event of what the position on the instrument and account is at the moment of the subscription (snapshot of position). This is the result of trade 1.
2	APE (No trade)	This will give you the event of what the position on the instrument and account is at the moment of the subscription (snapshot of position). This is the result of trade 2.
3	APE (No trade)	This will give you the event of what the position on the instrument and account is at the moment of the subscription (snapshot of position). This is the result of trade 3.
1	ATE	Contains the transferred trade details.
2	ATE	Contains the transferred trade details.
3	ATE	Contains the transferred trade details.
5	APE (with trade)	Trade within the APE populated with trade detail
6	APE (with trade)	Trade within the APE populated with trade detail

After inserting one trade to the system

Trade number	Account ID	Instrument ID	Reason
4	H0001	1000531	Trade

If the user sends a replay request on the account event flow, the user will receive the following events:

Trade number	Event	Comment
1	APE (No trade)	This will give you the event of what the position on the instrument and account is at the moment of the subscription (snapshot of position). This is the result of trade 1.
2	APE (No trade)	This will give you the event of what the position on the instrument and account is at the moment of the subscription (snapshot of position). This is the result of trade 2.
3	APE (No trade)	This will give you the event of what the position on the instrument and account is at the moment of the subscription (snapshot of position). This is the result of trade 3.
1	ATE	Contains the transferred trade details.
2	ATE	Contains the transferred trade details.
3	ATE	Contains the transferred trade details.
4	APE (with trade)	Trade within the APE populated with current day's trade details.

9.6 Removal of Account Position

Net zero positions are defined as follow:

Futures and Options

- Net quantity of the position is zero (long quantity + short quantity = 0, where short quantity is represented as a negative number), AND
- Nett market value of the position is zero (long market value + short market value = 0, where short market value is represented as a negative number).

CFDs:

- Net quantity of the position is zero (long quantity + short quantity = 0, where short quantity is represented as a negative number), AND
- Net market value of the position is zero (long market value + short market value = 0, where short market value is represented as a negative number)
- AND
- Net spread volume of the position is zero (long spread volume + short spread volume = 0, where short spread volume is represented as a negative number).

The removal of net 0 positions will be handled as follows in RTC:

RTC does not remove net 0 positions immediately but rather at SOD the following day. Any subsequent trades done on the same instrument and account after the position nets to zero will add on to the existing long and short quantities, long and short market value and long and short spread volume depending on the side (buy/sell) of the trade.

Example:

Trades on Instrument A (Non-CFD)	Long Qty	Short Qty	Long Market Value	Short Market Value	Comment
Trade 1: Buy 3 – Price R2	3	0	R6	R0	Position published on AccountPositionEvent.
Trade 2: Sell 5 – Price R2	3	5	R6	R10	Position published on AccountPositionEvent.

Trade 3: Buy 2 – Price R2	5	5	R10	R10	Position published on AccountPositionEvent Position is not removed.
Trade 4: Buy 8– Price R3	13	5	R24	R10	Position published on AccountPositionEvent; Positions continue to be added to the net 0 position.
Trade 5: Sell 2 – Price R2	13	7	R24	R14	Position published on AccountPositionEvent; Positions continue to be added to the net 0 position.

Any net 0 positions at EOD will be stored at EOD and will only be removed the following day before start of day positions are published.

The AccountPositionEvent messages for net 0 positions at EOD will be published as follows until removed:

Positions on Instrument B (Non-CFD)	Long Qty	Short Qty	Long Market Value	Short Market Value	Comment
At Start of EOD or CM Balancing1	5	5	R10	R10	Net Position is zero.
During EOD after CM Balancing2	5	5	R10	R10	AccountPositionEvent published for net zero position with position reason END_OF_DAY.
At System Start up	5	5	R10	R10	AccountPositionEvent published for net zero position with position reason SYSTEM_STARTUP.
At SOD					No AccountPositionEvent published as position is removed before SOD.

Note: CFD positions with net quantity of zero and long and short spread volumes that are not equal will be removed at SOD on the day following the expiry date of the CFD instrument.

The RTC functionality with regards to removal of net zero positions is also detailed in [Volume PT02 – Post-trade EMAPI Clearing](#), section 9.6.

Note: The long spread volume for a position is the sum of Interest spread * Remaining quantity for all Buy trades belonging to the position. The short spread volume is the sum of Interest spread * Remaining quantity for all Sell trades belonging to the position.

9.7 Manual upload of trades directly to RTC system

The JSE can perform trade management functions in instances that relate to corporate action adjustments and exceptional circumstances to protect the market. The process involves a manual upload of trades directly to RTC by the JSE Clearing Operations team and the JSE has created a trading user in the Clearing System for this specific function. The JSE Clearing Operations team will make use of this trading user when uploading manual trades, making it easier for members to identify the user.

The tradinguserID is 5 numeric value e.g “99999” and the resultant trades will be published as per the current process via Account position Event to the affected members with this specific tradinguserID.

10 END-OF-DAY PROCESSING

RTC enters normal end-of-day processing after the end of trade management. This section describes how to identify when RTC has started and concluded EOD processing; it also describes the steps in the process and any message exchanges between RTC and external participants.

Note: Under exceptional conditions, the JSE may need to re-run the EOD process to correct any errors; in addition, in the rare scenario that the JSE calls for intraday margin, some steps of the EOD process will be run. The steps that get executed under these exceptional scenarios are also provided in this section.

10.1 Normal Processing

The EOD schedule begins when the `currentRtcState` field of the `CurrentSystemState` message is set to `END_OF_TRADE_MANAGEMENT` and ends when the system is set to `POST END OF DAY`.

The `SchedulerState` field of `CurrentSystemState` is set to `NORMAL`.

A `CurrentSystemState` event is published on the `ReferenceDataFlow` indicating the current RTC system state.

10.2 Re-Run

The JSE may be required to repeat some steps of the end-of-day procedure to correct aspects of the end-of-day process such as pricing or margin parameters. The JSE manually cancels previously created settlement instructions with Strate.

The SchedulerState field of CurrentSystemState is set to RERUN EOD.

10.3 Intraday Margin Call

In certain market conditions, a margin call can be made during the day. IM, AM and VM will be calculated, and margin will be called from the CM.

When the call is made, the `RtcState` in the `CurrentSystemState` message published on the Reference Data Flow will remain in the OPEN state.

The SchedulerState of CurrentSystemState is set to INTRADAY MARGIN CALL.

Intraday Margin call includes risk calculation and settlement. `RiskNodeEvents` and `AccountPositionEvents` will be published as result from these steps.

When the Margin Call is finished, i.e. the last entry is finished, the Full Day schedule becomes the active one (`SchedulerState` = NORMAL).

10.4 End-of-Day Schedule

The following table contains the various steps of the EOD process run in RTC and the associated messages that are published to members on the various broadcast flows as well as any reference data published to IDP. The steps are executed in sequential order and all previous steps must have completed successfully before the current step can start.

Note: The columns *Re-Run* and *IMC (Intraday Margin Call)* indicate whether the step is executed in the case of an exceptional re-run or intraday margin call.

#	Step	Description	Associated messages	Re-Run	IMC
0.	Start of EOD	JSE starts EOD when system is in state End Of Trade Management and no more Trade Management functions are allowed.	The CurrentSystemState messages is published with the CurrentRtcState values of END_OF_DAY on the Reference Data Flow:	✓	✓
1.	EOD prices dissemination	Publish EOD prices to IDP.	JSE makes settlement prices available on Information Dissemination Portal (IDP).	✓	✓
2.	Option exercise (incl. options close out)	System generated option exercise.	AccountPositionEvents (positions and trades)	—	—
3.	Future close out	System generated future close out.	AccountPositionEvents (positions and trades)	—	—
4.	VM calculation	Calculate variation margin.	AccountPositionEvents (Type: settlement positions for the VM) RiskNodeEvent	✓	✓
5.	Risk arrays calculation	Calculate J-Span risk arrays and publish to IDP.	JSE makes risk arrays available on Information Dissemination Portal (IDP)	✓	✓
6.	Calculate IM, AM, SM and other J-SPAN components.	Calculate EOD Risk numbers for all risk nodes for all members.	RiskNodeEvent	✓	✓
7.	Calculate funding interest for CFDs	Calculate funding interest for CFDs.	AccountPositionEvents (settlement positions)	✓	—
8.	Calculate dividends for dividend neutrals	Calculate dividend payments in respect of dividend neutral instruments.	AccountPositionEvents (settlement positions) Calculated dividend factors are available on EMAPI using QueryDividendPaymentFactorsReq	✓	—
9.	Calculate Interest on Cash Collateral	Calculate interest on cash collateral in ZAR and on FX collateral in foreign currency.	AccountPositionEvents (settlement positions)	—	—

#	Step	Description	Associated messages	Re-Run	IMC
10.	CM Balancing1 Note: (See Section 12.1 for more details)	Clearing members confirm their balance on IM, AM, VM, funding interest and dividends for dividend neutrals.	CmBalancing1Event published to clearing members. Clearing members confirm using SetCmBalancingStatusReq , referencing CM balancing step 1.	✓	—
11.	Create price message to Strate	RTC updates reference data for EligibleSecurities (Prices and Yields).	Published as EligibleSecurity messages on the Reference Data Flow. Price file sent to sent Strate.	✓	✓
12.	Request securities collateral Note: (See Section 11.3.2 for more details)	JSE requests and processes securities collateral	AccountPositionEvents (collateral positions)	✓	✓
13.	Request FX collateral Note: (See Section 11.3.3 for more details)	RTC calculates the amount that a member is required to cover with FX collateral. Clearing members respond with RegisterFXCollateralReq .	RTC publishes ReadyConfirmAvailableFXEvent to clearing members. AccountPositionEvents (settlement positions in FX)	✓	✓
14.	Calculate ZAR collateral Note: (See Section 11.3.4 for more details)	Calculation of ZAR requirements after non-ZAR and securities collateral has been pledged	AccountPositionEvents (settlement positions in ZAR)	✓	✓
15.	Add fees	JSE billing system sends billing data to RTC.	AccountPositionEvents (settlement positions in ZAR)	✓	—
16.	Add external payments	JSE sends external payments from other systems (for markets not yet migrated to RTC) for the Clearing Members.	AccountPositionEvents (settlement positions in ZAR)	✓	—

#	Step	Description	Associated messages	Re-Run	IMC
17.	Aggregate Member Daily Summary	Consolidate numbers for Daily Account Summary.	Published on Settlement flow: DailyAccountSummaryDetailsEvent AggregatedSummaryTradingMemberEvent AggregatedSummaryclearingMemberEvent	✓	✓
18.	CM Balancing2 Note: (See Section 12.2 for more details)	Clearing members confirm their balancing on booking fees, risk fees, net payment after collateral process and net payments from other systems	CmBalancing2Event published to clearing members. Clearing members confirm using SetCmBalancingStatusReq , referencing CM balancing step 2. RTC publishes AccountPositionEvents (settlement positions for clearing members)	✓	—
19.	Determine net settlement amounts per CM	Netting of settlement amounts per clearing member.	AccountPositionEvents (settlement positions), for the netting of settlement amounts per clearing member.	✓	✓
20.	Generate and send payment instructions	JSE sends the payment instructions to Strate.	The settlement instructions are sent to Strate.	✓	✓
21.	Bulk Client Transfer of defaulting TM Note: (See section 13 for more details)	Transfer clients from a defaulting trading member to another trading member.	Reference data for new clients. AccountPositionEvents including trades for the movement.	—	—
22.	Member transfer Note: (See section 13 for more details)	Transfer member.	Reference data for new member. AccountPositionEvents including trades for the movement.	—	—

#	Step	Description	Associated messages	Re-Run	IMC
23.	Corporate actions	<p>Apply corporate actions for the next business day.</p> <p>Publish trades resulting from a corporate action.</p> <p><u>Publish during EoD on LDT (approximately 22:30) and during system startup on ex-date (approximately 06:00)</u></p>	<p>AccountPositionEvents messages will be published to reflect the updates of the old and new positions.</p> <p>AccountTradeEvent will be published on the Account Event Flow.</p> <p>The updates are effective the next business day.</p> <p>Note: The actual Corporate Action event reference data is only available from IDP (See 1.7).</p>	—	—
24.	Set State Post End of Day	The End Of Day sequence is ended.	<p>CurrentSystemState message with the CurrentRtcState equal to POST_END_OF_DAY is published on the Reference Data Flow.</p>	—	—

10.5 Start of Day schedule

At system startup, the [RtcState](#) in [CurrentSystemState](#) is [POST_END_OF_DAY](#), and the business date is yesterday's date.

While the state is [POST_END_OF_DAY](#), it is possible for the JSE to initiate a re-run of the previous day's End of Day process.

The Start of Day is started automatically at a predefined time. When the Start of Day process has been completed, the business date has changed to today's date and the [RtcState](#) is [OPEN](#). The [CurrentSystemState](#) is published on the Reference Data Flow.

Nr	Step	Description	Associated messages	Link
1	Roll business date	The business date is incremented from the previous business date to today's date.	The RtcState in CurrentSystemState remains in POST_END_OF_DAY . The business date is incremented. This is published on the Reference Data Flow.	
2	Store Start of Day Positions	Start of day positions are published on the Account Event flow	AccountPositionEvent messages with positionReason START_OF_DAY are published.	
3	Set State Open	The system state is changed to OPEN .	The RtcState in CurrentSystemState is changed to OPEN . This is published on the Reference Data Flow.	

10.6 Account Position Events and Sequence Numbers on Start of Day

Assume there are positions in the system from the previous day. At system startup, all position events are published with the sequence number starting at 1. The position events contain all types (ACTUAL and SETTLEMENT) with the position reason SYSTEM_STARTUP. At this point no user subscriptions has been setup. For example, the system publishes events for subscription group 1031 with sequence numbers 1-21.

Before the schedule entries Open and Roll Business Date, a user subscribes to the account position event with subscription type set to SUBSCRIPTION (future updates only and no current value) and subscription group 1031. At the schedule entry Store Start Day Positions, the user will receive account position events with type ACTUAL and position reason START_OF_DAY. Because the sequence number is incremented for each event on the flow, the first event from the Store Start Day Position will have the sequence number equal to 22. The user will only receive sequence numbers 22 and upwards.

To receive all events sent out, the ReplayReq shall be used.

11 COLLATERAL MANAGEMENT

11.1 Overview

The primary objective of Collateral Management is to provide functionality to manage collateral cover for margin requirements, i.e. Initial Margin (IM) and Additional Margin (AM).

Collateral may be posted in:

- Securities (for example, equities and bonds)
- Foreign currency, e.g. USD, EUR, GBP (by foreign clients only)
- ZAR Cash

The counterparts to handle these collateral types are:

- Strate, South Africa's Central Securities Depository (CSD) for securities collateral
- Authorised Dealers for foreign currency
- Settlement Banks, for ZAR currency

ZAR and FX can cover Additional Margin; securities collateral may be posted by local and foreign clients, whilst FX collateral may be posted by foreign clients only.

The following functions are available to members:

Period	Collateral Management Functions
End of Day	<ul style="list-style-type: none">• Pledge securities and FX• Set minimum cash limit <p>Note: For margin requirement calculated at EOD</p>
Intraday	<ul style="list-style-type: none">• Increase pledge of securities to release cash• Release securities (but members will be required to place more cash - ZAR or foreign cash)• Adjust pledged securities amount to compensate for a change of price of the underlying• Set minimum cash limit <p>Note: For margin requirement as calculated the previous EOD.</p>

11.2 Set Minimum Cash Limit

Scenario	Request/Response Message	Time of day	Notes
Set Minimum Cash Limit	CdSetMinimumZARLimitReq	Any time	<p>The clearing member and trading member sets the minimum limit in percent of the margin that must be covered in cash within the global parameters.</p> <p>Limit set by CM must be higher (more cash) than the limit set by the clearing house (JSE).</p> <p>TM limit must be higher than the CM limit.</p>

11.3 Collateral End-of-Day Process

11.3.1 Overview

The following diagrams illustrate at a high-level the various processes and interactions between RTC (JSE Clear) and other systems during the collateral process at EOD; the message exchange between clearing members and RTC using EMAPI are highlighted in red:

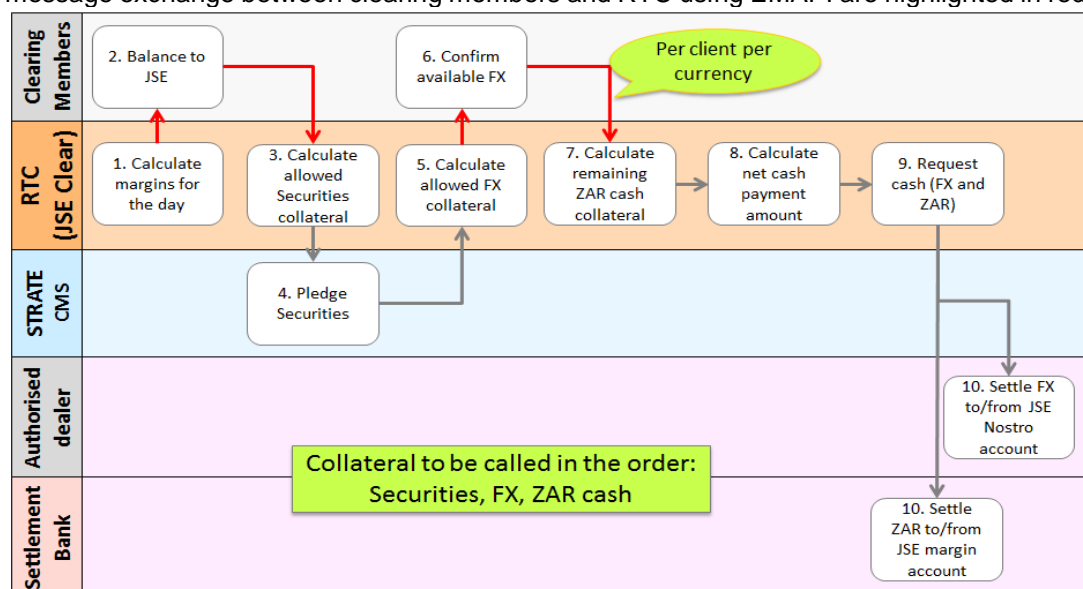


Figure 7 - Collateral EOD Process Overview

The following diagram provides another view of the EOD collateral process, including the interactions with the CSDP; messaging interactions between RTC and the clearing members via the EMAPI interface is highlighted in red:

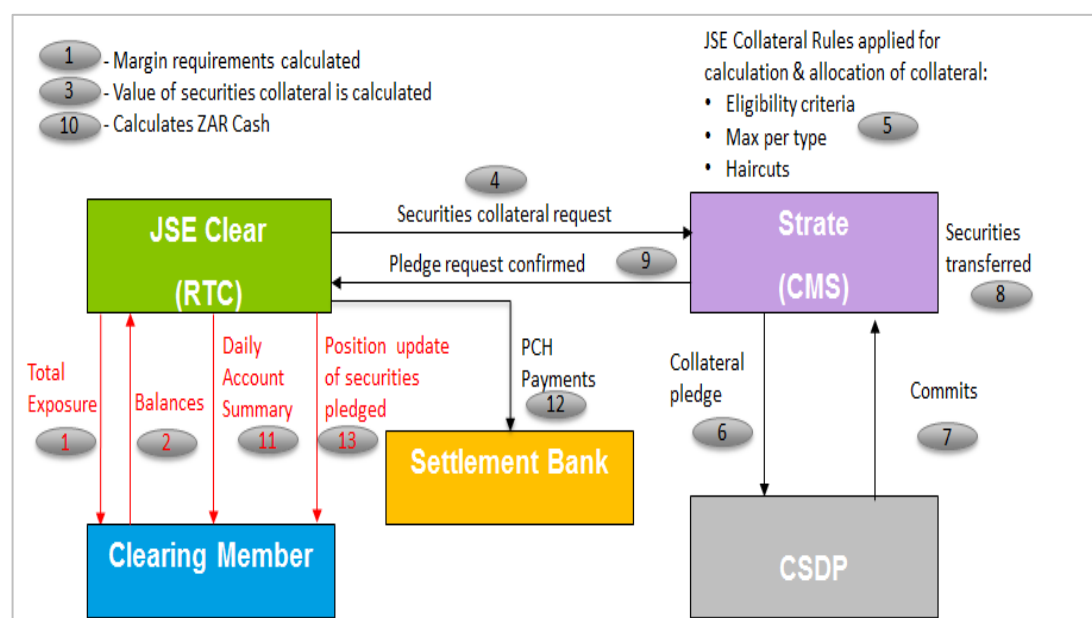


Figure 8 - Collateral EOD Process Overview

Note: From the overview diagrams above, it is clear that the overall collateral EOD process involves many processes and messages that don't rely on the EMAPI protocol covered in this document. Please refer to *Volume PT00 – Post-trade Services Overview* and other related documents (See 1.7) for a more comprehensive view of collateral management.

11.3.2 Securities Collateral Processing

After the margin requirements have been calculated by RTC and the clearing members have balanced with the JSE on CM Balancing 1 (See Section 12), the securities collateral process takes places as per the following table:

Scenario	Messages	Time of day	Notes
Get list of eligible securities	N/A	Any time	The securities allowed for collateral are published to trading and clearing members as EligibleSecurity messages on the Reference Data Flow.
Get list of eligible securities for collateral (Price File)	Strate Requests Price File Price File send to Strate	Any time	The resulting response contains one item per eligible security. Messaging with Strate is not via EMAPI.
Securities Collateral Request	RTC requests security collateral from Strate	End-of-Day	RTC sends EMAPI messages to JSE Integration, however when JSE Integration interfaces with Strate, the message exchanges are not done in EMAPI format. Note: For a client with no margin exposure, RTC will send an exposure amount of zero to Strate unless the client has been deregistered for securities collateral.
Pledge securities	Strate sends pledge request confirmation	End-of-Day	One request is sent for each risk node where the member client has an SDA account. RTC sends EMAPI messages to JSE Integration, however when JSE Integration interfaces with Strate, the message exchanges are not done in EMAPI format.

Note: During EOD securities collateral processing, [AccountPositionEvents](#) are sent to external participants to reflect the updated account positions.

11.3.3 Foreign Exchange Collateral Processing

After securities for collateral are pledged by Strate, RTC calculates the allowed FX for collateral and then requests clearing members to confirm the available FX.

The following table describes the FX Collateral processing and message interactions between RTC and the clearing members via the EMAPI interface:

Scenario	Request/Response Or Event Messages	Time of day	Notes
RTC indicates that clearing members can start the FX collateral process	ReadyConfirmAvailableFXEvent is published on the Settlement Event Flow.	End-of-Day	RTC calculates the amount that a member is allowed to cover with FX collateral.
Clearing member retrieves the amounts in ZAR that a member is allowed to cover with FX collateral	GetRequestsForFXCollateralReq GetRequestsForFXCollateralRsp , including an array of RequestForFXCollateral	End-of-Day	The resulting GetRequestsForFXCollateralRsp contains an array of collateral amounts per entity for the clearing member.
Confirms the amount of FX collateral for each trading member or client	RegisterFXCollateralReq RegisterFXCollateralRsp	End-of-Day	The clearing member registers FX collateral. Note: For an FX client where no FX is required, RTC will send an exposure amount of zero to the CM and expect a response (of zero) unless the client has been deregistered for FX.

Note: During the EOD Foreign Exchange Collateral process, [AccountPositionEvents](#) are sent to the external participants to reflect the updated account positions.

11.3.4 Settlement

After the FX Collateral process, RTC calculates the remaining ZAR cash collateral and net cash payment amounts and sends instructions to the Settlement Bank (ZAR) and Authorised Dealer (FX) once balancing confirmations have been received by Clearing Members on CM Balancing 2.

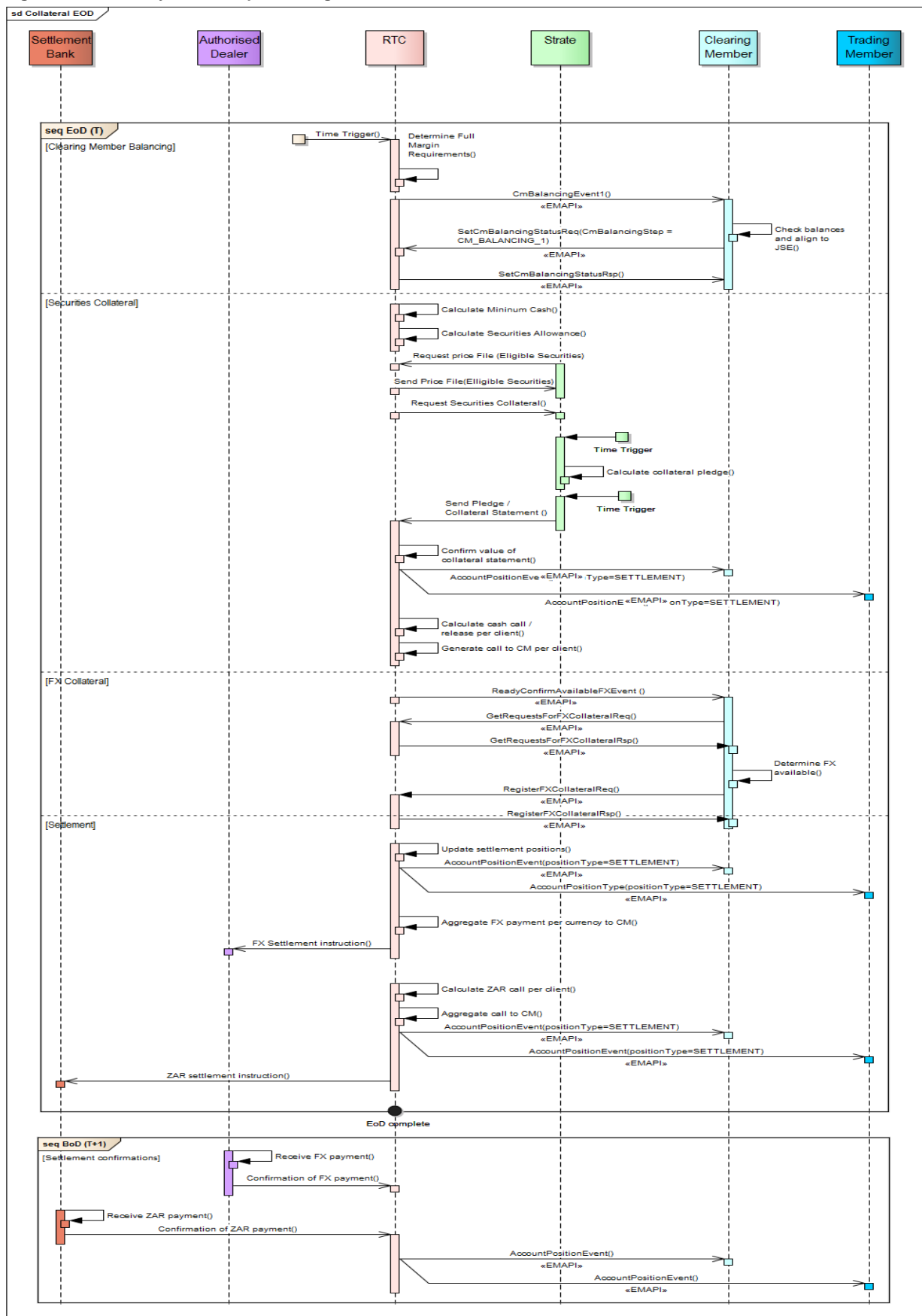
The ZAR cash will be settled to or from the JSE margin account while the FX will be settled to or from the JSE Nostro account.

Note: Please refer to Section 12 for additional information on settlement functions such as the daily account summary published to external participants.

11.3.5 EOD Collateral Sequence Diagram

The sequence diagram on the following page illustrates the message interactions and their timing for the EOD collateral process described in this section.

Figure 9 - End-of-Day collateral processing



11.4 Collateral Intraday Re-balancing

11.4.1 Overview

The collateral re-balancing process facilitates withdrawals, releases, top-ups and substitutions. It is based upon the risk calculated at EOD the night before but using the latest available prices for collateral. Withdrawals are based upon the Payment Advice message created by Strate. The re-balance process happens once a day (currently scheduled to start at 11h00).

The diagram below provides an overview of the rebalancing messaging between clearing members and RTC as well as RTC and Strate. Messaging via EMAPI is highlighted in red:

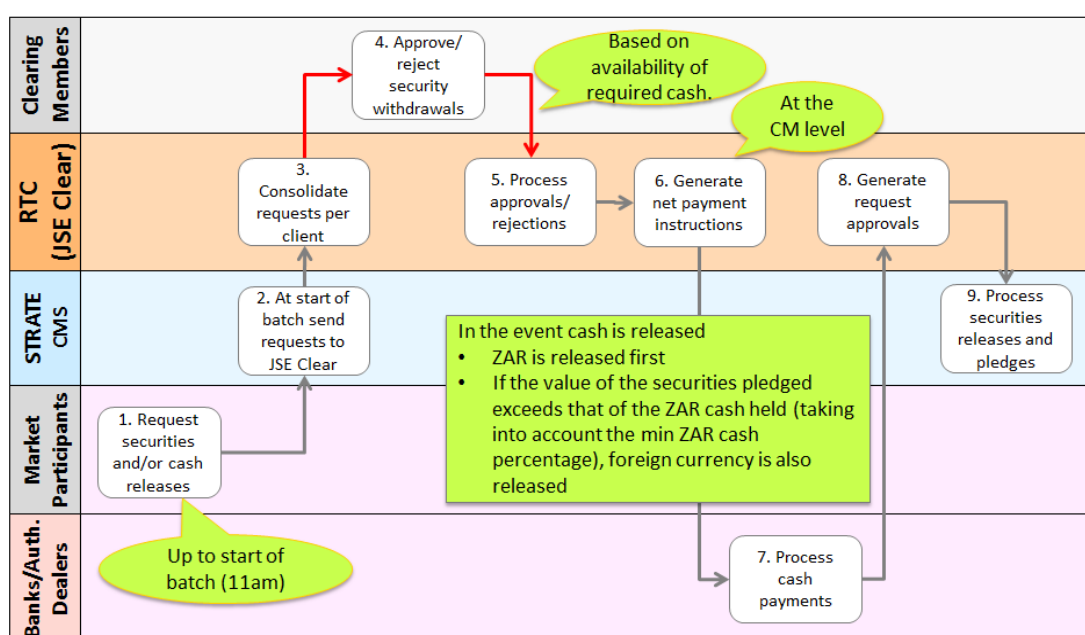


Figure 10 - Collateral Intraday overview

RTC (JSE Clear) sends Strate CMS updated prices at 11:00am to ensure substituting securities are equivalent in value as prices may have changed materially from the prices sent during EOD the day before.

If the value of the pledged securities on an account falls below a specified threshold due to price decreases, additional securities are pledged to ensure the exposure (IM) is covered. If securities are not available, a cash call is generated and is processed intraday if received prior to the start of the EOD batch process.

Note: From the overview diagram above, it is clear that the overall collateral intraday process involves many processes and messages that don't rely on the EMAPI protocol covered in this document. Please refer to Volume PT00 – Post-trade Services Overview and related documents (See 1.7) for a more comprehensive view of collateral management.

11.4.2 Withdrawals Processing

The table below describes the intraday collateral re-balancing scenarios involving the clearing members via the RTC EMAPI interface:

#	Scenario	Request/Response Message	Notes
1	A withdrawal notification event is sent out to CM.	WithdrawalNotificationEvent (on SETTLEMENT_EVENT_FLOW)	The CM receives a notification that there is a set of payment advices in RTC that needs confirmation.
2	CM to fetch the withdrawals that need confirmation	GetPaymentAdvicesReq GetPaymentAdvicesRsp	This operation can be performed any number of times. The response is paged (bookmarked), meaning there will be an indication as to whether there is more information to be retrieved from RTC.
3	CM to confirm or reject one or more withdrawals (payment advices)	ConfirmWithdrawalsReq includes an array of PaymentAdvice	The CM confirms or rejects one or more withdrawal (payment advice). The request has an indication as to whether it is a confirmation or rejection.

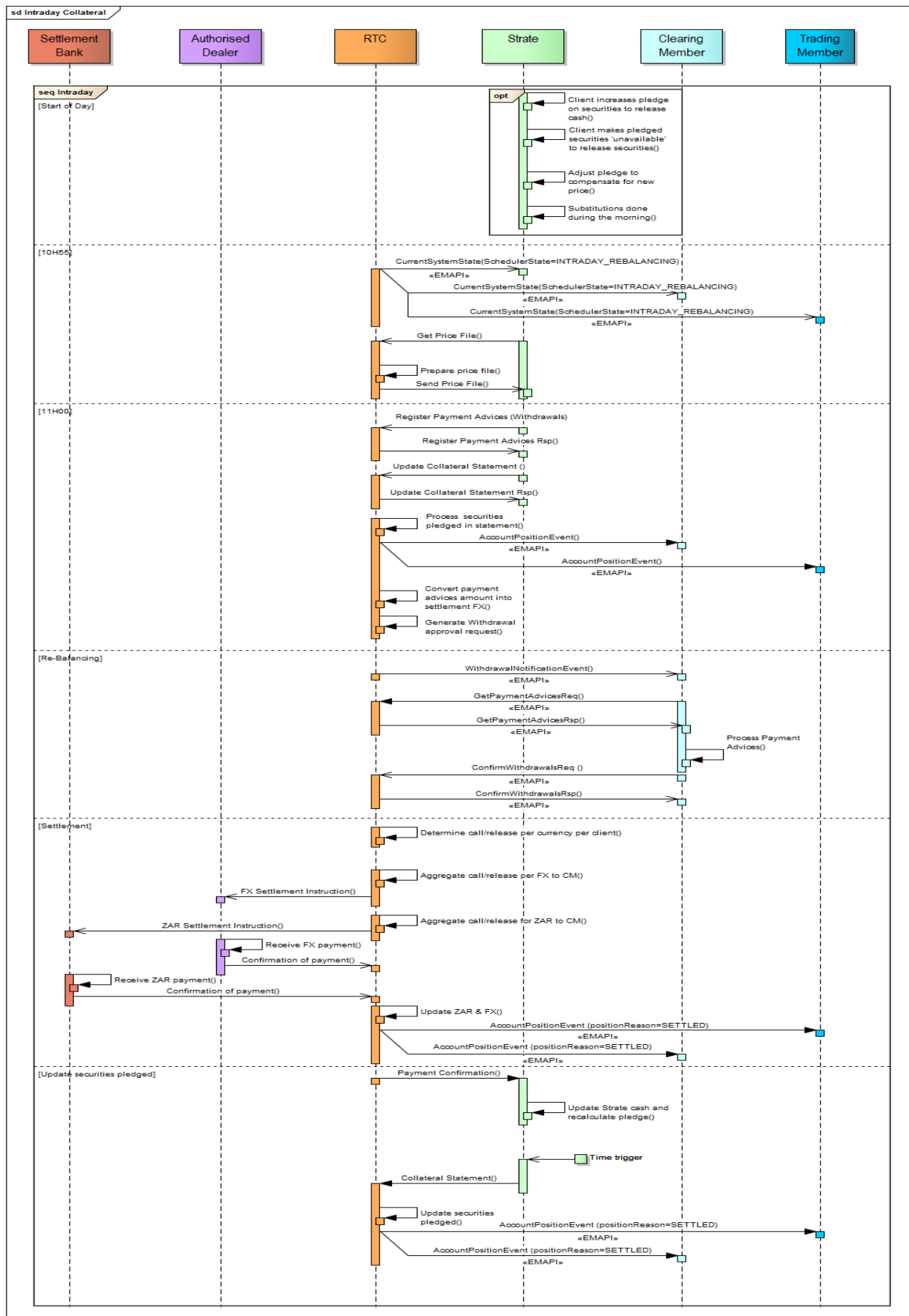
Note: During intraday collateral rebalancing, [AccountPositionEvents](#) are sent to external participants to reflect the updated account positions.

Note: Payments to/from the CM due to normal collateral price fluctuations will not be sent for confirmations; these payments will be made or requested automatically as is done at EOD.

11.4.3 Intraday Collateral Re-balancing Sequence Diagram

The sequence diagram on the following page illustrates the message interactions and their timing during the intraday collateral process described in this section.

Figure 12 - Intraday rebalancing sequence diagram



12 CLEARING MEMBER BALANCING

At two separate points of the end-of-day process, Clearing Members are required to balance to RTC (i.e. RTC waits for the approval on the balancing message), in order for subsequent steps of the EOD procedure to continue.

Balancing is divided into two steps. Once the first balancing request has been confirmed by all clearing members, securities collateral can be requested from Strate by RTC.

12.1 CM Balancing 1

Scenario	Request/Response Message	Time of day	Notes
RTC indicates that Clearing Members confirm their balance on IM, AM, VM, funding interest and dividends for dividend neutrals.	N/A	End of Day	CmBalancing1Event is sent to clearing members on the Settlement Event Flow. The embedded MemberBalance1 contains account summaries per TM. <i>Step #1 in Figure 8.</i>
Clearing members confirm the balancing.	SetCmBalancingStatusReq SetCmBalancingStatusRsp	End of Day	Indicate "balancing step 1" in the request. <i>Step #2 in Figure 8.</i>

12.2 CM Balancing 2

Scenario	Request/Response Message	Time of day	Notes
RTC indicates that Clearing members confirm their balancing on booking fees, risk fees, commissions, net payment after collateral process and net payments from other systems.	N/A	End of Day	CmBalancing2Event is sent to clearing member on the Settlement Event Flow. The embedded MemberBalance2 contains account summaries per TM. <i>Step #1 in Figure 8.</i>
Clearing members confirm the balancing.	SetCmBalancingStatusReq SetCmBalancingStatusRsp	End of Day	Indicate "balancing step 2" in the request. <i>Step #2 in Figure 8.</i>

12.3 CM Balancing Sequence Diagram

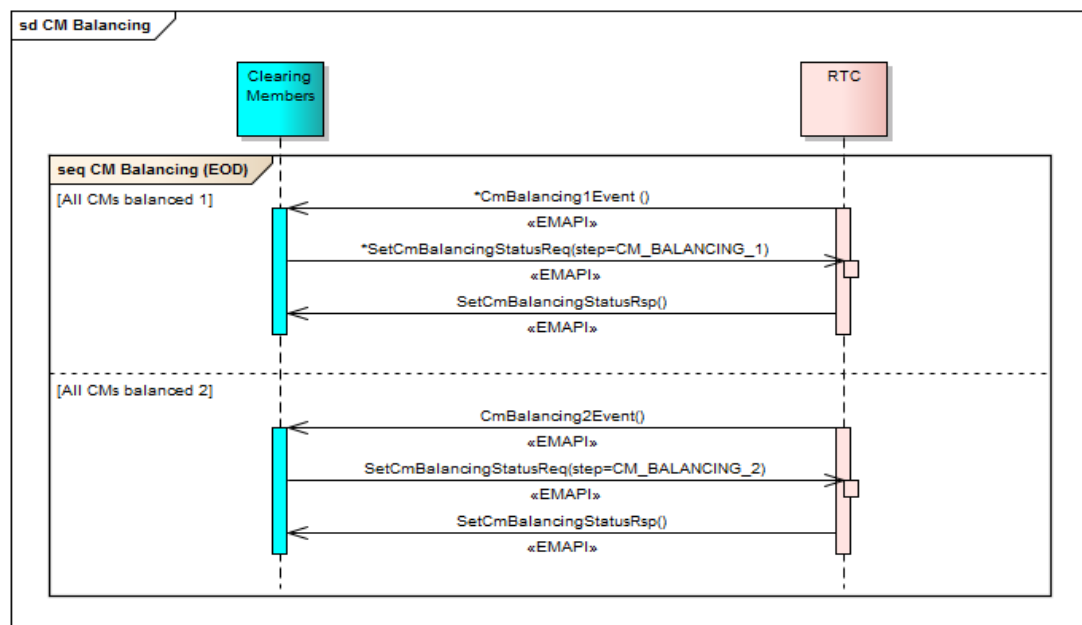


Figure 11 - CM Balancing Sequence Diagram

13 SETTLEMENT MANAGEMENT FUNCTIONS

The clearing system manages settlement of:

- Collateral based upon calculated margins (IM and AM)
- Securities (can only cover IM)
- Cash (ZAR)
- Foreign currency (to cover IM and AM)
- Variation margin
- Funding interest
- Dividend payments
- Interest on cash collateral
- Booking fees
- Netted payments from existing markets

RTC generates EMAPI messages that are converted by the JSE integration layer to the SWIFT messages used by the settlement banks, authorised dealers and Strate.

- For ZAR: messages are be sent to Settlement banks
- For FX: messages to be sent to Authorised dealers.
- Status of settlement instructions are updated in RTC when confirmation are received.
- For securities: messages to be sent Strate (CSD).

Scenario	Request/Response Message	Time of day	Notes
Clearing Members retrieve daily account summary	N/A	End of Day	AggregatedSummaryClearingMemberEvent and AggregatedSummaryTradingMemberEvent is published on the Settlement Event Flow. The subscription groups used are the same as for the Account Event Flow, i.e. one subscription group for each combination of clearing member and trading member or branch.
Clearing Members retrieve the settlement instructions (ZAR or FX)	GetSettlementInstructionsReq GetSettlementInstructionsRsp		The settlement instructions in RTC will be in state PENDING until settlement has been confirmed.

13.1 Signs of values in daily account summary messages

In the `DailyAccountSummaryDetailsEvent`, `AggregatedSummaryTradingMemberEvent` and `AggregatedSummaryclearingMemberEvent` messages, the sign indicates the direction of the cash flows between the client and the CH.

A positive value for the following attributes should be interpreted as money being paid by the client to the CH:

- `totalBookingFee`
- `netFromOtherSystems`
- `totalMemberZarCashMovement`
- `totalVariationMargin`
- `fundingInterest`
- `dividendPayment`
- `commission`
- `totalRiskFees`

A negative value for the following attributes should be interpreted as money being paid by CH to the client:

- `totalInterestAmountOnCashCollateral`
- `netFromOtherSystems`
- `totalMemberZARCashMovement`
- `totalVariationMargin`
- `fundingInterest`
- `dividendPayment`
- `commission`
- `totalRiskFees`

The following attributes are always positive:

- `totalMember_CF_Cash`
- `totalMember_BF_Cash`
- `totalMember_CF_Sec`
- `totalMember_BF_Sec`
- `registeredSecuritiesAmount`
- `totalMember_CF_FXCash`
- `totalMember_BF_FXCash`
- `initialMargin`
- `previousInitialMargin`
- `totalAdditionalMargin`
- `FX Collateral Qty`
- `FX Market Value`
- `FX Collateral Value`

The following attributes can be positive or negative:

- `totalMemberCashMovement` – positive if the current cash collateral exceeds the previous cash collateral; negative if it is the other way around
- `totalMemberSecMovement` – positive if the current securities collateral exceeds the previous cash collateral; negative if it is the other way around
- `totalMemberFXCashMovement` – positive if the current FX collateral exceeds the previous FX collateral; negative if it is the other way around
- `initialMarginMovementCash` – positive if the current Cash IM exceeds the previous Cash IM; negative if it is the other way around
- `initialMarginMovementSecurities` – positive if the current Sec IM exceeds the previous Sec IM; negative if it is the other way around
- `additionalMarginMovements` – positive if the current AM exceeds the previous AM; negative if it is the other way around

14 MEMBER AND CLIENT TRANSFERS

The JSE may transfer members, clients and positions under several scenarios with new event messages being published on the broadcast flows as described in the table below. Please note that even though the data is published during the EOD batch run, it is only effective from the effective date specified in the respective Reference Data Messages.

Scenario	Time of day	Notes
Transfer of a client (requested by JSE)	End of Day	<p>New reference data will be published on the <code>Reference Data Flow</code>, in the same way as if a new client had been added to the TM (See 6.4.1)</p> <p>Member Event</p> <p>Position Account</p> <p>The client will have a new risk node ID under the new member.</p> <p>All position and collateral accounts will be replicated under the new TM. The <code>InternalAccount</code> ID of the position accounts will be different under the new CM.</p> <p>If the client had positions under the previous trading member, <code>transfer_From</code> and <code>Transfer_To</code> trades will be booked to close out positions on the old account and open positions under the new trading member.</p> <p>The cash collateral is moved to the new collateral account. <code>AccountPositionEvent</code> messages will be published to reflect this. The previous member will continue to receive the member event and position accounts for the transferred client with old details</p> <p>The member and client needs to be enabled under the previous member until month end for the interest and risk fees up until the transfer to be settled under the previous member.</p> <p>If the client is disabled under the old TM before the month end, the DASs for the client will continue to be published. Also the interest on cash collateral will be calculated for the disabled client as long as the cash collateral was in the client's collateral account.</p> <p>Tripartite agreements for the client will be disabled manually by a clearing house user before the transfer.</p>
Adding reference data for a member transfer (done by JSE)	Before the step below	<p>The new link will have an effective date, indicating when the transfer is to take place. This can be added several days in advance as new reference data is created at this time.</p> <p>Both old and new clearing member links will be published. The link with the latest effective date, where the effective date is not on a future date, is defined to be the active link.</p>

Scenario	Time of day	Notes
Initiate transfer of members (scheduled event in RTC)	End of Day	<p>RTC will transfer members according to links that have the next business date as “effective date”.</p> <p>New reference data will be published on the <code>Reference Data Flow</code> in the same way as if a new link had been added directly.</p> <p>The transferring member and clients will have a new risk structure with new risk node IDs under the new CM.</p> <p>All position accounts and collateral accounts for the TM and clients will be replicated under the new CM. The InternalAccountIDs of the position accounts will be different under the new CM.</p> <p>Positions will be closed out in the old accounts and new positions are created in the new accounts with <code>Transfer_From</code> and <code>Transfer_To</code> trades.</p> <p><code>AccountPositionEvent</code> messages will be published to reflect this. Transfer_From trades will be published to previous CM and Transfer_To trades to the new CM. The TM will receive both types if subscribed to both old and new subscription groups.</p> <p>Cash collateral is transferred to the respective collateral accounts under the new CM. <code>AccountPositionEvent</code> messages will be published to reflect this.</p> <p>The previous Clearing member will continue to receive the member event and position accounts with old details.</p> <p>The transferred member and member clients need to be enabled under the previous member until month end for the interest and risk fees up until the transfer to be settled under the previous CM.</p> <p>Tripartite will move with the transfer.</p>
Publishing of trades resulting from a member or client transfer.	End-of-Day Next day	Trades will be included when subscribing to the <code>Account Event Flow</code> using the respective subscription groups.
Initiate bulk transfer of clients from defaulting TM (scheduled event in RTC)	End of Day	<p>RTC will transfer clients according to a list previously uploaded via the RTC front-end.</p> <p>New reference data will be published on the <code>Reference Data Flow</code>, in the same way as if a new client had been added directly.</p> <p>Positions will be closed out in the old accounts and new positions are created in the new accounts.</p> <p><code>AccountPositionEvent</code> messages will be published to reflect this.</p>

Scenario	Time of day	Notes
Publishing of trades resulting from a bulk client transfer.	End-of-Day Next day	<p>Trades will be included when subscribing to the Account Event Flow.</p> <p>AccountTradeEvents will be published on the Account Event Flow to reflect the transferred trades.</p> <p>AccountPositionEvents will be published on the Account Event Flow to reflect the current snapshot of the positions after the transfer.</p>
Transfer of a single position, is requested via the front-end.	End of Day	<p>Position will be closed out in the old account and a new position is created in the new account.</p> <p>AccountPositionEvent messages will be published to reflect this.</p>
Publishing of trades resulting from a position transfer	End-of-Day Next day	<p>Trades will be included when subscribing to the Account Event Flow.</p> <p>AccountTradeEvents will be published on the Account Event Flow.</p>

APPENDIX A - MESSAGE FORMATS

Note: This Appendix contains the messages in *EmapTransactionsForMember.html* for ease of reference in the document via hyperlinks. Please refer to the complete set of technical specification documents published on the ITaC website: <https://www.jse.co.za/services/itac>.

Note: At the time of writing this document, there are a few upcoming changes that are currently under development. These changes will be published in a subsequent update to the specifications. Please see full list of known changes below:

Functional Area	Business Entity	EMAPI Entity or Message	Attribute	Description of change
				.

EMAPI protocol specification

Version 1.31.0 (Build: not released)

[Messages](#)

- [by type](#)

- [by ID](#)

[Constants](#)

[Status codes](#)

Messages

General Messages

[CdResponse](#)

[ChangePasswordReq](#)

[GetSequenceNumbersReq](#)

[GetSequenceNumbersRsp](#)

[ResponseMessage](#)

[SimpleRsp](#)

[TaxEndSnapshot](#)

[TaxHeartbeatReq](#)

[TaxHeartbeatRsp](#)

[TaxLogonReq](#)

[TaxLogonRsp](#)

[TaxLogoutReq](#)

[TaxRemoveSubscriptionReq](#)

[TaxReplayEndEvent](#)

[TaxReplayReq](#)

[TaxReplayRsp](#)

[TaxReplayStartEvent](#)

[TaxSessionStatus](#)

[TaxSnapshotSubscribeReq](#)

[TaxSnapshotSubscribeRsp](#)

[TaxStartSnapshot](#)

General Messages (Internal)

[CdRequest](#)

[RequestMessage](#)

Reference Data Messages

[AccessGroup](#)

[CalendarDate](#)

[CashAccount](#)

[CdAddCashAccountReq](#)

[CdUpdateCashAccountReq](#)

[ClassSpreadGroup](#)

[ClearingMemberLink](#)

[CollateralAccount](#)

[CorporateAction](#)

[Country](#)

[Currency](#)

[CurrentSystemState](#)

[Curve](#)

[CurveConstituent](#) [DateCollection](#)

[Deposit](#)

[EligibleCurrency](#)

[EligibleSecurity](#)
[ForwardRateAgreement](#)
[Instrument](#)
[InterestRateSwap](#)
[Market](#)
[MarketList](#)
[Member](#)
[PositionAccount](#)
[RiskNode](#)
[RtcCalendar](#)
[Segment](#)
[SeriesSpreadGroup](#)
[SettlementAccount](#)
[SubscriptionGroup](#)
[Surface](#)
[TradableInstrument](#)
[TripartiteAgreement](#)

External Members

[CdAddRtcMemberClientClearingLinkReq](#)
[CdAddRtcMemberClientReq](#)
[CdAddRtcMemberClientRsp](#)
[CdAddRtcPositionAccountReq](#)
[CdAddRtcPositionAccountRsp](#)
[CdEnableDisableRtcMemberClientReq](#)
[CdEnableDisableRtcPositionAccountReq](#)
[CdEnableDisableRtcPositionAccountRsp](#)
[CdSetClientAMPercentageReq](#)
[CdSetClientRiskLimitReq](#)
[CdSetMinimumZARLimitReq](#)
[CdSetTradingMemberAMPercentageReq](#)
[CdSetTradingMemberRiskLimitReq](#)
[CdUpdateRtcMemberClientReq](#)

Event Messages

[AccountPositionEvent](#)
[AccountTradeEvent](#)
[AggregatedSummaryClearingMemberEvent](#)
[AggregatedSummaryTradingMemberEvent](#)
[AtmVolatilityEvent](#)
[CommissionEvent](#)
[CurveEvent](#)
[DailyAccountSummaryDetailsEvent](#)
[DividendEvent](#)
[OptionDataEvent](#)
[PriceEvent](#)
[RiskNodeEvent](#)
[SurfaceEvent](#)
[WithdrawalNotificationEvent](#)
[YieldEvent](#)

Event Messages (Internal)

[AggregatedSummaryDetails](#)
[CollateralPositionValue](#)
[FxDailyAccountSummaryDetails](#)
[NotionalValue](#)
[RtcTradeExternalData](#)
[Trade](#)

External Members

[CmBalancing1Event](#)
[CmBalancing2Event](#)
[GiveUpEvent](#)
[ReadyConfirmAvailableFXEvent](#)

Trade Management Messages

[AbandonOptionPositionReq](#)
[AbandonOptionPositionRsp](#)
[AcceptCommissionReq](#)
[AddCommissionReq](#)
[AggregateTradesReq](#)
[AggregateTradesRsp](#)
[AllocateTradeReq](#)
[AllocateTradeRsp](#)
[ApproveGiveUpReq](#)
[AssignTradeReq](#)
[CancelCommissionReq](#)
[CancelGiveUpReq](#)
[CorrectAllocationErrorReq](#)
[CorrectAllocationErrorRsp](#)
[CorrectPrincipalReq](#)
[CorrectPrincipalRsp](#)
[ExerciseOptionPositionReq](#)
[ExerciseOptionPositionRsp](#)
[ModifyPositionSubAccountReq](#)
[ModifyPositionSubAccountRsp](#)
[ModifyTradeSubAccountReq](#)
[ModifyTradeSubAccountRsp](#)
[QueryTradesReq](#)
[QueryTradesRsp](#)
[RejectCommissionReq](#)
[RejectGiveUpReq](#)
[TripartiteAllocationReq](#)
[TripartiteAllocationRsp](#)
[UpdateTradeReferenceReq](#)
[UpdateTradeReferenceRsp](#)

Trade Management Messages (Internal)

[TradeDestination](#)
[TradeRes](#)

External Members

[GetRiskArrayReq](#)
[GetRiskArrayRsp](#)

External Members (Internal)

[Contract](#)

Settlement Messages

[ConfirmWithdrawalsReq](#)
[GetPaymentAdvicesReq](#)
[GetPaymentAdvicesRsp](#)

[GetSettlementInstructionsReq](#)
[GetSettlementInstructionsRsp](#)
[PaymentAdvice](#)

Settlement Messages (Internal)

[SettlementInstruction](#)

External Members

[GetRequestsForFXCollateralReq](#)
[GetRequestsForFXCollateralRsp](#)
[QueryDividendPaymentFactorsReq](#)
[QueryDividendPaymentFactorsRsp](#)
[RegisterFXCollateralReq](#)
[RegisterFXCollateralRsp](#)
[RequestForFXCollateral](#)
[SetCmBalancingStatusReq](#)
[SetCmBalancingStatusRsp](#)

External Members (Internal)

[DividendFactor](#)
[FXCollateral](#)
[FXCollateralStatus](#)
[FxInterestRate](#)
[MemberBalance1](#)
[MemberBalance2](#)

General Messages

Message: CdResponse

Message ID: 227

Type: General Messages

Description: A response to be used as super class for all responses from CD.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
5	latestSSN		long	This is the latest and most likely the highest state sequence number, SSN, that has been assigned to the reference data.

Message: ChangePasswordReq

Message ID: 126

Type: General Messages

Description: A request to change the current password. The user does not have to be logged in in order to change the password.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag
12	memberId		String	The id of the user's member (firm). Required because usernames are only unique within a member firm.
13	userId		String	The identification of the user (username).
14	oldPassword		String	The user's old password, used for authentication.
15	newPassword		String	The new password to be set.

This request will normally return a response of type [CdResponse](#) .

Message: GetSequenceNumbersReq

Message ID: 10430

Type: General Messages

Description: Get sequence numbers for broadcast flows.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
6	broadcastFlowId	required	int	Broadcast Flow requested.
7	subscriptionGroupId	required	int	Request sequence number for this subscription group.

This request will normally return a response of type [GetSequenceNumbersRsp](#) .

Message: GetSequenceNumbersRsp

Message ID: 10431

Type: General Messages

Description: Response to a GetSequenceNumbersReq request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
6	sequenceNumber		long	Latest sequence number for the requested broadcast flow and subscription group.
7	broadcastFlowId		int	Broadcast Flow.
8	subscriptionGroupId		int	Subscription group.

Message: ResponseMessage

Message ID: 230

Type: General Messages

Description: General response for request messages that don't have a defined response. It may also be used when a fatal error occurs before or during the normal response handling on the server.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
5	messageReference		String	The message reference from the corresponding RequestMessage.

Message: SimpleRsp

Message ID: 231

Type: General Messages

Description: General response for request messages that don't have a defined response.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
5	reply		String	Generic single string reply

Message: TaxEndSnapshot

Message ID: 73

Type: General Messages

Description: Message ending a snapshot response

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	The overall status
2	message		String	Complementary text for extra context-dependent info
3	subCode		int []	Subcodes
4	flow		Integer	If this message is the result of a snapshot/subscribe operation on a flow then this field contains the flow id.
5	pollSequenceNumber		Long	Not used in this configuration of RTC.
6	subscriptionGroup		Integer	Identifying group of instruments in a current value response if applicable, otherwise zero
10008	snapshotSize		Long	Number of items published in snapshot

Message: TaxHeartbeatReq

Message ID: 75

Type: General Messages

Description: Heartbeat sent to gateway in order to verify a connection

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	userData		String	User supplied data. The data is returned in the response.

This request will normally return a response of type [TaxHeartbeatRsp](#) .

Message: TaxHeartbeatRsp

Message ID: 76

Type: General Messages

Description: Response returned from gateway

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
5	reply		String	Generic single string reply
6	timestamp		String	Current central system time. The format is "yyyy-MM-ddTHH:mm:ss.SSS". Example: 2009-07-16T19:20:30.045
7	userData		String	User-supplied data from the request

Message: TaxLogonReq

Message ID: 63

Type: General Messages

Description: Request to the gateway to log in a member/user

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	member	required	String	User's member firm
3	user	required	String	Mandatory user id. The user must belong to the member.
4	password	required	String	User's password
5	ticket		Long	Ticket received at pre-login

6	possDupSessId		Integer	Possible duplicate session id. If two sessions (that is, users) have the same possDupSessId it means that an unacknowledged request on one of the sessions can be resent on the other with the possDup flag set and the system will be able to resolve if it is a duplicate or not. Not used in this configuration of RTC.
7	majorVersion		int	EMAPI major version. If any of the version fields is non-zero, the gateway will validate against the current EMAPI version.
8	minorVersion		int	EMAPI minor version. If any of the version fields is non-zero, the gateway will validate against the current EMAPI version.
9	microVersion		int	EMAPI micro version. If any of the version fields is non-zero, the gateway will validate against the current EMAPI version.

This request will normally return a response of type [TaxLogonRsp](#) .

Message: TaxLogonRsp

Message ID: 64

Type: General Messages

Description: Sent from the gateway to the client as a response to TaxLogonReq.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
5	reply		String	Generic single string reply
6	logonAccepted		Boolean	Indicates whether the login was successful or not.
7	loginStatus		int	Login specific status code.
8	isTestSystem		Boolean	Indicates whether this system is a test system or not.
9	systemName		String	The name of the system.
10	partitionHbtInterval		Integer	The interval (in seconds) between partition heartbeats sent from the system. Partition heartbeats are sent out as Heartbeat events.
11	clientHbtInterval		Integer	The interval (in seconds) between which clients are expected to send in heartbeats. The client should use the TaxHeartbeatReq message to send in heartbeats.
12	maxLostHeartbeats		Integer	The maximum number of heartbeats to lose before the connection can be considered to be down.

Message: TaxLogoutReq

Message ID: 65

Type: General Messages

Description: Request from client to gateway in end a session. A simple response is sent as response.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
-----------	------------------	-------	-------------------	---------

This request will normally return a response of type [SimpleRsp](#) .

Message: TaxRemoveSubscriptionReq

Message ID: 71

Type: General Messages

Description: Removes an active subscription. A SimpleRsp is sent as response for this request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	handle		int	Subscription handle (subscription identifier) identifying the subscription request to be removed. The handle is received in the response when setting up the subscription.

This request will normally return a response of type [SimpleRsp](#) .

Message: TaxReplayEndEvent

Message ID: 235

Type: General Messages

Description: Framing message indicating the end of requested replay data. The TaxReplayEndEvent indicates the end of a replay sequence.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	subscriptionGroup		int	The subscription group the data is for. The identifier is always set to zero for global flows.
2	nextSequence		Long	When requesting a replay, the trading system may not deliver the full sequence in the first call. The application may need to issue multiple additional requests for retrieving all data. The field "nextSequence" indicates if all data has been retrieved. If so, the field is NULL. Otherwise, the field indicates the sequence number to be used when requesting the next/following batch of replay data.
3	statusCode		int	EMAPI status code telling if the replay was successful or not.
4	statusMessage		String	Status text associated with the EMAPI status code returned.
5	internalCode		int	Not used in this configuration of RTC.
6	flow		int	The flow the data is for.

Message: TaxReplayReq

Message ID: 232

Type: General Messages

Description: Request message sent to the RTC system to recover a sequence of messages published earlier. The replay request will recover earlier published messages on a replayable flow. The response back is a simple response indicating whatever the request was successfully queued to the RTC system. The actual replay data is delivered as unsolicited events, framed by TaxReplayStartEvent and TaxReplayEndEvent messages.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	flow		int	Specifies the logical stream of information of a certain type. Allowed values: see constant group BroadcastFlows
3	subscriptionGroup		Integer	The subscription group on the subscribed flow.
4	sequenceNumber		long	The sequence number from which messages should be recovered for the specified subscription group and flow.
6	member		String	Optional attribute defining the member for which the replay is to be applied for. Used for on-behalf-of replay. Note that the user requesting replay for another member must be authorized to do so. If this attribute is left empty, the logged in user's member is used.
7	endSequenceNumber		long	The sequence number up to which messages should be recovered for the specified subscription group and flow. The value for this attribute could be derived from the TaxSnapshotSubscribeRsp.
8	requestType		int	The type of replay request. Allowed values: see constant group ReplayRequestType
10009	segmentSize		Integer	Not used in this configuration of RTC.

This request will normally return a response of type [TaxReplayRsp](#) .

Message: TaxReplayRsp

Message ID: 233

Type: General Messages

Description: Response message sent back for a previously-submitted TaxReplayReq. The TaxReplayRsp response will not contain the actual data being requested. The response data is delivered to the application asynchronously.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
5	reply		String	Generic single string reply
6	handle		int	Subscription handle identifying the subscription request. The handle is used when removing the subscription.

Message: TaxReplayStartEvent

Message ID: 234

Type: General Messages

Description: Framing message indicating the start sequence of requested replay data. When issuing a replay request, the replay data is delivered as unsolicited messages. The TaxReplayStartEvent indicates the start of a replay sequence.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	subscriptionGroup		int	The subscription group the data is for.
2	flow		int	The broadcast flow for the start event.

Message: TaxSessionStatus

Message ID: 77

Type: General Messages

Description: Unsolicited message indicating session status.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	status		int	Session status Allowed values: see constant group SessionStatus

Message: TaxSnapshotSubscribeReq

Message ID: 69

Type: General Messages

Description: Request to retrieve information and/or activate subscription of future updates of the information specified

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	member		String	Not used in this configuration of RTC.
3	user		String	Not used in this configuration of RTC.
4	requestType		int	Type of subscription request Allowed values: see constant group SubscriptionRequestType
5	flow		int	Data flow being requested Allowed values: see constant group BroadcastFlows
6	key		int	Selection key, identifying the data being subscribed to. In many cases, this is the subscription group.
7	sequenceNumber		long	Not used in this configuration of RTC.
8	lastPollSequenceNumber		Long	Not used in this configuration of RTC.

This request will normally return a response of type [TaxSnapshotSubscribeRsp](#).

Message: TaxSnapshotSubscribeRsp

Message ID: 70

Type: General Messages

Description: Response to a subscription request (TaxSnapshotSubscribeReq).

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
5	reply		String	Generic single string reply
6	handle		int	Subscription handle identifying the subscription request. The handle is used when removing the subscription.
7	lastPublishedSeqNo		Long	Not used in this configuration of RTC.

Message: TaxStartSnapshot

Message ID: 72

Type: General Messages

Description: Message preceding a snapshot response

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	subscriptionGroup		Integer	Group of instruments in current value response if applicable, otherwise zero.
2	flow		int	The broadcast flow for the start event

General Messages (Internal)

Message: CdRequest

Message ID: 226

Type: General Messages

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: This message is not used in EMAPI.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag
2	member		String (64)	The member of the <i>user</i> .
				The responsible user.
3	user		String (64)	Shall only be set in requests if <i>actingUser</i> is acting on behalf of another user. Otherwise TRADExpress will set it to the <i>actingUser</i> .

6	actingUser		String (64)	<p>The user that initiated a request.</p> <p>Shall only be set in requests if (and only if) it is a gateway user that sends the request.</p> <p>Otherwise TRADExpress will set it to the user that established the session on which the request was received on.</p> <p>See also the <i>user</i> field.</p>
---	------------	--	-------------	---

Message: RequestMessage

Message ID: 237

Type: General Messages

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: This message is not used in EMAPI..

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag

Reference Data Messages

Message: AccessGroup

Message ID: 10051

Type: Reference Data Messages

Description: This object defines an access Group.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cached		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	<p>Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.</p> <p>Allowed values: see constant group CACHE_ACTION</p>
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly

				increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	accessGroupId		String	The id for the Access Group.
9	participantUnitId		String	Specifies the parent Participant Unit.
10	clearingMemberId	required	String	The clearing member for the Access group.
11	subscriptionGroup		int	The subscription group for the Access group.

Message: CalendarDate

Message ID: 139

Type: Reference Data Messages

Description: A calendar date of a certain type.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
7	date	required	String (32)	A calendar date in standard format: YYYY-MM-DD.

8	dateType	required	Integer	The type of this date, for example 'closed'. Allowed values: see constant group DATE_TYPE
10	calendarId	required	String (128)	The "identity" of this date. This field may be used for specifying a calendar date that is specific for an exchange/country etc. The id here is (may be) referenced from the orderbook parameters block.
12	displayName		String (255)	The display name for this date, Midsummer for example.
13	isValidTradingBusinessDate		Boolean	Indicate if a date is a valid trading business date or not.

Message: CashAccount

Message ID: 10264

Type: Reference Data Messages

Description: A member or client can have zero or more than one cash account defined, but not more than one per currency.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	settlementBank	required	String (100)	Authorized dealer for foreign currency or settlement bank for ZAR.
9	settlementBankBranch		String (100)	The code of the branch within the authorized dealer or settlement bank.

10	bic	required	String (100)	Swift BIC code.
11	settlementBankAccountId	required	String (100)	ID of account within the authorized dealer or settlement bank.
12	internalCashAccountId		Long	RTC internal Cash Account ID. Set at creation by RTC.
13	currency	required	String (3)	The account is valid for this currency.
14	participantUnitId	required	String (128)	Specifies the parent Participant Unit.
15	isNonResident		boolean	Flag is set from member, is set to true for non resident members.

Message: CdAddCashAccountReq

Message ID: 10267

Type: Reference Data Messages

Description: Request to add a Cash Account

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag
28	cashAccount	required	CashAccount	The cash account to add

This request will normally return a response of type [CdResponse](#) .

Message: CdUpdateCashAccountReq

Message ID: 10268

Type: Reference Data Messages

Description: Request to update a Cash Account. The cash account is identified by either (1) the internalCashAccountId, or (2) the combination of participantUnitId and currency.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag
28	cashAccount	required	CashAccount	The cash account to update

This request will normally return a response of type [CdResponse](#) .

Message: ClassSpreadGroup

Message ID: 10158

Type: Reference Data Messages

Description: Class Spread Group.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
-----------	------------------	-------	-------------------	---------

1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheld		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	csgld	required	String (100)	Uniquely identifies each Class Spread Group.
9	ssgld		Integer	The ID of the SSG that the CSG belongs to, if any.
11	ssmr		Long	The SSMR for the CSG within the SSG. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.

Message: ClearingMemberLink

Message ID: 10123

Type: Reference Data Messages

Description: This object defines a link from Trading Member to Clearing Member.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheld		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.

				Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	clearingMemberLinkId	required	Long	The id for the Clearing Member Link.
9	participantUnitId	required	String	Specifies the parent Participant Unit.
10	tradingMemberId	required	String (100)	The Trading Member ID.
11	clearingMemberId	required	String	The Clearing Member ID.
12	marketId	required	String	The Market ID.
16	effectiveDate	required	String	The effective date from which the link shall be used for capture of incoming trades. The format is yyyy-mm-dd.

Message: CollateralAccount

Message ID: 10093

Type: Reference Data Messages

Description: Represents a collateral account.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.

5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	accountId	required	Long	The id of the collateral account.
10	riskNodeid	required	Long	Specifies the parent risk calculation node.
13	isDisabled		Boolean	If set to true, this object has been disabled.
14	accessGroup		String	The access group for the Collateral Account.
16	accountName	required	String (100)	The name for the Collateral Account.
19	clearingMemberId	required	String (128)	The clearing member for the Collateral Account.
20	participantUnitId	required	String (128)	The owner of the account in the member structure.

Message: CorporateAction

Message ID: 10329

Type: Reference Data Messages

Description: A Corporate Action task definition.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null

				if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	corporateActionId	required	String (100)	Id of the Corporate Action. CA Serial no in MDS.
9	corporateActionType	required	String (5)	Type of the Corporate Action. One of CI,CM,CO,CP,CR,CV,IS,NC,RT,SC,SD,TE,UB. Note that the CA type is only used for information purpose in RTC and is not validated.
10	positionFactor		Long	Factor to use to adjust positions. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
11	fromInstrument	required	String (100)	Master ID of the TI on which the CA is taking place. The positions on this TI will be closed.
12	toInstrument		String (100)	Master ID of the new TI for the new position. New positions will be created on this TI. If missing, settlement positions will be created instead.
13	effectiveDate	required	String (10)	Corporate actions are applied at EOD on the last business day before the effective date. Format is YYYY-MM-DD.
14	ldtDate		String (10)	Last Day Traded. Format is YYYY-MM-DD.
15	price		Long	Price to use when positions should be closed out. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
16	status		int	to know the status of the corporate action. Allowed values: see constant group CorporateActionStatus
17	statusText		String	to know the status of the corporate action.
18	statusTimestamp		String (100)	Update timestamp, set by RTC.

Message: Country

Message ID: 316

Type: Reference Data Messages

Description: Holds basic information on a country, such as currency, time zone and holidays.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheld		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION

4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
7	countryId	required	String (16)	The id (country code) of the Country. Use ISO 3166 ISO 2-alpha codes.
8	name	required	String (128)	The name of the Country
9	currency	required	String (3)	The currency code according to ISO 4217.
10	timezoneOffset	required	Integer (32)	Timezone offset from UTC/GMT in minutes (positive value for countries east of Greenwich)

Message: Currency

Message ID: 95

Type: Reference Data Messages

Description: This object represent a Currency.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.

6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
7	currencyId	required	String (3)	The standard currency id according to ISO 4217. SEK, USD, GPB etc
8	longName		String (64)	The full name of the currency. Example: South African Rand
10010	currencyCode	required	long	ISO Currency Numeric Code, for example 710 for ZAR.
10012	notionalValueDecimals	required	Integer	Number of decimals used for notional in the currency. Only allowed to be set to 2.
10013	calendarID	required	String	ID of the calendar to be used for finding settlement days

Message: CurrentSystemState

Message ID: 10440

Type: Reference Data Messages

Description: This object holds a RTC System State.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	currentSystemStateId	required	Integer	The id, will only exist one record since only one state can be current.
9	currentRtcState	required	String	Current state in RTC. Allowed values: see constant group RtcState

10	businessDate		String	The business the state is set, in format YYYY-MM-DD.
11	schedulerState	required	Integer	Current scheduler state in RTC. Allowed values: see constant group SchedulerState
12	rerunReason		String	Reason for rerunning End of Day, only set when scheduler state is RERUN.

Message: Curve

Message ID: 10177

Type: Reference Data Messages

Description: A curve, y axis dependent on x axis, $y = f(x)$.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cachedId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	curvelId		Long	The internal curve Id for RTC.
9	externalCurvelId	required	String (100)	The external curve Id received from Master reference data system.
10	curveName	required	String (128)	User friendly name of the curve.
11	strippedInRtc	required	Boolean	If true, RTC is responsible to prepare the curve.
12	priceFormat	required	int	Number of decimals used.
13	bootStrappingMethod	required	String (100)	BootStrapping method.

				Allowed values: see constant group BOOTSTRAPPING_METHOD
14	dayCountConvention	required	String (100)	Day count convention. Allowed values: see constant group DAY_COUNT_CONVENTION
15	interpolationMethod	required	String (100)	Interpolation method. Allowed values: see constant group INTERPOLATION_METHOD
16	extrapolationMethod	required	String (100)	Extrapolation method. Allowed values: see constant group EXTRAPOLATION_METHOD
20	axisUnitX	required	String (100)	xAxis unit. Allowed values: see constant group AXIS_UNIT
21	axisUnitY	required	String (100)	yAxis unit. Allowed values: see constant group AXIS_UNIT
22	pePartitionId		Integer	Partition ID for the instrument in the RTC Price Engine. Not required, 1 will be used if the field is blank.
23	interestRateConvention	required	String (100)	Interest rate convention for the interest rate produced. Allowed values: see constant group INTEREST_RATE_CONVENTION

Message: CurveConstituent

Message ID: 10185

Type: Reference Data Messages

Description: Instruments to build the curve.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheld		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The

				sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	curveConstituentId		Long	The internal curve Constituent Id for RTC.
9	curveld		Long	Specifies the Curve that this CurveConstituent belongs to.
11	externalInstrumentId	required	String	Instrument Id (JSE Master ID) for the deposit, FRA or Swap.

Message: Deposit

Message ID: 10214

Type: Reference Data Messages

Description: Deposit instrument. Used as curve constituents.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheld		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	depositId		Long	The internal Deposit ID for RTC.

9	externalDepositId	required	String (100)	The external Deposit Id received from Master reference data system.
10	depositName	required	String (128)	User friendly name of the Deposit.
12	dayCountConvention	required	String	Day count convention. Allowed values: see constant group DAY_COUNT_CONVENTION
16	compoundingConvention	required	String	Compounding convention. Allowed values: see constant group CompoundingConvention
17	rtcCalendarId	required	String	Calendar used for holidays.
18	businessDayConvention	required	String	Day count convention. Allowed values: see constant group BusinessDayConvention
19	tenorPeriodType	required	String (100)	Period type for tenor. For instance M in 3M. Allowed values: see constant group PeriodType
20	tenorPeriod	required	int	Tenor period. For instance 3 in 3M.
21	rollsOn	required	String	Rolls on convention. Allowed values: see constant group RollsOnConvention
22	pePartitionId		Integer	Partition ID for the instrument in the RTC Price Engine. Not required, 1 will be used if the field is blank.

Message: EligibleCurrency

Message ID: 10355

Type: Reference Data Messages

Description: Currency eligible for FX collateral.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly

				increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	instrumentMasterId	required	String (128)	External instrument master ID for the currency pair.
10	priceCCY	required	String (3)	The second currency in the currency pair. Normally ZAR.
11	baseCCY	required	String (3)	The first currency in the currency pair, i.e. normal the collateral currency (when using EUR, GBP and USD).
12	haircut	required	Long	Haircut percent on the exchange rate. Support for four decimals. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
13	maxAmount	required	Long	Maximum allowed amount for the currency allowed as collateral. The amount is defined in ZAR before the haircut is applied..
14	prio		Integer	When a client has several Nostro accounts (cash accounts in foreign CCY) and there is a doubt which currency to use, they are used in prio order. FX collateral with the highest priority is used first. The lowest number has got the highest priority. Must be unique.

Message: EligibleSecurity

Message ID: 10256

Type: Reference Data Messages

Description: Security instrument eligible for collateral.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheld		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence

				number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	instrumentMasterId	required	String (255)	External instrument master ID.
10	isin	required	String (255)	ISIN of the instrument.
11	alphaCode	required	String (255)	Alpha Code of the instrument.
12	haircut	required	Long	Percentage that is subtracted from the portfolio value of an asset that is being used as collateral. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
13	maxAmount	required	Long	Maximum allowed value allowed to pledge as collateral for this ISIN per risk node and before haircut is applied.

Message: ForwardRateAgreement

Message ID: 10216

Type: Reference Data Messages

Description: A Forward Rate Agreement instrument.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing).

				This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	forwardRateAgreementId		Long	The internal FRA Id for RTC.
9	externalForwardRateAgreementId	required	String (100)	The external FRA Id received from Master reference data system.
10	forwardRateAgreementName	required	String (128)	User friendly name of the FRA.
12	dayCountConvention	required	String (100)	Day count convention. Allowed values: see constant group DAY_COUNT_CONVENTION
18	compoundingConvention	required	String (100)	Compounding convention. Allowed values: see constant group CompoundingConvention
20	rtcCalendarId	required	String (100)	Calendar used for holidays.
21	businessDayConvention	required	String (100)	Day count convention. Allowed values: see constant group BusinessDayConvention
23	rollsOn	required	String (100)	Rolls on convention. Allowed values: see constant group RollsOnConvention
24	tenorPeriodType	required	String (100)	Tenor period type. Example M in 3M. Allowed values: see constant group PeriodType
25	tenorPeriod	required	int	Tenor length. Example 3 in 3M.
26	resetLagPeriodType	required	String (100)	Reset lag. Allowed values: see constant group PeriodType
27	resetLagPeriod	required	int	Reset lag.
28	pePartitionId		Integer	Partition ID for the instrument in the RTC Price Engine. Not required, 1 will be used if the field is blank.

Message: Instrument

Message ID: 295

Type: Reference Data Messages

Description: The Instrument holds basic background information, such as instrument id, type and (optionally) primary market for an instrument. Since an instrument can be traded in different currencies and visibility (normal,

dark etc), the instrument has a set of child objects called TradableInstrument in which the actual trading takes place. An Instrument may reference another Instrument using the "parentInternalId" attribute. The parentInternalId is typically used by warrants or options to reference the underlying instrument.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
7	internalId		String (128)	A unique instrument identifier. May be an ISIN, CUSIP or symbol name. The InternalID is created using the InstrumentID and InstrumentIDType fields. Example: SE0000108656_ISIN
8	parentInternalId		String (128)	This is a reference to the parent instrument, if any. A parent instrument is typically an underlying instrument when trading derivatives. Shall be set to null if this instrument is a "root" instrument.
9	isEnabled		Boolean	The state of this item.
11	disabledCount		Integer	Not used in this configuration of RTC.
12	instrumentId	required	String (64)	The global identity of the instrument. The type of identifier (ISIN, CUSIP etc) is defined by the instrumentIDType attribute.
13	instrumentIDType	required	String (32)	The type of the InstrumentId (ISIN, CUSIP etc) Allowed values: see constant group InstrumentIDType
14	prevInstrumentId		String (64)	Not used in this configuration of RTC.
15	prevInstrumentIDType		String (32)	Not used in this configuration of RTC. Allowed values: see constant group InstrumentIDType
16	type	required	String (32)	The type of instrument (equity, warrant, future etc).

				Allowed values: see constant group InstrumentType
17	shortName	required	String (64)	The short display name for the instrument, ERICB for example.
18	name	required	String (255)	The complete instrument name, Ericsson B for example.
19	issuer		String (255)	The issuer of the instrument
20	sector		String (128)	The financial sector to which the instrument belong
21	primaryMarketId		String (32)	A unique id that defines market/market that is to be considered primary for the instrument, XLON, XSSE for example.)
22	adt		Long	Not used in this configuration of RTC. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
23	adtCurrency		String (3)	Not used in this configuration of RTC.
27	validFromDate	required	String (10)	The first date the instrument is valid. The format is yyyy-MM-dd.
28	validToDate		String (10)	The last date the instrument is valid. The format is yyyy-MM-dd.
30	listOfAliases		String (256)	A list of other markets/exchanges' ID of this instrument in the format: market1:id1,market2:id2,...
10047	pmPartitionId		Integer	The instrument partition for the Position Manager server.
10048	rtcInternalId		Long	The RTC internal ID for the underlying product.

Message: InterestRateSwap

Message ID: 10215

Type: Reference Data Messages

Description: A Interest rate swap instrument.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheld		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the

				sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	interestRateSwapId		Long	The internal IRS Id for RTC.
9	externalInterestRateSwapId	required	String (100)	The external IRS Id received from Master reference data system.
10	interestRateSwapName	required	String (128)	User friendly name of the IRS.
12	rollsOn	required	String (100)	Rolls on convention. Allowed values: see constant group RollsOnConvention
13	dayCountConvention	required	String (100)	Day count convention. Allowed values: see constant group DAY COUNT CONVENTION
19	compoundingConvention	required	String (100)	Compounding convention. Allowed values: see constant group CompoundingConvention
21	rtcCalendarId	required	String (100)	Calendar used for holidays.
22	businessDayConvention	required	String (100)	Day count convention. Allowed values: see constant group BusinessDayConvention
24	tenorPeriodType	required	String (100)	Tenor period type. Example M in 3M. Allowed values: see constant group PeriodType
25	tenorPeriod	required	int	Tenor length. Example 3 in 3M.
26	resetLagPeriodType	required	String (100)	Reset lag period. Example M in 3M. Allowed values: see constant group PeriodType
27	resetLagPeriod	required	int	Reset lag length. Example 3 in 3M.
28	pePartitionId		Integer	Partition ID for the instrument in the RTC Price Engine. Not required, 1 will be used if the field is blank.

Message: Market

Message ID: 299

Type: Reference Data Messages

Description: Defines a market.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
7	marketId	required	String (64)	A system-unique identifier of the Market. May only contain characters a-z, A-Z, 0-9, "_", "-", "+" and ".".
8	isEnabled		Boolean	The state of this item.
9	disabledCount		Integer	A count of how many times this element has been enabled/disabled. An element will not be enabled until disabledCount is zero.
10	name	required	String (255)	The name of the Market.
11	countryCode	required	String (16)	The country code for this market.
12	wwwPage		String (256)	A reference to a market specific internet site.
19	validFromDate		String (10)	The first date the market is valid. The format is yyyy-MM-dd.
10021	lookbackPeriod	required	Long	JSPAN attribute. The lookback period used to get historical prices. For example, 90 to use the last 90 days prices. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
10023	volatilityLookbackPeriod	required	Long	JSPAN attribute. The volatility lookback period used when calculating volatilities.

				This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
10026	maxScaleUp	required	Long	JSPAN attribute. The maximum amount volatility may be scaled up. This should be a number between 1 and 2. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
10027	maxScaleDown	required	Long	JSPAN attribute. The maximum amount volatility may be scaled down. This should be a number between 0 and 1. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
10028	confidencePercentile	required	Long	JSPAN attribute. A percentile stored as a decimal, e.g. 0.975 for 97.5%. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
10030	imrStatisticsPeriod	required	Long	Number of days to compare price and volatility move for. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
10031	manualExerciseEndTime	required	String (12)	The end time for option exercise on the expiration date. Must be in the format HH:MM:SS.
10032	optionAllocationModelType	required	Integer	Allocation model for options, eg Pro-rata or Random. Allowed values: see constant group OptionAllocationModelType
10033	dailyMaximumParticipationFactor	required	Long	Daily Maximum Participation Factor for Liquidation Period add-on. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
10034	nonTradingDaysBeforeDefault	required	Long	Used in the Liquidation Period Add-on calculation. Number of days it will take the JSE to confirm default before it starts to close out positions. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
10035	volRounding	required	Integer	Number of decimals for volatility calculated by RTC.

Message: MarketList

Message ID: 300

Type: Reference Data Messages

Description: The MarketList is a child object of a Market. The purpose of the MarketList is mainly to organize the different instruments on a market into separate lists. The actual interpretation of the MarketList is customer specific. Operations such as halt and enable/disable performed on a MarketList will affect all Segments and TradableInstruments within the MarketList.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cached		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
7	internalMarketListId		String (128)	The internal (unique) id of the MarketList. Set by RTC.
8	isEnabled		Boolean	The state of this item.
9	disabledCount		Integer	A count of how many times this element has been enabled/disabled. An element will not be enabled until disabledCount is zero.
10	marketListId	required	String (128)	The display id of the MarketList. Must be unique within the Market. May only contain characters: a-z, A-Z, 0-9, "_", "-", "+", and ".".
11	name	required	String (255)	This is the name of this Market
12	parentInternalId	required	String (128)	The parent market id (EMAPI).
20	validFromDate		String (10)	The first date the market list is valid. The format is yyyy-MM-dd.

Message: Member

Message ID: 101

Type: Reference Data Messages

Description: This object represents a member firm and holds all basic member data such as id, full name, mail addresses and contact persons etc.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheld		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
7	memberId	required	String (64)	The public ID of the participant. This id has to be unique.
8	fullName	required	String (128)	Complete name of the member firm
10	address		String (255)	Company postal address.
11	phone		String (64)	Company phone number.
12	fax		String (64)	Company fax address.
13	complianceContact		String (128)	Name of contact person in compliance matters
14	complianceContactPhone		String (64)	Phone number to the compliance contact person
15	complianceContactMail		String (128)	Mail address to the compliance contact person
16	matchingContact		String (128)	Matching/BackOffice contact person
17	matchingContactPhone		String (64)	Phone number to the Matching/BackOffice contact person
18	matchingContactMail		String (128)	Mail address to Matching/BackOffice contact person
19	isDisabled		Boolean	Set to true if this member has been disabled.

26	associatedMemberId		String (255)	The parent member, for example the parent of a trading member branch or a client.
30	memberType		Integer	The type of member. N.B. in the documentation member type is also known as Participant type. Allowed values: see constant group MemberType
33	validFromDate		String (10)	The date from which the member is valid. The format is yyyy-MM-dd
34	listOfAliases		String (256)	A list of other markets/exchanges' ID of this member in the format: market1:memberid1,market2:memberid2,... "maket1" is assumed to be a market defined in the "local" system and is automatically converted to uppercase, since this is the conversion for market ids in the system.
39	allowedOnBehalfOfMemberIdList		String (255)	A comma separated list of Member Ids for which this member may act on behalf of.
10043	participantUnitType		Integer	The type of this participant unit. Allowed values: see constant group ParticipantUnitType
10046	isStaff		Boolean	Is Staff, true or false
10047	isBeneficial		Boolean	Is Beneficial, true or false
10048	allowClientSubAccounts		Boolean	Sub accounts are allowed, true or false
10049	vatRegNumber		String	VAT registration number
10050	bdaCode		Integer	Broker Deal Account number
10051	email		String	Email
10052	country		String	Country Code, e.g. ZA
10053	isNonResident		Boolean	Is Non Resident is true if country for the client is not equal to ZA
10054	nominatedMember		String	MemberId of the nominated member, fulfilling physical settlement when the actual member is not in the Equities market. Otherwise, set to the member itself.
10055	strateCode		String (100)	Code of client or member at CSD.
10056	externalPayment		Boolean	If RTC should expect net payment from the JSE integration layer for this member. Required for clearing members.
10057	ownTM		String	This is set for clearing members only, to indicate its own trading member.
10058	allowFxCollateral		Boolean	This field is mandatory for CMs. If true, RTC will expect system-to-system communication on size of FX collateral payments.
10059	allowedMarkets		String	A comma-delimited list of market codes that the client is allowed to have trades and positions in.
10060	waitForCmBalancing		Boolean	This field is mandatory for CMs. If true, RTC will expect the CM to send in a response to a balancing event.
10061	clientType		String	For clients only - type of client. Information to surveillance. Required for all clients. Allowed values: see constant group ClientType
10062	idNumber		Long	For clients only - ID number. Required for local individual clients: Client Type = Individual AND isNonResident = FALSE

10063	passportNumber		String	For clients only - Passport number. Required for foreign individual clients: Client Type = Individual AND isNonResident = TRUE
10064	companyRegistrationNumber		String	For clients only - Company registration number. Required for all company clients: Client Type = Company
10065	isProfessional		Boolean	For clients only - Information to surveillance. Required for all clients.
10066	isShariah		Boolean	For clients only - Information to surveillance. Required for all clients.
10067	isDiscretionary		Boolean	Is Discretionary, true or false
10068	preferredCcy		String (3)	Currency used for intraday rebalancing. Needs to be an eligible FX collateral currency (or ZAR, this will be the default though).
10069	branchMemberNumber		String	For TM branches only. Unique within a TM. Valid number is between 01 and 99.
10070	cmMessageRef		String (5)	For CM only. Mandatory for CM. Number used when creating settlement instructions. This number is concatenated into the message reference no.

Message: PositionAccount

Message ID: 10045

Type: Reference Data Messages

Description: Position account is used to keep actual clearing positions and settlement positions.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has

				been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	positionAccountId	required	Long	The internal id of the position Account.
9	externalPositionAccountId	required	String (10)	The external id of the position account. The maximum length is 10 characters. House sub accounts are validated to be a maximum of 9 long.
11	positionAccountType	required	Integer	The type of this position account. Allowed values: see constant group PositionAccountType
12	accessGroup	required	String	The access group for the Position Account.
13	clearingMemberId	required	String	The clearing member for the Position Account.
14	riskNode	required	Long	The risk node for the Position Account.
15	participantUnitId		String	Specifies the parent Participant Unit.
16	isEnabled	required	Boolean	The state of this item.
17	positionAccountSubType	required	Integer	The sub type of this position account. Allowed values: see constant group PositionAccountSubType

Message: RiskNode

Message ID: 10046

Type: Reference Data Messages

Description: Risk node is the entity that defines the level for risk calculations. It has one or several accounts connected and form a tree structure with aggregated risk numbers on parent nodes.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be

				null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	riskNodeId	required	Long	The id for the Risk Node.
9	parentRiskNodeId		Long	Risk node Id of the parent.
10	riskNodeName	required	String (100)	The name for the Risk Node.
11	isDefault		Boolean	This is used to identify the default risk node for a member.
12	accessGroup	required	String	The access group for the Risk Node.
13	clearingMemberId	required	String	The clearing member for the Position Account.
14	participantUnitId	required	String	Specifies Participant Unit that this risk node belongs to.
15	positionAccountType		Integer	The account type for accounts connected to this risk node. Allowed values: see constant group PositionAccountType
18	isClearingHouse		Boolean	This is used to identify if the risk node is for clearing house.
19	minimumZARLimit		Integer	The minimum percentage of the collateral that must be in ZAR.
20	riskLimit		Long	The risk limit after which alert is triggered.
21	parentHouseRiskNodeId		Long	Risk node Id of the house risk node of the parent.
22	amPercentage		Integer	The AM percentage used to calculate additional margin.

Message: RtcCalendar

Message ID: 10145

Type: Reference Data Messages

Description: Represents a calendar in the RTC system.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cached		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.

6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	rtcCalendarId		String	The id of the RtcCalendar.
9	calendarDateCalendarId		String	Specifies which RtcCalendar to use for holidays and half-days .
10	displayName	required	String (128)	The descriptive name for this calendar.
11	defaultScheduleStartTime	required	String (32)	The default starting time for this calendar. Must be in the format HH:MM:SS.
12	timeZone	required	String (128)	The time zone for this calendar.
13	saturdaysSundaysClosed		Boolean	Set to true if Saturdays and Sundays are closed.

Message: Segment

Message ID: 302

Type: Reference Data Messages

Description: A Segment is a grouping of TradableInstruments that share the same trading rules. Operations on a segment, such as halt and enable/disable affects all tradable instruments related to the segment. A Segment is a child object to MarketList.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.

7	internalSegmentId		String (128)	The cache (unique) id of the Segment. The internalSegmentId field is assigned by the system.
8	isEnabled		Boolean	The state of this item.
9	disabledCount		Integer	A count of how many times this element has been enabled/disabled. An element will not be enabled until disabledCount is zero.
10	segmentId	required	String (128)	The display id of the Segment. Must be unique within the parent MarketList. May only contain characters: a-z, A-Z, 0-9, "_", "-", "+", and ".".
11	parentInternalId	required	String (128)	The parent Market List ID.
12	name	required	String (255)	The name of this Segment
22	validFromDate		String (10)	The first date the segment is valid. The format is yyyy-MM-dd.

Message: SeriesSpreadGroup

Message ID: 10159

Type: Reference Data Messages

Description: Series Spread Group.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	ssgId	required	Integer	Uniquely identifies each Series Spread Group.
9	groupName	required	String	Class Spread Group name.

Message: SettlementAccount

Message ID: 10082

Type: Reference Data Messages

Description: This object defines a settlement account.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	settlementAccountId		Long	The id for the Settlement Account.
9	externalSettlementAccountId		String (100)	The external id for the account.
10	settlementAccountName		String (100)	The name for the Settlement Account.
11	isNettingAccount	required	Boolean	If true, this account is a settlement account where positions are netted together before being sent to an external payment/delivery system.
12	isClearingHouseAccount	required	Boolean	If true, this is a Settlement Account for the Clearing House.
13	isEnabled	required	Boolean	Defines whether the account is enabled.
14	participantUnitId		String	Links to participantUnit if the Settlement Account is not a Clearing House Settlement Account
15	isDefaultClearingHouseAccount	required	Boolean	If true, this is the default Settlement Account for the Clearing House.
17	accessGroup	required	String	The access group for the Settlement Account.

Message: SubscriptionGroup

Message ID: 96

Type: Reference Data Messages

Description: The subscription group is used to filter objects on broadcast flows. When a subscription is set up for a subscription group the system controls the user access rights for that access group.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cachedId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
7	subscriptionGroupId	required	Integer	The business id of the subscription group.
8	description	required	String (255)	A text description of the set of order books contained in the group.
9	partitionId	required	Integer	The partition this subscription group belong to.
10012	accountAccessGroup		String	The account access group.

Message: Surface

Message ID: 10178

Type: Reference Data Messages

Description: A surface. Three axis, z dependent on x and y, $z = f(x, y)$.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
-----------	------------------	-------	-------------------	---------

1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cachedId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	surfaceId		Long	The internal surface Id for RTC.
9	externalSurfaceId	required	String (100)	The external surface Id received from Master reference data system.
10	surfaceName	required	String (128)	User friendly name of the surface.
11	dayCountConvention	required	String	Day count convention. Allowed values: see constant group DAY_COUNT_CONVENTION
12	interpolationMethod	required	String	Interpolation method. Allowed values: see constant group INTERPOLATION_METHOD
13	extrapolationMethod	required	String	Extrapolation method. Allowed values: see constant group EXTRAPOLATION_METHOD
19	axisUnitX	required	String	xAxis unit. Allowed values: see constant group AXIS_UNIT
20	axisUnitY	required	String	yAxis unit. Allowed values: see constant group AXIS_UNIT
21	axisUnitZ	required	String	zAxis unit. Allowed values: see constant group AXIS_UNIT
22	pePartitionId		Integer	Partition ID for the instrument in the RTC Price Engine. Not required, 1 will be used if the field is blank.

Message: TradableInstrument

Message ID: 296

Type: Reference Data Messages

Description: The TradableInstrument is a child object of an Instrument. The TradableInstrument holds trading information (order book id, currency, market, visibility etc) which is necessary for entering orders in a specific instrument. There is one TradableInstrument instance per market/currency/visibility combination. A TradableInstrument instance references a Segment, all trading rules for the referenced Segment applies to the instance.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cached		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
7	internalId		String (192)	The internal id of the TradableInstrument. This id is unique within the system and is created from the fields parentInternalId, market, currency and visibility.
8	parentInternalId		String (192)	The internal id of the parent Instrument. (The instrumentId and the instrumentIdType of the parent Instrument).
9	internalSegmentId		String (198)	A "cache" segment reference to which this tradable instrument belongs. Assigned by the server when the TradableInstrument is created. Validated against Market/MarketList/Segment attributes when updating a TradableInstrument.

10	isEnabled		Boolean	The state of this item.
12	disabledCount		Integer	A count of how many times this element has been enabled/disabled. An element will not be enabled until disabledCount is zero.
13	tradableInstrumentId	required	String (64)	The tradable instrument id. Typically an ISIN, CUSIP or symbol name. This is the JSE Master ID.
14	tradableInstrumentIdType	required	String (32)	The type of the TradableInstrumentId (ISIN, CUSIP etc) Allowed values: see constant group InstrumentIdType
15	currencyId	required	String (3)	The currency code according to ISO 4217
16	shortName	required	String (255)	Display name of this tradable instrument.
17	marketId	required	String (128)	The market where this tradable instrument is traded. This field refers to a Market object's marketId field.
18	marketListId	required	String (128)	The market list where this instrument is traded, reflecting a MarketList object's marketListId field. The market list must belong to the market specified by the marketId field.
19	segmentId	required	String (128)	The segment where this instrument is traded, reflecting a Segment object's segmentId field. The segment specified must belong to the market stated in the marketId field and belong to the market list stated by the marketListId field.
27	subscriptionGroupId		Integer	The id of the subscription group to which this tradable instrument belong
31	validFromDate		String (10)	The first date the tradable instrument is valid. The format is yyyy-MM-dd. If this optional data is not specified, the value from the tradable instrument's instrument will be used. The parent tradable instrument must have a validFrom date that is equal to or less than the validFrom date of the child tradable instrument.
32	validToDate		String (10)	The last date the tradable instrument is valid. The format is yyyy-MM-dd. If this optional data is not specified, the value from the tradable instrument's instrument will be used. The parent tradable instrument must have a validTo date that is equal to or greater than the validTo date of the child tradable instrument.
33	numberOfShares		Long (64)	The number of shares in the instrument. If the number of shares is not defined, then there will not be any qty checks for that instrument.
39	listOfAliases		String (256)	A list of other markets/exchanges' ID of this tradable instrument in the format: market1:id1,market2:id2,...
57	lastTradingDate		String (10)	The last date the instrument is open for trading. The format is yyyy-MM-dd.
58	corporateActionIndicator		String (24)	This attribute is used to indicate that this instrument has been subject to a Corporate Action.
59	validForTrading		Boolean	If this field is true, this TradableInstrument is valid for trading. ValidForTrading is typically used when a TradableInstrument is not traded on "this" exchange, but is needed as underlying for derivatives (carrying the price). The default value is true.
10061	volatilityScanningRange		Long	Volatility Scanning Range (VSR) This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.

10062	contractSize	required	Long	The size (quantity) of one traded contract.
10063	settlementCycle		Integer	Settlement cycle in days. Mandatory for instrument type Spot.
10064	classSpreadGroup		String (255)	The Class Spread Group (CSG) to which the contract belongs.
10065	classSpreadMarginRequirement		Long	Class Spread Margin Requirement (CSMR) This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
10068	expiryDate		String (10)	The expiration date for this contract. The format is yyyy-mm-dd. Used for options and futures.
10069	contractCode		String (100)	The Contract Code. Example: AGLS
10072	underlyingTradableInstrument		String (100)	The Underlying Tradable Instrument. Example: the future (AGLS-20141023) for an option or the underlying (ABL) for a future
10073	contractCategory		String (100)	The Contract Category. Example: SingleStock
10074	settlementType		Integer	Settlement type, Cash or Physical. If cash settled, only cash is settled at expiry. If physical, the underlying instrument is settled at expiry. Used for futures and options. Allowed values: see constant group SettlementType
10075	exerciseStyle		Integer	Exercise style, Americal or European. Used for options. Allowed values: see constant group ExerciseStyle
10076	isCall		Boolean	Call or Put option.
10077	strike		Long	Strike for option. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
10078	volatilitySurfaceId		String (100)	The Volatility Surface Id. Used for options.
10079	yieldCurveId		String (100)	The id of yield curve instrument used for valuation. Mandatory for future with valuation model set to RTC Mark-to-Model. Mandatory if InstrumentSubType is Dividend Neutral, International Dividend Neutral, Quanto International Dividend Neutral, Quanto Index Dividend Neutral or Contract for Difference, regardless of valuation model.
10080	contractDescription		String (255)	The Contract Description.
10081	instrumentType	required	String (32)	The type of instrument (equity, option, future etc). Allowed values: see constant group InstrumentType
10084	stressPeriodStartDate		String (10)	In addition to the look-back period other dates of price data can be added. This is the start date of the period used. The format is yyyy-mm-dd. Used for spot instruments.
10085	stressPeriodEndDate		String (10)	In addition to the look-back period other dates of price data can be added. This is the end date of the period used. The format is yyyy-mm-dd. Used for spot instruments.
10086	liquidationPeriod		Integer	The liquidation period used in the IMR calculation. Also used for the calculation of the Liquidity Add-on.

				For example, 1 for daily and 7 for weekly. Mandatory for spot instruments. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
10087	advt		Long	The Average Daily Value Traded, used for the calculation of the Liquidity Add-on. Mandatory for spot instruments. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
10088	imrOfficial		Long	Official IMR (Initial Margin Requirement) This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
10089	oneDayVar		Long	1-day VaR This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
10090	alphaCode	required	String (30)	The Alpha Code. Example: AGL
10091	optionStyle		Integer	Future style or Upfront premium. Used for options. Allowed values: see constant group OptionStyle
10092	atmCents		Integer	Cents from ATM. Used for options.
10093	isin		String (12)	ISIN for the instrument.
10094	valuationModelType	required	String (100)	Valuation Model Type. Allowed values: see constant group ValuationModelType
10095	valuationSubType		String (100)	Valuation Sub Type. Allowed values: see constant group ValuationSubType
10096	isTradable	required	Boolean	Indicates whether the instrument can be traded at JSE.
10097	baseCurrency		String (10)	Base currency, mandatory for FX spot instruments.
10098	priceCurrency		String (10)	Price currency, mandatory for FX spot instruments.
10099	maturityDate		String (10)	Maturity Date for Bonds. Format yyyy-mm-dd.
10100	dayCountConvention		String	The convention determines how interest accrues over time, e.g. number of days between two coupon payments. For Bonds. Allowed values: see constant group DAY_COUNT_CONVENTION
10101	businessDayConvention		String	Rules for date rolling when a payment day or date used to calculate accrued interest falls on a holiday. For Bonds. Allowed values: see constant group BusinessDayConvention

10103	rtcCalendarId		String	Calendar used for holidays. For Bonds.
10104	couponRate		Long	Annual Coupon rate for Bonds. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
10105	listingDate		String (10)	The date when the Bond was listed.
10107	couponFrequencyPeriod		Integer	Coupon frequency period length. How often coupons are paid annually. For Bonds.
10108	couponIndicator		String	Coupon rate indicator. For Bonds. Allowed values: see constant group CouponIndicator
10110	couponDates		String (1000)	For Bonds. Coupon dates, comma separated dates on format MMDD.
10111	bookClosingDates		String (1000)	For Bonds. Book Closing dates, comma separated dates on format MMDD.
10112	priceFormat	required	Integer	Number of decimals in prices.
10113	smrOfficial		Long	Official SMR (Settlement Margin Requirement) This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
10114	instrumentSubType	required	String (32)	The sub type of instrument (spot, index, CFD, etc.). Allowed values: see constant group InstrumentSubType
10115	assetClass	required	String (255)	The asset class this tradable instrument belongs to. Allowed values: see constant group AssetClass
10116	pePartitionId		Integer	Partition ID for the instrument in the RTC Price Engine. Not required, 1 will be used if the field is blank.
10117	assetSubClass	required	String	Asset sub class used for stress testing. Allowed values: see constant group AssetSubClass
10118	baseRateInstrument		String (100)	Master ID of the base rate instrument
10120	priceProxyMasterID		String (100)	The Master ID of another underlying instrument to use as proxy when calculating the IMR %. Will mostly be used for foreign instruments where the price history is not available.
10121	interestCommencementDate		String (10)	For Bonds. This date is used as last coupon date equivalent if no coupon has yet been paid.
10122	inwardListed	required	Boolean	Designated as Inward listed by the South African Reserve Bank.
10123	anyday	required	Boolean	If Anyday is true, the expiry date of the instrument may be on any business day.
10124	nominal		Long	The Nominal Amount for Bonds. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
10125	redemptionFraction		Long	Fraction of Nominal to be returned at maturity. For Bonds. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.

10126	compoundingConventionOfYield		String	Compounding Convention of yield. For Bonds. Allowed values: see constant group CompoundingConvention
10127	compoundingConventionOfRate		String	Compounding Convention of coupon rate. For Bonds, information only. Allowed values: see constant group CompoundingConvention
10128	exchange	required	String (100)	ID of the exchange where the instrument is traded
10129	country	required	String (100)	Country code of instrument.
10130	contractSizeType		Integer	Indicator if the contract is base (standard), mini/maxi etc. Allowed values: see constant group ContractSizeType
10132	negativePriceAllowed		Boolean	If NPA is True, prices in the tradable instrument may be negative, otherwise not. Only be allowed to be True for Spot and Future, used for structured products.

Message: TripartiteAgreement

Message ID: 10146

Type: Reference Data Messages

Description: This object defines agreement for tripartite.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	key		String	Is used to identify the parent object (is set to null if this is the root object). This field is set by RTC, only set on outgoing messages on the reference data flow.
2	cacheId		String	Is used to identify the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow.
3	action		Integer	Identify the reason for the cache action (CACHE_ACTION), i.e. if it is an addition of a new reference data object, an update of an existing object or a removal of an object from the reference data cache. This field is set by RTC, only set on outgoing messages on the reference data flow. Allowed values: see constant group CACHE_ACTION
4	stateSequenceNumber		long	A sequence number that is incremented with each reference data update, i.e. a version number for the cache contents. The sequence number series is common for all caches. This means that for a specific cache instance, the sequence number is not necessarily consecutive (but constantly increasing). This field is set by RTC, only set on outgoing messages on the reference data flow.
5	uniqueObjectId		String	The id is unique among all objects and may be used to retrieve a specific instance. Do not however, try to interpret the

				contents. This field is set by RTC, only set on outgoing messages on the reference data flow.
6	timeStamp		String (23)	The date and time of the latest modification for this reference data object. Format: yyyy-mm-ddTHH.MM.SS.sss. May be null if the object never has been updated. This field is set by RTC, only set on outgoing messages on the reference data flow.
8	tripartiteAgreementId	required	String (100)	Uniquely identifies each tripartite agreement.
9	initiator	required	String (100)	Initiator of the tripartite allocation. ID of Trading Member or Branch.
10	acknowledger	required	String (100)	Acknowledger (receiver) of the tripartite allocation. ID of Trading Member or Branch.
11	client	required	String (100)	Client of the tripartite allocation.
12	fromDate	required	String	From date of tripartite agreement. The format is yyyy-mm-dd.
13	toDate		String	To date of tripartite agreement. The format is yyyy-mm-dd.

External Members

Message: CdAddRtcMemberClientClearingLinkReq

Message ID: 10127

Type: External Members

Description: Request to add a Clearing member to a client

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag
12	memberClientId	required	String	The member id
13	clearingMemberId	required	String	The clearing member

This request will normally return a response of type [CdResponse](#) .

Message: CdAddRtcMemberClientReq

Message ID: 10031

Type: External Members

Description: Request to add a Client

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag
14	address	required	String	The address
15	phone	required	String	The phone
16	associatedMemberId	required	String	The parent member
18	email	required	String	The email address
20	clientId		String	The client id
21	clientName	required	String	The client name
22	vatRegNumber		String	VAT registration number
24	isStaff	required	Boolean	Is Staff, true or false

25	isBeneficial	required	Boolean	Is Beneficial, true or false
26	bdaCode		Integer	BDA Code
27	country	required	String	Country Code
28	strateCode		String	Code of client or member at CSD.
29	allowedMarkets	required	String	A comma-delimited list of market codes that the client is allowed to have trades and positions in.
30	clientType	required	String	Type of client. Information to surveillance. Allowed values: see constant group ClientType
31	idNumber		Long	ID number. Required for local individual clients.
32	passportNumber		String	Passport number. Required for foreign individual clients.
33	companyRegistrationNumber		String	Company registration number. Required for all company clients.
34	isProfessional		boolean	Information to surveillance.
35	isShariah		boolean	Information to surveillance.
36	isDiscretionary	required	Boolean	Is Discretionary, true or false
37	nominatedMember		String	The member id of the nominated member.
38	preferredCcy		String (3)	Currency used for intraday rebalancing. Needs to be an eligible FX collateral currency (or ZAR, this will be the default though).

This request will normally return a response of type [CdAddRtcMemberClientRsp](#).

Message: CdAddRtcMemberClientRsp

Message ID: 10474

Type: External Members

Description: The response to add rtc member client request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	latestSSN		long	This is the latest and most likely the highest state sequence number, SSN, that has been assigned to the reference data.
8	clientId		String	The client id generated by the system.

Message: CdAddRtcPositionAccountReq

Message ID: 10034

Type: External Members

Description: Request to add a Position Account

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag
12	participantUnitId	required	String	The owner unit.
14	positionAccountType	required	Integer	The type of the position account. Allowed values: see constant group PositionAccountType
15	clearingMemberId	required	String	The clearing member of the Position Account.
16	externalPositionAccountId		String	The external id of the position account.
17	positionAccountSubType	required	Integer	The sub type of the position account. Allowed values: see constant group PositionAccountSubType

This request will normally return a response of type [CdAddRtcPositionAccountRsp](#).

Message: CdAddRtcPositionAccountRsp

Message ID: 10035

Type: External Members

Description: The response to add position account request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	latestSSN		long	This is the latest and most likely the highest state sequence number, SSN, that has been assigned to the reference data.
9	positionAccountExternalId		String	The external id of the position account.

Message: CdEnableDisableRtcMemberClientReq

Message ID: 10152

Type: External Members

Description: Request to add a Client

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag
12	clientId	required	String	The client id
13	isDisabled	required	Boolean	Set as enabled or not

This request will normally return a response of type [CdResponse](#).

Message: CdEnableDisableRtcPositionAccountReq

Message ID: 10072

Type: External Members

Description: Request to update a position account

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag
12	positionAccountId	required	Long	The Position account id.
16	isEnabled		boolean	Enabled (true) or not (false).

This request will normally return a response of type [CdEnableDisableRtcPositionAccountRsp](#).

Message: CdEnableDisableRtcPositionAccountRsp

Message ID: 10073

Type: External Members

Description: The response enableDisable position account.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	latestSSN		long	This is the latest and most likely the highest state sequence number, SSN, that has been assigned to the reference data.

Message: CdSetClientAMPercentageReq

Message ID: 10443

Type: External Members

Description: Set additional margin percentage. Leaving clientId empty means trading member will set additional margin percentage on all clients for that trading member.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag
12	clearingMemberId		String	The clearing member id. The clearing member (or CM link) the client belongs to.
13	clientId		String	The client id. Only applicable for a clearing member that wants to set the AM percentage on a specific client.
14	amPercentage		Integer	The additional margin percentage used to calculate additional margin from the calculated initial margin.

This request will normally return a response of type [CdResponse](#).

Message: CdSetClientRiskLimitReq

Message ID: 10294

Type: External Members

Description: Set risk limit. Leaving clientId empty means trading member will set risk limit on all clients for that trading member.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag
12	clearingMemberId		String	The clearing member id. The clearing member (or CM link) the client belongs to.
13	clientId		String	The client id. Only applicable for a clearing member that wants to set the risk limit on a specific client.
14	riskLimit		Long	The risk limit which is validated against the calculated risk on a risk node to check if there is a breach and trigger alert if required. If this field is left blank then the risk limit is removed from the client

This request will normally return a response of type [CdResponse](#) .

Message: CdSetMinimumZARLimitReq

Message ID: 10273

Type: External Members

Description: Set minimum ZAR limit. Leaving all fields empty as a clearing member will set minimumZARLimit on all trading members for that clearing member. Leaving all fields empty as a trading member will set minimumZARLimit on all clients for that trading member.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag
12	clearingMemberId		String	The clearing member id. Only applicable for a trading member that wants to set the minimumZARLimit on a specific client.
13	tradingMemberId		String	The trading member id. Only applicable for a clearing member that wants to set the minimumZARLimit on a specific trading member.
14	clientId		String	The client id. Only applicable for a trading member that wants to set the minimumZARLimit on a specific client.
15	minimumZARLimit		Integer	The minimum ZAR limit in percent. Must be within the interval 0-100. Leaving this field empty means no specific limit is set and that a value from higher in the hierarchy will be used.

This request will normally return a response of type [CdResponse](#) .

Message: CdSetTradingMemberAMPercentageReq

Message ID: 10442

Type: External Members

Description: Set additional margin percentage. Leaving the tradingMemberId field empty means the clearing member will set additional margin percentage on all trading members for that clearing member.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag
12	tradingMemberId		String	The trading member id. Only applicable for a clearing member that wants to set the AM percentage on a specific trading member.
13	amPercentage		Integer	The additional margin percentage used to calculate additional margin from the calculated initial margin.

This request will normally return a response of type [CdResponse](#) .

Message: CdSetTradingMemberRiskLimitReq

Message ID: 10293

Type: External Members

Description: Set risk limit. Leaving the tradingMemberId field empty means the clearing member will set risk limit on all trading members for that clearing member.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag
12	tradingMemberId		String	The trading member id. Only applicable for a clearing member that wants to set the limit on a specific trading member.
13	riskLimit		Long	The risk limit which is validates against the calculated risk on a risk node to check if there is a breach and trigger alert if required. If this field is left blank then the risk limit is removed from the trading member

This request will normally return a response of type [CdResponse](#) .

Message: CdUpdateRtcMemberClientReq

Message ID: 10147

.

Type: External Members

Description: Request to update a Client

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag
12	clientId	required	String	The client id
13	clientName	required	String	The member name
14	address	required	String	The address
15	phone	required	String	The phone

16	email	required	String	The email address
17	vatRegNumber		String	VAT registration number
19	isStaff	required	Boolean	"Is Staff, true or false"
20	isBeneficial	required	Boolean	"Is Beneficial, true or false"
21	bdaCode		Integer	Broker Deal Account number
22	country	required	String	Country Code
23	strateCode		String	Code of client or member at CSD.
27	companyRegistrationNumber		String	Company registration number. Required for all company clients.
28	isProfessional		boolean	Information to surveillance.
29	isShariah		boolean	Information to surveillance.
30	isDiscretionary	required	Boolean	"Is Discretionary, true or false"
31	nominatedMember		String	The member id of the nominated member.
32	preferredCcy		String (3)	Currency used for intraday rebalancing.
33	idNumber		Long	ID number. Required for local individual clients.
34	clientType	required	String	Type of client. Information to surveillance. Allowed values: see constant group ClientType
35	passportNumber		String	Passport number. Required for foreign individual clients.
36	isNonResident	required	Boolean	Indication of the Client is resident of South Africa or not. If Country Code is ZA, the Client must be Resident. If Country Code is not ZA, the Client must be Non Resident, for the From Trade.

This request will normally return a response of type [CdResponse](#) .

Event Messages

Message: AccountPositionEvent

Message ID: 10032

Type: Event Messages

Description: Represents a change in a position for a specific instrument in a specific account.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
47	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
51	accountId	required	long	Account id.
53	positionType	required	Integer	The position type. Allowed values: see constant group PositionType
54	settlementDate		String	Settlement date, used for positions of type SETTLEMENT.
55	subId		long	In some cases, several positions in the same instrument and the same account are held as separate positions, although they both fit the instrument/account key. There may be technical or business logic reasons for this. The risk system can handle that multiple sources have positions in identical instruments and accounts and it will perform the aggregation of such positions itself for calculation purposes.

60	longQty		long	<p>The position long quantity in its canonical unit, i.e. longNominalQty multiplied by contract size of tradable instrument. A positive quantity represents a long position. E.g. a position of 10 futures contracts of an instrument with a contract size of 100 would be represented in this field as 1000 (divisor rule to be applied).</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.</p>
61	shortQty		long	<p>The position short quantity in its canonical unit, i.e. shortNominalQty multiplied by contract size of tradable instrument. A negative quantity represents a short position. E.g. a position of 10 futures contracts of an instrument with a contract size of 100 would be represented in this field as 1000 (divisor rule to be applied).</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.</p>
62	shortNominalQty		BigInteger	<p>The position quantity, i.e. number of contracts. Only set for positions in tradable instruments. A negative quantity represents a short position. E.g. a position of 10 futures contracts of an instrument with a contract size of 100 would be represented in this field as 10 (divisor rule to be applied).</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.</p>
63	shortInitialValue		BigInteger	<p>The value of the position, according to the prices of the trades that build up the position. Simply put, this value is the sum of the price * quantity of all trades in the position. Since long and short quantities have different signs, values of long positions are typically positive and values of short positions are typically negative (the opposite being true if the price is negative). The initial value of the position is expressed in the currency of the instrument the position belongs to.</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.</p>
64	shortMarketValue		BigInteger	<p>The value of the position, according to the market price that was used last time a cash settlement of the variation margin of the position was made. Since long and short quantities have different signs, values of long positions are typically positive and values of short positions are typically negative (the opposite being true if the price is negative). The market value is expressed in the currency of the instrument the position belongs to.</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.</p>
65	longNominalQty		BigInteger	<p>The position quantity, i.e. number of contracts. Only set for positions in tradable instruments. A positive quantity represents a long position. E.g. a position of 10 futures contracts of an instrument with a contract size of 100 would be represented in this field as 10 (divisor rule to be applied).</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.</p>
66	longInitialValue		BigInteger	<p>The value of the position, according to the prices of the trades that build up the position. Simply put, this value is the sum of the price * quantity of all trades in the position. Since long and short quantities have different signs, values of long positions are typically positive and values of short positions</p>

				are typically negative (the opposite being true if the price is negative). The initial value of the position is expressed in the currency of the instrument the position belongs to. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
67	longMarketValue		BigInteger	The value of the position, according to the market price that was used last time a cash settlement of the variation margin of the position was made. Since long and short quantities have different signs, values of long positions are typically positive and values of short positions are typically negative (the opposite being true if the price is negative). The market value is expressed in the currency of the instrument the position belongs to. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
68	transactionId		long	Internal RTC transaction ID.
69	positionReason		int	Allowed values: see constant group PositionReason
70	tradeId		long	Trade ID generated by the system, globally unique. A new ID will be created each time the trade is moved to a new account, i.e. when an equal opposite trade is created. Note that an ID will be created also at trade capture.
71	positionTimestamp		String	The time when the event occurred, on format yyyy-MM-dd'T'HH:mm:ss.SSS.
72	trade		Trade	The trade that caused this account position event.
74	subscriptionGroup		int	The id of subscription group the message is published on.
75	longSpreadVolume		BigInteger	Total spread volume for CFD contract. This is similar to Initial Value, but this is base rate from the trade multiplied with the quantity. Positive for long values, negative for short. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
76	shortSpreadVolume		BigInteger	Total spread volume for CFD contract. This is similar to Initial Value, but this is base rate from the trade multiplied with the quantity. Positive for long values, negative for short. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
77	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.
78	externalAccountId		String	The ID of the position account as seen by the member.
79	clearingMemberId		String	The member code of the clearing member.
81	collateralAccountName		String	The name of the collateral account.
82	tradingMember		String	TM that owns the trade.
83	tradingMemberBranch		String	TM branch that owns the trade.
84	clientId		String	Client ID added to trade by CS. Empty if trade is for TM house.

Message: AccountTradeEvent

Message ID: 10141

Type: Event Messages

Description: This event represents an insertion, deletion or update of a trade in an account. The message includes information on the position on the account after the trade event.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
4	subscriptionGroup		int	The id of subscription group the message is published on.
5	accountId	required	long	Account id.
7	positionType	required	Integer	The position type. Allowed values: see constant group PositionType
8	settlementDate		String	Settlement date, used for positions of type SETTLEMENT.
10	positionReason		int	Allowed values: see constant group PositionReason
11	positionTimestamp		String	The time when the event occurred, on format yyyy-MM-dd'T'HH:mm:ss.SSS.
12	trade		Trade	The referenced Trade.
13	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.
14	externalAccountId		String	The ID of the position account as seen by the member.
15	clearingMemberId		String	The member code of the clearing member.
17	collateralAccountName		String	The name of the collateral account.
18	tradingMember		String	TM that owns the trade.
19	tradingMemberBranch		String	TM branch that owns the trade.
20	clientId		String	Client ID added to trade by CS. Empty if trade is for TM house.

Message: AggregatedSummaryClearingMemberEvent

Message ID: 10350

Type: Event Messages

Description: Aggregated account summary event for clearing member.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
4	subscriptionGroup		int	The id of the subscriptionGroup the message is published on.
5	clearingMemberId		String	Clearing Member ID.
6	totals		AggregatedSummaryDetails	Aggregated trading member totals.
7	interestRateOnCashCollateral		Long	Interest rate for cash collateral for ZAR This field is a fixed point number with a scaling factor equal to 1/DIVISOR.INTEREST.
8	netFromOtherSystems		Long	Net amount from other system

				This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
9	cmMessageRef		String	Number used when creating settlement instructions. This number is concatenated into the message reference no.
10	tag		String	Price Tag that identifies the event. Allowed values: see constant group CondType
11	businessDate		String	Business Date.
12	timestamp		String	The time of the event.

Message: AggregatedSummaryTradingMemberEvent

Message ID: 10351

Type: Event Messages

Description: Aggregated account summary event for trading member.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
4	subscriptionGroup		int	The id of the subscriptionGroup the message is published on.
5	tradingMemberId		String	Trading Member ID.
6	clearingMemberId		String	Clearing Member ID.
7	totalClients		AggregatedSummaryDetails	Aggregated client accounts.
8	totalHouse		AggregatedSummaryDetails	Aggregated house accounts.
9	interestRateOnCashCollateral		Long	Interest rate for cash collateral for ZAR This field is a fixed point number with a scaling factor equal to 1/DIVISOR.INTEREST.
10	tag		String	Price Tag. Allowed values: see constant group CondType
11	businessDate		String	Business Date.
12	timestamp		String	The time of the event.

Message: AtmVolatilityEvent

Message ID: 10277

Type: Event Messages

Description: Event with ATM volatilities for an underlying instrument. The dates are expiry dates for the options.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
4	subscriptionGroupId		int	The id of the subscription group the message is published on.
5	condType		String	Price Condition type (tag). Allowed values: see constant group CondType
6	businessDate		String	Business date. Format is YYYY-MM-DD.
8	instrumentExternalId		String	The external instrument id. This is the JSE Master ID.
9	feedSource		String	The source. Allowed values: see constant group FeedSource
10	timestamp		String	The time of the market data event. The format is "yyyy-MM-ddTHH:mm:ss.SSS".
11	absoluteDates		String []	The absolute instrument expiry dates. Corresponds to the array of volatilities.
12	vols		long []	The volatilities. Corresponds to the array of dates. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.

Message: CommissionEvent

Message ID: 10514

Type: Event Messages

Description: Commission to be paid by destination TM to initiating TM.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
2	subscriptionGroup		int	The id of subscription group the message is published on.
3	commissionId	required	String	The commission id.
4	market	required	String	Market.
5	initiatingCM	required	String	CM that is associated with the initiating TM and will receive the payment.
6	initiatingTM	required	String	TM that sent the commission.
7	destinationCM	required	String	CM that will handle the payment.
8	destinationTM	required	String	TM that will carry the payment.
9	clientReference		String	Payer of the commission. Refer to EMAPI Clearing document, Commission Management section for guidance on the population of this field based on the various trade and deal management scenarios.
10	commissionReference		String	Identifier of trade or deal associated with the commission. Refer to EMAPI Clearing document, Commission Management section for guidance on the

				population of this field based on the various trade and deal management scenarios.
11	amount	required	Long	Amount to debit the destination TM. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
12	status	required	int	The status of the commission. Allowed values: see constant group CommissionStatus
13	enteredTimestamp	required	String	The timestamp when the commission first was entered.
14	cancelledTimestamp		String	The timestamp when the commission was either cancelled or rejected.
15	cancellationReference		String	Optional reference that can be supplied at cancellation or rejection requests.
16	acceptedTimestamp		String	The timestamp when the pending commission was accepted.
17	initiatingExternalAccountId		String	External ID of the initiator's sub account that is affected by the commission.
18	destinationExternalAccountId		String	External ID of the destination sub account that is affected by the commission.
19	secondaryFirmReference		String	Additional reference to the destination member. The recipient of the commission. Can be used to reference trading desk.
20	commissionVatType	required	String	VAT amount for commission. Allowed values: see constant group CommissionVatType

Message: CurveEvent

Message ID: 10227

Type: Event Messages

Description: Market data for a curve. See the Curve reference data object for a definition of the curve.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
4	subscriptionGroupId		int	The id of the subscription group the message is published on.
5	condType		String	Price Condition type (tag). Allowed values: see constant group CondType
6	businessDate		String	Business date. Format is YYYY-MM-DD.
8	curveExternalId		String	The curve external instrument id. This is the JSE Master ID.
9	feedSource		String	The source of the curve. Allowed values: see constant group FeedSource
10	timestamp		String	The time of the market data event. The format is "yyyy-MM-ddTHH:mm:ss.SSS".
11	absoluteDate		String []	The absolute date on the x-axis of the curve. This field is populated if the xAxis unit of the curve is absolute.

12	dateFractionOfAYear		long []	The date on the x-axis represented as fraction of a year using the day time convention on the discount curve. This field is populated if the xAxis unit of the curve is fraction of a year. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
13	rate		long []	The rate on the y-axis of the curve. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.

Message: DailyAccountSummaryDetailsEvent

Message ID: 10295

Type: Event Messages

Description: Daily account summary for client or members house nodes.

Field no.	Field name (tag)	Mand .	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
4	subscriptionGroup		int	The id of the subscriptionGroup the message is published on.
5	riskNodeId		long	The id of the risk node the message is published on.
6	businessDate		String	Business Date. The format is yyyy-MM-dd.
7	Tag		String	Price Tag. Allowed values: see constant group CondType
8	clearingMemberId		String	Clearing Member Id.
9	tradingMemberId		String	Trading Member Id.
10	clientId		String	Client Id.
11	strateCode		String	Strate Code.
12	collateralAccountId		Long	Collateral Account Id.
13	minimumCashLimit		Long	Percentage of initial margin that must be covered with cash collateral. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
14	initialMargin		Long	Initial Margin. This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
15	additionalMargin		Long	Additional Margin.

				This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
16	requestedSecuritiesAmount		Long	Value request from Strate. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
17	receivedSecuritiesAmount		Long	Received value from strate, before haircut. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
18	registeredSecuritiesAmount		Long	Market Value of the positions in security collateral in ZAR (market value of the security collateral position). This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
19	registeredCashAmount		Long	Registered Cash Amount. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
20	securitiesAmountCF		Long	Current securities amount, after haircut. This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
21	securitiesAmountBF		Long	Previous securities amount, after haircut. This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
22	cashAmountCF		Long	Current cash collateral amount (for ZAR), after haircut. This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
23	cashAmountBF		Long	Previous cash collateral amount (for ZAR), after haircut. This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

24	State		String	State. Valid values are: REGISTERED, COLLATERAL_CALCULATED, COLLATERAL_SEC_REGISTERED, COLLATERAL_SEC_STEPOVER, COLLATERAL_FX_REGISTERED, COLLATERAL_FX_STEPOVER, COLLATERAL_CASH_SETTLED, CLIENT_PAYMENTS,REBALANCING.
25	Status		String	Status of execution. Valid values are: SUCCESS, ERROR, OVERRIDE, VALIDATION_FAILED, VALIDATION_WARNING,
26	statusText		String	Status Text. Description of validation errors.
27	variationMargin		Long	Variation Margin. A Positive value means the money being paid by the Client to the CH and a Negative value means the money being paid by the CH to the Client This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
28	fundingInterestAmount		Long	Funding Interest amount. A Positive value means the money being paid by the Client to the CH and A Negative value means the money being paid by the CH to the Client This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
29	dividendAmount		Long	Dividend amount. A Positive value means the money being paid by the Client to the CH and a Negative value means the money being paid by the CH to the Client This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
32	settlementDate		String	Settlement Date. The format is yyyy-MM-dd.
33	previousInitialMargin		Long	Previous Business Day Initial Margin. This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
34	interestAmountOnCashCollateral		Long	Interest amount earned on cash collateral for ZAR. A Negative value means the money being paid by the CH to the Client This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

35	interestRateOnCashCollateral		Long	Interest rate on cash collateral for ZAR. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.INTEREST.
36	fxDailyAccountSummaryDetails		FxDailyAccountSummaryDetails []	The daily account summary details for FX currencies.
37	previousAdditionalMargin		Long	Previous Business Day Additional Margin. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
38	fxCashAmountCF		Long	Current cash collateral amount for FX currency (in ZAR), after haircut. This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
39	fxCashAmountBF		Long	Previous cash collateral amount for FX currency (in ZAR), after haircut. This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
40	netAmount		Long	Net amount from other systems. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
41	securitiesAmountMovement		Long	Securities amount movement. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
42	cashAmountMovement		Long	Cash collateral amount (for ZAR) movement. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
43	fxCashAmountMovement		Long	Cash collateral amount movement for FX currency (in ZAR) This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
44	totalZarCashMovement		Long	Sum of all ZAR movements: Cash Collateral Movement + VM + Booking fees incl. VAT + Risk fees incl. VAT + Commissions + Funding interest + Dividend payment + Interest amount on Cash Collateral. A Positive value means the money being paid by the Client to the CH and a Negative value

				means the money being paid by the CH to the Client This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
45	timestamp		String	The time of the event. The format is yyyy-MM-ddTHH:mm:ss.SSS.
46	commission		Long	Sum of all commissions with status New. A Positive value means the money being paid by the Client to the CH and a Negative value means the money being paid by the CH to the Client This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
47	bookingFeeAmount		Long	Booking fee amount excluding VAT. A Positive value means the money being paid by the Client to the CH This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
48	bookingFeeVatAmount		Long	VAT for booking fee This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
49	riskFeeAmount		Long	Interest Consideration at 17bps. The interest consideration is exempt from VAT. A Positive value means the money being paid by the Client to the CH and a Negative value means the money being paid by the CH to the Client. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
50	riskFeeVatAmount		Long	Interest consideration is exempt from VAT. This field is disseminated with a 0 value. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

Message: DividendEvent

Message ID: 10229

Type: Event Messages

Description: Dividend information for a tradable instrument. The instrument is a spot used as underlying for futures. The dividends are used in the future valuation.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
4	subscriptionGroupId		int	The id of the subscription group the message is published on.
5	condType		String	Price Condition type (tag). Allowed values: see constant group CondType
6	businessDate		String	Business date. Format is YYYY-MM-DD.
8	instrumentExternalId		String	The external instrument id. This is the JSE Master ID.
9	feedSource		String	The source of the price. Allowed values: see constant group FeedSource
10	timestamp		String	The time of the market data event. The format is "yyyy-MM-ddTHH:mm:ss.SSS".
11	exDate		String	The date on or after which a security is traded without rights to a previously declare dividend.
12	paymentDate		String	The date on which a declared stock dividend is scheduled to be paid.
13	dividend		long	The dividend value. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

Message: OptionDataEvent

Message ID: 10276

Type: Event Messages

Description: Valuation information for option.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
4	subscriptionGroupId		int	The id of the subscription group the message is published on.
5	condType		String	Price Condition type (tag). Allowed values: see constant group CondType
6	businessDate		String	Business date. Format is YYYY-MM-DD.
8	externalInstrumentId		String	The external instrument id. This is the JSE Master ID.
9	price		long	The option price. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
10	delta		long	The delta for the option. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
11	volatility		long	The volatility for the option.

				This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
12	feedSource		String	The source of the curve. Allowed values: see constant group FeedSource
13	timestamp		String	The time of the market data event. The format is "yyyy-MM-ddTHH:mm:ss.SSS".
14	gamma		long	The gamma for the option. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
15	vega		long	The vega for the option. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

Message: PriceEvent

Message ID: 10074

Type: Event Messages

Description: Market data for a tradable instrument.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
9	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
12	subscriptionGroupId		int	The id of the subscription group the message is published on.
14	price		long	The price. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
17	condType		String	Price Condition type (tag). Allowed values: see constant group CondType
18	businessDate		String	Business date. Format is YYYY-MM-DD.
19	externalInstrumentId		String	The external instrument id. This is the JSE Master ID.
20	feedSource		String	The source of the curve. Allowed values: see constant group FeedSource
21	timestamp		String	The time of the market data event. The format is "yyyy-MM-ddTHH:mm:ss.SSS".

Message: RiskNodeEvent

Message ID: 10033

Type: Event Messages

Description: This event contains calculated risk values for a Risk Node.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
23	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
27	riskNodeId		long	The id of the risk node the message is published on.
28	channel		String	The channel this result was calculated for. DEFAULT is the normal channel.
29	currency		String	The currency code according to ISO 4217.
30	portfolioValue		long	This value is not used in the JSE implementation of RTC.
31	variationMargin		long	Any profit or loss given the current market value compared to the previous mark-to-market value (or trade value). This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
32	portfolioRisk		long	The Initial Margin for the risk node, defined as J-SPAN IM + Liquidation Period Add-On + Large Position Add-On + SM. A positive value indicates a risk. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
33	collateralValue		long	The sum of the values of all the collateral positions in the collateral account of the risk node. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
34	subscriptionGroup		int	The id of the subscriptionGroup the message is published on.
35	liquidationAddOn		long	The liquidation period add-on value for the risk node. The liquidation period add-on is an amount that gets added to the margin calculated by J-SPAN. The Liquidation Period add-on increases the margin requirement when the client's notional exposure in a particular underlying forms a significant portion of the value that gets traded in the market on a daily basis. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
36	largePositionAddOn		long	The large position add-on value on the risk node. This is an additional IM to compensate for large positions or concentration risk. This calculation takes into account position size thresholds which will be defined by the JSE. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
37	jspanValue		long	The J-SPAN IM value, based on the positions on the risk node and the CSE risk arrays. The J-SPAN algorithm uses the

				<p>netted positions of all the accounts under the risk node.</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
38	additionalMargin		long	<p>The additional margin value on the risk node. Additional Margin is a margin that is added on top of IM and calculated as a percentage on IM. Different members and clients can have different additional margin percentages.</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
39	additionalMarginPercentage		long	<p>The additional margin percentage value on the risk node.</p>
40	riskLimit		long	<p>The risk limit value on the risk node. The global risk limit for the clearing house, or a more strict limit for the Clearing Member, Trading Member or Client.</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
41	valueAgainstLimit		long	<p>The value against limit value on the risk node, calculated as (IM + AM) - (VM + Collateral value).</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
42	alert		boolean	<p>The alert indicator for the risk node. True if valueAgainstLimit is larger than riskLimit, false otherwise.</p>
43	settlementMargin		long	<p>The settlement margin on the risk node. For risk nodes with settlement positions for physical delivery positions: SM = official SMR * quantity (netted on risk node).</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
44	equityNotionalValue		long	<p>The notional exposure for asset class Equity for this risk node.</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
45	fxNotionalValue		long	<p>The notional exposure for asset class FX for this risk node.</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
46	notionalValues		NotionalValue []	Notional value per underlying.
47	collateralPositions		CollateralPositionValue []	Values for collateral positions.

Message: SurfaceEvent

Message ID: 10228

Type: Event Messages

Description: Market data for a surface. See the Surface reference data object for a definition of the surface.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
4	subscriptionGroupld		int	The id of the subscription group the message is published on.
5	condType		String	Price Condition type (tag). Allowed values: see constant group CondType
6	businessDate		String	Business date. Format is YYYY-MM-DD.
9	feedSource		String	The source of the price. Allowed values: see constant group FeedSource
10	timestamp		String	The time of the market data event. The format is "yyyy-MM-ddTHH:mm:ss.SSS".
11	absoluteDate		String []	The absolute date on the x-axis of the surface. This field is populated if the xAxis unit of the surface is absolute.
12	dateFractionOfAYear		Long []	The date on the x-axis represented as fraction of a year using the day time convention on the surface. This field is populated if the xAxis unit of the surface is fraction of a year. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
13	strikeOrMoneyness		Long []	Value on the y-axis: strike or moneyness. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
14	volatility		Long []	Value on the z-axis: volatility in point (x, y). This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
15	surfaceExternalId		String	The surface external instrument id. This is the JSE Master ID.

Message: WithdrawalNotificationEvent

Message ID: 10486

Type: Event Messages

Description: A notification event sent to the Clearing Member that there is a number of payment advices that the CM must confirm.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
4	subscriptionGroup		int	The id of the subscriptionGroup the message is published on.
6	clearingMember		String	Clearing Member ID.

Message: YieldEvent

Message ID: 10437

Type: Event Messages

Description: The yield for an instrument. The instrument could be of different types, for instance a Bond or a Deposit.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
4	subscriptionGroupId		int	The id of the subscription group the message is published on.
5	condType		String	Price Condition type (tag). Allowed values: see constant group CondType
6	businessDate		String	Business date. Format is YYYY-MM-DD.
8	instrumentExternalId		String	The external instrument id.
9	feedSource		String	The source of the price. Allowed values: see constant group FeedSource
10	timestamp		String	The time of the market data event. The format is "yyyy-MM-ddTHH:mm:ss.SSS".
11	yieldType		String	Allowed values: see constant group YieldType
12	yield		long	The dividend expressed in yield (% on decimal form). This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.

Event Messages (Internal)

Message: AggregatedSummaryDetails

Message ID: 10352

Type: Event Messages

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: Object to hold aggregated account summary details.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	totalMember_CF_Cash		Long	The total ZAR Collateral Cash registered for the current day. This value will always be Positive

				This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
2	totalMember_BF_Cash		Long	<p>The total ZAR Collateral Cash registered for the previous day. This value will always be Positive</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
3	totalMember_CF_Sec		Long	<p>The total Collateral Security registered for the current day. This value will always be Positive</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
4	totalMember_BF_Sec		Long	<p>The total Collateral Security registered for the previous day. This value will always be Positive</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
5	initialMarginMovementCash		Long	<p>The total overall IM Cash movement balance per member (TM + Client), i.e. C/F (today) Cash Collateral - B/F (previous days) Cash Collateral = Cash IM movement. This value is calculated the same way as totalMember_Cash_Movement. This value is Positive if the current Cash Collateral exceeds the previous Cash Collateral; negative if it is the other way around.</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
6	initialMarginMovementSecurities		Long	<p>The total overall IM Securities movement balance per member (TM + Client), i.e. C/F (today) Collateral Securities - B/F (previous days) Collateral Securities = Securities movement. This value is calculated the same way as totalMember_Sec_Movement. This value is Positive if the current Collateral Securities exceeds the previous Collateral Securities; negative if it is the other way around</p>

				This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
7	totalVariationMargin		Long	<p>Total variation margin for all clients of a trading member and clients of the trading member's branches. A Positive value means the money being paid by the Client to the CH and a Negative value means the money being paid by the CH to the Client</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
8	totalAdditionalMargin		Long	<p>Total additional margin. This value will always be Positive</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
9	fundingInterest		Long	<p>The interest calculated from CFDs. ((base rate + interest spread) X nominal). A Positive value means the money being paid by the Client to the CH and a Negative value means the money being paid by the CH to the Client</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
10	dividendPayment		Long	<p>This is calculated from the dividend neutrals journal transactions. A Positive value interprets as money being paid by the Client to the CH and a Negative value means the money being paid by the CH to the Client</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
11	totalBookingFees		Long	<p>Total booking fees excluding VAT. A Positive value means the money being paid by the Client to the CH</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
12	totalBookingFeesVAT		Long	<p>Total VAT for booking fees</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>

13	initialMargin		Long	Initial Margin. This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
14	previousInitialMargin		Long	Previous Business Day Initial Margin. This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
15	totalInterestAmountOnCashCollateral		Long	Interest amount earned on cash collateral for ZAR. A Negative value means the money being paid by the CH to the Client This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
16	fxDailyAccountSummaryDetails		FxDailyAccountSummaryDetails []	The daily account summary details for FX currencies.
17	totalMember_CF_FxCash		Long	The total FX Collateral Cash registered (in ZAR value) for the current day. This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
18	totalMember_BF_FxCash		Long	The total FX Collateral Cash registered (in ZAR value) for the previous day. This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
19	additionalMarginMovements		Long	Additional Margin Movements (from yesterday). This value can be Positive if the current AM exceeds the previous AM; negative if it is the other way around This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
20	totalMember_Cash_Movement		Long	The total ZAR Collateral Cash movement balance per member (TM + Client), i.e. C/F (today) Cash Collateral - B/F (previous days) Cash Collateral = Total Cash movement. This value is calculated the same way as initialMarginMovementCash. This value is positive if the

				<p>current cash collateral exceeds the previous cash collateral and negative if it is the other way around.</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
21	totalMember_FxCash_Movement		Long	<p>The total FX Collateral Cash registered (in ZAR value) Movement. This value is positive if the current FX collateral exceeds the previous FX collateral and negative if it is the other way around</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
22	totalMember_Sec_Movement		Long	<p>The total Collateral Security movement balance per member (TM + Client), i.e. C/F (today) Collateral Securities - B/F (previous days) Collateral Securities = Securities movement. This value is calculated the same way as initialMarginMovementCash. This value is Positive if the current securities collateral exceeds the previous securities collateral; negative if it is the other way around.</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
23	totalZARCashMovement		Long	<p>Sum of all ZAR movements for the Member: Net Amount from other system + Member Cash Collateral Movement + Clients Cash Collateral Movement + Member VM + Clients VM + Net Booking Fees including VAT + Risk Fees including VAT + Commissions + Funding interest + Dividend payment + Interest amount on Cash Collateral. A Positive value means the money being paid by the Client to the CH and a Negative value means the money being paid by the CH to the Client</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
24	commission		Long	<p>Sum of all commissions with status New. A Positive value means the money being paid by the Client to the CH and a</p>

				<p>Negative value means the money being paid by the CH to the Client</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
25	totalRiskFees		Long	<p>Total Interest Consideration at 17bps. The interest consideration is exempt from VAT. A Positive value means the money being paid by the Client to the CH and A Negative value means the money being paid by the CH to the Client.</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
26	totalRiskFeesVAT		Long	<p>Interest consideration is exempt from VAT. This field is disseminated with a 0 value.</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>
27	registeredSecuritiesAmount		Long	<p>Market Value of the positions in security collateral in ZAR (market value of the security collateral position). This value will always be Positive</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.</p>

Message: CollateralPositionValue

Message ID: 10290

Type: Event Messages

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: Collateral position values before and after haircut.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	isCash		boolean	True if it is a cash position.
3	quantity		long	<p>The collateral position quantity.</p> <p>This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.</p>
4	marketValue		long	Market value of the collateral position.

				This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
5	valueAfterHaircut		long	Collateral position value after applying haircut and maximum amount. The haircut and maximum amount are taken from the EligibleSecurity list or EligibleCurrency object. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
6	externalInstrumentId		String	The JSE Master ID of the collateral security, or the currency if the collateral position is in ZAR or FX.
7	price		long	Price of the collateral position. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
8	haircut		long	Haircut of the collateral position. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

Message: FxDailyAccountSummaryDetails

Message ID: 10376

Type: Event Messages

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: FX part of the daily account summary.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	currencyCode		String	The Currency Code, ISO 4217 alphabetic Code. Example: USD
2	interestRateOnCashCollateral		Long	Interest rate for the currency. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.INTEREST.
3	interestAmountOnCashCollateral		Long	Interest amount for the currency. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
4	cashCollateralMovement		Long	Total movement in Cash Collateral for the current day that will be settled in the given currency This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
5	fxCollateralQty		Long	FX Collateral position quantity (in FX currency). This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.

6	fxMarketValue		Long	FX collateral position market value (in ZAR). This is the value before haircut: qty * price. This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
7	fxCollateralValue		Long	FX collateral value (in ZAR). This is the value calculated during the collateral process (using haircut%, max value). This value will always be Positive This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

Message: NotionalValue

Message ID: 10289

Type: Event Messages

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: Object to hold notional values per underlying.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	netNotionalValue		long	The net notional exposure of the specified underlying instrument. The Net Notional Exposure per Underlying is defined as the aggregated notional exposure (with sign) of all positions in tradable instruments with the same underlying spot. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
3	grossNotionalValue		long	The gross notional exposure of the specified underlying instrument. Gross is defined as the sum of the absolute notional exposure per position in instruments with the same underlying spot. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
4	externalInstrumentId		String	The external instrument id. This is the JSE Master ID.
5	liquidationPeriod		long	The calculated liquidation period. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

Message: RtcTradeExternalData

Message ID: 10085

Type: Event Messages

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: RtcTradeExternalData contains trade attributes specific for this configuration of RTC.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
10	tradingSystemMatchId		String	A system set identifier of the matched trade that was used to enter this trade into the system.
11	tradingSystemLinkId		String	A system set identifier of the matched trade that was used to enter this trade into the system.
12	aggressor		Boolean	Aggressor from Trading System.
13	capacity		Integer	Capacity from Trading System. Allowed values: see constant group Capacity
14	tradeType		String	Trade Type from Trading System. Allowed values: see constant group RtcTradeType
15	tradingSystemTradeHalfId		String	A system set identifier of the trade half from the external trading system.
16	clientOrderId		String	Reference to an order in the trading system.
17	outsidePriceBand		boolean	Flag to indicate that the trade price is outside price band. Used both on trades from trading system and on trades created in Deal Management.
18	zeroFeeFlag		boolean	Flag to indicate that the trade is marked for zero fee.
19	timeOfEntry		String	Timestamp from Trading System. Format 2014-11-18 13:24:21.
20	tradingUser		String	The dealer in the trading system.
21	onBookQuantity		Long	Quantity from trading system for on-book trades. 0 if trade is Off-book. Kept on each side and not changed after trade management unless for accumulations. When an accumulated trade is partly allocated, on-book and off-book quantities are no longer maintained. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
22	offBookQuantity		Long	Quantity from trading system for reported trades. 0 if trade is On-book. Kept on each side and not changed after trade management unless for accumulations. When an accumulated trade is partly allocated, on-book and off-book quantities are no longer maintained. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
23	originalTradingSystemMatchId		String	Refers to the original trade. Only used when receiving cancelled trades from the trading system.
24	agreedTime		String	Time agreed between TM1 and TM2. Only applicable for reported trades. The format is yyyyMMdd-HH:mm:ss.fff.
25	reportedTime		String	The time the reported trade was received on the trading system (system generated). The format is yyyyMMdd-HH:mm:ss.fff..
26	optionDelta		Long	Option delta from the trading system. Valid for Options.

				This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
27	firmTradeId		String	Free text field carried from the trading system for off book trades.
28	tradeReportId		String	Free text field carried from the trading system for on and off book trades.
29	eventLinkId		String	Unique key for all trades generated from a single event. Will only be populated for strategy trades.
30	tradeSubType	required	String	Trade sub type used for trades.

Message: Trade

Message ID: 10015

Type: Event Messages

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: This class represents a Trade, i.e. one side of a Matched Trade: the buyer's or the seller's.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	tradeId		long	Trade id. Internal RTC trade Half ID.
2	accountId		long	Account id. Internal RTC ID.
4	dealId		long	Internal RTC ID for both sides of a matched trade.
6	price		BigInteger	Price. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
7	tradeTimestamp		String	The time the trade occurred. The format is yyyy-MM-ddTHH:mm:ss.SSS.
10	isBuy		boolean	To identify whether buy or sell trade
11	tradeBusinessDate		String	The business date the trade occurred. The format is yyyy-MM-dd.
12	originalQuantity		BigInteger	Original quantity of the trade. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
13	remainingQuantity		BigInteger	Remaining quantity of the trade after allocation(if any). This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
14	tradeExternalData		RtcTradeExternalData	External information on trade from Trading System.
15	nextTradeIds		long []	List of forward referenced trade IDs
16	previousTradeIds		long []	List of backward referenced trade IDs
17	activeQuantity		BigInteger	Active quantity of the trade after allocation (if any).

				This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
18	externalMatchedTradeId		String	Trade ID set by the client entering the trade into the clearing system.
19	reservedQuantity		BigInteger	Reserved quantity for assign operation (if any). This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
21	initialValue		BigInteger	Value of the trade This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
22	interestRateSpread		BigInteger	Interest Rate Spread for trade if CFD. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
23	externalInstrumentId		String	The external instrument id. This is the JSE Master ID.
24	reference		String	The reference supplied with the trade.

External Members

Message: CmBalancing1Event

Message ID: 10411

Type: External Members

Description: Account balancing 1 message to the Clearing Member.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
4	subscriptionGroup		int	The id of the subscriptionGroup the message is published on.
6	clearingMember		String	Clearing Member ID.
7	businessDate		String	Business date which have generated the various payments. Note it is not the actual payment date, which happens at the business day after. The format is yyyy-MM-dd.
8	settlementDate		String	Date when the payment is due. The format is yyyy-MM-dd.
9	balances		MemberBalance1 []	List of balances for the clearing member or trading member. The totals for all TMs clearing through the CM are available under the CM ID.

Message: CmBalancing2Event

Message ID: 10413

Type: External Members

Description: Account balancing 2 message to CM.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
4	subscriptionGroup		int	The id of the subscriptionGroup the message is published on.
6	clearingMember		String	Clearing Member ID.
7	businessDate		String	Business date which have generated the various payments. Note it is not the actual payment date, which happens at the business day after. The format is yyyy-MM-dd.
8	settlementDate		String	Date when the payment is due. The format is yyyy-MM-dd.
9	interestRates		FxInterestRate []	Interest rates used for the day for interest on cash collateral calculations.
10	balances		MemberBalance2 []	List of balances for the clearing member or trading member. The totals for all TMs clearing through the CM are available under the CM ID.

Message: GiveUpEvent

Message ID: 10124

Type: External Members

Description: This event is for Assign and Tripartite flow from RTC.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
5	fourEyesId		long	The ID of the Assign or Tripartite.
6	initiator		String	The initiator of the request.
7	state		int	The current state of the operation. Allowed values: see constant group FourEyesState
8	tradeId		long	Trade id.
9	fromAccountId		Long	The current account for the Trade.
11	quantity		Long	The trade quantity. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
13	reference		String	An optional free text field.
14	acknowledger		String	The destination member Id.
17	timeInitiated		String	Time the action was initiated
18	timeComplete		String	Time the action was completed
19	timeCancelled		String	Time the action was cancelled
23	client		String	The destination client Id. Applicable for tripartite allocation only
24	subscriptionGroup		int	The id of the subscription group the message is published on.
25	price		Long	The trade price. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
26	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.

28	isBuy		Boolean	True if this trade is a buy trade, false if it is a sell trade.
29	commissionAmount		Long	A sight of the commission amount before the actual commission is submitted. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

Message: ReadyConfirmAvailableFXEvent

Message ID: 10377

Type: External Members

Description: Event published to the CMs to inform that RTC is ready to receive information about FX collateral.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	sequenceNumber		long	The sequence number is a unique number for all events published for the same flow.
4	subscriptionGroup		int	The id of the subscriptionGroup the message is published on.

Trade Management Messages

Message: AbandonOptionPositionReq

Message ID: 10188

Type: Trade Management Messages

Description: Abandon an option position.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	accountId	required	long	The account of the position.
3	quantity	required	Long	The abandon quantity. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
5	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.
6	updateId	required	String	Unique id that should be set for the request. If the system has already processed an update with this id, a success message is returned without any position change.

This request will normally return a response of type [AbandonOptionPositionRsp](#) .

Message: AbandonOptionPositionRsp

Message ID: 10189

Type: Trade Management Messages

Description: Response to the AbandonOptionPositionReq request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	tradeIds		Long []	The IDs of the trades that were created as part of this abandon of an option position.
7	transactionId		long	The ID of the transaction that updated the positions affected by this abandon of an option position.

Message: AcceptCommissionReq

Message ID: 10543

Type: Trade Management Messages

Description: Request for the destination member to accept a pending commission.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	possDup		boolean	The possible duplicate flag. Not used for this message.
2	commissionId	required	String	Id of the Commission.
3	destinationExternalAccountId		String	External ID of the destination sub account that is affected by the commission.
4	market		String	Market id. Required if destinationExternalAccountId is not left empty.
5	destinationTM		String	TM that will carry the payment. Required if requesting user is a CM user and if destinationExternalAccountId is not left empty.

This request will normally return a response of type [ResponseMessage](#) .

Message: AddCommissionReq

Message ID: 10515

Type: Trade Management Messages

Description: Request to add a Commission

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	market	required	String	Market.
3	initiatingTM	required	String	TM or Branch that sent the commission.
4	destinationTM	required	String	TM or Branch that will carry the payment.
5	clientReference		String	Payer of the commission. Refer to EMAPI Clearing document, Commission Management section for guidance

				on the population of this field based on the various trade and deal management scenarios.
6	commissionReference		String	Identifier of trade or deal associated with the commission. Refer to EMAPI Clearing document, Commission Management section for guidance on the population of this field based on the various trade and deal management scenarios.
7	amount	required	Long	Amount to debit the destination TM. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.DECIMAL.
8	updateId	required	String	Unique id that should be set for the commission request. If the system has already processed an update with this id, a success message is returned without any position change.
10	initiatingExternalAccountId		String	External ID of the initiator's sub account that is affected by the commission.
11	secondaryFirmReference		String	Additional reference to the destination member. The recipient of the commission. Can be used to reference trading desk.
12	commissionVatType	required	Int	VAT type for commission. Allowed values: see constant group CommissionVatType

This request will normally return a response of type [ResponseMessage](#) .

Message: AggregateTradesReq

Message ID: 10049

Type: Trade Management Messages

Description: A number of trades can be accumulated into a single trade with a volume weighted average price. The trades need to be on the same account, the same contract and the same side (only buy or only sell) from current day.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	moveId		Long	Must be unique for the referenced trade. Used to prevent duplicate move requests.
3	tradeIds	required	long []	Trade ids.
5	reference		String	An optional free text field.
6	destinationReference		String	The destination for this operation.
8	accountId	required	long	The account id for the trades.
9	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.
10	tradingUserId	required	String	The trading user id.

This request will normally return a response of type [AggregateTradesRsp](#) .

Message: AggregateTradesRsp

Message ID: 10050

Type: Trade Management Messages

Description: Response to the AggregateTrades request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	tradeId		Long	The IDs of the trade that were created as part of this move trade operation.
7	transactionId		long	The ID of the transaction that updated the positions affected by this move trade operation.

Message: AllocateTradeReq

Message ID: 10104

Type: Trade Management Messages

Description: The purpose of trade allocation is to allocate or split a trade from one account to an another account. Allocation is performed by making an opposite trade on the original account and an equal trade on the receiving account.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	tradeId	required	long	Trade id.
3	fromAccountId	required	long	The account to allocate the Trade from
6	destinations	required	TradeDestination []	One or more destinations for this move operation.
7	moveId		Long	Must be unique for the referenced trade. Used to prevent duplicate move trade requests.
8	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.
9	tradingUserId	required	String	The trading user id.

This request will normally return a response of type [AllocateTradeRsp](#) .

Message: AllocateTradeRsp

Message ID: 10105

Type: Trade Management Messages

Description: Response to the AllocateTradeReq request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.

6	tradeIds		Long []	The IDs of the trades that were created as part of this move trade operation.
7	transactionId		long	The ID of the transaction that updated the positions affected by this move trade operation.

Message: ApproveGiveUpReq

Message ID: 10130

Type: Trade Management Messages

Description: Request to approve give-up request. The give-up request has been published to the member using GiveUpEvent.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	fourEyesId	required	long	The ID of the Assign or Tripartite.
3	initiator	required	String	The initiator member Id.
4	acknowledger	required	String	The destination member Id.
5	client		String	The destination client Id.
8	destinations	required	TradeDestination []	One or more destinations for this move operation.
11	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.
12	tradingUserId	required	String	The trading user id.

This request will normally return a response of type [ResponseMessage](#).

Message: AssignTradeReq

Message ID: 10114

Type: Trade Management Messages

Description: Request to assign trade to another member.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	tradeId	required	long	Trade id.
3	fromAccountId	required	long	The current account for the Trade.
6	reference		String	An optional free text field.
8	quantity	required	Long	The trade quantity. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
9	acknowledger	required	String	The destination member Id.
10	moveId		Long	Must be unique for the referenced trade. Used to prevent duplicate move requests.
14	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.
15	tradingUserId	required	String	The trading user id.
16	commissionAmount		Long	A sight of the commission amount before the actual commission is submitted. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

This request will normally return a response of type [ResponseMessage](#) .

Message: CancelCommissionReq

Message ID: 10516

Type: Trade Management Messages

Description: Request for TM to cancel a commission that has been sent earlier the same day.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	commissionId	required	String	Id of the Commission.
3	cancellationReference		String	Optional reference that can be supplied at cancellation or rejection requests.

This request will normally return a response of type [ResponseMessage](#) .

Message: CancelGiveUpReq

Message ID: 10128

Type: Trade Management Messages

Description: Request to cancel giveup request by the initiator. The recipient of the GiveUpEvent will then be notified with a new GiveUpEvent where the state is set to CANCELLED.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	fourEyesId	required	long	The ID of the Assign or Tripartite.
3	initiator	required	String	The initiator member Id.
4	acknowledger	required	String	The destination member Id.
5	reason		String	An optional free text field.
7	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.

This request will normally return a response of type [ResponseMessage](#) .

Message: CorrectAllocationErrorReq

Message ID: 10108

Type: Trade Management Messages

Description: To correct when a trade has erroneously been allocated to wrong client, i.e. to move the trade from one client to another.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	tradeId	required	long	Trade id.
3	fromAccountId	required	long	The current account for the Trade.

5	reference		String	An optional free text field.
6	toAccountId	required	long	The new account for the Trade.
7	moveId		Long	Must be unique for the referenced trade. Used to prevent duplicate move trade requests.
8	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.
9	tradingUserId	required	String	The trading user id.

This request will normally return a response of type [CorrectAllocationErrorRsp](#) .

Message: CorrectAllocationErrorRsp

Message ID: 10109

Type: Trade Management Messages

Description: Response to the AllocateTradeReq request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	tradeIds		Long []	The IDs of the trades that were created as part of this move trade operation.
7	transactionId		long	The ID of the transaction that updated the positions affected by this move trade operation.

Message: CorrectPrincipalReq

Message ID: 10110

Type: Trade Management Messages

Description: To move a trade from a client account to a member main or sub account.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	tradeId	required	long	Trade id.
3	fromAccountId	required	long	The current account for the Trade.
5	reference		String	An optional free text field.
6	toAccountId	required	long	The new account for the Trade.
7	moveId		Long	Must be unique for the referenced trade. Used to prevent duplicate move trade requests.
8	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.
9	tradingUserId	required	String	The trading user id.

This request will normally return a response of type [CorrectPrincipalRsp](#) .

Message: CorrectPrincipalRsp

Message ID: 10111

Type: Trade Management Messages

Description: Response to the CorrectPrincipalReq request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	tradeIds		Long []	The IDs of the trades that were created as part of this move trade operation.
7	transactionId		long	The ID of the transaction that updated the positions affected by this move trade operation.

Message: ExerciseOptionPositionReq

Message ID: 10186

Type: Trade Management Messages

Description: Exercise an option position. For American style options, this is allowed at any time during the contract's lifetime. For European style options, this can only be done on the expiry day.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	accountId	required	long	The account of the position.
3	quantity	required	Long	The exercise quantity. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
5	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.
6	updateId	required	String	Unique id that should be set for the request. If the system has already processed an update with this id, a success message is returned without any position change.

This request will normally return a response of type [ExerciseOptionPositionRsp](#) .

Message: ExerciseOptionPositionRsp

Message ID: 10187

Type: Trade Management Messages

Description: Response to the ExerciseOptionPositionReq request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.

2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	tradeIds		Long []	The IDs of the trades that were created as part of this exercise of an option position.
7	transactionId		long	The ID of the transaction that updated the positions affected by this exercise of an option position.

Message: ModifyPositionSubAccountReq

Message ID: 10112

Type: Trade Management Messages

Description: Request to move a position from a house main/house sub/house suspense/client suspense account to a house main or house sub account, or move a position from branch main/branch sub/branch clients suspense account to a branch main or branch sub account.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	fromAccountId	required	long	The current account for the position.
3	quantity	required	Long	The quantity to move. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
5	reference		String	An optional free text field.
6	toAccountId	required	long	The new account for the position.
7	moveId		Long	Must be unique for the referenced trade. Used to prevent duplicate move requests.
9	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.
10	tradingUserId	required	String	The trading user id.
11	price		BigInteger	Optional price to be used for the created trades. If no price, EoD settlement price from the previous business day for the instrument is used. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

This request will normally return a response of type [ModifyPositionSubAccountRsp](#).

Message: ModifyPositionSubAccountRsp

Message ID: 10113

Type: Trade Management Messages

Description: Response to the ModifyPositionSubAccountReq request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.

3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	tradeIds		Long []	The IDs of the trades that were created as part of this move trade operation.
7	transactionId		long	The ID of the transaction that updated the positions affected by this move trade operation.

Message: ModifyTradeSubAccountReq

Message ID: 10148

Type: Trade Management Messages

Description: To move a trade from house account to other house accounts.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	tradeId	required	long	Trade id.
3	moveId		Long	Must be unique for the referenced trade. Used to prevent duplicate move trade requests.
4	fromAccountId	required	long	The account to allocate the Trade from
7	destinations	required	TradeDestination []	One or more destinations for this move operation.
8	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.
9	tradingUserId	required	String	The trading user id.

This request will normally return a response of type [ModifyTradeSubAccountRsp](#) .

Message: ModifyTradeSubAccountRsp

Message ID: 10149

Type: Trade Management Messages

Description: Response to the ModifyTradeSubAccountReq request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	tradeIds		Long []	The IDs of the trades that were created as part of this move trade operation.
7	transactionId		long	The ID of the transaction that updated the positions affected by this move trade operation.

Message: QueryTradesReq

Message ID: 10258

Type: Trade Management Messages

Description: Query trades from previous days. If the flag hasMore is set in the response, there are too many trades matching the search criteria. The client needs to specify narrower criteria and submit the query again.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	tradeBusinessDateFrom	required	String	Query trades from date. Can be current or previous business day. The format is yyyy-MM-dd.
3	tradeBusinessDateTo		String	Query trades to date. The format is yyyy-MM-dd.
4	tradeTimeFrom		String	Query trades from time. The format is yyyy-MM-ddTHH:mm:ss.SSS.
5	tradeTimeTo		String	Query trades to time. The format is yyyy-MM-ddTHH:mm:ss.SSS.
6	clearingMemberId		String	Query trades with Clearing Member. If requested by a CM user then mandatory, needs to be set to the same CM as the logged in user.
7	tradingMember		String	Query trades with Trading Member. If requested by TM user then mandatory, needs to be set to the same TM as the TM of the logged in user.
8	tradingMemberBranch		String	Query trades with branch. If requested by Branch user then mandatory, needs to be set to the same Branch as the Branch of the logged in user.
9	clientId		String	Query trades with client.
10	internalAccountId		Long	Query trades with account. If Client specified: need to be an account of the client. If Branch but not Client specified: needs to be a house account of the Branch. If TM but not Branch or Client specified: needs to be a house account of the TM.
11	tradingUser		String	Query trades with Trading User.
12	internalTradableInstrumentId		Long	Query trades with Tradable Instrument.
13	alphaCode		String	Query trades with Alpha Code.
14	previousTradeIds		String	Query trades with Previous Trade Ids.
15	nextTradeIds		String	Query trades with Next Trade Ids.
16	tradingSystemMatchId		String	Query trades with Trade Id.
17	tradingSystemTradeLinkId		String	Query trades with Trade Link Id.
18	dealId		Long	Query trades with internal Deal Id.
19	tradeId		Long	Query trades with internal Trade Id.
20	externalInstrumentId		String	Query trades with Tradable Instrument. This is the JSE Master ID.

This request will normally return a response of type [QueryTradesRsp](#) .

Message: QueryTradesRsp

Message ID: 10259

Type: Trade Management Messages

Description: Query trades response.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
-----------	------------------	-------	-------------------	---------

1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	trades		TradeRes []	Trades
7	hasMore		boolean	A flag indicating whether or not the response was truncated by the server

Message: RejectCommissionReq

Message ID: 10517

Type: Trade Management Messages

Description: Request for the Destination TM to reject a received commission.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	commissionId	required	String	Id of the Commission.
3	cancellationReference		String	Optional reference that can be supplied at cancellation or rejection requests.

This request will normally return a response of type [ResponseMessage](#) .

Message: RejectGiveUpReq

Message ID: 10132

Type: Trade Management Messages

Description: Reject assigned or tripartite trade as receiver. The initiator will be notified by a GiveUpEvent on the GiveUp Event Flow.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	fourEyesId	required	long	The ID of the Assign or Tripartite.
3	initiator	required	String	The initiator member Id.
4	acknowledger	required	String	The destination member Id.
5	reason		String	An optional free text field.
7	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.

This request will normally return a response of type [ResponseMessage](#) .

Message: TripartiteAllocationReq

Message ID: 10134

Type: Trade Management Messages

Description: Tripartite allocation to another member. Tripartite agreement must exist. Tripartite requests that are not handled during the day are removed by the system.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	tradeId	required	long	Trade id.
3	fromAccountId	required	long	The current account for the Trade.
5	quantity	required	Long	The trade quantity.
				This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
6	reference		String	An optional free text field.
7	acknowledger	required	String	The destination member Id.
8	client	required	String	The destination client Id.
11	moveId		Long	Must be unique for the referenced trade. Used to prevent duplicate move requests.
14	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.
15	tradingUserId	required	String	The trading user id.
16	commissionAmount		Long	A sight of the commission amount before the actual commission is submitted. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

This request will normally return a response of type [TripartiteAllocationRsp](#) .

Message: TripartiteAllocationRsp

Message ID: 10135

Type: Trade Management Messages

Description: Response to the TripartiteAllocationReq request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.

Message: UpdateTradeReferenceReq

Message ID: 10544

Type: Trade Management Messages

Description: Request to update the reference on trade.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	tradeId	required	long	Trade id.
3	externalInstrumentId	required	String	The external instrument id. This is the JSE Master ID.
4	accountId	required	long	The account of the trade to update.
5	reference		String	The new reference for the trade.

This request will normally return a response of type [UpdateTradeReferenceRsp](#)

Message: UpdateTradeReferenceRsp

Message ID: 10545

Type: Trade Management Messages

Description: Response to the UpdateTradeReferenceReq request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.

Trade Management Messages (Internal)

Message: TradeDestination

Message ID: 10018

Type: Trade Management Messages

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: This object describes on which account a trade should be booked and at what quantity.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	accountId	required	long	The account to book a Trade to.
2	quantity	required	BigInteger	The quantity to book in the account. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
3	reference		String	An optional free text field.

Message: TradeRes

Message ID: 10260

Type: Trade Management Messages

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: Trade resulted from query in history.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	tradeId		long	Internal identifier for a trade created by the clearing system. Updates of a trade (if supported) should keep the same TradeId. Unique over time.
2	origTradeId		Long	Used to preserve the original trade id when original trade is being referenced in a subsequent trade transaction such as a transfer. For example when moving a trade this refers to the previous trade id. It works for moved/allocated trades as long as the new trade only consist of quantity from one trade. It does not work for aggregations since those trades comes from multiple trades. In case a trade is moved in multiple steps origTradeId points to the previous trade and not the initial trade.
3	initialTradeId		Long	If there has been multiple moves this points to the initial trade.
8	businessDate		String	Business date of the transaction according. The format is yyyy-MM-dd.
9	tradeDate		String	The trade date of the trade. The format is yyyy-MM-dd.
10	dealId		Long	Internal identifier for linking the trade to a deal.
11	clientDealId		String	Reference to the deal id specified by the client.
12	clOrdId		String	A optional reference set by the trading member to backtrack the trade to an order at the trading venue
14	originalQuantity		BigInteger	The original quantity on the trade, differs from LastQty if quantity has been moved from/to the trade after entering it This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
15	activeQuantity		BigInteger	The current active quantity of the trade This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
16	reservedQuantity		BigInteger	The current reserved quantity of the trade This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
17	remainingQuantity		BigInteger	The current remaining quantity of the trade This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
18	price		BigInteger	Price of the trade. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
19	originalPrice		BigInteger	Price of the original trade. Needed for audit since price might be modified in trade management. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

20	currency		String	The currency of LastPx, either add it to the trade or take it from the instrument.
21	previousTradeIds		String	Comma separated list of trade ids.
22	nextTradeIds		String	Comma separated list of trade ids.
23	text		String	Optional free text field from the reference of the original trade.
24	initialValue		BigInteger	Initial value of the trade (price*qty*contract size) This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
25	tradeTimestamp		String	Trade time according to the clearing system. The format is yyyy-MM-ddTHH:mm:ss.SSS.
28	lastMkt		String	Market of the trade. Internally set in case of trade management operations.
29	positionReason		Integer	Contains all different types of event reasons also for all trade management reasons. Can be used for billing and surveillance purpose. Allowed values: see constant group PositionReason
30	isBuy		boolean	Side of trade (buy/sell)
31	trader		String	ID of the trader of this trade. For new trades caused by deal management it should be the id of the user doing the operation. In case of updating the trade (moving qty for example) the trader should remain as the original trader, i.e. this attribute should never be updated once created. The user from the position will state who caused the event, i.e. the user triggering the action.
33	underlyingSymbol		String	Human readable representation of the underlying instrument
34	contractSize		BigInteger	Size of contract This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
35	instrumentType		String	Type of instrument according to system
36	instrumentCurrency		String	Currency of the instrument.
37	marketList		String	Name of the market list
38	marketSegment		String	Name of the market segment
39	market		String	Name of the market
40	accountId		long	Unique identifier of the account for trade.
42	accountType		String	The type of account for the trade. References name to constant PositionAccountType.
43	accountSubType		String	The sub type of account for the trade. References name to constant PositionAccountSubType.
44	clearingMember		String	The clearing member for this trade
45	tradingMember		String	The member who owns this trade
46	tmBranch		String	Trading member branch if applicable
47	client		String	Client ID if applicable
48	callPut		String	If this is an option, it represents its type (call or put)
49	strike		BigInteger	The strike price of the instrument if it is an option. Null otherwise This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
50	expirationDate		String	The expiry date of the instrument
51	alphaCode		String	Common identifier code for derivative instruments

52	tsTradingMatchId		String	Trading Id from the Trading System
53	tsTradingLinkId		String	Trading Link Id from the Trading System
54	clientType		String	For clients only - type of client. Information to surveillance. Allowed values: see constant group ClientType
55	idNumber		Long	For clients only - ID number.
56	passportNumber		String	For clients only - Passport number.
57	companyRegistrationNumber		String	For clients only - Company registration number.
58	isProfessional		Boolean	For clients only - Information to surveillance.
59	isShariah		Boolean	For clients only - Information to surveillance.
60	externalInstrumentId		String	The external instrument id. This is the JSE Master ID.
61	timeOfEntry		String	Trade time according to the Trading system. The format is yyyy-MM-ddTHH:mm:ss.SSS.
62	clientName		String	Client Name if applicable
63	tsTradingHalfId		String	Trading Half Id from the Trading System
64	isStaff		Boolean	For clients only - Information to surveillance.
65	isBeneficial		Boolean	For clients only - Information to surveillance.
66	isDiscretionary		Boolean	For clients only - Information to surveillance.
67	aggressor		Boolean	Aggressor from Trading System.
68	capacity		Integer	Capacity from Trading System. Allowed values: see constant group Capacity
69	onBookQuantity		Long	The On Book Quantity. One unit of the currency is expressed by DIVISOR.QTY. This field represents a decimal value. The value of the field is the decimal value multiplied by the constant DIVISOR.QTY. Example: The value "12.50" is represented as 12500000 in this field. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
70	offBookQuantity		Long	The Off Book Quantity. One unit of the currency is expressed by DIVISOR.QTY. This field represents a decimal value. The value of the field is the decimal value multiplied by the constant DIVISOR.QTY. Example: The value "12.50" is represented as 12500000 in this field. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
71	tradeType		String	Trade Type from Trading System. Allowed values: see constant group RtcTradeType
72	zeroFeeFlag		boolean	Flag to indicate that the trade is marked for zero fee.
73	interestRateSpread		Long	Rate added to base rate on a CFD to get the Funding rate. Numeric, positive or negative. The value "2.0%" is represented as 2000000 in this field. Mandatory if the type of the instrument is CFD. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
74	isin		String	ISIN code
75	shortName		String	The user friendly name the tradable instrument. Is not unique.

76	instrumentSubType		String	JSE Instrument Type.
77	inwardListed		boolean	Inward Listed according to the South African Reserve Bank.
78	baseRate		String	The base rate name for a CFD.
79	tradingUser		String	ID of the trader of this trade. Carried from the trading system. Also known as dealer. For new trades caused by deal management it should be the id of the user doing the operation. In case of updating the trade (moving qty for example) the trader should remain as the original trader, i.e. this attribute should never be updated once created. The user from the position will state who caused the event, i.e. the user triggering the action.
80	externalPositionAccount		String	External ID of the position account. CM/TM link table is used to find the right risk tree.
81	participantUnitId		String	The owner of the account in the member structure. This is the ID of the lowest applicable level in the TM/Branch/Client member structure. Contains the Client ID for client accounts.
82	vatRegistrationNumber		String	The VAT Registration number. If the same legal Client is using several TMs, each TM will manage their own instance of the Client. The different instances of the Client in the system will then share the same VAT Registration Number.
83	isNonResident		boolean	Indication of the Client is resident of South Africa or not. If Country Code is ZA, the Client must be Resident. If Country Code is not ZA, the Client must be Non Resident.
84	address		String	Address of the client.
85	country		String	Registered Country for client.
86	bdaCode		Integer	Broker Dealer Accounting system (BDA) account number.
89	assetClass		String	Asset Class Allowed values: see constant group AssetClass
90	assetSubClass		String	Asset Sub Class Allowed values: see constant group AssetSubClass
91	reference		String	Reference field that can be used by user during trade management activities
92	agreedTime		String	Time agreed between TM1 and TM2. Only applicable for reported trades. The format is yyyyMMdd-HH:mm:ss.fff.
93	reportedTime		String	The time the reported trade was received on the trading system (system generated). yyyyMMdd-HH:mm:ss.fff.
94	fromTradeId		Long	Trade ID of original trade in case this trade is created from a trade management activity.
95	fromTradeTime		String	The time new trades are created or the time a trade management activity is accepted.
96	fromRemainingQuantity		BigInteger	The quantity moved to this trade This field is a fixed point number with a scaling factor equal to 1/DIVISOR.QTY.
97	fromTM		String	The TM of the From trade. Will be different from the TM for assigns and tripartite.
98	fromBranch		String	The Branch of the From trade.
99	fromCM		String	The CM of the From trade.

100	fromAccount		Long	The account ID of the From trade
101	fromAccountType		String	Account Type of the From trade. References name to constant PositionAccountType.
102	fromAccountSubType		String	Account Sub Type of the From trade. References name to constant PositionAccountSubType.
103	fromAccountOwner		String	The owner of the account in the member structure. This is the ID of the lowest applicable level in the TM/Branch/Client member structure. Contains the Client ID for client accounts.
104	fromClientName		String	The name of the to client for the From trade
105	fromClientType		String	Type of client. Used mainly for surveillance when creating reports. Allowed values: see constant group ClientType
107	fromIsStaff		boolean	True for Staff clients, for the From Trade.
108	fromIsBeneficial		boolean	True for Beneficial Account clients, for the From Trade.
109	fromIsProfessional		boolean	True for professional clients, for the From Trade.
110	fromIsShariah		boolean	True for Shariah clients, for the From Trade.
111	fromIsNonResident		boolean	Indication of the Client is resident of South Africa or not. If Country Code is ZA, the Client must be Resident. If Country Code is not ZA, the Client must be Non Resident, for the From Trade.
112	fromBdaCode		Integer	Broker Dealer Accounting system (BDA) account number, for the From trade
113	fromIsDiscretionary		boolean	True for Discretionary clients, for the From Trade.
114	clearingMemberName		String	The long name of the CM.
115	tradingMemberName		String	The long name of the TM.
116	branchName		String	The long name of the Branch.
117	clientPhone		String	The phone number of the Client
118	fromExternalPositionAccount		String	External ID of the position account of the From trade.
119	fromTradingUser		String	Trading user of the From trade
120	fromPositionReason		Integer	Position reason of the From trade Allowed values: see constant group PositionReason
121	nominatedMemberId		String	Member handling physical delivery
122	clientNominatedMemberId		String	Member handling physical delivery for the client.
123	optionDelta		BigInteger	Option delta from the trading system. Valid for Options. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
124	firmTradeId		String	Free text field carried from the trading system for off book trades.
125	tradeReportId		String	Free text field carried from the trading system for on and off book trades.
127	eventLinkId		String	Unique key for all trades generated from a single event. Will only be populated for strategy trades.
128	tradeSubType	required	String	Trade sub type used for trades.

External Members

Message: GetRiskArrayReq

Message ID: 10270

Type: External Members

Description: Query JSPAN risk arrays available in the system.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
3	allInstruments		boolean	If true, the values in externalInstrumentIds will be ignored. Risk arrays will be returned for all instruments with risk arrays, but startInstrumentOffset and maxNumberOfInstrumentsReturned are used for paging the response.
4	externalInstrumentIds		String []	List of instruments for which risk array will be retrieved. JSE Master IDs.
5	startInstrumentOffset	required	Integer	Offset of first instrument to include. Starts on zero. This field is used to page the response to avoid overflow.
6	maxNumberOfInstrumentsReturned	required	Integer	Maximum number of rows in response. Must be 500 or smaller. This field is used to page the response to avoid overflow.

This request will normally return a response of type [GetRiskArrayRsp](#) .

Message: GetRiskArrayRsp

Message ID: 10271

Type: External Members

Description: Response to a GetRiskArrayReq request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	contracts		Contract []	Risk arrays for all contracts.

External Members (Internal)

Message: Contract

Message ID: 10272

Type: External Members

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: Risk array information for one contract. Sub-object in GetRiskArrayRsp.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	businessDate		String	Business date. Format is YYYY-MM-DD.
3	expiryDate		String	Expiry date of the contract. Format is YYYY-MM-DD.
4	mtmPrice		Long	Mark-to-Market price. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
5	strikePrice		Long	Strike price of option contracts. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
6	volatility		Long	Volatility of the contract. MTM volatility for options. ATM volatility for futures that has options. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
7	riskArray		Long []	Risk array of the contract.
8	externalInstrumentId		String	The external instrument id. This is the JSE Master ID.
9	timeStamp		String	Risk array generation time. The format is "yyyy-MM-ddTHH:mm:ss.SSS".
10	condType		String	Price condition type. Allowed values: see constant group CondType

Settlement Messages

Message: ConfirmWithdrawalsReq

Message ID: 10487

Type: Settlement Messages

Description: Used for a Clearing Member to confirm or reject payment advices.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	confirmed	required	Boolean	True if the Payment Advice was confirmed by the CM, false if it was rejected by the CM.
3	withdrawals	required	PaymentAdvice []	List of withdrawals that are confirmed or rejected by CM.

This request will normally return a response of type [ResponseMessage](#).

Message: GetPaymentAdvicesReq

Message ID: 10491

Type: Settlement Messages

Description: Get payment advices for a particular clearing member.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	bookmark		String	The bookmark marks a specific item in a list of data. The bookmark received in the response should be used in next request to get next page of information.
3	clearingMember		String	The clearing member ID.
4	settlementDate		String	The settlement date. The format is yyyy-MM-dd.
5	pageSize		Integer	The preferred page size, this means max number of items in the response. If not set, the default pagesize is used.

This request will normally return a response of type [GetPaymentAdvicesRsp](#) .

Message: GetPaymentAdvicesRsp

Message ID: 10492

Type: Settlement Messages

Description: Response to the GetPaymentAdvicesReq request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	bookmark		String	The bookmark marks a specific item in a list of data on the server. The bookmark received in the response should be used in next request to get next page of information.
7	paymentAdvices		PaymentAdvice []	The payment advices.

Message: GetSettlementInstructionsReq

Message ID: 10301

Type: Settlement Messages

Description: Request to get settlement instructions.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	clearingMember		String	The requested clearing member ID.
3	settlementDate	required	String	The requested settlement date. The format is yyyy-MM-dd.
4	settlementRunId		Long	Settlement run id.
5	settlementInstructionState		String	Instruction state. Allowed values: see constant group SettlementInstructionState

6	bookmark		String	The bookmark marks a specific item in a list of data. The bookmark received in the response should be used in next request to get next page of information.
---	----------	--	--------	---

This request will normally return a response of type [GetSettlementInstructionsRsp](#).

Message: GetSettlementInstructionsRsp

Message ID: 10302

Type: Settlement Messages

Description: Response to GetSettlementInstructionsReq.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	instructions		SettlementInstruction []	Settlement instructions.
7	bookmark		String	The bookmark marks a specific item in a list of data on the server. The bookmark received in the response should be used in next request to get next page of information.

Message: PaymentAdvice

Message ID: 10488

Type: Settlement Messages

Description: An indication that the Clearing Member will deposit additional cash collateral.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	strateReferenceNo	required	String	The payment reference generated by the CSD.
2	strateCode	required	String	Strate code for the Client or Trading Member.
3	riskNodeID	required	long	Risk node ID, RTC internal ID.
4	amount	required	Long	The amount in ZAR that must be called for in cash collateral. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
5	currency	required	String	The currency code for the amount field, according to ISO 4217. Normally ZAR.
6	settlementAmount		Long	The amount in 'settlementCurrency' that must be called for in cash collateral. If the settlementCurrency is not ZAR, this amount has been calculated by RTC using the most recent exchange rate. This field is set by RTC.

				This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
7	settlementCurrency		String	The currency used to settle this withdrawal. Is the preferred currency for the member/client or ZAR if preferred currency is not set. This field is set by RTC.
8	clearingMember		String	Clearing Member ID. Set on outbound messages.
9	paymentAdviceState		int	The payment advice state Allowed values: see constant group PaymentAdviceState
10	senderRef		String	Unique Id generated by the clearing system.
11	participantUnitId		String	The owner (member or client) of the risk node. Set on outgoing messages from RTC.

Settlement Messages (Internal)

Message: SettlementInstruction

Message ID: 10303

Type: Settlement Messages

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: Settlement instructions suitable to pass on to the settlement systems. The settlement instructions in RTC will be in state PENDING until settlement has been confirmed.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	instructionId		long	System generated unique ID.
2	settlementDate		String	Settlement date. The format is yyyy-MM-dd.
3	referenceNo		String	Message reference no. The reference is built out of 3 components 1. 76 (CM receives payment from JSE) or 77 (CM payments, JSE receives) 2. CM template (from member) 3. Settlement date E.g. 760210515, Merrill Lynch receives payment on May 21, 2015
4	sendCode		String	JSE BIC.
5	minusAccount		String	The receiving member account where the position effect should be booked due to this instruction.
6	senderBIC		String	BIC code of the settlement bank of the paying member. Note that this can be the BIC of the settlement bank of JSE.
7	senderBranch		String	Branch no of the sender (used for SWIFT).
8	senderAccountId		String	Account number that the receiving member has in the settlement bank. Note that this can be the account of the clearing house.
9	externalFromAccount		String	The external account from which the amount should be moved.
10	plusAccount		String	The sending member account where the position effect should be booked due to this instruction.
11	receiverBIC		String	BIC code of the settlement bank of the receiving member. Note that this can be the BIC of the settlement bank of JSE.
12	receiverBranch		String	Branch no of the receiver (used for swift).

13	receiverAccountId		String	Account number that the receiving member has in the settlement bank. Note that this can be the account of the clearing house.
14	externalToAccount		String	The external account to which the amount should be moved.
15	amount		long	The amount to move. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
16	unconfirmedSettledAmount		long	The sum of reported settled amounts that are waiting for confirmation on a position update. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
17	settledAmount		long	The sum of reported settled amounts that have been matched to this instruction. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
18	currencyId		String	Currency in which the amount is settled. The currency code according to ISO 4217.
19	settlementInstructionState		String	Instruction state. Allowed values: see constant group SettlementInstructionState
20	settlementRunId		long	Refers to the parent settlement run.

External Members

Message: GetRequestsForFXCollateralReq

Message ID: 10384

Type: External Members

Description: Request for the clearing member to get the information about the amounts per client/house that can be covered with FX collateral. Next step is for the clearing member to send in a confirmation with the different amounts in FX with RegisterFXCollateralReq.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	valueDate	required	String	The requested value date.
3	clearingMemberId		String	Request FX Collateral for a CM.
4	bookmark		String	The bookmark from a paged response.

This request will normally return a response of type [GetRequestsForFXCollateralRsp](#).

Message: GetRequestsForFXCollateralRsp

Message ID: 10385

Type: External Members

Description: Response to GetRequestsForFXCollateralReq request. Includes values in ZAR that could be covered with FX, per client/house.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	fxRequests		RequestForFXCollateral []	The requests for FX Collateral
7	bookmark		String	Bookmark to use in query for next batch.

Message: QueryDividendPaymentFactorsReq

Message ID: 10527

Type: External Members

Description: Query the factors used in the calculation of dividend payments.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	exDate	required	String	The ex-date to get dividend payment factors for.
3	alphaCode		String	Alpha code for the instrument where dividend is paid. If blank, all dividend factors for the ex-date will be returned.

This request will normally return a response of type [QueryDividendPaymentFactorsRsp](#) .

Message: QueryDividendPaymentFactorsRsp

Message ID: 10528

Type: External Members

Description: Response message for the QueryDividendPaymentFactorsReq

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	dividendFactors		DividendFactor []	An array of the dividend factors matching the search criteria.

Message: RegisterFXCollateralReq

Message ID: 10386

Type: External Members

Description: For clearing member to register FX Collateral.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	valueDate	required	String	The requested value date. The format is yyyy-MM-dd.
3	clientId		String	The client ID.
4	tradingMemberId	required	String	The ID of the Trading Member.
5	clearingMemberId	required	String	The ID of the Clearing Member.
6	fxCollateral		FXCollateral []	The received currency collateral

This request will normally return a response of type [RegisterFXCollateralRsp](#) .

Message: RegisterFXCollateralRsp

Message ID: 10387

Type: External Members

Description: Response to RegisterFXCollateralReq request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.
6	fxCollateralStatus		FXCollateralStatus []	The status of registered FX collateral
7	statusText	required	String	Description of the execution status.

Message: RequestForFXCollateral

Message ID: 10383

Type: External Members

Description: Amount in ZAR per entity that can be covered with FX collateral. Returned as an array in GetRequestsForFXCollateralRsp.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	valueDate	required	String	Value date , format YYYY-MM-DD.
2	clientId	required	String	The client ID / TM House.
3	tradingMemberId	required	String	The ID of the Trading Member.
4	clearingMemberId	required	String	The ID of the Clearing Member.
5	amount	required	long	Amount in ZAR that can be covered with FX. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
6	ccy	required	String	Valued Currency.

Message: SetCmBalancingStatusReq

Message ID: 10420

Type: External Members

Description: Request to set balance status for a CM.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	step	required	String	The balancing step to set the status for. Allowed values: see constant group CmBalancingStep
3	clearingMember	required	String	ID of the clearing member.
4	balanced		boolean	True if the CM is balanced, otherwise false.

This request will normally return a response of type [SetCmBalancingStatusRsp](#) .

Message: SetCmBalancingStatusRsp

Message ID: 10421

Type: External Members

Description: Response to a SetCmBalancingStatusReq request.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	code		int	Status code. Code 3001 indicates that the request was processed successfully. For other codes, see the Status Code list in the EMAPI HTML description.
2	message		String	A textual description of the status code above.
3	subCode		int []	Status code for each leg of the request. Only used for batched requests.

External Members (Internal)

Message: DividendFactor

Message ID: 10529

Type: External Members

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: The dividend factors used for calculating settlement amounts for dividends.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	exDate		String	The ex-date for the dividend.
2	ldtDate		String	The LDT date for the dividend.

3	settlementDate	String	The settlement date for the dividend payment.
4	externalInstrumentId	String	The external instrument id. This is the JSE Master ID.
5	alphaCode	String	Common identifier code for derivative instruments
6	presentValue	Long	The present value factor (DivPV) for this dividend. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
7	forwardValue	Long	The forward value factor (DivFV) for this dividend. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
8	dividendAmount	Long	The dividend amount. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
9	currency	String	Currency of the dividend amount.
10	timestamp	String	Time stamp when the dividend calculation was started.

Message: FXCollateral

Message ID: 10388

Type: External Members

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: FX Collateral input.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
2	allowedFXAmount	required	long	Amount of allowed FX in ZAR. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
3	availableFXAmount	required	long	Amount of available Collateral FX Currency. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
4	currency	required	String	Available Currency. The alphabetic currency code according to ISO 4217.
5	valuationPrice	required	long	Valuation price of FX Currency. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

Message: FXCollateralStatus

Message ID: 10389

Type: External Members

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: Status of requested registration of FX collateral.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	postedQtyFx	required	long	The resulting posted collateral quantity per FX. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
2	postedValueFx	required	long	The resulting posted collateral value in ZAR per FX, after haircut. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
3	currency	required	String	Available Currency. The alphabetic currency code according to ISO 4217.
4	statusText	required	String	Description of status of the transaction.

Message: FxInterestRate

Message ID: 10414

Type: External Members

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: Keeps information about interest rate for a currency

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	currencyId		String	ID of the currency. ISO 4217 alphabetic code.
2	interestRate		Long	Interest rate for the currency. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

Message: MemberBalance1

Message ID: 10412

Type: External Members

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: CM balance 1 information for the CM or one of its cleared TM members.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	memberId		String	ID of clearing member or trading member.

2	initialMargin		long	Total Initial margin for TM house and clients accumulated to the Trading member. Always a positive value. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
3	additionalMargin		Long	Total additional margin for TM house and clients accumulated to the Trading member. Always a positive value. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
4	variationMargin		long	Total Variation margin for TM house and clients. Negative if the total net is a loss. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
5	dividends		Long	Sum of all Dividends for dividend neutral contracts. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
6	fundingInterest		Long	Sum of Interest on CFD contracts. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

Message: MemberBalance2

Message ID: 10415

Type: External Members

This message can only appear as a sub-object in other messages; it can never be used as a stand-alone message.

Description: CM balance 2 information for the CM or one of its cleared TM members.

Field no.	Field name (tag)	Mand.	Type (max length)	Comment
1	memberId		String	ID of clearing member or trading member.
3	commissions		Long	Total commission for the specified member ID. A negative number means that the member will pay the amount for commissions. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
4	bookingFees		Long	Total net booking fee, including VAT, for the specified member ID. A negative number means that the member will pay the amount. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.
5	riskFees		Long	Total net Interest Consideration at 17bps, for the specified member ID. The interest consideration is exempt from VAT. A negative number means that the member will pay the amount. This field is a fixed point number with a scaling factor equal to 1/DIVISOR.PRICE.

Messages by ID

ID	Message name
63	TaxLogonReq
64	TaxLogonRsp
65	TaxLogoutReq
69	TaxSnapshotSubscribeReq
70	TaxSnapshotSubscribeRsp
71	TaxRemoveSubscriptionReq
72	TaxStartSnapshot
73	TaxEndSnapshot
75	TaxHeartbeatReq
76	TaxHeartbeatRsp
77	TaxSessionStatus
95	Currency
96	SubscriptionGroup
101	Member
126	ChangePasswordReq
139	CalendarDate
226	CdRequest
227	CdResponse
230	ResponseMessage
231	SimpleRsp
232	TaxReplayReq
233	TaxReplayRsp
234	TaxReplayStartEvent
235	TaxReplayEndEvent
236	ProteusRefDataMessage
237	RequestMessage
295	Instrument
296	TradableInstrument
299	Market
300	MarketList
302	Segment
316	Country
10015	Trade
10018	TradeDestination
10031	CdAddRtcMemberClientReq
10032	AccountPositionEvent
10033	RiskNodeEvent
10034	CdAddRtcPositionAccountReq
10035	CdAddRtcPositionAccountRsp
10045	PositionAccount
10046	RiskNode
10049	AggregateTradesReq
10050	AggregateTradesRsp
10051	AccessGroup
10072	CdEnableDisableRtcPositionAccountReq
10073	CdEnableDisableRtcPositionAccountRsp

10074	PriceEvent
10082	SettlementAccount
10085	RtcTradeExternalData
10093	CollateralAccount
10104	AllocateTradeReq
10105	AllocateTradeRsp
10108	CorrectAllocationErrorReq
10109	CorrectAllocationErrorRsp
10110	CorrectPrincipalReq
10111	CorrectPrincipalRsp
10112	ModifyPositionSubAccountReq
10113	ModifyPositionSubAccountRsp
10114	AssignTradeReq
10123	ClearingMemberLink
10124	GiveUpEvent
10127	CdAddRtcMemberClientClearingLinkReq
10128	CancelGiveUpReq
10130	ApproveGiveUpReq
10132	RejectGiveUpReq
10134	TripartiteAllocationReq
10135	TripartiteAllocationRsp
10141	AccountTradeEvent
10145	RtcCalendar
10146	TripartiteAgreement
10147	CdUpdateRtcMemberClientReq
10148	ModifyTradeSubAccountReq
10149	ModifyTradeSubAccountRsp
10152	CdEnableDisableRtcMemberClientReq
10158	ClassSpreadGroup
10159	SeriesSpreadGroup
10177	Curve
10178	Surface
10185	CurveConstituent
10186	ExerciseOptionPositionReq
10187	ExerciseOptionPositionRsp
10188	AbandonOptionPositionReq
10189	AbandonOptionPositionRsp
10214	Deposit
10215	InterestRateSwap
10216	ForwardRateAgreement
10227	CurveEvent
10228	SurfaceEvent
10229	DividendEvent
10256	EligibleSecurity
10258	QueryTradesReq
10259	QueryTradesRsp
10260	TradeRes
10264	CashAccount
10267	CdAddCashAccountReq
10268	CdUpdateCashAccountReq
10270	GetRiskArrayReq
10271	GetRiskArrayRsp

10272	Contract
10273	CdSetMinimumZARLimitReq
10276	OptionDataEvent
10277	AtmVolatilityEvent
10289	NotionalValue
10290	CollateralPositionValue
10293	CdSetTradingMemberRiskLimitReq
10294	CdSetClientRiskLimitReq
10295	DailyAccountSummaryDetailsEvent
10301	GetSettlementInstructionsReq
10302	GetSettlementInstructionsRsp
10303	SettlementInstruction
10329	CorporateAction
10350	AggregatedSummaryClearingMemberEvent
10351	AggregatedSummaryTradingMemberEvent
10352	AggregatedSummaryDetails
10355	EligibleCurrency
10376	FxDailyAccountSummaryDetails
10377	ReadyConfirmAvailableFXEvent
10383	RequestForFXCollateral
10384	GetRequestsForFXCollateralReq
10385	GetRequestsForFXCollateralRsp
10386	RegisterFXCollateralReq
10387	RegisterFXCollateralRsp
10388	FXCollateral
10389	FXCollateralStatus
10411	CmBalancing1Event
10412	MemberBalance1
10413	CmBalancing2Event
10414	FxInterestRate
10415	MemberBalance2
10420	SetCmBalancingStatusReq
10421	SetCmBalancingStatusRsp
10430	GetSequenceNumbersReq
10431	GetSequenceNumbersRsp
10437	YieldEvent
10440	CurrentSystemState
10442	CdSetTradingMemberAMPercentageReq
10443	CdSetClientAMPercentageReq
10474	CdAddRtcMemberClientRsp
10486	WithdrawalNotificationEvent
10487	ConfirmWithdrawalsReq
10488	PaymentAdvice
10491	GetPaymentAdvicesReq
10492	GetPaymentAdvicesRsp
10514	CommissionEvent
10515	AddCommissionReq
10516	CancelCommissionReq
10517	RejectCommissionReq

Constants

[AssetClass](#)
[AssetSubClass](#)
[AXIS_UNIT](#)
[BOOTSTRAPPING_METHOD](#)
[BroadcastFlows](#)
[BusinessDayConvention](#)
[CACHE_ACTION](#)
[Capacity](#)
[ClearingMemberLinkStatus](#)
[ClientType](#)
[CmBalancingStep](#)
[CommissionStatus](#)
[CompoundingConvention](#)
[CondType](#)
[ContractSizeType](#)
[CorporateActionStatus](#)
[CouponIndicator](#)
[DATE_TYPE](#)
[DAY_COUNT_CONVENTION](#)
[DIVISOR](#)
[ExerciseStyle](#)
[EXTRAPOLATION_METHOD](#)
[FeedSource](#)
[FourEyesState](#)
[InstrumentIdType](#)
[InstrumentSubType](#)
[InstrumentType](#)
[INTEREST_RATE_CONVENTION](#)
[INTERPOLATION_METHOD](#)
[LoginStatus](#)
[MemberType](#)
[OptionAllocationModelType](#)
[OptionStyle](#)
[ParticipantUnitType](#)
[PaymentAdviceState](#)
[PeriodType](#)
[PositionAccountSubType](#)
[PositionAccountType](#)
[PositionReason](#)
[PositionType](#)
[ReplayRequestType](#)
[RollsOnConvention](#)
[RtcState](#)
[RtcTradeType](#)
[SchedulerState](#)
[SessionStatus](#)
[SettlementInstructionState](#)
[SettlementType](#)
[SubscriptionRequestType](#)
[ValuationModelType](#)
[ValuationSubType](#)
[YieldType](#)

Constant group: AssetClass

Description: Describes how the asset class available in the system.

Constant name	Type	Value	Comment
EQUITY	String	"EQUITY"	Equity.
FX	String	"FX"	FX.
FIXED_INCOME	String	"FIXED_INCOME"	Fixed Income.

Constant group: AssetSubClass

Description: Describes how the asset sub class available in the system.

Constant name	Type	Value	Comment
LE	String	"LOCAL_EQUITY"	Local equity.
FE	String	"FOREIGN_EQUITY"	Foreign equity.
FX	String	"FX"	FX.
NI	String	"NOMINAL_INTEREST_RATE"	Nominal interest rate.

Constant group: AXIS_UNIT

Description: Specifies the unit of an axis.

Constant name	Type	Value	Comment
ABSOLUTE_DATE	String	"ABSOLUTE_DATE"	Absolute date.
FRACTION_OF_YEAR	String	"FRACTION_OF_YEAR"	Fraction of year.
YIELD_PERCENTAGE	String	"YIELD_PERCENTAGE"	Yield (%).
STRIKE	String	"STRIKE"	Strike.
MONEYNESS	String	"MONEYNESS"	Moneyiness.
VOLATILITY	String	"VOLATILITY"	Volatility.

Constant group: BOOTSTRAPPING_METHOD

Description: Method used for bootstrapping.

Constant name	Type	Value	Comment
SWAP	String	"Swap"	Swap.
BOND	String	"Bond"	Bond.
INFLATION_LINKED_BOND	String	"Inflation linked Bond"	Inflation linked Bond.

Constant group: BroadcastFlows

Description: Defines broadcast flows

Constant name	Type	Value	Comment
PUBLIC_GLOBAL_REFERENCE_DATA_FLOW	int	11	Global reference data flow.
ACCOUNT_EVENT_FLOW	int	301	Account event flow.
RISK_EVENT_FLOW	int	302	Risk event flow.
MARKETDATA_EVENT_FLOW	int	303	Market Data event flow.
GIVEUP_EVENT_FLOW	int	304	GiveUp event flow.
SETTLEMENT_EVENT_FLOW	int	305	Settlement event flow.

Constant group: BusinessDayConvention

Description: Business Day convention.

Constant name	Type	Value	Comment
NONE	String	"NONE"	None.
FOLL_GOOD	String	"FOLL_GOOD"	Following Good.
MOD_FOLL	String	"MOD_FOLL"	Modified Following.

Constant group: CACHE_ACTION

Description: Defined cache actions

Constant name	Type	Value	Comment
ADD	int	1	Add to cache EMAPI - interpret as Add
UPDATE	int	2	Update cache EMAPI - interpret as Update
BOOTLOAD	int	3	Add to cache with bootloader EMAPI - interpret as Add
REMOVE_CACHE_DB	int	4	Remove from cache and db, does not remove if there are references to object. Return status code ValidationHasReference if referenced. EMAPI - interpret as Remove
REMOVE_CACHE_DB_FORCED	int	5	Remove from cache and db, removes even if there are references to object. EMAPI - interpret as Remove
REMOVE_CACHE	int	6	Remove from cache (does not remove object from db), does not remove if there are references to object. Return status code ValidationHasReference if referenced. The isDeleted attribute is set to BOOLEAN.TRUE EMAPI - interpret as Remove

REMOVE_CACHE_FORCED	int	7	Remove from cache (does not remove object from db), removes even if there are references to object. The isDeleted attribute is set to BOOLEAN.TRUE EMAPI - interpret as Remove
---------------------	-----	---	--

Constant group: Capacity

Description: Capacity on trade (from Trading System).

Constant name	Type	Value	Comment
Principal	Integer	1	Principal.
Agent	Integer	2	Agent.

Constant group: ClientType

Description: Type of client, information to surveillance.

Constant name	Type	Value	Comment
INDIVIDUAL	String	"INDIVIDUAL"	Individual.
COMPANY	String	"COMPANY"	Company.
HEDGE_FUND	String	"HEDGE_FUND"	Hedge fund.
STATE_ENTERPRISE	String	"STATE_ENTERPRISE"	State enterprise.
TRUST	String	"TRUST"	Trust.
CLOSED_CORPORATION	String	"CLOSED_CORPORATION"	Closed corporation.
ASSET_MANAGER	String	"ASSET_MANAGER"	Asset manager.
INVESTMENT_MANAGER	String	"INVESTMENT_MANAGER"	Investment manager.

Constant group: CmBalancingStep

Description: Enumeration specifying CM balancing steps.

Constant name	Type	Value	Comment
CM_BALANCING_1	String	"CM_BALANCING_1"	Step 1, involves IM, AM, VM, dividends, funding etc.
CM_BALANCING_2	String	"CM_BALANCING_2"	Step 2, involves fees.

Constant group: CompoundingConvention

Description: Compounding Period Used.

Constant name	Type	Value	Comment
NACC	String	"NACC"	NACC
NACQ	String	"NACQ"	NACQ
NACA	String	"NACA"	NACA
NACS	String	"NACS"	NACS
SIMPLE	String	"SIMPLE"	Simple

Constant group: CondType

Description: Price condition types. The value is a price tag that marks the market data values.

Constant name	Type	Value	Comment
ANY	String	"ANY"	Used to get the latest entry regardless of condition.
INDICATIVE	String	"INDICATIVE"	Indicative.
END_OF_DAY_SETTLEMENT	String	"END_OF_DAY_SETTLEMENT"	EoD Margin Call.
INTRA_DAY_1_SETTLEMENT	String	"INTRA_DAY_1_SETTLEMENT"	Intra day margin call.
INTRA_DAY_2_SETTLEMENT	String	"INTRA_DAY_2_SETTLEMENT"	Intra day margin call.
INTRA_DAY_3_SETTLEMENT	String	"INTRA_DAY_3_SETTLEMENT"	Intra day margin call.
INTRA_DAY_4_SETTLEMENT	String	"INTRA_DAY_4_SETTLEMENT"	Intra day margin call.
INTRA_DAY_5_SETTLEMENT	String	"INTRA_DAY_5_SETTLEMENT"	Intra day margin call.
INTRA_DAY_6_SETTLEMENT	String	"INTRA_DAY_6_SETTLEMENT"	Intra day margin call.
INTRA_DAY_7_SETTLEMENT	String	"INTRA_DAY_7_SETTLEMENT"	Intra day margin call.
INTRA_DAY_8_SETTLEMENT	String	"INTRA_DAY_8_SETTLEMENT"	Intra day margin call.
INTRA_DAY_9_SETTLEMENT	String	"INTRA_DAY_9_SETTLEMENT"	Intra day margin call.
INTRA_DAY_10_SETTLEMENT	String	"INTRA_DAY_10_SETTLEMENT"	Intra day margin call.

Constant group: ContractSizeType**Description:** Enumeration for contract size type

Constant name	Type	Value	Comment
BASE	Integer	1	Base
MINI	Integer	2	Mini
MAXI	Integer	3	Maxi
SUPER	Integer	4	Super

Constant group: CorporateActionStatus**Description:** Status of a Corporate Action task.

Constant name	Type	Value	Comment
NEW	int	0	Task registered and pending for action.
EXECUTED	int	1	Task executed.
CANCELLED	int	2	Task cancelled.
FAILED	int	-1	Task failed.

Constant group: CouponIndicator**Description:** Coupon rate indicator for Bonds.

Constant name	Type	Value	Comment
FIXED	String	"FIXED"	Coupon is Fixed.

Constant group: DATE_TYPE**Description:** Defines the different date types.

Constant name	Type	Value	Comment
CLOSED	int	1	On this day, the exchange is closed.
HALF_DAY	int	2	On this day, the exchange uses a half-day schedule.
NORMAL	int	3	On this day, the exchange uses the normal schedule.

Constant group: DAY_COUNT_CONVENTION**Description:** Method used for bootstrapping.

Constant name	Type	Value	Comment
ACTUAL_360	String	"ACTUAL_360"	Actual 360.
ACTUAL_365	String	"ACTUAL_365"	Actual 365.

Constant group: DIVISOR**Description:** There are integer/long fields that represent decimal numbers. These need to be divided with the following constants.

Constant name	Type	Value	Comment
QTY	int	1000000	Divisor for quantity field
PRICE	int	1000000	Divisor for price fields
INTEREST	int	1000000	Divisor for interest fields.
DELTA	int	1000000	Divisor for delta fields.
DECIMAL	int	1000000	Divisor for decimal value fields.

Constant group: ExerciseStyle**Description:** Defines the type of option

Constant name	Type	Value	Comment
EUROPEAN	int	1	
AMERICAN	int	2	

Constant group: EXTRAPOLATION_METHOD**Description:** Method used for extrapolation.

Constant name	Type	Value	Comment
LINEAR	String	"LINEAR"	Linear extrapolation.
FLAT	String	"FLAT"	Use nearest interpolated value.
FLAT_FORWARD	String	"FLAT_FORWARD"	Float forward volatility extrapolation.

Constant group: FeedSource**Description:** Market data pricing tag.

Constant name	Type	Value	Comment
RTC	String	"RTC"	Rtc.
PRICING_SYSTEM	String	"PRICING_SYSTEM"	JSE Pricing system.

Constant group: FourEyesState**Description:** State of FourEyes operation.

Constant name	Type	Value	Comment
UNKNOWN_REASON	int	0	
RECEIVED	int	1	
INITIATED	int	2	
CONFIRMED	int	3	
COMPLETED	int	4	
EXPIRED	int	5	
REQUEST_FAILED_VALIDATION	int	6	
CONFIRM_FAILED_VALIDATION	int	7	
ERROR	int	8	
CANCELLED	int	9	
REJECTED	int	10	

Constant group: InstrumentIdType**Description:** Defines the type of the InstrumentId (ISIN, CUSIP etc)

Constant name	Type	Value	Comment
ISIN	String	"ISIN"	ISIN identifier
CUSIP	String	"CUSIP"	CUSIP identifier
SYMB	String	"SYMB"	SYMBOL identifier

Constant group: InstrumentSubType**Description:** Defines the type of Instrument

Constant name	Type	Value	Comment
EQUITY	String	"EQUITY"	Equity.
INDEX	String	"INDEX"	Index.
SINGLE_STOCK	String	"SINGLE_STOCK"	Single Stock.
CFD	String	"CFD"	Contract For Difference.
DIVIDEND_NEUTRAL	String	"DIVIDEND_NEUTRAL"	Dividend Neutral.
FOREX	String	"FOREX"	Forex.
FOREX_PAIR	String	"FOREX_PAIR"	Forex Pair.
BOND	String	"BOND"	Bond.
BASKET	String	"BASKET"	Basket.
EXOTIC	String	"EXOTIC"	Exotic.
FOREX_INDEX	String	"FOREX_INDEX"	Forex index.
FWDFWD	String	"FWDFWD"	FwdFwd.
EXOTIC_OPTION	String	"EXOTIC_OPTION"	Exotic option.
INTERNATIONAL_DIVIDEND_NEUTRAL	String	"INTERNATIONAL_DIVIDEND_NEUTRAL"	International dividend neutral.
INTERNATIONAL_EQUITY	String	"INTERNATIONAL_EQUITY"	International equity.
INTERNATIONAL_INDEX	String	"INTERNATIONAL_INDEX"	International index.
INVERTED	String	"INVERTED"	Inverted.
OTHER	String	"OTHER"	Other.
QUANTO	String	"QUANTO"	Quanto.

QUANTO_INTL	String	"QUANTO_INTL"	Quanto international.
QUANTO_INDEX_DIVIDEND_NEUTRAL	String	"QUANTO_INDEX_DIVIDEND_NEUTRAL"	Quanto index dividend neutral.
QUANTO_INTL_DIVIDEND_NEUTRAL	String	"QUANTO_INTL_DIVIDEND_NEUTRAL"	Quanto international dividend neutral.
VARIANCE	String	"VARIANCE"	Variance.

Constant group: InstrumentType

Description: Defines the type of Instrument

Constant name	Type	Value	Comment
FUTURE	String	"FU"	Future.
BOND	String	"BO"	Bond
OPTION	String	"OPT"	Parent of OptionTradableInstruments
SPOT	String	"SPOT"	Spot type instrument
CFD	String	"CFD"	Contract For Difference.

Constant group: INTEREST_RATE_CONVENTION

Description: Interest rate convention for the interest rate produced.

Constant name	Type	Value	Comment
NACC	String	"NACC"	NACC.

Constant group: INTERPOLATION_METHOD

Description: Method used for interpolation.

Constant name	Type	Value	Comment
LINEAR	String	"LINEAR"	Linear.
FLAT_FORWARD	String	"FLAT_FORWARD"	Float forward volatility interpolation.
NATURAL_CUBIC_SPLINE	String	"NATURAL_CUBIC_SPLINE"	Natural cubic spline.
MONOTONE_PRESERVING	String	"MONOTONE_PRESERVING"	Variant of cubic spline that ensures positive and continuous forward rates. For yield curve interpolation.
MONOTONE_CONVEX	String	"MONOTONE_CONVEX"	Ensures a positive and continuous forward rate. For yield curve interpolation..

Constant group: LoginStatus

Description: Provides the result of a login request.

Constant name	Type	Value	Comment
LOGIN_ACCEPTED	int	0	The login is accepted.
LOGIN_REJECTED	int	-1	The login is rejected due to invalid password or invalid user id.
USER_ACCOUNT_LOCKED	int	-2	User account is locked due to too many erroneous login attempts.
PASSWORD_EXPIRED	int	-3	The password has expired.
LOGIN_ACCESS_DENIED	int	-4	User does not have access to login service for this application.
WRONG_VERSION	int	-5	Client and TAX server versions are not compatible.
INITIAL_LOGIN	int	-6	Initial login, password must be changed.
USER_ACCOUNT_DISABLED	int	-7	Account disabled by operational staff.

Constant group: MemberType

Description: Defines the different member/participant types.

Constant name	Type	Value	Comment
MARKETPLACE	Integer	1	The Clearing House itself.
INFORMATION_VENDOR	Integer	5	An Information Vendor.
MEMBER_UNIT	Integer	7	A member unit is a type of member that must be connected to a parent member, for example to divide an organization into different departments. Trading Member Branches and Clients are both of the type MEMBER_UNIT.
CLEARING_ONLY_MEMBER	Integer	8	A Clearing Member.

TRADING_ONLY_MEMBER	Integer 9	A Trading Member.
---------------------	-----------	-------------------

Constant group: OptionAllocationModelType**Description:** Allocation model for option exercise.

Constant name	Type	Value	Comment
PRO_RATA	int	1	Pro-rata allocation.
RANDOM	int	2	Random allocation.

Constant group: OptionStyle**Description:** Style of the Option

Constant name	Type	Value	Comment
FUTURE_STYLE	Integer	1	Future styled
UPFRONT_PREMIUM	Integer	2	Upfront premium

Constant group: ParticipantUnitType**Description:** Participant type. Defines the type of participant a member has in the member tree.

Constant name	Type	Value	Comment
CLEARING_MEMBER	Integer	1	Clearing Member.
TRADING_MEMBER	Integer	2	Trading Member
CLIENT	Integer	3	Client.
TRADING_MEMBER_BRANCH	Integer	4	Trading Member Branch.
INFORMATION_VENDOR	Integer	5	Information Vendor.

Constant group: PaymentAdviceState**Description:** The states of a Payment Advice.

Constant name	Type	Value	Comment
RECEIVED	int	0	
NOTIFIED_TO_CM	int	1	
REJECTED_BY_CM	int	2	
CONFIRMED_BY_CM	int	3	
SETTLED_BY_CM	int	4	
CANCELLED	int	5	

Constant group: PeriodType**Description:** Period type.

Constant name	Type	Value	Comment
DAYS	String	"DAYS"	Days.
MONTHS	String	"MONTHS"	Months.
YEARS	String	"YEARS"	Years.

Constant group: PositionAccountSubType**Description:** The sub type of the position account.

Constant name	Type	Value	Comment
SUSPENSE	Integer	1	Suspense
MAIN	Integer	2	Main
SUB	Integer	3	Sub

Constant group: PositionAccountType**Description:** The type of the position account.

Constant name	Type	Value	Comment
HOUSE	Integer	1	House
CLIENT	Integer	2	Client

Constant group: PositionType**Description:** Defines different Position types.

Constant name	Type	Value	Comment
---------------	------	-------	---------

ACTUAL	int	1	A position of type ACTUAL represents ownership of the position.
SETTLEMENT	int	2	A position of type SETTLEMENT represents changed of the position on the settlement date.

Constant group: PositionReason

Description: Reason for position update

Constant name	Type	Value	Comment
UNKNOWN_REASON	int	0	Unknown reason, an internal error has occurred.
FEE	int	1	Booking Fee.
TRADE	int	2	New trade.
EXERCISE	int	5	The position in the option was closed out due to early exercise, or the option is in-of-money and the option is exercised automatically. The future trade is the result of an option exercise. (early or in-the-money automatically by the system)
CASH_SETTLEMENT_VM	int	7	Cash settlement of variation margin
SETTLEMENT_NETTING	int	17	Positions are concentrated to the settlement accounts and netted out on the original accounts.
SETTLED	int	18	A payment or delivery has been processed by an external system and the settlement position is netted out.
COLLATERAL	int	19	Collateral position update.
ALLOCATED_FROM	int	26	The trade is allocated from a trading member account to a client account.
ALLOCATED_TO	int	27	The trade origins from an allocation from a trading member account to a client account.
ALLOCATION_CORRECTION_FROM	int	28	The deal is erroneously moved from the client account to another client account.
ALLOCATION_CORRECTION_TO	int	29	The deal originates from a move from one client account to another client account.
PRINCIPAL_CORRECTION_FROM	int	30	A position is moved from a house main account to a house sub account or from a house sub account to a house main account by creating a new deal.
PRINCIPAL_CORRECTION_TO	int	31	The deal originates from a move from a client account.
ACCUMULATED_FROM	int	32	The deal was aggregated to another deal.
ACCUMULATED_TO	int	33	The deal originates from a deal aggregation activity.
POS_SUBACCOUNT_MOD_FROM	int	34	The position sub account was moved from this account..
POS_SUBACCOUNT_MOD_TO	int	35	The position sub account was moved to this account.
ASSIGNED_FROM	int	36	The deal assign source.
ASSIGNED_TO	int	37	The deal originates from a deal assign activity.
ASSIGN_INITIATED	int	40	The deal assign initiated from.
ASSIGN_REJECTED	int	41	The deal assign rejected by receiver.
ASSIGN_CANCELLED	int	42	The deal assign cancelled by initiator.
ASSIGN_EXPIRED	int	43	The deal assign has expired.
START_OF_DAY	int	44	Start of day position snapshot.
TRIPARTITE_FROM	int	45	The deal has been assigned using Tripartite agreement.
TRIPARTITE_TO	int	47	The deal origins from an assign using Tripartite agreement.
TRIPARTITE_INITIATED	int	49	Tripartite assign has been initiated.
TRIPARTITE_APPROVED	int	50	Tripartite assign has been approved.
TRIPARTITE_REJECTED	int	51	Tripartite assign has been rejected.
TRIPARTITE_CANCELLED	int	52	Tripartite assign has been cancelled.
TRIPARTITE_EXPIRED	int	53	Tripartite assign has expired.
TRADE_SUBACCOUNT_MOD_FROM	int	54	Position has been moved from this account as Sub account modification.
TRADE_SUBACCOUNT_MOD_TO	int	55	Position has been moved to this account as Sub account modification.
ZERO_FEE	int	56	Trade updated for zero fee.

ABANDON	int	59	Option abandon.
TRANSFERRED_SP_FROM	int	60	Position has been moved from this account by Transfer Single Position.
TRANSFERRED_SP_TO	int	61	Position has been moved to this account by Transfer Single Position.
CLOSE_OUT	int	62	A close-out deal is created by the system to close open positions in a future on expiration of a tradable instrument.
PARTIALLY_SETTLED	int	63	Partially settled payment.
NET_PAYMENT	int	64	Payment from external system.
FEE_VAT	int	65	VAT amount for booking fee.
CANCELLED	int	66	Trade cancelled.
CANCELLED_BUST	int	67	Trade busted.
CANCELLED_PRICE_ADJUST	int	68	Trade cancelled for price adjust
DIVIDEND	int	69	Dividend payment for dividend neutral contracts
CLOSE_OUT_CA	int	70	Close out position for Corporate Action.
NEW_POSITION_CA	int	71	Created position for Corporate Action.
INTEREST_ON_COLLATERAL	int	73	Interest amount on collateral.
FUNDING_INTEREST	int	74	Funding interest payment for CFDs.
DEPOSIT	int	76	A deposit of an asset in an account.
WITHDRAWAL	int	77	A withdrawal of an asset from an account.
TRANSFERRED_FROM	int	78	An opposite trade was created at the original account as a result of a client or TM position transfer.
TRANSFERRED_TO	int	79	A trade was created at the destination account as a result of a client or TM position transfer.
END_OF_DAY	int	80	End of day position snapshot.
SYSTEM_STARTUP	int	81	RTC system startup snapshot.
DEFAULT_FROM	int	82	The position was transferred due to a defaulted member or client.
DEFAULT_TO	int	83	The position was transferred due to a defaulted member or client.
MANUAL	int	84	Event due to a manual update.
MIGRATION	int	85	An add/update due to migration.
COMMISSION	int	86	Commissions added by members.
RISK_FEE	int	88	Risk Fee.
RISK_FEE_VAT	int	90	VAT amount for risk fee.
REVERSE_TRANSACTION	int	91	A position was updated due to a revert of the EOD step.
DELETED	int	92	A close out, option exercise or abandon trade was marked as deleted due to a revert of the EOD step.

Constant group: RollsOnConvention

Description: Rolls on convention. Needed to determine coupon dates.

Constant name	Type	Value	Comment
DAY	String	"DAY"	Day(m).
START_OF_MONTH	String	"START_OF_MONTH"	Start of month.
END_OF_MONTH	String	"END_OF_MONTH"	End of month.
IMM_DAY	String	"IMM_DAY"	IMM Day. (3rd Monday of month).

Constant group: CommissionStatus

Description: The state of the commission.

Constant name	Type	Value	Comment
NEW	int	1	New
CANCELLED	int	2	Cancelled
REJECTED	int	3	Rejected
PENDING	int	4	The commission is pending, not yet applied
EXPIRED	int	5	Pending commission has expired

Constant group: ReplayRequestType**Description:** Literals describing the type of replay request

Constant name	Type	Value	Comment
REPLAY	int	0	Request to replay specific events; no future updates
REPLAY_UNSEGMENTED	int	1	Request to replay specific events without having to issue requests for new segments
REPLAY_SUBSCRIPTION	int	2	Request for unsegmented replay of events up to the latest and for subsequent subscription to future updates

Constant group: RtcState**Description:** System state in RTC.

Constant name	Type	Value	Comment
OPEN	String	"OPEN"	Open.
END_OF_TRADE_MANAGEMENT	String	"END_OF_TRADE_MANAGEMENT"	Trade management is no longer allowed.
END_OF_DAY	String	"END_OF_DAY"	End of Day process started.
POST_END_OF_DAY	String	"POST_END_OF_DAY"	End of Day process completed.

Constant group: RtcTradeType**Description:** Trade type (from Trading System).

Constant name	Type	Value	Comment
ORDER	String	"ORDER"	Order.
REPORT	String	"REPORT"	Report.
CANCEL_TRADE	String	"CT"	Cancel trade.
TRADE_BUST	String	"TB"	Trade bust.
CANCEL_PRICE_ADJUST	String	"PA"	Cancel price adjust.

Constant group: SessionStatus**Description:** Session status

Constant name	Type	Value	Comment
FORCED_LOGOFF_BY_NEW_LOGIN	int	1	The session has been terminated due a new login with the same user.
FORCED_LOGOFF_USER_DISABLED	int	2	The session has been terminated because the user has been disabled.
FORCED_LOGOFF_USER_DELETED	int	3	The session has been terminated because the user has been deleted.
FORCED_LOGOFF	int	4	User session logout was forced. Caused by an operator terminating the session.
DISCONNECT	int	5	User session disconnected
NORMAL_LOGOFF	int	6	Normal user requested logout

Constant group: SettlementType**Description:** Defines the type of option

Constant name	Type	Value	Comment
CASH	int	1	
PHYSICAL	int	2	
CASH_OR_PHYSICAL	int	3	

Constant group: SchedulerState**Description:** State of the scheduler.

Constant name	Type	Value	Comment
NORMAL	Integer	1	Normal state during daily operations.
RERUN_EOD	Integer	2	This state is used during End of Day rerun.
INTRADAY_MARGIN_CALL	Integer	3	This current system state is used during Intraday Margin Call.

REBALANCING	Integer 4	This current system state is used during Intraday Collateral Rebalancing.
-------------	-----------	---

Constant group: SubscriptionRequestType

Description: Literals describing the type of subscription request

Constant name	Type	Value	Comment
CURRENT_VALUE	int	1	Request for current values only; no future updates
SUBSCRIPTION	int	2	Request for subscription of future updates only; no current value
CURRENT_VALUES_AND_SUBSCRIPTION	int	3	Request for current values and future updates

Constant group: ValuationModelType

Description: Valuation Model Type.

Constant name	Type	Value	Comment
MARK_TO_MODEL	String	"MARK_TO_MODEL"	Mark to model.
MARK_TO_MARKET	String	"MARK_TO_MARKET"	Mark to market.

Constant group: ValuationSubType

Description: Valuation Sub Type.

Constant name	Type	Value	Comment
COST_OF_CARRY_DIVIDEND_PROJECTION	String	"COST_OF_CARRY_DIVIDEND_PROJECTION"	Cost of Carry dividend projection.
COST_OF_CARRY_DIVIDEND_YIELD	String	"COST_OF_CARRY_DIVIDEND_YIELD"	Cost of Carry dividend yield.
COST_OF_CARRY_DIVIDEND_NEUTRAL	String	"COST_OF_CARRY_DIVIDEND_NEUTRAL"	Cost of Carry dividend neutral.
BLACK76	String	"BLACK76"	Black76.

Constant group: YieldType

Description: Yield Type.

Constant name	Type	Value	Comment
YIELD_TO_MATURITY	String	"YIELD_TO_MATURITY"	fixed income done in yield to maturity
DIVIDEND_YIELD	String	"DIVIDEND_YIELD"	Percentage value represented as the annual dividend payouts (cash flow) of the instrument in relation to the current market price of the instrument.
INTEREST_RATE	String	"INTEREST_RATE"	Interest specified as a percentage value.

Constant group: ClearingMemberLinkStatus

Description: The status of the clearing member link. The value may be null and means a new link has been added.

Constant name	Type	Value	Comment
IN_PROGRESS	Integer	1	The EOD transfer of member is in progress. This happens when a link shall be changed according to effective date of the link.
ACTIVE	Integer	2	The EOD transfer of member is finished and the new link became active.
OLD	Integer	3	The link that became old after the EOD transfer of member.
FAILED	Integer	4	The EOD transfer of member failed.
NEW	Integer	5	The EOD transfer of member is not applicable for this link. It is a new link and nothing to transfer from.

Constant group: SettlementInstructionState

Description: Settlement Instruction State.

Constant name	Type	Value	Comment
CREATED	String	"CREATED"	The instruction has been created but processing has not yet started.
PENDING	String	"PENDING"	The processing has started for the instruction.
PARTIALLY_SETTLED	String	"PARTIALLY_SETTLED"	The instruction has been partially settled.
SETTLED	String	"SETTLED"	The instruction has been completely settled, i.e. the settled amount is the same as the requested amount. This is an end state.
CANCELLED	String	"CANCELLED"	The instruction has been cancelled. This is an end state.
FAILED	String	"FAILED"	The instruction has failed. This may be used by an external part if there is something wrong with the instruction, for example an erroneous account number. Not used in JSE.

Constant group: CommissionVatType

Description: VAT for commission.

Constant name	Type	Value	Comment
VAT_AT_STANDARD_RATE	int	1	VAT at standard rate.
VAT_AT_ZERO_PERCENT	int	2	VAT at zero percent.

Status Codes

Value	Name	Text	Comment
3001	Ok	OK	OK
3002	Warning	The processing succeeded but some errors occurred	Warning
3003	Deferred	The request is queued	Deferred
3004	DeferredWithWarning	The request is enqueued but some errors occurred	DeferredWithWarning
4001	AfwFailMsgVal	Message validation failed, see explanation	Message validation failed,
1200 1	AccNoRightsDefined	User is not authorized for this service	
1200 2	AccNoUserFound	No user found	
1200 3	AccNoRulesDefined	User is not authorized for this service	
1200 4	AccUserDeniedService	The service is not allowed	
1200 5	AccUserDeniedServiceForObject	The service is not allowed for this object	
1600 1	AfwQueueInfoMissing	The queue information in the configuration is missing. Please check your configuration file.	The queue information in the configuration is missing. Please check your configuration file.
1600 2	AfwFailDynVal	Dynamic validation error: This failure should be replaced by a specific one.	Dynamic validation error: This failure should be replaced by a specific one.
1600 3	AfwFailStatVal	Static validation error: This failure should be	Static validation error: This failure should be

		replaced by a specific one.	replaced by a specific one.
16004	AfwDynValTapStatusErr	Program error, dynamic validation inconsistency for TapStatus	Program error, dynamic validation inconsistency for TapStatus
16005	AfwAppITapStatusErr	Program error, application service inconsistency for TapStatus	Program error, application service inconsistency for TapStatus
16006	AfwAppINullOrReturn	Program error, null not allowed return from service	Program error, null not allowed return from service
16007	AfwFailValTokenValue	Token value must not be undefined	Token value must not be undefined
16008	AfwFailValMessage	The message must not be null	The message is not allowed to be null
16009	AfwFailValState	The request is not allowed during this trading state	The request is not allowed during this trading state
16010	AfwFailValUser	User is not allowed to perform request	User is not allowed to perform request
16011	AfwNoResponseFound	No response message found within specified time interval	No response message found within specified time interval
16012	AfwQueued	The request is queued for processing	The request is queued for processing
16013	AfwCancelled	The request is cancelled	The request is cancelled
16014	AfwExecuting	The request is executing	The request is executing
16015	AfwNotFound	The request is not found	The request is not found
16016	AfwVersionMismatchThrowaway	The request could not be served by the server	The request could not be served by the server
16017	AfwServiceNotAvailable	The requested service is not available in the recipient server	The requested service is not available in the recipient server
16018	AfwInvalidStateForCatchUp	Cannot perform catch-up when in current state	Cannot perform catch-up when in current state
16019	AfwInitiateCatchUpFailed	Failed to initiate catch-up	Failed to initiate catch-up
16020	AfwInvalidStateForRecoveryNegotiation	Cannot perform recovery negotiation when in current state	Cannot perform recovery negotiation when in current state
16021	AfwInitiateRecoveryNegotiationFailed	Failed to initiate recovery negotiation	Failed to initiate recovery negotiation
16022	AfwDbClassNoMsgIf	The requested object class is not a MessageIf	The requested object class is not a MessageIf
16023	AfwDbClassNoDbTable	The requested object class is not a database table	The requested object class is not a database table
16024	AfwDbUnrecognizedChildColumn	The requested child attribute is not a database column	The requested child attribute is not a database column
16025	AfwDbChainError	There was an error in the in-chain database service	There was an error in the in-chain database service
16026	AfwCvSegmentTooOld	The requested snapshot segment is too old and	The requested snapshot segment is too old and

		has been dropped by the server	has been dropped by the server
16027	AfwCvSegmentTooMany	The server cannot allocate more snapshot segments at this time. Try again later.	The server cannot allocate more snapshot segments at this time. Try again later.
16028	AfwFailedToUnpackRawRequest	Failed to unpack raw request	Failed to unpack raw request
16029	AfwUnknowRawRequestUnpacker	Failed to find unpacker for a raw request	Failed to find unpacker for a raw request
16030	AfwUnknowRawRequestDestination	Failed to find destination for a raw request	Failed to find destination for an unpacked raw request
16031	AfwXioConnectRefused	Connection to the xio messaging server was refused	Connection to the xio messaging server was refused
16032	AfwXioConnectFailed	Failed to connect to the xio messaging server	Failed to connect to the xio messaging server
16033	AfwInitiateCatchUpNeedTruncate	Standby object store might need to be truncated	Standby object store might need to be truncated
17001	TaxSessionMissing	There is no valid session available for the user.	There is no valid session available for the user.
17002	TaxGateletFailure	The gatelet failed to forward the request.	The gatelet failed to forward the request.
17003	TaxServiceFailure	The service returned a null response.	The service returned a null response.
17004	TaxMissingMessage	Cannot service request: Unknown message	The incoming message is not included in the classpath for TAX.
17005	TaxMissingGatelet	Cannot service request: No gatelet registered for the message.	The incoming message does not have an associated gatelet.
17006	TaxComponentActionFailed	Cannot service request: Failed to start/stop component.	Illegal state transition
17007	TaxComponentNotFound	Cannot service request: Component not found.	No such component
17008	TaxMissingTepsSubscription	No TEPS subscription connection	No TEPS subscription connection
17009	TaxInvalidTepsSubscription	Failed to activate TEPS subscription	Failed to activate TEPS subscription
17010	TaxUnknownClientSubscription	Failed to find client subscription	Failed to find client subscription
17011	TaxUserDisabled	Session no longer valid - user disabled	Session no longer valid - user disabled
17012	TaxUserDeleted	Session no longer valid - user deleted	Session no longer valid - user deleted
17013	TaxSessionThrottled	Session over-utilized - throttled	Session throttled, request delayed
17014	TaxRejectedThrottled	Request rejected - rate too high	Request rejected - rate too high
17015	TaxSessionThrottledResetOnly	Session over-utilized: throttled, delayed - timestamp reset	Session throttled, return code not returned, but timestamp was reset.
17016	TaxTargetPartitionDown	Cannot service request. There are no servers available for the target partition.	Cannot service request. There are no servers available for the target partition.

17017	TaxFailoverInProgress	Cannot service request: A failover is currently in progress.	Cannot service request: A failover is currently in progress.
17018	TaxServiceException	The request resulted in an exception in the target service.	The request resulted in an exception in the target service.
17019	TaxTransportInterrupted	Cannot service request: Internal communication was interrupted.	Cannot service request: Internal communication was interrupted.
18001	AlertUnknownId	Unknown alert id	The id of the alert is not known.
52001	CdInternalJavaError	CD internal java error, unexpected exception occurred.	CD internal java error, unexpected exception occurred.
52002	CdInvalidFirmId	The provided member ID does not exist.	The provided member ID does not exist.
52003	CdIllegalFirmId	The provided member ID is illegal.	The provided member ID is illegal.
52004	CdInvalidUserId	The provided user ID does not exist.	The provided user ID does not exist.
52005	CdIllegalUserId	The provided user ID is illegal.	The provided user ID is illegal.
52006	CdInvalidPwd	Invalid user ID or password.	For some reason the user action failed.
52007	CdInvalidPwdSameAsBefore	Invalid password, new password cannot be same as old.	Invalid password, new password cannot be same as old.
52008	CdInvalidLoginTicket	Login ticket from pre-login is not valid	Invalid login ticket
52009	CdInvalidPwdFormat	Invalid password format	Invalid password format
52010	CdUserDisabled	User account is disabled	User account is disabled
52011	CdUserPasswordNotInitialized	User password not initialized must be changed before login	User password not initialized must be changed before login
52012	CdUserPasswordHasExpired	User password has expired	User password has expired
52013	CdNoSuchUser	User does not exists.	For some reason the user action failed.
52014	CdUnknownError	An unexpected error occurred.	Use this error code with care. It is better to construct a specific message.
52015	CdUserActionFailed	The user action failed.	For some reason the user action failed.
52016	CdPreLoginNotUsed	Pre-Login service is not configured and used	Pre-Login service is not configured and used
52017	CdPwdLocked	Account locked.	For some reason the user action failed.
52018	CdPwdExpired	Password expired.	For some reason the user action failed.
52019	CdAccProfileUnavailable	Unable to fetch the acc profile.	For some reason the user action failed.
52020	CdAlreadyLoggedIn	The user is already logged on.	User attempted to logon but is already logged on.
52021	CdInvalidRoleChangeFIX	It's not allowed to change rules of a FIX_Trader.	.

5202 2	CdInvalidRoleChangeNonFIX	It's not allowed to change rules for a non-FIX_Trader to a FIX_Trader.	.
5202 3	CdInvalidFIXLogin	User is FIX user. Not allowed to login via GUI.	.
5202 4	CdInvalidPwdProperties	Invalid value: Minimum value is 1.	.
5202 5	CdInvalidOrderBookRuleGroup	The supplied OrderBookRuleGroup does not exist	The supplied OrderBookRuleGroup does not exist
5202 6	CdInvalidOrderBook	The supplied OrderBook does not exist	The supplied OrderBook does not exist
5202 7	CdInvalidCombinationOrderBookLeg	The supplied Combination OrderBook Leg does not exist	The supplied OrderBook does not exist
5202 8	CdInvalidSubscriptionGroup	The supplied SubscriptionGroup does not exist	The supplied SubscriptionGroup does not exist
5202 9	CdInvalidCurrency	The supplied CurrencyId does not exist	The supplied CurrencyId does not exist
5203 0	CdInvalidTickSizeTableId	The supplied TickSizeTableId does not exist	The supplied TickSizeTableId does not exist
5203 1	CdInvalidTradingScheduleTableId	The supplied TradingScheduleTableId does not exist	The supplied TradingScheduleTableId does not exist
5203 2	CdInvalidAllowedRequestGroupId	The supplied AllowedRequestGroupId is invalid	The supplied AllowedRequestGroupId does not exist
5203 3	CdInvalidProcessingSequenceId	The supplied processingSequenceId is invalid	The supplied processingSequenceId is invalid
5203 4	CdInvalidUserPropertiesCategory	The supplied UserPropertiesCategory is invalid	The supplied UserPropertiesCategory is invalid
5203 5	CdInvalidServiceProfileId	The supplied ServiceProfileId is invalid	The supplied ServiceProfileId is invalid
5203 6	CdInvalidServiceProfileGroupId	The supplied ServiceProfileGroupId is invalid	The supplied ServiceProfileGroupId is invalid
5203 7	CdInvalidServiceProfileRefId	The supplied ServiceProfileRefId is invalid	The supplied ServiceProfileRefId is invalid
5203 8	CdIllegalServiceProfileId	The supplied ServiceProfileId contains illegal characters	The supplied ServiceProfileId contains illegal characters
5203 9	CdIllegalServiceProfileGroupId	The supplied ServiceProfileGroupId contains illegal characters	The supplied ServiceProfileGroupId contains illegal characters
5204 0	CdIllegalServiceProfileRefId	The supplied ServiceProfileRefId contains illegal characters	The supplied ServiceProfileRefId contains illegal characters
5204 1	CdIllegalServiceProfileEntryDelayClassId	The supplied ServiceProfileEntryDelayClassId contains illegal characters	The supplied ServiceProfileEntryDelayClassId contains illegal characters

52042	CdInvalidServiceProfileEntryDelayClassId	The supplied ServiceProfileEntryDelayClassId is invalid	The supplied InvalidServiceProfileRefId contains illegal characters
52043	CdIllegalServiceProfileEntryInfoLevelClassId	The supplied ServiceProfileEntryInfoLevelClassId contains illegal characters	The supplied ServiceProfileEntryInfoLevelClassId contains illegal characters
52044	CdInvalidServiceProfileEntryInfoLevelClassId	The supplied ServiceProfileEntryInfoLevelClassId is invalid	The supplied ServiceProfileEntryInfoLevelClassId is invalid
52045	CdCombinationOrdersLegsMustBelongToTheSameSubscriptionGroup	All legs in a combination order must belong to the same subscription group	
52046	CdCombinationOrdersLegsMustUseSameSorting	All legs in a combination order must belong to order books using the same type of sorting	
52047	CdCombinationOrderLegsMustNotShareTheSameOrderBook	Combination order legs may not share the same order book	
52048	CdTooManyCombinationOrderLegs	There were too many legs in the combination order	
52049	CdCombinationOrderLegsMustNotBeCombinationOrder	Combination order legs may not be a combination order	
52050	CdInvalidServerGroupId	The supplied ServerGroupId is invalid	The supplied ServerGroupId is invalid
52051	CdIllegalServerGroupId	The supplied ServerGroupId contains illegal characters	The supplied ServerGroupId contains illegal characters
52052	CdInvalidServerProcessId	The supplied ServerProcessId is invalid	The supplied ServerProcessId is invalid
52053	CdIllegalServerProcessId	The supplied ServerProcessId contains illegal characters	The supplied ServerProcessId contains illegal characters
52054	CdInvalidCalendarDate	The supplied CalendarDate is invalid	The supplied CalendarDate is invalid
52055	CdIllegalRemoveOperationType	The supplied cache remove type is invalid	The supplied cache remove type is invalid
52056	CdPartitionMismatch	Invalid Partition/Subscription group/Orderbook rule group/Orderbook combination	There is a mismatch between the defaultPartition in the OrderbookRulegroup and the partition for the supplied subscription group
52057	CdInvalidPwdUserOrFirm	Invalid userid, password or member firm	The login parameters are invalid
52058	CdInvalidPartitionTableId	Invalid partition table ID	The supplied partition table is invalid
52059	CdInvalidPartitionId	Invalid partition ID	The supplied partition ID does not exist
52060	CdInvalidTickSizeTable	The supplied ticksize table has no rows	The supplied ticksize table has no rows.
52061	CdInvalidLowerLimit	The supplied lower limit is invalid	The supplied lower limit is invalid.

52062	CdInvalidFractionalTick	The supplied fractional tick is invalid	The supplied fractional tick is invalid.
52063	CdInvalidTradeTick	The supplied trade tick is invalid	The supplied trade tick is invalid.
52064	CdTransactionValidationError	Transaction failed validation	Transaction failed validation. One or more mandatory fields are missing or have illegal values.
52065	CdInvalidInstrumentId	The supplied instrumentId is invalid	The supplied InstrumentId is invalid
52066	CdInvalidDisplayName	The supplied displayName is invalid	The supplied DisplayName is invalid
52067	CdInvalidOrderBookParameterId	The supplied OrderBookParameter ID is invalid	The supplied OrderBookParameter ID is invalid
52068	CdInvalidOrderBookRuleGroupParameterId	The supplied OrderBookRuleGroupParameter is invalid	The supplied OrderBookRuleGroupParameter is invalid
52069	CdInvalidModificationSchedule	The supplied serviceId is invalid	The supplied serviceId is invalid
52070	CdInvalidModificationScheduleDates	The supplied startTime and endTime are invalid	The supplied startTime and/or endTime is invalid
52071	CdUserNotAuthorizedForAnySessionType	The user is not authorized to use any session type	The user is not authorized to use any TAX-type
52072	CdNoConnectorForSessionTypeAvailable	No connectors for the user's session types are available	The user is not authorized to use any TAX-type
52073	CdInvalidUserPropertiesCategoryId	The supplied UserPropertiesCategory ID is invalid	The supplied UserPropertiesCategory ID is invalid
52074	CdInvalidExternalMarketId	The supplied ExternalMarketId is invalid	The supplied ExternalMarketId is invalid
52075	CdInvalidServerPartitionTable	The supplied ServerPartitionTable is invalid	The supplied ServerPartitionTable is invalid
52076	CdInvalidServerPartitionTableId	The supplied ServerPartitionTableId is invalid	The supplied ServerPartitionTableId is invalid
52077	CdInvalidServerPartition	The supplied ServerPartition is invalid	The supplied ServerPartition is invalid
52078	CdInvalidServerPartitionId	The supplied ServerPartition ID is invalid	The supplied ServerPartitionId ID is invalid
52079	CdInvalidGlobalPasswordProperties	The supplied GlobalPasswordProperties is invalid	The supplied GlobalPasswordProperties is invalid
52080	CdInvalidDefaultSvcProfHandling	Invalid default ServiceProfile handling	Invalid default ServiceProfile handling
52081	CdXmlParsingFailure	The supplied document could not be parsed	There where fatal syntax errors in the provided XML document.
52082	CdXmlLoadPartialFailure	Some entities were not loaded into CD	The loading of CD data was not a complete success.
52083	CdInvalidPostBootAction	The specified post boot action is unknown.	The specified post boot action is unknown and could not be executed.

52084	CdInvalidDateOrTime	The specified date or time has incorrect format or is inappropriate	The specified post boot action is unknown and could not be executed.
52085	CdScheduledUpdateError	Scheduled reference data update transaction failed.	Scheduled reference data update transaction failed.
52086	CdScheduledUpdatePreValidationError	Scheduled reference data update pre-validation failed.	Scheduled reference data update pre-validation failed.
52087	CdScheduledUpdatePreValidationWarning	Scheduled reference data update pre-validation resulted in warnings.	Scheduled reference data update pre-validation resulted in warnings.
52088	CdPendingScheduledUpdateWarning	A scheduled update has been issued for this object. The scheduled update may be lost.	A scheduled update has been issued for this object. The scheduled update may be lost.
52089	CdInvalidCombinationOrderLegDisplayOrder	Duplicate display order numbers on combination legs are not allowed. All numbers null is allowed.	
52090	CdUpdateWarning	The validation of the request resulted in warning.	The validation of the request resulted in warning(s). This code is returned when a reference data operation may conflict with other operations or is potentially dangerous. The actual transaction is not performed. To override the warning (and execute the transaction anyway), use the set the overrideWarning attribute to true in the request.
53001	CdMimInvalidMarketId	The marketId field is invalid	The marketId field is invalid
53002	CdMimInvalidName	The name field is invalid	The name field is invalid
53003	CdMimInvalidCountryCode	The countryCode field is invalid	The name countryCode is invalid
53004	CdMimInvalidMarketListId	The marketListId field is invalid	The marketListId field is invalid
53005	CdMimInvalidSegmentId	The supplied segmentId is invalid	The supplied SegmentId is invalid
53006	CdMimInvalidInstrumentId	The supplied instrumentId is invalid	The supplied InstrumentId is invalid
53007	CdMimInvalidTradableInstrumentId	The supplied tradableInstrumentId is invalid	The supplied TradableInstrumentId is invalid
53008	CdMimInvalidViewId	The supplied viewId is invalid	The supplied ViewId is invalid
53009	CdMimInvalidViewElementId	The supplied viewElementId is invalid	The supplied ViewElementId is invalid
53010	CdMimInvalidInstrumentType	The supplied instrumentType is invalid	The supplied InstrumentType is invalid
53011	CdMimInvalidSegmentType	The supplied segmentType is invalid	The supplied SegmentType is invalid

53012	CdMimInvalidValidFromDate	The supplied validFromDate field is invalid	The supplied validFromDate is unacceptable
53013	CdMimInvalidValidToDate	The supplied validToDate field is invalid	The supplied validToDate is a mess
53014	CdMimInvalidNavigationLevel	The supplied mimLevel field is invalid	The supplied mimLevel is not a valid MarketInstrumentModel Level literal.
53015	CdMimInvalidAction	The supplied action field is invalid	The supplied action is not a valid MarketInstrumentModel Action literal.
53016	CdMimXmlParsingFailure	The supplied document could not be parsed	There were fatal syntax errors in the provided XML document.
53017	CdMimXmlLoadPartialFailure	Some entities were not loaded into CD	The loading of CD data was not a complete success.
53018	CdMimInvalidRoleList	Invalid comma-separated role list	The format of the supplied role list is invalid.
53019	CdMimInvalidMemberType	Invalid member type	The supplied member types does not match any of the values in MemberType.
53020	CdMimInvalidListOfAliases	The supplied listOfAliases is invalid	The supplied listOfAliases is invalid
53021	CdMimRecursiveEnableFailed	Failed to enable some of the requested Elements	Failed to enable some of the requested Elements, sorry
53022	CdMimRecursiveDisableFailed	Failed to disable some of the requested Elements	Failed to disable some of the requested Elements, sorry
53023	CdMimInvalidHolidayScheduleId	Invalid holidayScheduleId field	Invalid holidayScheduleId field supplied
53024	CdMimInvalidHolidayId	Invalid holidayId field	Invalid holidayId field supplied
53025	CdMimInvalidTime	Invalid time format or value	A specified time either had an invalid format or value
53026	CdMimInvalidConvertibleData	Invalid ConvertibleData	The supplied ConvertibleData is invalid
53027	CdMimInvalidRightData	Invalid RightData	The supplied RightData is invalid
53028	CdMimInvalidSubscriptionOptionData	Invalid SubscriptionOptionData	The supplied SubscriptionOptionData is invalid
53029	CdMimInvalidInterimShareData	Invalid InterimShareData	The supplied InterimShareData is invalid
53030	CdMimInvalidWarrantData	Invalid WarrantData	The supplied WarrantData is invalid
53031	CdMimInvalidCertificateData	Invalid CertificateData	The supplied CertificateData is invalid
53032	CdMimInvalidIndustrySector	Invalid IndustrySector	The supplied IndustrySector is invalid

53033	CdMimInvalidTradableInstrumentData	Invalid TradableInstrumentData	The supplied TradableInstrumentData is invalid
53034	CdMimInvalidGenericInstrumentData	Invalid GenericInstrumentData	The supplied GenericInstrumentData is invalid
53035	CdMimInvalidInstrumentTypeData	Invalid InstrumentTypeData	The supplied InstrumentTypeData is invalid
53036	CdMimInvalidTradableInstrumentTypeData	Invalid TradableInstrumentType Data	The supplied TradableInstrumentType Data is invalid
53037	CdMimInvalidScheduledCorporateActionId	Invalid ScheduledCorporateActionId	The supplied TradableInstrumentType Data is invalid
53038	CdMimInvalidCorporateActionEffectId	Invalid CorporateActionEffectId	The supplied TradableInstrumentType Data is invalid
53039	CdMimNoGlobalTradingProtectionLimitDefined	No global Trading Protection Limit defined	There is no Global Trading Protection Limit Defined in the system.
53040	CdMimRatioCombLegError	A ratio combination has to have exactly two legs.	A ratio combination has to have exactly two legs.
69001	CacheConcurrentModification	Error updating cache, recursive call to updateCache	
69002	CacheError	Error updating the Cache. Possible causes: \n 1) A cache Add on something that already exists in the cache. \n 2) A cache update on something that does not exist in the cache.	
69003	CacheInvalidKey	Key node not found	Supplied key does not match a cache node
69004	CacheDbError	Exception from db	Cache has encountered a db exception while updating
69005	CacheEmptyUpdateReq	Update Req did not supply any update messages	Update Req did not supply any update messages
69006	CacheNoSuchNode	Update req, node not found	No node found matching update req
69007	CacheNodeAlreadyExists	Add req, cache does already contain node	Cache already contains node
69008	CacheValidationFailed	Cache failed validation	Cache failed validation
69009	CacheValidationIsReferenced	Node is referenced and can not be removed	Node is referenced and can not be removed
69010	CacheInvalidId	Id can not be null	MessageDataId Id can not be null
69011	CacheBlobError	Error encoding or decoding a blob	A BlobMangler implementation has thrown an exception
72001	TaxUnknownFlow	unknown broadcast flow	Client has specified and unknown broadcast flow when setting up a subscription

7200 2	TaxSubscriptionHandleNotFound	no active subscription found matching provided subscription handle	Client has provided an subscription handle that can not be found
7200 3	TaxUnknownInvalidRqst	unknown or invalid EMAPI request	EMAPI message is not known or is invalid
7200 4	TaxUnknownInvalidRsp	was not able to translate TAP response to EMAPI response	EMAPI message is not known or is invalid
7200 5	TaxUncompleteSnapshot	not all information could be collected successfully in the snapshot	Snapshot did not complete successfully
7200 6	TaxUncompleteRecovery	not all information could be collected successfully in the recovery	Recovery did not complete successfully
7200 7	TaxLoginFailed	user login request failed	TAX server was not able to logon the user
7200 8	TaxUserSessionAlreadyLoggedIn	user session already logged in	User has already logged in the session
7200 9	TaxPreLoginServiceNotUsed	PreLogin service is not used	PreLogin service is not used
7201 0	TaxUserHeartbeatInactivity	Have not received heartbeats for user, session will be disconnected	Have not received heartbeats for user, session will be disconnected
7201 1	TaxServiceNotAvailable	The TAX server has yet not reached routing state Active, the client should retry later	The TAX server has not reached routing state Active. Thus it can not service the request yet. The client should retry later.
7201 2	TaxSubscriptionNotAllowed	Subscription not allow, it may still be possible to retrieve the current values	Client has to poll for data instead of setting up a subscription
7201 3	TaxClientVersionMismatch	The client version is not compatible with the server, logon was rejected.	
7201 4	TaxInvalidOrderBookFilter	Invalid or zero length order book filter.	Invalid or zero length order book filter.
7201 5	TaxUserSessionLoginInternalError	login service failed internally	Login service failed internally
7201 6	TaxUnknownPartition	Can't map request to any partition	Partition mapping failed
7201 7	TaxUnknownRequestType	Unknown subscription request type	Client has specified an unknown request type when setting up a subscription
7201 8	TaxOwnerMappingNoUser	Can't determine user	User can't be determined from the request
7201 9	TaxOwnerMappingNoTradingMember	Can't determine member	Member can't be determined from the request
7202 0	TaxSystemReverse	The system has reverted to a prior state. Some previously sent updates may be obsolete. Reverse your data structures accordingly and re-synchronize	The system has reverted to a prior state. This typically happens when an emergency server is activated, which means that some previously sent updates may be

			obsolete and must be reverted.
72021	TaxTradeReportCounterpartyMemberUnknown	Can't determine the counter party member of the trade report	TradeReportReqProcessor, which fills in trading member information in trade reports, can not find the specified counter party member in the reference data.
85001	AhsInternalError	Internal error	
85002	AhsInvalidSql	The SQL query does not make sense	
1503001	CdRtcMessageValidationFailed		Generic no-text error message, text will be set by reporting code.
1503002	CdRtcCacheModificationError		Generic no-text error message, text will be set by reporting code.
1503003	CdRtcTransactionNotAllowedInCurrentSystemState	The transaction is not allowed in the current system state	Some transactions, particularly reference data update transactions, are only allowed in the system maintenance state.
1503004	CdRtcMulipleErrors		Multiple errors occurred. More information is provided in the text message.
1503005	CdRtcSwitchScheduleError	Error when trying to switch the current rtc schedule to another schedule. More information is provided in the text message.	Error when trying to switch the current rtc schedule to another schedule. More information is provided in the text message.
1504001	RtcClearingInvalidTradableInstrumentOnTrade	Tradable instrument on trade is missing	Tradable instrument on trade is invalid.
1504002	RtcClearingInvalidBuyerTradingMember	Buyer trading member on trade is invalid	Buyer trading member on trade is invalid.
1504003	RtcClearingInvalidSellingTradingMember	Selling trading member on trade is invalid	Selling trading member on trade is invalid.
1504004	RtcClearingDealManagementNotAllowedForAccount	The service is not allowed for this combination of from and/or to accounts	The service is not allowed for this combination of from and/or to accounts.
1504005	RtcClearingInvalidMember	Member Specified in the request is not valid.	Member Specified in the request is not valid.
1504006	RtcClearingNoValidCmLinkForBuyerTm	Buyer Trading Member Specified in the request has no valid Clearing Member Link.	Buyer Trading Member Specified in the request is not valid.
1504007	RtcClearingNoValidCmLinkForSellerTm	Seller Trading Member Specified in the request has no valid Clearing Member Link.	Seller Trading Member Specified in the request is not valid.
1504008	RtcClearingInvalidAccount	Account Specified in the request is not valid.	Account Specified in the request is not valid.
1504009	RtcClearingNotHouseAccount	Account Specified is not a house account.	Account Specified is not a house account.
1504010	RtcClearingMessageValidationFailed		Generic no-text error message, text will be set by reporting code.

1504 011	RtcClearingCashSettlementVMFailed	An error occurred during the execution of cash settlement variation margin	An error occurred during the execution of cash settlement variation margin.
1504 012	RtcClearingInvalidCollateralAccount	Collateral Account Specified in the request is not valid.	Collateral Account Specified in the request is not valid.
1504 013	RtcClearing_UNKNOWN_INSTRUMENT		Generic no-text error message, text will be set by reporting code.
1504 014	RtcClearing_UNKNOWN_ACCOUNT		Generic no-text error message, text will be set by reporting code.
1504 015	RtcClearing_INVALID_QUANTITY		Generic no-text error message, text will be set by reporting code.
1504 016	RtcClearing_INVALID_PRICE		Generic no-text error message, text will be set by reporting code.
1504 017	RtcClearing_MISSING_ALLOCATIONS		Generic no-text error message, text will be set by reporting code.
1504 018	RtcClearing_MESSAGE_VALIDATION_FAILED		Generic no-text error message, text will be set by reporting code.
1504 019	RtcClearing_INVALID_ALLOCATION		Generic no-text error message, text will be set by reporting code.
1504 020	RtcClearing_NOT_VALID_FOR_CASH_SETTLEMENT		Generic no-text error message, text will be set by reporting code.
1504 021	RtcClearing_NO_MARKET_PRICE		Generic no-text error message, text will be set by reporting code.
1504 022	RtcClearing_CASH_SETTLEMENT_FAILED		Generic no-text error message, text will be set by reporting code.
1504 023	RtcClearing_EXPIRATION_FAILED		Generic no-text error message, text will be set by reporting code.
1504 024	RtcClearing_EXPIRATION_NOT_SUPPORTED_IN_CURRENT_STATE		Generic no-text error message, text will be set by reporting code.
1504 025	RtcClearing_EXPIRATION_NOT_SUPPORTED		Generic no-text error message, text will be set by reporting code.
1504 026	RtcClearing_INVALID_INSTRUMENT_STATE		Generic no-text error message, text will be set by reporting code.
1504 027	RtcClearing_INVALID_INSTRUMENT_PLUGIN		Generic no-text error message, text will be set by reporting code.
1504 028	RtcClearing_INVALID_BASIC_ATTRIBUTES		Generic no-text error message, text will be set by reporting code.
1504 029	RtcClearing_INVALID_ADDITIONAL_ATTRIBUTES		Generic no-text error message, text will be set by reporting code.
1504 030	RtcClearing_INVALID_PRODUCT_REFERENCE		Generic no-text error message, text will be set by reporting code.

1504 031	RtcClearing_INVALID_CURRENCY_REFERENCE		Generic no-text error message, text will be set by reporting code.
1504 032	RtcClearing_NO_BUSINESS_DATE		Generic no-text error message, text will be set by reporting code.
1504 033	RtcClearing_UNBALANCED_TRANSACTION		Generic no-text error message, text will be set by reporting code.
1504 034	RtcClearing_REVERSE_OPERATION_FAILED		Generic no-text error message, text will be set by reporting code.
1504 035	RtcClearing_INVALID_PM_PARTITION		Generic no-text error message, text will be set by reporting code.
1504 036	RtcClearing_INVALID_PRODUCT_NAME		Generic no-text error message, text will be set by reporting code.
1504 037	RtcClearing_INVALID_SYMBOL		Generic no-text error message, text will be set by reporting code.
1504 038	RtcClearing_INVALID_NAME_FOR_ID		Generic no-text error message, text will be set by reporting code.
1504 039	RtcClearing_OPERATION_PENDING		Generic no-text error message, text will be set by reporting code.
1504 040	RtcClearing_INVALID_TRADABLE_INSTRUMENT_ATTRIBUTES		Generic no-text error message, text will be set by reporting code.
1504 041	RtcClearing_ID_NOT_UNIQUE		Generic no-text error message, text will be set by reporting code.
1504 042	RtcClearing_INVALID_ID_FOR_NAME		Generic no-text error message, text will be set by reporting code.
1504 043	RtcClearing_INVALID_PRODUCT_ID		Generic no-text error message, text will be set by reporting code.
1504 044	RtcClearing_INVALID_TRADABLE_INSTRUMENT_ID		Generic no-text error message, text will be set by reporting code.
1504 045	RtcClearing_INVALID_ACCOUNT_ID		Generic no-text error message, text will be set by reporting code.
1504 046	RtcClearing_INVALID_ACCOUNT_NAME		Generic no-text error message, text will be set by reporting code.
1504 047	RtcClearing_TRADE_NOT_FOUND		Generic no-text error message, text will be set by reporting code.
1504 048	RtcClearing_INVALID_TRADE_DESTINATION		Generic no-text error message, text will be set by reporting code.
1504 049	RtcClearing_INVALID_TRADE_DESTINATION_ACCOUNT		Generic no-text error message, text will be set by reporting code.
1504 050	RtcClearing_INVALID_AUTOMATIC_MOVE_ACCOUNT_ID		Generic no-text error message, text will be set by reporting code.

1504 051	RtcClearing_POSITION_NOT_FOUND		Generic no-text error message, text will be set by reporting code.
1504 052	RtcClearing_EXERCISE_FAILED		Generic no-text error message, text will be set by reporting code.
1504 053	RtcClearing_TIMESTAMP_IN_THE_PAST		Generic no-text error message, text will be set by reporting code.
1504 054	RtcClearing_FOUR_EYES_NOT_FOUND		Generic no-text error message, text will be set by reporting code.
1504 055	RtcClearing_OPTION_IS_NOT_AMERICAN		Generic no-text error message, text will be set by reporting code.
1504 056	RtcClearing_INSTRUMENT_IS_NOT_AN_OPTION		Generic no-text error message, text will be set by reporting code.
1504 057	RtcClearing_INVALID_POSITION		Generic no-text error message, text will be set by reporting code.
1504 058	RtcClearing_FOUR_EYES_EXPIRED		Generic no-text error message, text will be set by reporting code.
1504 059	RtcClearing_INVALID_FOUR_EYES_STATE		Generic no-text error message, text will be set by reporting code.
1504 060	RtcClearing_INVALID_FOUR_EYES_FILTER		Generic no-text error message, text will be set by reporting code.
1504 061	RtcClearing_DUPLICATE_MOVE_TRADE		Generic no-text error message, text will be set by reporting code.
1504 062	RtcClearing_SAVE_OBJECT_FAILED		Generic no-text error message, text will be set by reporting code.
1504 063	RtcClearing_CALENDAR_UNDEFINED		Generic no-text error message, text will be set by reporting code.
1504 064	RtcClearing_TRANSACTION_NOT_FOUND		Generic no-text error message, text will be set by reporting code.
1504 065	RtcClearing_INVALID_SEQ_NO		Generic no-text error message, text will be set by reporting code.
1504 066	RtcClearing_INVALID_SOURCE_ID		Generic no-text error message, text will be set by reporting code.
1504 067	RtcClearing_INVALID_SUBSCRIPTION_ID		Generic no-text error message, text will be set by reporting code.
1504 068	RtcClearing_NOT_SAME_ACCOUNT		Generic no-text error message, text will be set by reporting code.
1504 069	RtcClearing_NOT_SAME_INSTRUMENT		Generic no-text error message, text will be set by reporting code.
1504 070	RtcClearing_NOT_SAME_SIDE		Generic no-text error message, text will be set by reporting code.

1504 071	RtcClearing_NO_REMAINING_QTY		Generic no-text error message, text will be set by reporting code.
1504 072	RtcClearing_CASH_SETTL_OF_VM_INVALID		Generic no-text error message, text will be set by reporting code.
1504 073	RtcClearing_INVALID_CLIENT_DEAL_ID		Generic no-text error message, text will be set by reporting code.
1504 074	RtcClearing_CONCURRENT_POSITION_MODIFICATION		Generic no-text error message, text will be set by reporting code.
1504 075	RtcClearing_SESSION_TOKEN_REQUIRED		Generic no-text error message, text will be set by reporting code.
1504 076	RtcClearing_GATHER_CASH_NOT_ALLOWED		Generic no-text error message, text will be set by reporting code.
1504 077	RtcClearing_GATHER_CASH_FAILURE		Generic no-text error message, text will be set by reporting code.
1504 078	RtcClearing_AMBIGUOUS_INSTRUMENT_ROUTING		Generic no-text error message, text will be set by reporting code.
1504 079	RtcClearing_INVALID_INSTRUMENT		Generic no-text error message, text will be set by reporting code.
1504 080	RtcClearing_INVALID_BOOKMARK		Generic no-text error message, text will be set by reporting code.
1504 081	RtcClearing_DUPLICATE_TRADE_ID		Generic no-text error message, text will be set by reporting code.
1504 082	RtcClearing_INVALID_INSTRUMENT_LIST		Generic no-text error message, text will be set by reporting code.
1504 083	RtcClearing_INVALID_DEAL_BUNDLE_LIST		Generic no-text error message, text will be set by reporting code.
1504 084	RtcClearing_INVALID_HISTORY_FILTER		Generic no-text error message, text will be set by reporting code.
1504 085	RtcClearing_INVALID_UNDERLYING_REFERENCE		Generic no-text error message, text will be set by reporting code.
1504 086	RtcClearing_NOT_VALID_FOR_GENERIC_EVENT		Generic no-text error message, text will be set by reporting code.
1504 087	RtcClearing_GENERIC_EVENT_FAILED		Generic no-text error message, text will be set by reporting code.
1504 088	RtcClearing_INVALID_INSTRUMENT_SPECIFIC_ATTRIBUTES		Generic no-text error message, text will be set by reporting code.
1504 089	RtcClearing_SERVICE_UNAVAILABLE		Generic no-text error message, text will be set by reporting code.
1504 090	RtcClearing_UNKNOWN_REQUEST_TYPE		Generic no-text error message, text will be set by reporting code.

1504 091	RtcClearing_INTERNAL_DATABASE_ERROR		Generic no-text error message, text will be set by reporting code.
1504 092	RtcClearing_INTERNAL_TIMEOUT_ERROR		Generic no-text error message, text will be set by reporting code.
1504 093	RtcClearing_INTERNAL_ERROR		Generic no-text error message, text will be set by reporting code.
1504 094	RtcClearing_NOT_INITIALIZED		Generic no-text error message, text will be set by reporting code.
1504 095	RtcClearing_TIMED_OUT_OPERATION		Generic no-text error message, text will be set by reporting code.
1504 096	RtcClearing_CONCURRENT_OPERATION		Generic no-text error message, text will be set by reporting code.
1504 097	RtcClearing_INVALID_FROM_ACCOUNT		Generic no-text error message, text will be set by reporting code.
1504 098	RtcClearing_TRADE_IN_THE_PAST		Generic no-text error message, text will be set by reporting code.
1504 099	RtcClearing_DEAL_MANAGEMENT_NOT_ALLOWED		Generic no-text error message, text will be set by reporting code.
1504 100	RtcClearing_FOUR_EYES_ALREADY_EXISTS		Generic no-text error message, text will be set by reporting code.
1504 101	RtcClearing_INVALID_INITIATOR		Generic no-text error message, text will be set by reporting code.
1504 102	RtcClearing_INVALID_CLIENT		Generic no-text error message, text will be set by reporting code.
1504 103	RtcClearing_MTM_PRICE_MISSING_FOR_INSTRUMENT		Generic no-text error message, text will be set by reporting code.
1504 104	RtcClearing_INVALID_DELIVERY_INSTRUCTION_ID		Generic no-text error message, text will be set by reporting code.
1504 105	RtcClearing_ONGOING_FUTURES_EXPIRY		Generic no-text error message, text will be set by reporting code.
1504 106	RtcClearing_ONGOING_CASH_SETTLEMENT		Generic no-text error message, text will be set by reporting code.
1504 107	RtcClearing_CALC_DIVIDEND_FAILED		Generic no-text error message, text will be set by reporting code.
1504 108	RtcClearing_TRADE_MANAGEMENT_NOT_OPEN		Generic no-text error message, text will be set by reporting code.
1504 109	RtcClearing_INVALID_ORIGINAL_TRADING_SYSTEM_TRADE_MATCH_ID		Generic no-text error message, text will be set by reporting code.
1504 110	RtcClearing_SUB_ACCOUNT_MODIFICATION_NOT_ALLOWED		Generic no-text error message, text will be set by reporting code.

1504 111	RtcClearing_INSTRUMENT_NOT_EXPIRED		Generic no-text error message, text will be set by reporting code.
1504 112	RtcClearing_AUTHORIZATION_FAILED		Generic no-text error message, text will be set by reporting code.
1504 113	RtcClearing_UNKNOW_COMMISSION		Generic no-text error message, text will be set by reporting code.
1504 114	RtcClearing_EXISTING_COMMISSION		Generic no-text error message, text will be set by reporting code.
1504 115	RtcClearing_INVALID_STATUS		Generic no-text error message, text will be set by reporting code.
1504 116	RtcClearing_INVALID_MEMBER		Generic no-text error message, text will be set by reporting code.
1504 117	RtcClearing_OPEN_INTEREST_NETTING_GROUP_ALREADY_EXISTS		Generic no-text error message, text will be set by reporting code.
1506 001	RtcRisk_INVALID_QUANTITY		Generic no-text error message, text will be set by reporting code.
1506 002	RtcRisk_INVALID_VALUE		Generic no-text error message, text will be set by reporting code.
1506 003	RtcRisk_INVALID_SETTLEMENT_DATE		Generic no-text error message, text will be set by reporting code.
1506 004	RtcRisk_NO_BUSINESS_DATE		Generic no-text error message, text will be set by reporting code.
1506 005	RtcRisk_INVALID_INSTRUMENT_PLUGIN		Generic no-text error message, text will be set by reporting code.
1506 006	RtcRisk_INVALID_BASIC_ATTRIBUTES		Generic no-text error message, text will be set by reporting code.
1506 007	RtcRisk_INVALID_ADDITIONAL_ATTRIBUTES		Generic no-text error message, text will be set by reporting code.
1506 008	RtcRisk_INVALID_PRODUCT_REFERENCE		Generic no-text error message, text will be set by reporting code.
1506 009	RtcRisk_INVALID_CURRENCY_REFERENCE		Generic no-text error message, text will be set by reporting code.
1506 010	RtcRisk_INVALID_TRADABLE_INSTRUMENT_ATTRIBUTES		Generic no-text error message, text will be set by reporting code.
1506 011	RtcRisk_DUPLICATE_ID		Generic no-text error message, text will be set by reporting code.
1506 012	RtcRisk_INVALID_NAME_FOR_ID		Generic no-text error message, text will be set by reporting code.
1506 013	RtcRisk_IDENTICAL_OPERATION_PENDING		Generic no-text error message, text will be set by reporting code.

1506 014	RtcRisk_UNKNOWN_RISK_NODE		Generic no-text error message, text will be set by reporting code.
1506 015	RtcRisk_UNKNOWN_ACCOUNT		Generic no-text error message, text will be set by reporting code.
1506 016	RtcRisk_UNKNOWN_INSTRUMENT		Generic no-text error message, text will be set by reporting code.
1506 017	RtcRisk_INVALID_PARTITION		Generic no-text error message, text will be set by reporting code.
1506 018	RtcRisk_INVALID_PRODUCT_NAME		Generic no-text error message, text will be set by reporting code.
1506 019	RtcRisk_INVALID_SYMBOL		Generic no-text error message, text will be set by reporting code.
1506 020	RtcRisk_ID_NOT_UNIQUE		Generic no-text error message, text will be set by reporting code.
1506 021	RtcRisk_INVALID_ID_FOR_NAME		Generic no-text error message, text will be set by reporting code.
1506 022	RtcRisk_INVALID_PRODUCT_ID		Generic no-text error message, text will be set by reporting code.
1506 023	RtcRisk_INVALID_TRADABLE_INSTRUMENT_ID		Generic no-text error message, text will be set by reporting code.
1506 024	RtcRisk_INVALID_ACCOUNT_ID		Generic no-text error message, text will be set by reporting code.
1506 025	RtcRisk_INVALID_ACCOUNT_NAME		Generic no-text error message, text will be set by reporting code.
1506 026	RtcRisk_INVALID_CALCULATOR_PROVIDER_ID		Generic no-text error message, text will be set by reporting code.
1506 027	RtcRisk_INVALID_CALCULATOR_CONFIG_ATTRIBUTES		Generic no-text error message, text will be set by reporting code.
1506 028	RtcRisk_UNKNOWN_CALCULATOR_ID		Generic no-text error message, text will be set by reporting code.
1506 029	RtcRisk_INVALID_FOREX_ID		Generic no-text error message, text will be set by reporting code.
1506 030	RtcRisk_INVALID_FOREX_ATTRIBUTES		Generic no-text error message, text will be set by reporting code.
1506 031	RtcRisk_UNKNOWN_FILTER_ID		Generic no-text error message, text will be set by reporting code.
1506 032	RtcRisk_UNKNOWN_EVALUATOR_ID		Generic no-text error message, text will be set by reporting code.
1506 033	RtcRisk_DUPLICATE_CHANNEL_NAME		Generic no-text error message, text will be set by reporting code.

1506 034	RtcRisk_INVALID_RISK_LIMIT		Generic no-text error message, text will be set by reporting code.
1506 035	RtcRisk_RISK_NODE_NOT_READY_FOR_UPDATE		Generic no-text error message, text will be set by reporting code.
1506 036	RtcRisk_UNKNOWN_CHANNEL		Generic no-text error message, text will be set by reporting code.
1506 037	RtcRisk_DEFAULT_CHANNEL_MUST_EXIST		Generic no-text error message, text will be set by reporting code.
1506 038	RtcRisk_UNKNOWN_OBJECT		Generic no-text error message, text will be set by reporting code.
1506 039	RtcRisk_INVALID_FILTER_CONFIGURATION		Generic no-text error message, text will be set by reporting code.
1506 040	RtcRisk_INVALID_EVALUATOR_CONFIGURATION		Generic no-text error message, text will be set by reporting code.
1506 041	RtcRisk_EVALUATOR_IS_IN_USE		Generic no-text error message, text will be set by reporting code.
1506 042	RtcRisk_FILTER_IS_IN_USE		Generic no-text error message, text will be set by reporting code.
1506 043	RtcRisk_CALCULATOR_IS_IN_USE		Generic no-text error message, text will be set by reporting code.
1506 044	RtcRisk_INVALID_BOOKMARK		Generic no-text error message, text will be set by reporting code.
1506 045	RtcRisk_MARKETDATA_MISSING		Generic no-text error message, text will be set by reporting code.
1506 046	RtcRisk_POSITION_MISSING		Generic no-text error message, text will be set by reporting code.
1506 047	RtcRisk_DUPLICATE_FOREX		Generic no-text error message, text will be set by reporting code.
1506 048	RtcRisk_INVALID_BUSINESS_DATE		Generic no-text error message, text will be set by reporting code.
1506 049	RtcRisk_INVALID_TRACEABLE_CALC_ID		Generic no-text error message, text will be set by reporting code.
1506 050	RtcRisk_INVALID_BACKTESTING_ID		Generic no-text error message, text will be set by reporting code.
1506 051	RtcRisk_SUBSCRIPTION_ERROR		Generic no-text error message, text will be set by reporting code.
1506 052	RtcRisk_INVALID_UNDERLYING_REFERENCE		Generic no-text error message, text will be set by reporting code.
1506 053	RtcRisk_SERVICE_UNAVAILABLE		Generic no-text error message, text will be set by reporting code.

1506 054	RtcRisk_FAILED_RISK_CALCULATION		Generic no-text error message, text will be set by reporting code.
1506 055	RtcRisk_CALCULATOR_ERROR		Generic no-text error message, text will be set by reporting code.
1506 056	RtcRisk_CANNOT_HANDLE_RISKNODE		Generic no-text error message, text will be set by reporting code.
1506 057	RtcRisk_TIMED_OUT_OPERATION		Generic no-text error message, text will be set by reporting code.
1506 058	RtcRisk_CONCURRENT_OPERATION		Generic no-text error message, text will be set by reporting code.
1506 059	RtcRisk_CALCULATION_RESULT_EVALUATOR_ERROR		Generic no-text error message, text will be set by reporting code.
1506 060	RtcRisk_INTERNAL_ERROR		Generic no-text error message, text will be set by reporting code.
1506 061	RtcRisk_CURRENCY_CONVERSION_ERROR		Generic no-text error message, text will be set by reporting code.
1506 062	RtcRisk_MULTIPLE_CALCULATOR_ERROR		Generic no-text error message, text will be set by reporting code.
1506 063	RtcRisk_MISSING_HISTORICAL_PRICES		Generic no-text error message, text will be set by reporting code.
1506 064	RtcRisk_MISSING_CLOSING_PRICES		Generic no-text error message, text will be set by reporting code.
1506 065	RtcRisk_MISSING_JSPAN_ATTRIBUTES		Generic no-text error message, text will be set by reporting code.
1506 066	RtcRisk_DB_OPERATION_FAILED		Generic no-text error message, text will be set by reporting code.
1506 067	RtcRisk_IMR_CALCULATION_FAILED		Generic no-text error message, text will be set by reporting code.
1506 068	RtcRisk_INVALID_MARKET		Generic no-text error message, text will be set by reporting code.
1506 069	RtcRisk_FAILED_RISK_ARRAY_CALCULATION		Generic no-text error message, text will be set by reporting code.
1506 070	RtcRisk_INVALID_MARKETDATAPRICEKEYS		Generic no-text error message, text will be set by reporting code.
1506 071	RtcRisk_INVALID_DELETE		Generic no-text error message, text will be set by reporting code.
1506 072	RtcRisk_NO_MARKET_PRICE		Generic no-text error message, text will be set by reporting code.
1506 073	RtcRisk_NO_INSTRUMENTS_FOR_BACKTESTING		Generic no-text error message, text will be set by reporting code.

1506 074	RtcRisk_HISTORICAL_DATA_NOT_FOUND		Generic no-text error message, text will be set by reporting code.
1506 075	RtcRisk_UNKNOWN_STRESS_SCENARIO_FACTOR		Generic no-text error message, text will be set by reporting code.
1506 076	RtcRisk_HISTORICAL_STRESSTEST_NOT_PERMITTED		Generic no-text error message, text will be set by reporting code.
1506 077	RtcRisk_INVALID_CALCULATOR_CONFIG		Generic no-text error message, text will be set by reporting code.
1506 078	RtcRisk_AUTHORIZATION_FAILED		Generic no-text error message, text will be set by reporting code.
1506 079	RtcRisk_SERVICE_NOT_SUPPORTED		Generic no-text error message, text will be set by reporting code.
1506 080	RtcRisk_MULTIPLE_PARTITIONS_REQUESTED		Generic no-text error message, text will be set by reporting code.
1506 081	RtcRisk_INVALID_LAMBDA_NAME		Generic no-text error message, text will be set by reporting code.
1506 082	RtcRisk_INVALID_WHAT_IF_CALC_ID		Generic no-text error message, text will be set by reporting code.
1506 083	RtcRisk_INVALID_WHAT_IF_REQUEST		Generic no-text error message, text will be set by reporting code.
1507 001	RtcMarketData_PARTIAL_SUCCESS		Generic no-text error message, text will be set by reporting code.
1507 002	RtcMarketData_INTERNAL_SERVER_ERROR		Generic no-text error message, text will be set by reporting code.
1507 003	RtcMarketData_NO_RECORD_FOUND		Generic no-text error message, text will be set by reporting code.
1507 004	RtcMarketData_TARGET_SERVER_NOT_AVAILABLE		Generic no-text error message, text will be set by reporting code.
1507 005	RtcMarketData_REQUEST_INTERNAL_TIMEOUT		Generic no-text error message, text will be set by reporting code.
1507 006	RtcMarketData_REFERENCE_DATA_MISSING		Generic no-text error message, text will be set by reporting code.
1507 007	RtcMarketData_MARKET_DATA_MISSING		Generic no-text error message, text will be set by reporting code.
1507 008	RtcMarketData_VALUATION_FAILED		Generic no-text error message, text will be set by reporting code.
1507 009	RtcMarketData_DISPATCH_ERROR		Generic no-text error message, text will be set by reporting code.
1507 010	RtcMarketData_MISSING_HISTORICAL_PRICES		Generic no-text error message, text will be set by reporting code.

1507 011	RtcMarketData_MISSING_CLOSING_PRICES		Generic no-text error message, text will be set by reporting code.
1507 012	RtcMarketData_MISSING_JSPAN_ATTRIBUTES		Generic no-text error message, text will be set by reporting code.
1507 013	RtcMarketData_DB_OPERATION_FAILED		Generic no-text error message, text will be set by reporting code.
1507 014	RtcMarketData_IMR_CALCULATION_FAILED		Generic no-text error message, text will be set by reporting code.
1507 015	RtcMarketData_INVALID_MARKET		Generic no-text error message, text will be set by reporting code.
1507 016	RtcMarketData_EXTERNAL_USER_ERROR		Generic no-text error message, text will be set by reporting code.
1507 017	RtcMarketData_ILLEGAL_ARGUMENT_VALUE		Generic no-text error message, text will be set by reporting code.
1507 018	RtcMarketData_ERRONEOUS_SUBSCRIPTION_ID		Generic no-text error message, text will be set by reporting code.
1507 019	RtcMarketData_SERVICE_UNAVAILABLE		Generic no-text error message, text will be set by reporting code.
1507 020	RtcMarketData_INVALID_BOOKMARK		Generic no-text error message, text will be set by reporting code.
1507 021	RtcMarketData_INSTRUMENT_DOES_NOT_EXIST		Generic no-text error message, text will be set by reporting code.
1507 022	RtcMarketData_INCORRECT_INSTRUMENT_TYPE		Generic no-text error message, text will be set by reporting code.
1507 023	RtcMarketData_MISSING_REQUIRED_FIELD		Generic no-text error message, text will be set by reporting code.
1507 024	RtcMarketData_REFERENCE_DATA_UPDATING		Generic no-text error message, text will be set by reporting code.
1507 025	RtcMarketData_UNHANDLED_INSTRUMENT_CLASSES		Generic no-text error message, text will be set by reporting code.
1507 026	RtcMarketData_PARTIALSUCCESS		Generic no-text error message, text will be set by reporting code.
1507 027	RtcMarketData_AUTHORIZATION_FAILED		Generic no-text error message, text will be set by reporting code.
1508 001	RtcSettlement_INVALID_NETTING_RULE_ID		Generic no-text error message, text will be set by reporting code.
1508 002	RtcSettlement_INVALID_NETTING_RULE_NAME		Generic no-text error message, text will be set by reporting code.
1508 003	RtcSettlement_INVALID_NETTING_RULE_ACCOUNT		Generic no-text error message, text will be set by reporting code.

1508004	RtcSettlement_INVALID_EXTERNAL_ACCOUNT_REF		Generic no-text error message, text will be set by reporting code.
1508005	RtcSettlement_CONCURRENT_POSITION_MODIFICATION		Generic no-text error message, text will be set by reporting code.
1508006	RtcSettlement_INVALID_ACCOUNT_ID		Generic no-text error message, text will be set by reporting code.
1508007	RtcSettlement_EXTERNAL_REF_NOT_DEFINED		Generic no-text error message, text will be set by reporting code.
1508008	RtcSettlement_NO_CLEARING_HOUSE_ACCOUNT		Generic no-text error message, text will be set by reporting code.
1508009	RtcSettlement_INVALID_INSTRUCTION_ID		Generic no-text error message, text will be set by reporting code.
1508010	RtcSettlement_INVALID_NETTING_RULE_TYPE		Generic no-text error message, text will be set by reporting code.
1508011	RtcSettlement_INVALID_NETTING_RULE_FILTER		Generic no-text error message, text will be set by reporting code.
1508012	RtcSettlement_INVALID_AMOUNT		Generic no-text error message, text will be set by reporting code.
1508013	RtcSettlement_INVALID_ACCOUNT		Generic no-text error message, text will be set by reporting code.
1508014	RtcSettlement_DUPLICATE_CLIENT_DEPOSIT_ID		Generic no-text error message, text will be set by reporting code.
1508015	RtcSettlement_TOO_LARGE_WITHDRAWAL		Generic no-text error message, text will be set by reporting code.
1508016	RtcSettlement_NO_USER_SPECIFIED		Generic no-text error message, text will be set by reporting code.
1508017	RtcSettlement_INVALID_BOOKMARK		Generic no-text error message, text will be set by reporting code.
1508018	RtcSettlement_INVALID_SETTLEMENT_RUN_ID		Generic no-text error message, text will be set by reporting code.
1508019	RtcSettlement_INVALID_INSTRUCTION		Generic no-text error message, text will be set by reporting code.
1508020	RtcSettlement_UNKNOWN_ACCOUNT		Generic no-text error message, text will be set by reporting code.
1508021	RtcSettlement_UNKNOWN_SSI		Generic no-text error message, text will be set by reporting code.
1508022	RtcSettlement_UNKNOWN_INSTRUCTION		Generic no-text error message, text will be set by reporting code.
1508023	RtcSettlement_UNKNOWN_SETTLEMENT_RUN		Generic no-text error message, text will be set by reporting code.

1508 024	RtcSettlement_INVALID_SSI_ID		Generic no-text error message, text will be set by reporting code.
1508 025	RtcSettlement_MESSAGE_ALREADY_PROCESSED		Generic no-text error message, text will be set by reporting code.
1508 026	RtcSettlement_INVALID_INSTRUCTION_STATE		Generic no-text error message, text will be set by reporting code.
1508 027	RtcSettlement_EXTERNAL_ACCOUNT_NOT_FOUND		Generic no-text error message, text will be set by reporting code.
1508 028	RtcSettlement_SETTLEMENT_RUN_IN_STARTED_STATE		Generic no-text error message, text will be set by reporting code.
1508 029	RtcSettlement_ADD_SETTLED_AMOUNT_FAILED		Generic no-text error message, text will be set by reporting code.
1508 030	RtcSettlement_INVALID_INSTRUCTION_AMOUNT		Generic no-text error message, text will be set by reporting code.
1508 031	RtcSettlement_INVALID_SETTLEMENT_RUN_STATE		Generic no-text error message, text will be set by reporting code.
1508 032	RtcSettlement_SETTLEMENT_RUN_NOT_IN_CREATED_STATE		Generic no-text error message, text will be set by reporting code.
1508 033	RtcSettlement_SETTLEMENT_RUN_IS_CANCELED		Generic no-text error message, text will be set by reporting code.
1508 034	RtcSettlement_INVALID_SETTLEMENT_RUN_FILTER		Generic no-text error message, text will be set by reporting code.
1508 035	RtcSettlement_INVALID_INSTRUCTION_FILTER		Generic no-text error message, text will be set by reporting code.
1508 036	RtcSettlement_INVALID_TRADABLE_INSTRUMENT_ID		Generic no-text error message, text will be set by reporting code.
1508 037	RtcSettlement_SERVICE_UNAVAILABLE		Generic no-text error message, text will be set by reporting code.
1508 038	RtcSettlement_INTERNAL_ERROR		Generic no-text error message, text will be set by reporting code.
1508 039	RtcSettlement_REQUEST_TIMED_OUT		Generic no-text error message, text will be set by reporting code.
1508 040	RtcSettlement_REQUEST_TIMED_OUT_STATE_MAY_HAVE_CHANGED		Generic no-text error message, text will be set by reporting code.
1508 041	RtcSettlement_VALIDATION_ERROR		Generic no-text error message, text will be set by reporting code.
1508 042	RtcSettlement_NO_INSTRUMENTS		Generic no-text error message, text will be set by reporting code.
1508 043	RtcSettlement_AUTHORIZATION_FAILED		Generic no-text error message, text will be set by reporting code.

1508 044	RtcSettlement_DAS_OBJECT_NOT_CREATED		Generic no-text error message, text will be set by reporting code.
1510 001	BdxRtcMessageValidationFailed		Generic no-text error message, text will be set by reporting code.
1512 001	RtcDatawarehouse_VALIDATION_ERROR		Generic no-text error message, text will be set by reporting code.
1512 002	RtcDatawarehouse_SERVICE_UNAVAILABLE		Generic no-text error message, text will be set by reporting code.
1512 003	RtcDatawarehouse_INTERNAL_ERROR		Generic no-text error message, text will be set by reporting code.
1512 004	RtcDatawarehouse_SUCCESS		Generic no-text error message, text will be set by reporting code.
1513 001	TaxEndOfDayServiceNotAvailable	The end of day service is not available in this TAX server	End of day service not available.
1513 002	Tax_MORE_THAN_EXPECTED_POSITIONS		Generic no-text error message, text will be set by reporting code.
1513 003	TaxMessageValidationFailed		Generic no-text error message, text will be set by reporting code.

APPENDIX B - COMMISSIONS

1. Process for Assigns and Tripartite Allocations:

Creation of pending commission

When the assign/tripartite request is created or submitted, the front end can create a pending commission entry. One option is to include the value of the “reference” field on the deal management request message (field #6) on the pending commission. This can be used later to link the deal confirmation to the associated pending commission. If the ‘Reference’ field is being used for other purposes, the front end will need to link the deal confirmation to the relevant pending commission entry using the combination of accountID, instrument, price and quantity or through another linking method.

The client reference can also be populated on the pending commission entry:

- For an assign the `clientReference` must be populated with the code of the counterparty Trading Member or Branch i.e. code provided in the counterparty codes file published on IDP e.g. PRSXXXTMT01, ABL2.
- For a tripartite the `clientReference` must be populated with the “ClientID” of the tripartite client e.g. AAA523

Submission of commission

When the assign/tripartite is accepted by the counterparty and the [AccountPositionEvent](#) ‘From’ deal is received from RTC, the ‘Reference’ field on this message can be used to link to the deal management request and in turn the pending commission that must be submitted into the Clearing System.

The `CommissionReference` on the commission message must be populated with the value in the ‘NextTradeIds’ field (field #15) of the Trade sub-message (10015), which is field# 72 of the [AccountPositionEvent](#) message (10032). This ‘NextTradeIDs’ value references the ‘TradeID’ of the ‘To’ deal from RTC which will allow the counterparty to link the commission he receives with the deal that he receives.

Note: The initiator will receive two [AccountPositionEvent](#) ‘From’ deals. The one will be an update of the trade that is being assigned/tripartite allocated*, with the position reason updated to ‘Allocate From’. The other will be the equal and opposite deal which will also have a position reason of “Allocate From”. The relevant deal for purposes of linking to the pending commission is the equal and opposite deal – its ‘Reference’ field will match the ‘Reference’ field of the deal management request and the pending commission.

Note: * The trade being assigned/tripartite allocated may be a trade that has not been deal managed (original trade), a deal that resulted from an accumulation or a trade that has been partially deal managed.

Example Commission with an Assign

Position reason	RTC assigned identifier	TM/Branch	Position Account	NextTradeIds	PreviousTradeIds
APE – Field 69	“TradeId” –APE - Trade submessage- Field 1	APE – Field 82 or 83	“AccountId” APE – Field 51	APE – Trade submessage – Field 15	APE – Trade submessage – Field 16
Trade	4530689	ABL2	2590479616820789		
Assign From*	4530689	ABL2	2590479616820789	4530690	
Assign From	4530690	ABL2	2590479616820789	4530691	4530689
Assign To	4530691	CRCXXXTMT01	2590464575745882		4530690

* Original trade with updated position reason (as it will be received in live subscription)

AddCommissionReq:

Field	Value
InitiatingTM	ABL2
destinationTM	CRCXXXTMT01
clientReference	CRCXXXTMT01
commissionReference	4530691

Example: Commission with a Tripartite Allocation

Position reason	RTC assigned identifier	TM/Branch	Position Account	NextTrade IDs	PreviousTradeIDs
APE – Field 69	“TradeID” – APE - Trade submessage- Field 1	APE – Field 82 or 83	“AccountId” APE – Field 51	APE – Trade submessage – Field 15	APE – Trade submessage – Field 16
Trade	4530689	ABL2	2590479616820789		
Tripartite From	4530689	ABL2	2590479616820789	4530690	
Tripartite From	4530690	ABL2	2590479616820789	4530691	4530689
Tripartite To	4530691	CRCXXXTMT01	2590464575745882		4530690

AddCommissionReq:

Field	Value
InitiatingTM	ABL2
destinationTM	CRCXXXTMT01
clientReference	AAA523
commissionReference	4530691

2. Process for Allocations:

Note: For own-client commissions, two options are possible. See below:

- a) Commissions calculated at EOD in the clearing software through the application of commission fee tables provided by the TM to his CM beforehand (once off setup)

This method is currently used for certain commissions/brokerage fees today. One of the existing commission/brokerage fees can be used to incorporate the new 'allocation' commission i.e. the commission fee that replacing the current turn. Note if this option is used for commissions charged in lieu of the current turn then own client commission entries will not be submitted to the JSE clearing system

- b) Commissions submitted at time of trading or deal management.

The option to be catered for in front ends will depend on Trading Member requirements – do they have a requirement for commissions charged to their own clients to be submitted at the time of trading or deal management or can this be done at EOD, as is currently the case for certain types of commissions and brokerage fees charged by TMs. The rules below pertain to option 2 above.

Creation of pending commission

When the allocation request is created or submitted, the front end can create a pending commission entry. The `clientReference` must be populated with the 'accountID'* of the payer of the commission e.g. 2537111731090004

Submission of commission

When the `AccountPositionEvent` 'To' deal is received on confirmation of the allocation, the front end will need to link it to the relevant pending commission entry using the combination of accountID, instrument, price and quantity or through another linking method.

The `CommissionReference` on the Commission message must be populated with the TradeID of the 'To' deal i.e. 'TradeID' field (1) of the Trade sub-message (10015), which is field# 72 of the `AccountPositionEvent` message (10032).

Note: When submitting deal management instructions, it is the accountID that is included on the instruction hence the recommendation to use this field when submitting commissions and not the ExternalAccountID

Example: Commission with an Allocation

Position reason APE – Field 69	RTC assigned identifier “TradeID” – APE - Trade submessage- field 1	TM/Branch APE – Field 82 or 83	Position Account “AccountID” APE – Field 51	NextTradeID s APE – Trade submessage - Field 15	PreviousTradeID Ds APE – Trade submessage - Field 16
Trade	4530689	CRCXXXTMT01	2590479616820006*		
Allocate From	4530689	CRCXXXTMT01	2590479616820006	4530690	
Allocate From	4530690	CRCXXXTMT01	2590479616820006	4530691	4530689
Allocate To	4530691	CRCXXXTMT01	2590563853059535		4530690

* Internal ID of 99999 suspense account

AddCommissionReq:

Field	Value
InitiatingTM	CRCXXXTMT01
destinationTM	CRCXXXTMT01
clientReference	2590563853059535
commissionReference	4530691

3. Process for Off-book trades done directly on client account:

Creation of pending commission

When the Off book trade is booked, the front end can create a pending commission entry. If not used for another purpose either of the available free-text reference fields on the Off book trade capture message can be populated with a value that can also be populated on the pending commission to allow linking of the RTC deal to the pending commission later in the process.

The `clientReference` must be populated with the "accountID"* of the payer of the commission e.g. 2537111731090004.

Submission of commission

When the `AccountPositionEvent` deal is received from RTC (position reason of Trade), the front end will need to link it to the pending commission entry. Per above if one of the free-text fields on the reported trade is used for purposes of linking to the pending commission this can be used to match the resultant deal from RTC with the pending commission. Alternatively, the resultant deal can be linked to the pending commission using the combination of accountID, instrument, price and quantity or another linking method.

The `commissionReference` field of the commission message must be populated with the 'Trade ID' field (1) of the Trade sub-message (position reason of 'Trade') of the corresponding `AccountPositionEvent` message.

Note: When submitting off-book trades to MIT, the external account ID is provided. However RTC returns both the external account ID and internal account ID in the account position event deal. To be consistent with the previously mentioned deal management activities, the internal account ID should be populated.

4. Process for On-book trades done directly on client account:

Creation of pending commission

When the On book order is captured, the front end can create a pending commission entry and as the order is executed (filled) the relevant part of the commission can be processed. One option is to use the `ClientOrderID` in the execution report and trade capture report from MIT as well as on the RTC deal message to link the trade to the pending commission.

The `clientReference` must be populated with the "accountID"* of the payer of the commission e.g. 2537111731090004.

Submission of commission

When the `AccountPositionEvent` deal is received from RTC (position reason of Trade), the front end will need to link it to the pending commission entry. Per above if `ClientOrderID` value is used for purposes of linking to the pending commission this can be used to match the resultant deal from RTC with the pending commission. Alternatively, the resultant deal can be linked to the pending commission using the combination of accountID, instrument, price and quantity or through another linking method.

The `CommissionReference` field of the commission message must be populated with the 'Trade ID' field (1) of the Trade sub-message (position reason of 'Trade') of the corresponding `AccountPositionEvent` message.

Note: When submitting on-book trades to MIT, the external account ID is provided. However RTC returns both the external account ID and internal account ID in the account position event deal. To be consistent with the previously mentioned deal management activities, the internal account ID should be populated.

Example: Commission submitted associated with a trade (on or off-book) executed in the trading system directly on the client's account

Buy Side Trade				Sell Side Trade			
Position Reason	RTC assigned identifier	TM/Branch	Position Account	Position Reason	RTC assigned identifier	TM/Branch	Position Account
APE – Field 69	"TradeID" – APE - Trade submessage-field 1	APE – Field 82 or 83	"AccountID" APE – Field 51	APE – Field 69	"TradeID" – APE - Trade submessage-field 1	APE – Field 82 or 83	"AccountID" APE – Field 51
Trade	4530689	CRCXXXTMT01	2537111731090004	Trade	4530689	ABMXXXTMT01	2590479616820004

AddCommissionReq:

Field	Value
InitiatingTM	CRCXXXTMT01
destinationTM	CRCXXXTMT01
clientReference	2537111731090004
commissionReference	4530689

Note: Similarly a commission can be captured for the sell side by the counterparty if sell-side TM acting in an agency capacity.

5. Bulk Commissions:

In addition to a commission per deal, the JSE commission solution also allows bulk commissions i.e. a single commission entry applicable to multiple trades/deals. If submitting a bulk commission **the `clientReference` field must be populated as specified in sections above** depending on whether the destination is a Trading Member, Branch, own client or tripartite client. The `CommissionReference` will not and does not need to be populated with a specific, individual deal identifier and the clearing software will not attempt to link and report on the commission against a specific trade/deal. The clearing software will however feed these commissions into EOD settlement amounts.

6. Allocation Correction:

The commission submitted for the incorrect allocation must be cancelled.

Creation of pending commission

When the Correct Allocation Error request is created or submitted, the front end can create a pending commission entry. One option is to include the value of the "Reference" field on the deal management request message (field #5) on the pending commission. This can be used later to link the deal confirmation to the associated pending commission. If the 'Reference' field is being used for other purposes, the front end will need to link the deal confirmation to the relevant pending commission entry using the combination of `accountID`, instrument, price and quantity or through another linking method.

The `clientReference` must be populated with the '`accountID`'* of the payer of the commission e.g. 2537111731090004.

Submission of commission

When the `AccountPositionEvent` 'To' deal is received on confirmation of the allocation correction, the front end will need to link it to the relevant pending commission entry using the reference captured on the deal management request or combination of `accountID`, instrument, price and quantity or through another linking method.

The `CommissionReference` on the Commission message must be populated with the `TradeID` of the 'To' deal i.e. '`TradeID`' field (1) of the Trade sub-message (10015), which is field# 72 of the `AccountPositionEvent` message (10032).

Note: When submitting deal management instructions, it is the `accountID` that is included on the instruction hence the recommendation to use this field when submitting commissions and not the `ExternalAccountID`

Example: Commission for an Allocation Correction Error

The commission submitted for the incorrect allocation must be cancelled.

Position reason	RTC assigned identifier	TM/Branch	Position Account	NextTradeIDs	PreviousTradeIDs
APE – Field 69	“TradeID” – APE - Trade submessage-Field 1	APE – Field 82 or 83	“AccountId” APE – Field 51	APE – Trade submessage – Field 15	APE – Trade submessage – Field 16
Allocate To	4530689	CRCXXXTMT01	2590479616820006		
Allocation Correction From	4530689	CRCXXXTMT01	2590479616820006	4530690	
Allocation Correction From	4530690	CRCXXXTMT01	2590479616820006	4530691	4530689
Allocation Correction To	4530691	CRCXXXTMT01	2590563853059535		4530690

AddCommissionReq:

Field	Value
InitiatingTM	CRCXXXTMT01
destinationTM	CRCXXXTMT01
clientReference	2590563853059535
commissionReference	4530691

7. Principal Correction:

The commission submitted for the original trade to the client account must be cancelled.

8. Trade Cancellation:

The commission submitted for the original trade must be cancelled.

- A new commission for the new/replacement trade if applicable must be submitted

9. Price Adjust:

The commission submitted for the incorrect trade must be cancelled and a new pending commission created for the price adjust trade.

Creation of pending commission

When the Price Adjust trade is booked, the front end can create a pending commission entry. If not used for another purpose either of the available free-text reference fields on the Off book trade capture message can be populated with a value that can also be populated on the pending commission to allow linking of the RTC deal to the pending commission later in the process.

The `clientReference` must be populated with the "accountID"* of the payer of the commission e.g. 2537111731090004.

Submission of commission

When the `AccountPositionEvent` deal is received from RTC (position reason of Price Adjust), the front end will need to link it to the pending commission entry. Per above if one of the free-text fields on the reported trade is used for purposes of linking to the pending commission this can be used to match the resultant deal from RTC with the pending commission. Alternatively, the resultant deal can be linked to the pending commission using the combination of accountID, instrument, price and quantity or another linking method.

The `commissionReference` field of the commission message must be populated with the 'Trade ID' field (1) of the Trade sub-message (position reason of 'Price_Adjust') of the corresponding `AccountPositionEvent` message.

Note: When submitting off-book trades to MIT, the external account ID is provided. However RTC returns both the external account ID and internal account ID in the account position event deal. To be consistent with the previously mentioned deal management activities, the internal account ID should be populated.

Example: Commission for a Price Adjust Trade

The commission submitted for the incorrect trade must be cancelled and a new pending commission created for the price adjust trade.

Buy Side Trade				Sell Side Trade			
Position Reason APE – Field 69	RTC assigned identifier “TradeID” –APE - Trade submessage-field 1	TM/Branch APE – Field 82 or 83	Position Account “AccountID” APE – Field 51	Position Reason APE – Field 69	RTC assigned identifier “TradeID” –APE - Trade submessage-field 1	TM/Branch APE – Field 82 or 83	Position Account “AccountID” APE – Field 51
Cancelled_Price_Adjust	4530689	CRCXXXTMT01	253711731090004	Cancelled_Price_Adjust	4530689	ABMXXXTMT01	2590479616820004

AddCommissionReq for Buy side trade::**

Field	Value
InitiatingTM	CRCXXXTMT01
destinationTM	CRCXXXTMT01
clientReference	2537111731090004
commissionReference	4530689

**Similarly a commission can be captured for the sell side by the counterparty if sell-side TM acting in an agency capacity

10. Commission charged to counterparty for reported trade.

Charging a commission to the counterparty member (or branch) in a reported trade. There are certain business transactions which are affected through a reported trade (or Off Book trade) and in which a commission may be charged to the counterparty. Examples include:

- A member brokers a deal between his own client and another member, charging commissions to both his own client and the other member
- An inter-dealer broker executes on behalf of a member, charging the member a commission when he gives up the trade

c) A member's client executes through another member, the executing broker. When the executing broker gives up the trade to the client's member he charges a commission. This scenario has two variations:

- a. Member's client places order directly with another member
- b. Member outsources the execution of his client's order to an executing broker

Note: The give-ups in scenarios (b) and (c) above should be achieved through the assign mechanism in deal management. However in the case where the executing broker executes through a negotiated trade and cannot wait for the counterparty to accept his leg of the trade, the executing broker can affect the give-up through a reported trade instead of an assign. The purpose of this document is to specify how the two reference fields on the commission entry message must be populated in the case where a member charges a commission to the counterparty in a reported trade.

Client Reference

The ClientReference field of the commission message must be populated with the code of the counterparty Trading Member or Branch i.e. code provided in the counterparty codes file published on IDP e.g. PRSXXXTMT01, ABL2.

Commission Reference

The CommissionReference field of the commission message must be populated with the unique identifier of the matched trade that is common to both the buy and sell legs of the trade. This unique identifier is available and can be taken from either:

i) System/API: Trading System

Message: Trade Capture Report (AE) message i.e. Trade Capture Report for the confirmation of the acceptance of the Trade
Field: TradeID (Tag 1003)

ii) System/API: Real-time Clearing System (RTC)

Message: AccountPositionEvent (10032) message, Trade (10015) sub message, RTCTradeExternalData (10085) sub-message
Field: TradingSystemMatchID (Field 10)

Example: This example uses fields off the RTC deal message to populate the commission reference fields. As stated above one can also use fields off the Trade Capture Report message.

Buy Leg			
Position Reason APE – Field 69	RTC assigned identifier “TradeID” –APE - Trade submessage-field 1	TM/Branch APE – Field 82 or 83	TradingSystemMatchID RTCTradeExternalData – Field 10
Trade	4530689	CRCXXXTMT01	M8zplwlr6

Sell Leg			
Position Reason APE – Field 69	RTC assigned identifier “TradeID” –APE - Trade submessage-field 1	TM/Branch APE – Field 82 or 83	TradingSystemMatchID RTCTradeExternalData – Field 10
Trade	4530702	ABMXXXTMT01	M8zplwlr6

AddCommissionReq (assuming Member on buy side is charging commission to Member on sell side)

Field	Value
InitiatingTM	CRCXXXTMT01
destinationTM	ABMXXXTMT01
clientReference	ABMXXXTMT01
commissionReference	M8zplwlr6

APPENDIX C – UPDATEABLE POSITION REASONS

Note: This Appendix contains the position reasons in *EmapiTransactionsForMember.html* which when linked to trades, those particular trades references can be updated using the *UpdateTradeReferenceReq* message.

Note: The complete technical specification documents can be found here: <https://www.jse.co.za/services/itac>

Position Reason	Description	Allow reference to be edited
ACCUMULATED_FROM	The deal was aggregated to another deal.	yes
ACCUMULATED_TO	The deal originates from a deal aggregation activity.	yes
ALLOCATED_FROM	The trade is allocated from a trading member account to a client account.	yes
ALLOCATED_TO	The trade origins from an allocation from a trading member account to a client account.	yes
ALLOCATION_CORRECTION_FROM	The deal is erroneously moved from the client account to another client account.	yes
ALLOCATION_CORRECTION_TO	The deal originates from a move from one client account to another client account.	yes
ASSIGN_CANCELLED	The deal assign cancelled by initiator.	yes
ASSIGN_EXPIRED	The deal assign has expired.	yes
ASSIGN_INITIATED	The deal assign initiated from.	yes
ASSIGN_REJECTED	The deal assign rejected by receiver.	yes
ASSIGNED_FROM	The deal assign source.	yes
ASSIGNED_TO	The deal originates from a deal assign activity.	yes
CANCELLED	Trade cancelled.	yes
POS_SUBACCOUNT_MOD_FROM	The position sub account was moved from this account..	yes
POS_SUBACCOUNT_MOD_TO	The position sub account was moved to this account.	yes
PRINCIPAL_CORRECTION_FROM	A position is moved from a house main account to a house sub account or from a house sub account to a house main account by creating a new deal.	yes
PRINCIPAL_CORRECTION_TO	The deal originates from a move from a client account.	yes
TRADE	New trade.	yes
TRADE_SUBACCOUNT_MOD_FROM	Position has been moved from this account as Sub account modification.	yes
TRADE_SUBACCOUNT_MOD_TO	Position has been moved to this account as Sub account modification.	yes
TRIPARTITE_APPROVED	Tripartite assign has been approved.	yes
TRIPARTITE_CANCELLED	Tripartite assign has been cancelled.	yes
TRIPARTITE_EXPIRED	Tripartite assign has expired.	yes
TRIPARTITE_FROM	The deal has been assigned using Tripartite agreement.	yes
TRIPARTITE_INITIATED	Tripartite assign has been initiated.	yes
TRIPARTITE_REJECTED	Tripartite assign has been rejected.	yes
TRIPARTITE_TO	The deal origins from an assign using Tripartite agreement.	yes